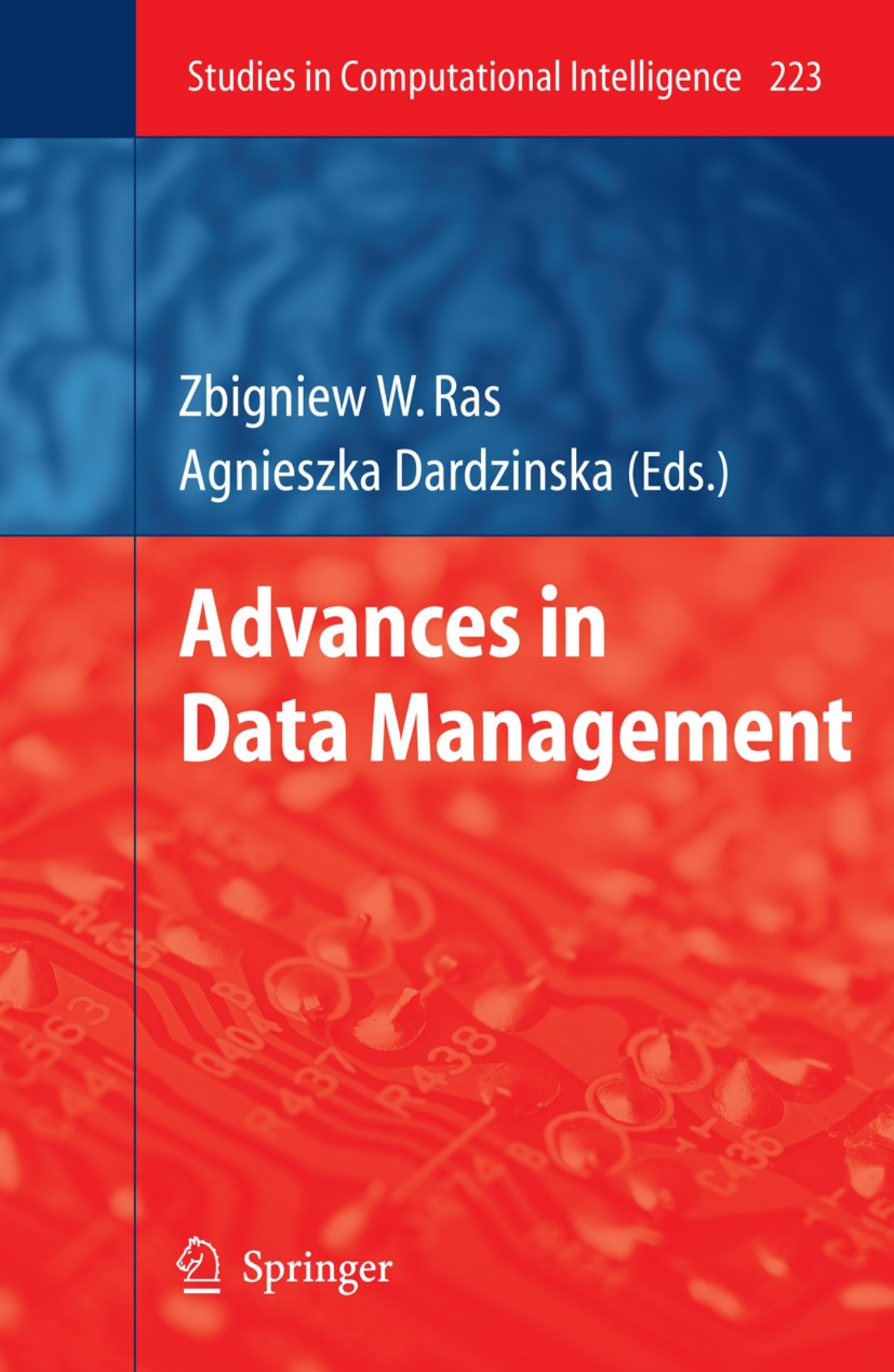
Studies in Computational Intelligence 223

Zbigniew W. Ras
Agnieszka Dardzinska (Eds.)

Advances in Data Management



Springer

Studies in Computational Intelligence, Volume 223

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 202. Aboul-Ella Hassanien, Ajith Abraham, and Francisco Herrera (Eds.)
Foundations of Computational Intelligence Volume 2, 2009
ISBN 978-3-642-01532-8

Vol. 203. Ajith Abraham, Aboul-Ella Hassanien, Patrick Siarry, and Andries Engelbrecht (Eds.)
Foundations of Computational Intelligence Volume 3, 2009
ISBN 978-3-642-01084-2

Vol. 204. Ajith Abraham, Aboul-Ella Hassanien, and André Ponce de Leon F. de Carvalho (Eds.)
Foundations of Computational Intelligence Volume 4, 2009
ISBN 978-3-642-01087-3

Vol. 205. Ajith Abraham, Aboul-Ella Hassanien, and Václav Snášel (Eds.)
Foundations of Computational Intelligence Volume 5, 2009
ISBN 978-3-642-01535-9

Vol. 206. Ajith Abraham, Aboul-Ella Hassanien, André Ponce de Leon F. de Carvalho, and Václav Snášel (Eds.)
Foundations of Computational Intelligence Volume 6, 2009
ISBN 978-3-642-01090-3

Vol. 207. Santo Fortunato, Giuseppe Mangioni, Ronaldo Menezes, and Vincenzo Nicosia (Eds.)
Complex Networks, 2009
ISBN 978-3-642-01205-1

Vol. 208. Roger Lee, Gongzu Hu, and Huaikou Miao (Eds.)
Computer and Information Science 2009, 2009
ISBN 978-3-642-01208-2

Vol. 209. Roger Lee and Naohiro Ishii (Eds.)
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2009
ISBN 978-3-642-01202-0

Vol. 210. Andrew Lewis, Sanaz Mostaghim, and Marcus Randall (Eds.)
Biologically-Inspired Optimisation Methods, 2009
ISBN 978-3-642-01261-7

Vol. 211. Godfrey C. Onwubolu (Ed.)
Hybrid Self-Organizing Modeling Systems, 2009
ISBN 978-3-642-01529-8

Vol. 212. Viktor M. Kureychik, Sergey P. Malyukov, Vladimir V. Kureychik, and Alexander S. Malyoukov
Genetic Algorithms for Applied CAD Problems, 2009
ISBN 978-3-540-85280-3

Vol. 213. Stefano Cagnoni (Ed.)
Evolutionary Image Analysis and Signal Processing, 2009
ISBN 978-3-642-01635-6

Vol. 214. Been-Chian Chien and Tzung-Pei Hong (Eds.)
Opportunities and Challenges for Next-Generation Applied Intelligence, 2009
ISBN 978-3-540-92813-3

Vol. 215. Habib M. Ammari
Opportunities and Challenges of Connected k-Covered Wireless Sensor Networks, 2009
ISBN 978-3-642-01876-3

Vol. 216. Matthew Taylor
Transfer in Reinforcement Learning Domains, 2009
ISBN 978-3-642-01881-7

Vol. 217. Horia-Nicolai Teodorescu, Junzo Watada, and Lakhmi C. Jain (Eds.)
Intelligent Systems and Technologies, 2009
ISBN 978-3-642-01884-8

Vol. 218. Maria do Carmo Nicoletti and Lakhmi C. Jain (Eds.)
Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control, 2009
ISBN 978-3-642-01887-9

Vol. 219. Maja Hadzic, Elizabeth Chang, Pornpit Wongthongtham, and Tharam Dillon
Ontology-Based Multi-Agent Systems, 2009
ISBN 978-3-642-01903-6

Vol. 220. Bettina Berendt, Dunja Mladenic, Marco de de Gemmis, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtech Svatek, and Filip Zelezny (Eds.)
Knowledge Discovery Enhanced with Semantic and Social Information, 2009
ISBN 978-3-642-01890-9

Vol. 221. Tassilo Pellegrini, Sören Auer, Klaus Tochtermann, and Sebastian Schaffert (Eds.)
Networked Knowledge - Networked Media, 2009
ISBN 978-3-642-02183-1

Vol. 222. Elisabeth Rakus-Andersson, Ronald R. Yager, Nikhil Ichalkaranje, and Lakhmi C. Jain (Eds.)
Recent Advances in Decision Making, 2009
ISBN 978-3-642-02186-2

Vol. 223. Zbigniew W. Ras and Agnieszka Dardzinska (Eds.)
Advances in Data Management, 2009
ISBN 978-3-642-02189-3

Zbigniew W. Ras and Agnieszka Dardzinska (Eds.)

Advances in Data Management

Springer

Prof. Zbigniew W. Ras
University of North Carolina at Charlotte
College of Computing and Informatics
Charlotte, N.C. 28223
USA
E-mail: ras@uncc.edu

Dr. Agnieszka Dardzinska
Wydzial Informatyki
Politechnika Bialostocka
ul. Wiejska 45a
15-351 Bialystok
Poland

ISBN 978-3-642-02189-3 e-ISBN 978-3-642-02190-9

DOI 10.1007/978-3-642-02190-9

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: Applied for

Typeset & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

Data management comprises all the disciplines related to managing data as a valuable resource of knowledge. This volume is focusing on its recent advances in five major directions: Information Retrieval & Web Intelligence, Knowledge Discovery, Information Integration & Data Security, Intelligent Agents, and Data Management in Medical Domain.

The first part of the book contains five contributions in the area of Information Retrieval & Web Intelligence.

The chapter, written by P. Kolaczkowski and Henryk Rybinski, presents a novel approach to solving Index Selection Problem, a well known problem in the database area. Opposite to other known approaches, the method searches the space of possible query execution plans, instead of searching the space of index configurations. The space is searched by an evolutionary algorithm, and as a result the set of indexes used by the best query execution plans is obtained. The algorithm converges to the optimal solution, as opposite to greedy heuristics, which for performance reasons tend to reduce the space of candidate solutions, possibly discarding optimal solutions. Although the search space is very large and grows exponentially with the size of input workload, searching the space of the query plans allows to direct more computational power to the most costly plans, thus yielding very fast convergence to ”good enough” solutions.

In the Chapter titled "Integrated Retrieval from Web of Documents and Data", K. Thirunarayan and T. Immaneni present a unified web model that integrates the HTML Web and the Semantic Web, formalizing the connection between them. Their hybrid query language to retrieve documents and data improves recall for legacy documents and provides keyword-based search capability for the semantic web. The prototype system SITAR implements their approach inlcuding the novel wordset pair queries.

The third chapter, written by S. Zadrozny and J. Kacprzyk, concerns the difficulty of expressing user requirements (information needs) in standard query languages. Authors show how the use of fuzzy logic makes possible to model and properly process linguistic terms in queries. They look at various ways of how to understand bipolarity in database querying, propose fuzzy counterparts of some crisp approaches and study their properties.

In the next chapter, T. Andreasen and H. Bulskov present an approach where conceptual summaries are provided through a conceptualization as given by ontology. The idea is to restrict a background ontology to the set of concepts that appear in the text to be summarized and the same provide a structure, called instantiated ontology, that is specific to the domain of the text and can be used to

condense to a summary not only quantitatively but also it conceptually covers the subject of the text.

The fifth chapter is written byY. Yao and N. Zhong and concerns the World Wide Web which due to its huge scale and complexity, one may find extremely difficult to search for simple theories and models for explaining it. Clearly, more complicated theories and methodologies are needed, so that the Web can be examined from various perspectives. There are two purposes of this chapter. One is to present an overview of the triarchic theory of granular computing, and the other is to examine granular computing perspectives on Web Intelligence (WI).

The second part of the book contains four contributions in the area of Knowledge Discovery.

The purpose of its first chapter is to explore how visualization techniques can enhance classifier evaluation. To this day, the most commonly used evaluation tools in machine learning are scalar metrics (e.g., Accuracy, AUC, RMSE, Precision, Recall etc.) which assign a single performance value to a classifier. Such metrics are very practical since they allow the user to easily rank classifiers according to their performance. At the same time, however, they are not as informative as could be because they summarize a lot of information in a single value (sometimes they even summarize the performance of a classifier on many different domains through averaging). To counter this issue, researchers often include the results obtained by different metrics on different domains. This, however, gives rise to a lot of information that may be difficult for human being to process. Rather than data-mining these results, authors decided to explore what visualization techniques can do for us.

They present a case study demonstrating the kind of information that can be gathered about classifiers using visualization approaches. In particular, they show the strength of their approach when applied to various classifiers on different domains and on multiclass domains. They also show how their visualization tool can allow us to analyze the results with respect to the domains' characteristics, answering questions that cannot usually be answered by other evaluation methods.

The second chapter, written by J. Stefanowski and S. Wilk, deals with inducing rule-based classifiers from imbalanced data, where one class (a minority class) is under-represented in comparison to the remaining classes (majority classes). Authors discuss reasons for bias of standard classifiers, in particular rule-based ones, toward the majority classes resulting in misclassification of examples from the minority class. To avoid limitations of sequential covering approaches, which are commonly applied to induce rules, a new approach to improve sensitivity of a rule-based classifier is presented. It involves modifying the structure of sets of rules, where for the majority classes minimal sets of rules are still induced, while rules for the minority class are generated by the EXPLORE algorithm. This algorithm produces rules being more general and supported by more learning examples than rules from a minimal set.

The trird chapter provides a comprehensive list of ways of organizing state sequences as well as event sequences. The proposed methods are semi-automatic allowing the user to control the conversion process. It is shown how by specifying

one or two reasonably sized matrices we can, for states to events conversions associate a set of events with the transition between any pair of states, or, for events to states transformations impose restriction mechanisms to limit the number of different states generated.

The fourth chapter, written by J. Rauch, deals with the GUHA method of mechanizing hypothesis formation. It can be seen as one of the first data mining strategies. Applications of modern and enhanced implementation of GUHA confirmed the generally accepted need to use domain knowledge in the process of data mining. Moreover it inspired considerations on the application of logical calculi for dealing with domain knowledge in data mining. The author of this chapter presents these considerations.

The third part of the book contains three contributions in the area of Information Integration and Data Security.

In its first chapter, authors discuss conflicts during integration of security solutions with business solutions covering the wide spectrum of social, socio-technical and purely technical perspectives. The investigated recent approaches for automated detection of conflicts are also discussed in brief. The ultimate objective is to discover the best suited approaches for detecting conflicts by software developers. It spans over approaches from cryptographic level to policy level weaving over the feature interaction problem typically suited for software systems. The assessment of these approaches is demonstrated by a remote healthcare application.

The second chapter deals with a monitoring-based approach for privacy data management. In the last ten years, the service oriented paradigm for information systems development has emerged as a powerful solution to provide seamless integration between organizations that provide their services as web-enabled software services (e.g., Web Services). The open world perspective enables services to collaborate and interact in highly distributed environments, cutting across the boundaries of various organizations (including enterprises, governmental and non-profit organizations), accessing, querying and sharing data. At the beginning, the interest of researchers and practitioners has converged on the functional aspects of those software services and their description. Because of the increasing agreement on the implementation and management of the functional aspects of those services, such as the adoption of Web Service Description Language (WSDL) for service description, Simple Object Access Protocol (SOAP) for Web service messaging, or Web Services Business Process Execution Language (WS-BPEL) for Web service composition, the interest of researchers is shifting toward the 'non-functional' or quality aspects of web-enabled services including security, privacy, availability, accessibility, etc. Hence, as the amount of exchanged information exponentially grows, privacy has emerged as one of the most crucial and challenging issues and is today one of the major concerns of users exchanging information through the web.

The third chapter, written by D. Maluf and C. Knight, describes the conceptualization, design, implementation and application of an approach to scalable and cost-effective information integration for large-scale enterprise information management applications. This work was motivated by requirements in the United States National

Aeronautics and Space Administration (NASA) enterprise where many information and process management applications demand access to and integration of information from large numbers of information sources (in some cases up to as many as 50 different sources) across multiple divisions and with information of different kinds in different formats.

The fourth part of the book contains three contributions in the area of Intelligent Agents.

In the first chapter of this section authors present a survay of current trends in pervasive environment management through database principles. The main components of their ongoing project SoCQ, devoted to bridging the gap between pervasive computing and databases, are sketched.

The second chapter, writen by E. Suzuki, describes a novel design method of swarm robots based on the dynamic Bayesian network. The method makes a principled use of data with a probabilistic model and it is expected to lead to a reduced number of experiments in the design. A simple but a real life example using two swarm robots is described as an application.

The last chapter of this section, written by H. Prade, provides an introductory survey to possibilistic logic, which offers a bipolar representation setting for the qualitative handling of uncertainty and preferences, and its application to information fusion and multiple-agent systems; it also outlines a treatment of uncertainty in databases.

The last part of the book contains three contributions in the area of Data Management in Medical Domain.

The first chapter, written by P. Berka and M. Tomeckova, describes step-by-step the process of building a rule-based system for classifying patients according to the atherosclerosis risk. They build the set of rules in two steps. First, they create the initial set of rules from data using machine learning algorithm called Kex. Then, they refine this set of rules according to suggestions of domain expert and according to further testing. Finally they describe the rule based expert system shell called Nest.

The second chapter is written by M. Kwiatkowska, M. Riben (epidemiologist), and K. Kielan (clinical psychiatrist). Authors revisited the definition of imprecision and closely related concepts of incompleteness, uncertainty, and inaccuracy in the context of medical data. They describe syntactic, semantic, and pragmatic aspects of imprecision, and they argue that interpretation of imprecision is highly contextual, and, furthermore, that medical data cannot be decoupled from their meanings and their intended usage. To address the contextual interpretation of imprecision, they present a representational framework for knowledge-based modeling of medical data. This new framework brings together three approaches to representation of medical concepts: a semiotic approach, a fuzzy-logic approach, and a multidimensional approach.

The third chapter of this section deals with information retrieval from a large volume of biomedical literature such as PubMed. It is important to have efficient passage retrieval systems that allow researchers to quickly find desired information. Aspect-level performance means that top-ranked passages for a topic should cover diverse aspects of that topic. Authors propose the HIERDENC text

retrieval system that ranks the retrieved passages, achieving efficiency and improved aspect-level performance over other methods.

We wish to express our thanks to all authors who contributed the above eighteen chapters to this book.

April 2009

Z.W. Raś
A. Dardzińska

Contents

Part I
Information Retrieval and Web Intelligence

Automatic Index Selection in RDBMS by Exploring Query Execution Plan Space*

Piotr Kołaczkowski and Henryk Rybiński

Abstract. A novel approach to solving Index Selection Problem (ISP) is presented. In contrast to other known ISP approaches, our method searches the space of possible query execution plans, instead of searching the space of index configurations. An evolutionary algorithm is used for searching. The solution is obtained indirectly as the set of indexes used by the best query execution plans. The method has important features over other known algorithms: (1) it converges to the optimal solution, unlike greedy heuristics, which for performance reasons tend to reduce the space of candidate solutions, possibly discarding optimal solutions; (2) though the search space is huge and grows exponentially with the size of the input workload, searching the space of the query plans allows to direct more computational power to the most costly plans, thus yielding very fast convergence to "good enough" solutions; and (3) the costly reoptimization of the workload is not needed for calculating the objective function, so several thousands of candidates can be checked in a second. The algorithm was tested for large synthetic and real-world SQL workloads to evaluate the performace and scalability.

1 Introduction

Relational Database Management Systems (RDBMS) have been continuously developed for more than three decades now and became very complex. To administer them, much experience and knowledge is required. The costs of employing professional database administrators are often much higher than the costs of database software licensing [12]. The total administration costs are especially large for large databases containing hundreds of tables and executing millions of queries a day.

Piotr Kołaczkowski
Warsaw University of Technology
e-mail: pkolaczk@ii.pw.edu.pl

Henryk Rybiński
Warsaw University of Technology
e-mail: hrb@ii.pw.edu.pl

* The work has been granted by Polish Ministry of Education (grant No 3T11C 002 29).

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 3–24.
springerlink.com

Recently we observe a high demand on solutions reducing these costs. Especially intelligent, automatic tools for solving complex administration problems are very helpful. One of such complex problems is performance tuning. In this paper we consider the aspect of proper index selection, which often significantly affects the overall database application performance. The importance of proper index selection increases with the size of a database.

Indexes are used by the RDBMS to accelerate query processing. Below we illustrate some cases, where they are especially useful:

- A query fetches only a small fraction of tuples stored in the database, determined by predicates with small selectivity:

```
SELECT * FROM person WHERE person_id = 12345;
SELECT * FROM person WHERE age > 100;
```

- Tuples should be returned in the same order as an order defined by an index. For example if a B+ tree index on the column `birth_date` is present, it can be potentially used for executing the query:

```
SELECT * FROM person ORDER BY birth_date LIMIT 10;
```

  Using indexes for ordering tuples may also be sane when the query requires sorting as a preparatory step, e.g. before joining relations, or grouping and aggregating tuples.
- A query requires joining two relations - a small one with a huge one. Then it may benefit from an index on the primary or foreign key of the latter one:

```
SELECT * FROM person p
  JOIN departement d ON (p.dept_id = d.dept_id)
  WHERE person_id = 12345;
```

- A query processes a subset of columns that is totally contained in the index, so that accessing the table can be avoided. Index-only scans are usually much faster than table scans, because indexes are usually smaller than tables. Besides, the physical order of tuples is rarely the same as the order of tuples in the index, so avoiding the table access also reduces large amount of costly random block fetches.
- A query is best handled by some dedicated kind of index, e.g. it contains a full-text-search, spacial search, skyline computation etc. These cases require some extensions to the SQL and are not covered by this paper, though the presented methods can be easily extended to support these cases.

There can be usually more than one execution plan available for a single query. These different plans may use different indexes. Each plan may use more than one index at a time, but also a single index may be used in several plans. The physical ordering of rows in the table often affects gains the application has from using a given index, so some database systems enable creation of the so called *correlated* or *clustered* indexes, which force the table rows to have the same ordering as the ordering of the index. There can be at most one clustered index per table.

One should note, however, that indexes induce an extra maintenance cost, as their existence may slow down inserting, deleting and updating the data. Additionally, the chosen indexes require some storage space, which may be limited. Therefore one can pose the following problem:

> Find such an index set, that minimizes the cost of processing of the input workload and that requires less storage space than the specified limit.

The problem is known in the literature as *Index Selection Problem* (ISP). According to [10] it is NP-hard. Note that in practice the space limit in the ISP is soft, because databases usually grow, thus the space limit is specified in such a way that a significant amount of storage space remains free even if the limit were totally used. Therefore, slightly exceeding the storage space limit is usually acceptable if only this provides strong performance improvements.

The ISP problem resembles the well known *Knapsack Problem* (KP). However, it is more complex, because the performance gain of a candidate index depends on the other indexes in the index configuration, while in the traditional KP the values and weights of items are independent of each other. Additionally, the process of calculating the total gain of the given candidate index configuration is computationally costly. Actually, given a candidate index configuration, it is necessary to calculate the optimal plan and its cost for each query in the input workload. To cope with the complexity, most researchers concentrated on efficient greedy heuristics that select a small subset of possibly useful index candidates in the first phase and then use some kind of heuristic search strategy to find a good index configuration within the specified space limit in the second phase [3, 8, 21, 24, 25, 26]. The essential disadvantage of this two-phase process is that by aggressive pruning the result space beforehand, one may remove the optimal solution and possibly also some other good solutions from the searched space. It is illustrated in Fig. 1, where the optimal solution denoted by the black triangle is located outside the searched space. To this end, we concentrate in the paper on finding a method that converges to the global optimum in a continuous, one-phase process.

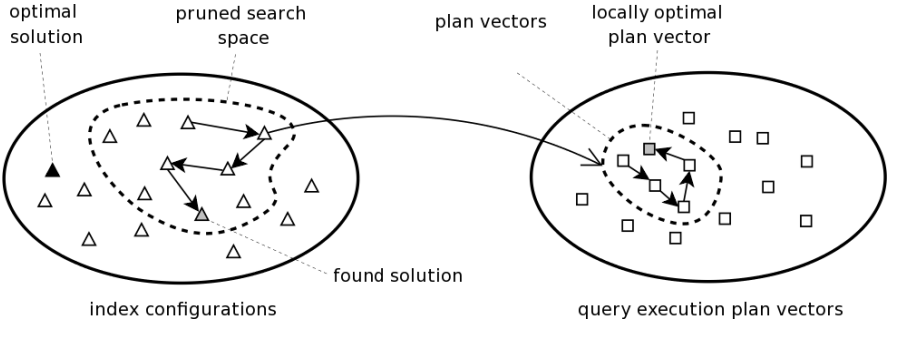


Fig. 1 Heuristic search of the index configuration space

Let us note that the index configuration being the global optimal solution of ISP determines the optimal set of the query execution plans, obtained for the input query workload. Also, the globally optimal set of plans determines the optimal index configuration. Thus we can formulate a *dual problem to ISP*:

> Among all possible sets of query execution plans for the given workload, find the one that minimizes the sum of query processing times plus the sum of maintenance times of used indexes and utilizes the indexes of a total size that does not exceed the specified storage space limit.

The method proposed in the paper solves the dual ISP problem, and gets the final index configuration from its result set of query plans. Because the search space is extremely large even for a small number of queries, an evolution strategy is used to drive the search. The approach is illustrated in Fig. 2. Here, the optimizing process consists in searching the space of plan vectors (by means of an evolution strategy). Each candidate plan vector has the relevant index configuration directly assigned. To this end, in addition to a good global query plan, a set of picked indexes is returned as a result. Our algorithm proposes also clustered indexes and takes index maintenance costs into account.

Evolutionary strategies and genetic algorithms were recognized as feasible to solving ISP [11]. It was also shown that they can be superior to some other heuristic search strategies for this application [18]. However, the previous works concentrated on searching the index space, rather than the plan space, and this disallowed to focus more on the optimization of costly query plans. We believe that focusing on the most costly plans is crucial for achieving high performance of the index selection tool and makes it possible to solve larger problems. The evolutionary strategy approach has also an advantage over other heuristic search strategies, such as simulated annealing [15], tabu-search [13] or hill-climbing [23], by the fact that the fitness function and the problem and solution spaces may dynamically change while the problem is being solved and, if only the mutation intensity is high enough, the population will follow the changes and converge to the new solution without the need to restart the whole process. This makes the evolutionary approach ideal for autonomic, self-tuning database systems.

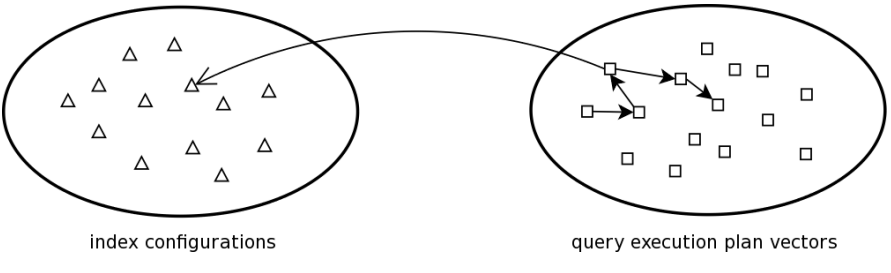


Fig. 2 Heuristic search of the query execution plan space

The method was inspired by our observations of professional database administrators. We were wondering, how they tune their database systems. At first, they usually investigate the most costly queries. These are the queries that require a long time for each execution and/or are executed very frequently. Concentrating on these queries can lead to significant performance gains with only a little effort on the optimization, as usually the number of costly queries is small relative to the number of all queries. Then, they display a plan for each query by using tools that come with the database management system (usually called `EXPLAIN` or similar), and try to figure out, why the plan is bad and how it can be improved. When they know, how the query should be optimally processed, they create all needed structures like indexes or materialized views and eventually give some hints to the query optimizer to make it choose their plan. Of course, sometimes the administrator's best plan is not the optimal one, or the database query planner does not do what it is expected to, so for complicated queries some trial and error method is used. However, less experienced admistrators tend to jump right to the trial-and-error phase, creating various possibly useful structures and checking whether they improve the query execution times or not.

The contributions of this paper are the following. First we describe a novel algorithm for solving the ISP in the query plan space. Second, we provide a theoretical proof of convergence of the method to the optimal solution and analyze its other properties. Third we demonstrate the results of experiments showing that the algorithm works correctly and achieves higher performance and better results than the state-of-art commercial index advisor.

2 Related Work

There are many papers that refer to the problem of finding an optimal configuration of indexes. The research on ISP has been conducted since mid seventies of the previous century. In order to reduce the solution search space, already in [7] some simple heuristics have been proposed. Several papers [2, 9, 14, 29] recognize ISP as a variation of a well-known Knapsack Problem and propose some heuristic methods to solve it. However, none of them uses a query plan optimizer to estimate gains of the selected indexes nor to propose index candidates.

In [10] a method is described that suggests various index configurations, and uses an R-system query plan optimizer to evaluate their gains. The candidate configurations are generated by heuristic rules, based on the SQL queries in the workload, so the whole solution is quite complex. The authors of the Index Tuning Wizard for MS SQL Server [8] took a very similar approach, but they used simpler heuristics for the index candidate selection. They start from the indexes on all indexable columns used in each query, and later prune this set by choosing only the best index configuration for each query.

The idea of employing the query plan optimizer was further extended in [26]. In this method, the query plan optimizer not only evaluates the gains of possible index configurations, but also generates the best index configuration for each query. This technique reduces the number of required calls to the query plan optimizer. Only

one call is required to get the best index configuration for a given query. The final solution is evaluated by transforming the ISP problem to the Knapsack Problem and applying a simple heuristic. Our solution is somehow based on this idea. However, our optimizer does it for many queries in parallel, and it is not limited to selecting locally optimal index configurations for one query.

In some approaches [4, 5, 6, 17, 18] the authors concentrate on the ISP problem itself, independently of its database environment. These methods assume the gains of indexes or their configurations as explicitly given, and treat queries and indexes as set elements, without diving into their structure. Optimal solutions for such defined synthetic problems involving several thousands of queries and candidate indexes have been reported. However, none of these papers analyzes the problem of generating the candidate index configurations or evaluating their gains for a real-world application.

Some local search heuristics have been applied to solving a variation of ISP where the workload and data can change over time [24, 25]. Candidate indices are proposed by the query optimizer as in [26]. Each candidate index is given a rank based on the total performance gain it causes. The performance gain is estimated by analyzing selectivity of the query predicates and by the query plan optimizer. Old queries influence the total index gain less than the recent ones. Given a ranking of indexes, the result index set is chosen by heuristics that solves a Knapsack Problem.

The algorithms from [8, 26] usually solve some variations of the Knapsack Problem by starting from an empty index set and adding candidate indexes to it, until the space limit is exceeded. Another approach has been presented in [3]. Here, the optimization process starts from the set of indexes and materialized views that generate the highest total performance gain for the `SELECT` queries in the workload. Then, the elements are removed from this set or merged together as long as the space constraint is violated. The method uses the query optimizer to evaluate the gains of the candidate structures. Some heuristics for reducing the number of calls to the query optimizer are discussed. These heuristics were further improved in [21] in such a way that they do not incur query cost misestimations while requiring 1.3–4 times less optimizer calls than the original method [3] thanks to reusing some parts of the locally optimal plans. The improvement can be used by any "what-if" ISP solver, allowing it potentially to examine much more candidate index configurations, and yielding better final results.

In the papers [4, 17, 20] the ISP problem is formulated as an *Integer Linear Programming* problem. The authors use an exact algorithm based on branch-and-bound with various kinds of constraint relaxation to solve it. Unlike the heuristic approach, this class of methods is able to give an exact upper bound on how far the actual solution is from the optimum. The performed experiments have shown that these methods yield results with acceptable error for the ISP instances of practical size very quickly, though finding the exact optimum usually requires many hours of computation. However, the ILP approach requires additionally running a preparatory step for a careful candidate index set selection, just as the previously mentioned heuristics ([4, 5, 6, 17, 18, 24, 25, 26]). Feeding all the numerous possible index configurations to the ILP solver is impractical, as it could easily create ILP instances having much too many variables and constraints to be solvable in a

reasonable amount of time [20]. On the other hand, pruning this set too much may cause good solutions to be missed. The accurate candidate index selection itself is a complex problem and some work has been done in the context of the OLAP workloads [27], but to our best knowledge it remains open for the general case.

3 Formal Problem Definition

The database system processes a set of n tasks $Q = \{q_1, q_2, \ldots, q_n\}$. Each task can select, insert, update or delete data. For simplicity, whenever we use word *query*, we also mean the tasks that modify data. By D we denote a set of all possible indexes that can be created for a given database, whereas certain D_x, $D_x \subseteq D$, will be called index configuration. Each query can be executed according to its *query execution plan*. The query execution plan consists of instructions for the *query executor*, along with a list of indexes which are to be used while performing the query. The formalism below describes this mechanism.

By P_Q we denote a space of all possible query execution plans that can be generated for the queries from Q, assuming any index from D is available. As in our optimization problem Q is well defined and does not change, in the sequel we simply write P. Let $P(q) \subseteq P$ be a set of all possible query execution plans for the query q, $P(q) = \{p_1, p_2, ..., p_m\}$ and $P(q, D_x)$ be a set of plans valid for the D_x index configuration.

Given a configuration D_x and a query $q \in Q$, the query planner generates a plan according to the function $\pi : Q \times 2^D \rightarrow P$. An ideal planner would optimize the execution of q and provide the plan $p^*_{q,D_x} = \pi(q, D_x)$, $p^*_{q,D_x} \in P(q, D_x)$, which minimizes the execution cost of the query q. Each plan p has an associated *execution cost* $\text{cost}(p, D_x)$. Let us note that for a plan p there is a minimal subset of indexes $D_l(p)$, which are indispensable for this plan. Obviously $D_x \supseteq D_l(p)$. The index maintenance costs are included in the cost calculations, so the cost of a given plan may depend on the other indexes, not used by the plan itself. The objective is to find such a subset D_x of D that minimizes the total cost of the query executions:

$$\min_{D_x \subseteq D} \sum_{q \in Q} w_q \text{cost}(p^*_q, D_x) \tag{1}$$

$$\text{size}(D_x) \leq s_{\text{max}} \tag{2}$$

where $\text{size}(D_x)$ is the total size of the used indexes, s_{max} is the size limit and w_q is a weight of query q in the input workload, usually related to the frequency of that query.

4 The Algorithm

For searching the space of index configurations we use a standard steady-state evolution strategy with a constant population size, a constant mutation rate, and neither elitist subpopulation, nor crossover.

Individual Representation and Fitness Function

According to the cost function (1), a naïve approach would consist in directly seeking for a subset $D_x \subseteq D$, such that the value of the total cost is minimal. However, this would involve calling the query optimizer for every individual index configuration D_x possibly many times, which is a very costly operation. Actually, many calls would be performed only to find out that the new index configuration does not change the best plans, even though only a small subset of queries actually needs to be reoptimized at every D_x change. Note, that the number of various possible index configurations used by a single query grows exponentially with the number of query predicates and joins. This problem was thoroughly discussed in [3].

In our approach, each individual in the population is a vector of query execution plans $v = [v_1, v_2, \ldots, v_n]$ where $v_i \in P(q_i)$ for $i = 1 \ldots n$. Each plan can use any indexes from the set D. We minimize a function:

$$\min_{v} \sum_{i=1}^{n} [\text{cost}(v_i, D_x) + g(D_x)], \tag{3}$$

where $g(D_x)$ is a penalty for exceeding the size limit s_{max}:

$$g(D_x) = \max\{0, \alpha[s(D_x) - s_{\text{max}}]\} \tag{4}$$

The value of α can be set large enough to make exceeding the size limit non-profitable. The set D_x is a function of the vector v:

$$D_x(v) = \bigcup_{i=1}^{n} D_l(v_i) \tag{5}$$

Note that the set of possible v values is a superset of the set of all possible plan configurations established during the optimization process of the cost function (1), because v may contain locally suboptimal query plans.

The indexes used by v are also kept as a part of the representation of an individual. Each index has an associated reference counter so that it is easy to tell, when the index is not used any more and can be safely discarded.

Evolution Process

Now we can describe in more detail the evolution process. The optimization is performed as follows:

1. a population is initialized with a single plan vector containing optimal plans for the existing set of indexes in the database;
2. in each iteration:
   a. a random individual is selected with a probability determined by its fitness;
   b. the reproduced individual is mutated;

c. the mutated individual is added to the population;
d. if the maximum size of the population is exceeded, a random individual is removed from the population.

To reduce the overhead of copying large query execution plans and index metadata on each individual reproduction phase, the plans and indexes are lazily copied. The actual copying is done only for small parts of the vector that is mutated.

Selection

For the selection phase we use a simple tournament selection. The selection pressure can be adjusted by one parameter $e \in \mathbf{R}^+$ that influences the tournament size and the probability of worse individual being selected over a better one. The exact procedure is performed as follows:

1. Let $e' := e$.
2. Select an individual v randomly with a uniform probability distribution.
3. While $e' > 0$, repeat:

   a. Select an individual v' randomly with a uniform probability distribution.
   b. If v' is better than v, assign $v := v'$ with probability $e' - \lceil e' - 1 \rceil$.
   c. Assign $e' := e' - 1$.

4. Return v.

For $e = 0$ we get a uniform selection. For $e = 0.5$ we get a 2-way tournament, where the best individual has 50% chances to win. For $e = 1$ we get a classical 2-way tournament where the best individual always wins. The larger e, the larger is the selection pressure.

Mutation

The basic mutation of an individual v consists of applying some small *atomic transformations* to a randomly chosen subset of plans in v. Every transformation guarantees that the transformed plan is semantically equivalent to the original one. If this cannot be satisfied, it fails and a transformation of another class is chosen. There is a fixed number of atomic transformation classes.

- The *join reordering* transformation changes the execution order of two random adjacent join operators. The transformation may fail for some coincidence of joins of different types (inner, left outer, right outer, full).
- The *join algorithm change* transformation chooses a different algorithm for joining two relations. So far, our experimental optimizer handles a few well known join algorithms: a *block nested loops join*, an *index nested loops join*, *sort merge join* and a *hash join*. None of these algorithms is the best one in all possible situations, so selecting a proper one is essential in creating a good plan. For the index nested loops join, a new index may be introduced as well as some index

introduced earlier for another plan can be reused. The transformation cannot introduce an index identical to any of the indexes used by the other plans.
- The *table access change* modifies leaves of the plan by choosing one of the following relation access methods: full table scan, full index scan and range index scan. This transformation may introduce new indexes, as well as, may choose any index already used by some other plan, and eventually extend it with more columns. The column set of the index is selected from the columns in the query predicates, columns used in the `GROUP BY` or `ORDER BY` clauses, or all columns used in the plan.

A *mutation of the plan* consists of at least one successful atomic transformation. All the transformation classes are allotted with the same probability. The number of successful single plan transformations k_t performed per mutation is a random number, generated with a probability distribution, such that numbers near 1 are the most probable. This makes most of the mutations small and enables the algorithm to perform *hill climbing*. However, it is essential that sometimes the mutation is large, in order to avoid premature convergence to a locally optimal solution and to ensure the whole search space is explored. This we achieve by using the positive part of a Gaussian distribution:

$$K(\sigma) = \lfloor |N(0, \sigma^2)| + 1 \rfloor \tag{6}$$

The number of plans k_p mutated in each individual in one iteration of the evolutionary algorithm is calculated from the same distribution (6). The algorithm was insensitive to the exact setting of σ.

The plans to be mutated can be selected in one of the following ways:

- The probabilities of selecting each of the plans are equal. We will call it *uniform selection*.
- The probability of selecting a plan is proportional to its cost. We will call it *proportional selection*. This type of selection makes mutating costly plans more frequent than the cheap ones and greatly increases the convergence rate of the algorithm. However, in some rare cases a very cheap plan may be indirectly responsible for a high total cost due to requirement of the index with a high maintenance cost caused by frequent updates. In such cases, the uniform plan selection may be better.
- Uniform or proportional selection is chosen with probability of 0.5. We will call it *mixed selection*. This seems to have advantages of both the uniform and proportional selection. The costly plans are mutated much more often than the cheap ones, while each of the cheap plans is guaranteed a mutation rate only 2 times lower than it is for the uniform plan selection.

After completing the mutation phase, a cost of the new plan is estimated. Due to the fact that mutations are often local, there is usually no point in recalculating the costs of most of the nodes in the query plan tree. The transformation operators mark only changed parts of the trees and all their ascendants to be recalculated. For instance, changing the table access method in one of the leaves of the plan does not cause

changing the costs of other leaves. This partial plan caching technique is somehow similar to that used in [21], but may be more aggressive due to the fact that the result plan need not be optimal. For large plans, e.g., as the ones from the TPC-H benchmark [28], the partial plan caching improves the performance of the mutation step by 3 to 10 times. In the future, this technique can be improved to also avoid recalculating the ascendants of the changed parts in most cases.

Replacement

When the population reaches a defined size, the mutated individual replaces a randomly chosen individual from the population. The first version of the algorithm made uniformly a choice of individuals to be replaced. However, this often led to replacing some very good individuals with bad ones, and the algorithm converged very slowly. Therefore for the selection of individuals to kill, we use the same tournament selection scheme as we use in the selection of individuals for mutating, though "in the opposite direction" – this time the worse individual has higher chances to be chosen. Thus, the good individuals are very unlikely to be removed from the population pool, but still the probability of removing them is slightly greater than zero.

Clustered Indexes

The basic method of plan transformations can advise only non-clustered indexes. The transformations do not introduce new clustered indexes. It would be tempting to allow them to do so, but this could result in a possibility of having more than one clustered index on the same table, because the transformations are performed independently of each other. It is possible to add additional checking code for each mutation to avoid this, but we decided to make the process of introducing clustered indexes as a separate mutation, so that its rate could be controlled separately.

We extend the mutation procedure with a possibility to *mutate indexes* directly by means of changing the types of used indexes D_x, so that the restriction for at most one clustered index per table always holds. The complete mutation procedure is as follows:

1. Generate a random number k_t using the formula (6).
2. Repeat k_t times:
   a. with probability $(1 - \mu)$ mutate a random plan in v;
   b. with probability μ change a random index in D_x from non-clustered to clustered or the other way around. If needed, change one index from clustered to non-clustered, so that there is at most one clustered index per table. Recalculate the costs of all plans that use the changed indexes.

The μ parameter is usually a small positive number. We used in our experiments $\mu = 0.05$. The higher value of μ may cause extra overheads of the algorithm, because the index type change requires calculating costs of many plans, so it significantly increases the average computational cost of a single iteration. One index may be used by several queries. Thus, modifying one index type requires more CPU cycles

than modifying a single plan. Besides, performing index mutations too often does not decrease the total number of iterations that must be executed to achieve good results.

Redundant Indexes Problem

In some cases the proposed algorithm does not converge to the good solution quickly, and often it proposes overlapping, redundant indexes. Consider m semi-identical queries in the workload in the sense they are querying one table and having 2 predicates A_1 and A_2 with the same selectivity. There may be two optimal two-column indexes for each query, differing only with the order of columns in the index key. Each index alone would reduce the cost of each query by the same amount. The optimal solution should contain only one such index, assuming the considered indexes are not useful for any other query in the workload. Unfortunately the random nature of the algorithm makes it unlikely that the same optimal plan for all m queries is chosen, especially if m is large. Probably approximately half of the plans would use the index on (A_1, A_2) and the other half – the index on (A_2, A_1). Note that changing a single query plan to use one index over another does not change the fitness of the solution, and the objective function forms a plateau here. Without a special handling of this situation, the solution might contain both of the indexes and the maintenance cost would be higher than actually required.

To this end we introduce yet another type of individual mutation that removes a random index from the solution by transforming a set of plans that use it. The removal of index usage is performed by the standard plan transformations: (1) the join algorithm change to replace index nested loops joins, and (2) the table access change to change the index used in the index scans. This time the transformations are not random – they are given an exact tree node to operate on. If there exists an alternative index in D_x, which gives similar benefits to the plans as the removed index, it is used. There is no need to perform this kind of mutation as frequently as the other types of mutations, due to the fact that redundant indexes are not created very often. We set the probability of triggering this type of mutation in each iteration to 0.01, and it turns out to be sufficient.

5 Analysis

The presented algorithm resembles the *evolution strategies*(ES), as presented in [1]. It was proven that the ES algorithm with a mutation operator using Gaussian distribution guarantees finding the optimum of a *regular* optimization problem. However, the algorithm in [1] represents each solution as a vector of real numbers, but the domain of our problem is non-linear and each solution is represented by a vector of trees. Thus, our optimization problem does not meet the requirements to be regular, as stated in [1], and the convergence theorem cannot be applied.

Fortunately, the domain of our problem can be represented by a graph, where each vertex represents a single query plan vector and edges connect solutions (states)

that can be directly transformed into each other by applying a single atomic transformation (as described in Section 4). Each mutation step in our ES can be viewed as a random walk on this graph. Due to the fact that the following holds:

- the probability of each transformation is stationary and greater than 0,
- the current solution depends only on the preceding solution and transformation chosen,
- the graph is finite and connected,

the process forms a finite-state Markov chain [19]. Additionally, each transformation has a corresponding co-transformation that brings the individual back to its original state. Thus, there are no absorbing or transient states in the system, and the Markov chain is irreducible (but not necessarily ergodic). This guarantees that by performing enough state transitions, each possible solution will be reached. Due to the fact, that the number of transformations in each mutation step given by the equation (6) is unlimited, each possible solution can be reached in a single mutation step of the ES, which guarantees finding the optimal solution after some (sufficiently large) number of iterations. Note, that for this to hold, the atomic transformations must be able to create any feasible solution. For instance, if the optimal solution required scanning an index with 10 columns, and there was no transformation that could introduce such 10-column index, the assumption that the solution graph is connected would not be met and the method would fail to find the optimum plan.

The average and pessimistic time complexity to find the global optimum is exponential with respect to the number of joins, predicates and used columns in the queries. Unless the complexity classes **P** = **NP**, this property is true for any ISP solver claiming global convergence. However, as shown further in the Section 6, the ES is capable of finding good solutions in a very short time even for large and complex ISP instances.

The time complexity of each iteration of the presented ES is as follows:

- the selection step requires selecting a constant number of random plans; if plans are stored in an array with random access, the complexity of the selection step is $O(1)$;
- the reproduction step requires copying the plan vector and indexes; even though it is lazily-copied, the complexity of this step is $O(n + l)$, where l is the number of indexes in the individual;
- the mutation step requires selecting some number of plans in the plan vector to be mutated and then actually performing the mutations on them; the average number of selected plans is constant; the complexity of each mutation is proportional to the number of atomic transformations applied to the plan and to the complexity of each transformation; the average number of atomic transformations is constant, while the complexity of the transformation is proportional to the number of nodes plus the number of predicates in the query plan tree; assuming the query plans are of limited size, the complexity of the mutation step is $O(1)$;
- the replacement (recombination) step is similar to the selection step and has the complexity $O(1)$.

The asymptotic complexity of the single ES iteration is $O(n + l)$, which means the algorithm will get slower the more queries in the workload are given and the more indexes are selected. However, for the typical number of queries and plans, the reproduction step would be fast enough not to become a bottleneck, as making a copy of an individual is a relatively simple operation. The most complex and computationally costly step is the mutation, requiring not only modifying some plans, but also estimating the costs of changed plan tree nodes.

The memory requirements for the algorithm are quite low in comparison to the algorithms that require the index configuration candidate selection step, as no numerous candidates need to be remembered. Actually, the memory is used for storing the population, i.e. the query plans and the selected indexes. Thus the required memory size is proportional to the number of queries in the input workload, final result size, and the user defined maximal number of individuals in the population.

6 Implementation and Experiments

Our first attempt at implementing our algorithm was to modify an existing query planner found in a widely used open-source RDBMS such as PostgreSQL or MySQL, so that it could use our approach to select indexes. However, at this time their planners lacked important features, like support for covering indexes or proper query rewrite engine. These features are essential for achieving high performance in complex analytical benchmarks. Even though these features could be added, still a lot of further effort would be required to implement the query execution plan mutation procedure in a foreign environment, probably by modifying large parts of the existing code. Partial query execution plan caching, as proposed in this paper in order to increase the performance of the index selection tool, would probably complicate things even more. Thus, we have implemented our algorithm as a standalone tool, separated from any RDBMS index selection tool, possibly duplicating some work already done in the existing query planners. To minimize the effort, we have decided to reuse as much concepts and code from the PostgreSQL database system as possible, including the formulas for cardinality and cost estimation, and directly accessing metadata gathered by the PostgreSQL's `ANALYZE` tool. PostgreSQL was chosen because it seemed to provide the most advanced query planner among the open-source solutions available. Moreover, we had access to its large and advanced commercial applications.

In order to be able to perform experiments using synthetic benchmarks, we have added missing support for index-only-scans, and improved the query rewrite engine, as well as, the PostgreSQL query execution cost estimation model, so that now it better matches the one of commercial database systems, in particular IBM DB/2.

The tool was entirely implemented in the Java programming language, with help of the ANTLR [22] parser generator for creating the SQL parser, JDBC API to import the PostgreSQL's metadata and the SWING library to create an easy to use user interface. The application with source code will be freely available for academic research.

The experiments were divided into two groups. The first group of experiments consisted in running the tool for a very small synthetic workloads and simple database schemas, to check if the optimal index recommendations are found. The results were compared with the results obtained from the DB/2 index advisor [26]. The second group of experiments consisted in running the tool for two different real-world transactional workloads, as well as, a selected set of TPC-H queries. Below we describe the results in more detail.

6.1 Simple Workloads

A single table *table* with 3 integer columns $c1, c2, c3$ and a workload consisting of 2 queries were sufficient to show the most interesting properties of the algorithm. Every column in this test had the same width of 4 bytes, and evenly distributed values in range $[1, 1000000]$. There were 1 million rows in the table.

One of the tests was to check, how frequent updates of one column might affect the index recommendations given by the algorithms tested. It consisted of two workloads. The first workload consisted of only one query q_A with the weight $w_A = 1$:

```
SELECT c1, c2 FROM table WHERE c1 < 1000 AND c2 < 2000;
```

The storage space limit was set enough high not to affect the result. The obvious recommendation given both by our tool and the DB/2 advisor was a covering index on (c1, c2). Adding an update statement q_B with a very large weight $w_B = 10000$:

```
UPDATE table SET c1 = <constant> WHERE c3 = 10;
```

changed the recommendations. Both index advisors recommended the index on the column (c3), to accelerate q_B. However, the DB/2 advisor did not provide any index to accelerate the query q_A, while our tool recommended a clustered index on the column c2. Such index is still much better than no index at all, and does not impose the index maintenance cost that would be required for the index on (c1, c2) due to the frequent updates q_B. Note, that such recommendation for q_A would be suboptimal if q_B did not exist in the workload or had a smaller weight. As the initial index candidate sets generated by the methods [3, 8, 26] are based on the indexes picked by the query planner for each of the queries separately, the recommendations given by these algorithms cannot be optimal for the whole workload in this case. Actually, the recommendation given by our tool is much better than the one of the DB/2 advisor, even according to the original query plan cost model, which is used by the DB/2 query planner.

A similar behavior was observed in the case of a single `Select` statement, and a certain storage size limit. Let us consider a query:

```
SELECT c1, c2 FROM table WHERE c1 = <constant>;
```

With a relaxed storage limit, both advisors recommended creating a covering index on $(c1, c2)$, so that it can be used to quickly locate tuples matching the query predicate. Then we set the storage limit slightly below the total size of the database with the recommended index, and repeated the optimizing procedure. This time our

tool recommended a smaller, clustered index on the single column c1, while the DB/2 advisor did not recommend any index. Obviously the fitness of both solutions differed by a factor of more than 500, because the DB/2 recommendation forces the query planner to choose a costly full table scan, which could have been easily avoided.

Another test has shown the value of index merging technique, originally introduced in [3], and used in our approach by the table access change transformation to create new index candidates. Consider two queries:

```
SELECT avg(c1) FROM table;
SELECT avg(c2) FROM table;
```

Due to the fact that answering these queries requires reading every tuple of the table, and assuming the database storage is row-based, the only possibility to accelerate the execution of these queries is to create covering indexes, so that one avoids reading not needed columns. Obviously, with no storage space limit 2 separate indexes, one on the column $c1$, and the other one on $c2$ constitute the optimal solution. This recommendation was given by the tested index advisors. However, if the storage space limit was reduced, so that one of the recommended indexes did not fit, the optimal solution would be an index on $(c1, c2)$ or $(c2, c1)$. This correct solution was given by our tool, in contrast to the solution given by the DB/2 advisor. The algorithm in [3] obviously should also find the optimal solution in this simple case.

6.2 Complex Workloads

For the experiments, we used two real-world server side transactional applications: a commercial multiplayer network game with 100,000 users and 0.7 GB of data in 108 tables (further referred to as MG) and a smaller mobile web application of one of the Polish telecom operators, using 33 tables (further referred to as WA). MG executed much more update statements than WA, which was mostly read-only (see Tab. 1). The two applications already had some indexes created by experts. MG was a mature application running in production for over 3 years, while WA was in its beta stage, and has been optimized very roughly by the application designers without looking at the workload. There existed indexes for primary keys and for some most often executed queries. Each type of measurement that we performed, was repeated 20 times, and the best, average, and the worst results were recorded.

The automatic index selector was used to select indexes for the copies of the databases in two different situations:

- hot start: after leaving the database ”as is”, without dropping any indexes
- cold start: after dropping all the indexes

Each workload consisted of 50000 SQL queries recorded from the database system log files. Because it took too long to estimate the cost of such a large workload and then to automatically select indexes, both workloads were compressed by a

Table 1 Characteristic of the workloads

Statement type	Share [%] MG	WA
Single-table `SELECT`	66.95	78.73
Multi-table `SELECT`	18.93	16.14
Aggregate `SELECT`	2.84	3.30
`INSERT`	0.92	1.83
`UPDATE`	12.71	0.98
`DELETE`	0.49	2.31

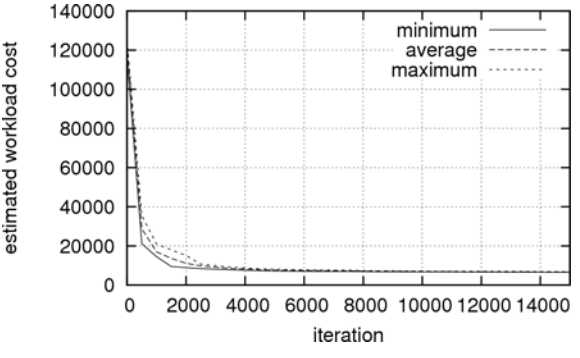


Fig. 3 Final workload cost as a function of number of iterations for MG database for cold start

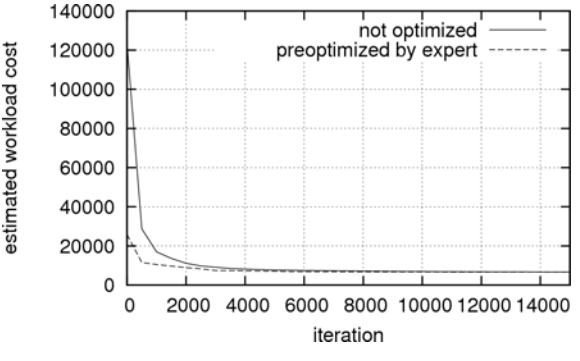


Fig. 4 Average convergence curves for hot and cold start

method presented in [16]. The final numbers of queries after compression were 289 for MG, and 62 for WA.

The algorithm was insensitive to a wide range of settings such as the population size and environment pressure. We have obtained good results for the populations of the size 10, as well as 80. Also setting the environment pressure e between 1.0 and 2.0 did not significantly affect the quality of the results and the speed of convergence. We present below the results for $n = 20$ and $e = 1.5$. The size limit $s_{\max}$ was set to 150% of the database size without indexes, but the limit was never reached

during the experiments. The size of the database with the materialized solution index configuration never exceeded 130% of the original database size.

We have observed a very fast convergence of the estimated workload cost. For the cold start, the first 500 iterations reduced the workload cost by over 80% for MG (Fig. 3). In the next 5,000 iterations, the workload cost dropped by 10% of the original workload cost, which was already very close to the cost obtained at the end of the experiment, after 150,000 iterations. For the hot start, the initial workload cost was much lower, so the performance improvement achieved by the index selection was not as large as for the cold start (Fig. 4), but again most of the improvement was achieved by the first few hundred iterations. For MG, our implementation did about 450 iterations per second on average on a single core AMD Athlon 3700+ processor. The test program was executed by the Sun Java 6 virtual machine with 250 MB of heap space. It was possible to use a 64 MB heap, but the performance was slightly lower – about 280 iterations per second. The compressed workload of WA was smaller, and our index selection tool was able to run over 3100 iterations per second. Moreover, it was possible to get "good enough" index configurations in less time than it took the optimizer to calculate the minimum cost of the whole workload for the one, fixed index configuration, which involved generating an optimal plan for each query in the workload.

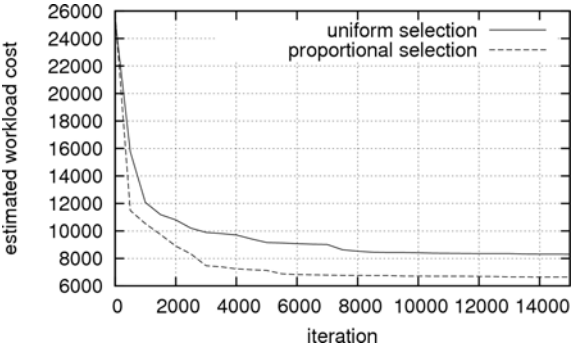


Fig. 5 Influence of the plan selection type on the average convergence speed

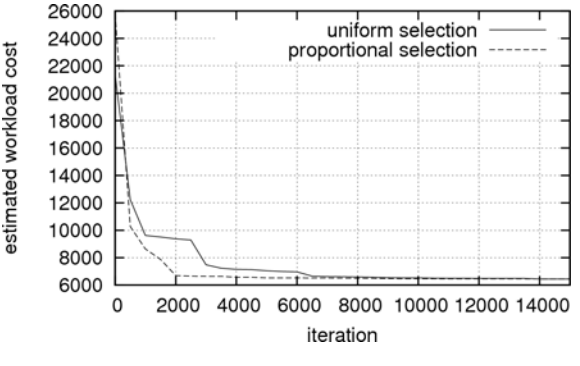


Fig. 6 Influence of the plan selection type on the maximum convergence speed

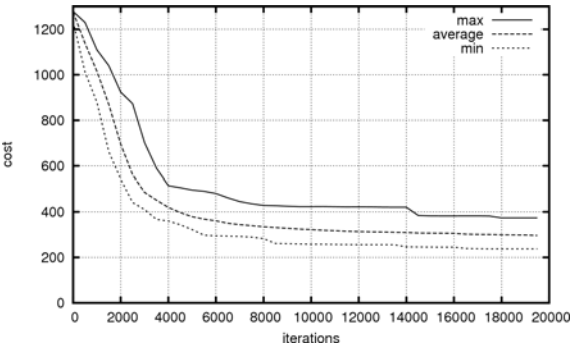


Fig. 7 Convergence curves for the TPC-H workload

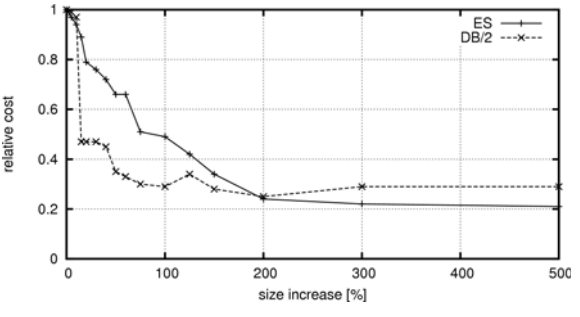


Fig. 8 Estimated costs of TPC-H workload as a function of the storage space limit

We have also checked how the plan selection algorithm affects the performance of the method. As a reference, we have implemented the uniform plan selection algorithm. The proportional selection gave much better average results over the uniform selection (Fig. 5). When using the uniform selection, sometimes running even 100,000 iterations did not produce optimal results. The best recorded convergence curve for the uniform selection was also much worse than that for the proportional selection (Fig. 6). Therefore we conclude the proper plan selection is crucial to achieving fast and stable convergence.

Additionally, to test if our approach is feasible also for analytical workloads, we used 20 different queries from the standard TPC-H database benchmark [28] and a sample TPC-H database generated with a factor of 0.1 (about 100 MB). For this experiment, the database contained only the primary key indexes at the start. In contrast to the workloads MG and WA, this test was performed both by the DB/2 advisor, and by our implementation of the presented ES algorithm. Obviously, due to the fact that the ES implementation is based on the cost model of PostgreSQL, it would be the best to compare it to a reference index advisor for PostgreSQL. Unfortunately, there is no such index advisor for any recent PostgreSQL version yet.

As for the MA and MG workloads, we measured the maximal, minimal and average convergence rate for 20 runs of our tool, 20000 iterations each (Fig. 7). The solution converged slower than in the case of the transactional workloads, but good

results were obtained in less than 6000 iterations. The performance was about 1500 iterations per second, and it took about 10 seconds to give almost optimal solution. This was similar to the times required for running the DB/2 advisor, which were between 5 and 12 seconds. We predict the performance of our tool, measured as a rate of executed iterations, can be much better, because no extensive code optimizations has been made yet and a lot of possibilities of improvement has not been tried. Due to the fact that (1) the query execution plans for the TPC-H workload are often large and contain many joins, (2) the plan transformations are often local, more aggresive caching of join cardinality and cost calculations can bring possibly essential benefits. The convergence rate in this experiment was rather insensitive to the exact settings of the parameters. In particular, the solution converged well for the population size set to 10 or 100, and the environment pressure set to 0.5 or 1.8.

The TPC-H workload required much more space for indexes than MG and WA. Without the storage space limit, the DB/2 advisor recommended indexes having a total size of over 200 MB, while our tool — of about 250MB. Reducing the storage size limit for the recommended indexes increased the final estimated costs of workload for both solutions (Fig. 8). The recommended index sets and their estimated benefits were similar, but not exactly the same in both solutions. . The observed differences were presumably caused by the query plan cost models that did not *exactly* match. Materializing the indexes recommended by our tool for the storage size limit of 40%, applied for the DB/2 database, improved the estimated workload cost by over 55%, but obviously not as much as the original DB/2 recommendation. The same situation was observed when applied the DB/2 recommendation to our cost model — the value of the objective function was better for our solution than for the DB/2 advisor's one.

7 Conclusions

We have presented a novel approach to solving the index selection problem. The main idea of the presented algorithm was that instead of seeking the index space we suggest to search the space of query execution plans. In order to avoid preliminary pruning we have decided to apply an evolutionary strategy. The experiments proved acceptable performance and good reliability of the developed algorithms for real-world index selection problems for medium-size relational databases and also feasibility of the approach for complex analytical workloads. We proved theoretically that the method converges to the global optimum. We also presented examples where our algorithm actually gives better solutions than the solutions given by the state-of-the-art heuristics which explore the index configuration space. The method does not require any phase for preparatory candidate index selection. In addition it shows very low sensitivity to parameter settings, therefore it can be especially useful in the implementations of self-tuning, autonomic database systems.

References

1. Back, T., Hoffmeister, F., Schwefel, H.P.: A survey of evolution strategies. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 2–9. Morgan Kaufmann, San Francisco (1991)
2. Barcucci, E., Pinzani, R., Sprugnoli, R.: Optimal selection of secondary indexes. IEEE Trans. Softw. Eng. 16(1), 32–38 (1990), http://dx.doi.org/10.1109/32.44361
3. Bruno, N., Chaudhuri, S.: Automatic physical database tuning: a relaxation-based approach. In: SIGMOD 2005: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 227–238. ACM Press, New York (2005), http://doi.acm.org/10.1145/1066157.1066184
4. Caprara, A., Fischetti, M., Maio, D.: Exact and approximate algorithms for the index selection problem in physical database design. IEEE Trans. on Knowl. and Data Eng. 7(6), 955–967 (1995), http://dx.doi.org/10.1109/69.476501
5. Caprara, A., González, J.J.S.: Separating lifted odd-hole inequalities to solve the index selection problem. Discrete Appl. Math. 92(2-3), 111–134 (1999), http://dx.doi.org/10.1016/S0166-218X(99)00050-5
6. Caprara, A., Salazar, J.: A branch-and-cut algorithm for a generalization of the uncapacitated facility location problem. TOP 4(1), 135–163 (1996), citeseer.ist.psu.edu/caprara95branchcut.html
7. Chan, A.Y.: Index selection in a self-adaptive relational data base management system. Tech. rep., Cambridge, MA, USA (1976)
8. Chaudhuri, S., Narasayya, V.R.: An efficient cost-driven index selection tool for Microsoft SQL Server. In: VLDB 1997: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 146–155. Morgan Kaufmann Publishers Inc., San Francisco (1997)
9. Choenni, S., Blanken, H.M., Chang, T.: Index selection in relational databases. In: International Conference on Computing and Information, pp. 491–496 (1993), citeseer.ist.psu.edu/choenni93index.html
10. Finkelstein, S., Schkolnick, M., Tiberio, P.: Physical database design for relational databases. ACM Trans. Database Syst. 13(1), 91–128 (1988), http://doi.acm.org/10.1145/42201.42205
11. Fotouhi, F., Galarce, C.E.: Genetic algorithms and the search for optimal database index selection. In: Proceedings of the The First Great Lakes Computer Science Conference on Computing in the 90's, pp. 249–255. Springer, London (1991)
12. Ganek, A.G., Corbi, T.A.: The dawning of the autonomic computing era. IBM Syst. J. 42(1), 5–18 (2003)
13. Glover, F.: Tabu search – part i. ORSA Journal on Computing 1(3), 190–206 (1989)
14. Ip, M.Y.L., Saxton, L.V., Raghavan, V.V.: On the selection of an optimal set of indexes. IEEE Trans. Softw. Eng. 9(2), 135–143 (1983), http://dx.doi.org/10.1109/TSE.1983.236458
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
16. Kołaczkowski, P.: Compressing very large database workloads for continuous online index selection. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 791–799. Springer, Heidelberg (2008)
17. Kormilitsin, M., Chirkova, R., Fathi, Y., Stallman, M.: Plan-based view and index selection for query-performance improvement. Tech. Rep. 18, NC State University, Dept. of Computer Science (2008)

18. Kratica, J., Ljubić, I., Tošić, D.: A genetic algorithm for the index selection problem (2003), citeseer.ist.psu.edu/568873.html
19. Meyn, S.P., Tweedie, R.: Markov Chains and Stochastic Stability. Springer, Heidelberg (1993)
20. Papadomanolakis, S., Ailamaki, A.: An integer linear programming approach to database design. In: Workshop on Self-Managing Database Systems (2007)
21. Papadomanolakis, S., Dash, D., Ailamaki, A.: Efficient use of the query optimizer for automated physical design. In: VLDB 2007: Proceedings of the 33rd international conference on Very large data bases, pp. 1093–1104. VLDB Endowment (2007)
22. Parr, T.: ANTLRv3: Another tool for language recognition (2003–2008), http://www.antlr.org/
23. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, London (2003), http://portal.acm.org/citation.cfm?id=773294
24. Sattler, K.U., Schallehn, E., Geist, I.: Autonomous query-driven index tuning. In: IDEAS 2004: Proceedings of the International Database Engineering and Applications Symposium (IDEAS 2004), pp. 439–448. IEEE Computer Society Press, Los Alamitos (2004), http://dx.doi.org/10.1109/IDEAS.2004.15
25. Schnaitter, K., Abiteboul, S., Milo, T., Polyzotis, N.: Colt: continuous on-line tuning. In: SIGMOD 2006: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pp. 793–795. ACM Press, New York (2006), http://doi.acm.org/10.1145/1142473.1142592
26. Skelley, A.: DB2 advisor: An optimizer smart enough to recommend its own indexes. In: ICDE 2000: Proceedings of the 16th International Conference on Data Engineering, p. 101. IEEE Computer Society, Washington (2000)
27. Talebi, Z.A., Chirkova, R., Fathi, Y., Stallmann, M.: Exact and inexact methods for selecting views and indexes for olap performance improvement. In: EDBT 2008: Proceedings of the 11th international conference on Extending database technology, pp. 311–322. ACM, New York (2008), http://doi.acm.org/10.1145/1353343.1353383
28. Transaction Performance Council: The TPC-H decision support benchmark (2001–2008), http://www.tpc.org/tpch/
29. Whang, K.-Y.: Index selection in relational databases. In: FODO, pp. 487–500 (1985)

Integrated Retrieval from Web of Documents and Data

Krishnaprasad Thirunarayan and Trivikram Immaneni

Abstract. The Semantic Web is evolving into a property-linked web of data, conceptually different from but contained in the Web of hyperlinked documents. Data Retrieval techniques are typically used to retrieve data from the Semantic Web while Information Retrieval techniques are used to retrieve documents from the Hypertext Web. We present a Unified Web model that integrates the two webs and formalizes connection between them. We then present an approach to retrieving documents and data that captures best of both the worlds. Specifically, it improves recall for legacy documents and provides keyword-based search capability for the Semantic Web. We specify the Hybrid Query Language that embodies this approach, and the prototype system SITAR that implements it. We conclude with areas of future work.

Keywords: Data Retrieval, Information Retrieval, Hypertext Web, Semantic Web, Unified Web, Hybrid Query Language.

1 Introduction

Information Retrieval systems allow the user to communicate his/her information need using a simple keyword based query mechanism, and typically return a set of documents that are *likely* to satisfy the users' information need as represented by the query. Such systems need to *interpret* the user query and the contents of the document repository (such as Hypertext Web), due to inherent ambiguity in the natural language (and the varying reliability of the sources). To bridge the gap between the information need and the semantics of the query and the documents (and to take into account the nature of the sources), an IR system conceptually ranks *all* the documents in the repository based upon their *likelihood* of satisfying the user's information need and returns the top *n* documents from the ranked list. Furthermore, in practice, if the documents being returned by the system are not directly fulfilling the user information need, these documents are at least helping the user probe and understand the nature of the repository and the nature of the system itself which in turn leads to effective query formulation.

Krishnaprasad Thirunarayan and Trivikram Immaneni
Kno.e.sis Center
Department of Computer Science and Engineering
Wright State University,
3640 Colonel Glenn Highway, Dayton, OH 45435, USA
e-mail: {t.k.prasad,immaneni.2}@wright.edu

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 25–48.
springerlink.com

Semantic Web [1] is a term used to describe the family of description languages, standards, and other technologies which aim at "extending" the current web by making its content machine accessible. Since the Resource Description Framework (RDF) forms the foundation of this "extension", we can visualize the Semantic Web (SW) as a labeled graph with resources as nodes and binary predicates as edges (web of data). This is in contrast to the Hypertext Web (HW) which is a graph with resources (usually documents) as nodes and hyperlinks as edges (web of documents).

An interesting question that arises is as to where the Web documents fit into the SW and how they can be retrieved. Intuitively, since the Web documents are resources that are identified by their URIs, we can view them as nodes in the SW graph. The document *content* has to be explicitly incorporated into the SW as literals. We can then use RDF query languages such as SPARQL [2], which enable RDF graph traversal and support regular expression matching of strings (literals), to retrieve the documents based upon their neighborhood as well as their content. This is classic Data Retrieval (DR).

Arguably, for this method of retrieving Web documents to have any remote chance of out-performing current Information Retrieval (IR) techniques, each and every Web document should have highly useful *semantic descriptions*. (As it stands, the textual content manifests itself as unstructured literals residing in structured data, whose implicit semantics is inaccessible to the machines.) Another issue here is that query languages such as SPARQL require the users to have intimate knowledge of underlying schema (exact URIs) to compose queries. For example, to search for course pages that talk about retrieval, the user or agent can compose the following query, where "i" is for case insensitive matching, and the FILTER statement is akin to LIKE statement in SQL:

```
PREFIX univ: <http://www.univ.com/>
    Select ?g
    Where
    {
          ?x rdf:type univ:course .
          ?x univ:coursePage ?g .
              FILTER regex(?g, "retrieval", "i")
    }
```

Unfortunately, such semantic metadata can neither be manually created with reasonable effort for legacy documents, nor automatically extracted using extant NLP techniques. The simple keyword-based interfaces that systems such as Yahoo! and Google expose to their users is another compelling reason to stick to IR techniques for retrieving Web documents. So, we seem to be better off retrieving data from the SW using DR techniques and retrieving documents from the HW using keyword-based IR techniques. In this sense, when seen from (data or document) retrieval perspective, the Semantic Web is, conceptually, a web of data that is estranged from the web of documents that is the Hypertext Web. Note that, some technologies such as RDFa [3,25] are trying to change this. But unless we find a

way to (automatically) create semantic descriptions of all of the existing HW documents, there will always be a large corpus of documents isolated from the SW.

Our high level goal is to *view* the Semantic Web and the Hypertext Web as a unified whole and retrieve data and documents from this Unified Web (UW) [4, 5, 6]. This way, we can utilize the *available* semantic descriptions to enhance Web document retrieval and will also have the option of using the information from (unstructured documents of) the HW to improve the SW data retrieval, as explained later.

The web documents can be broadly divided into the following three categories – those meant primarily for human consumption (HTML, plain text, jpg, etc.), those meant primarily for machine consumption (RDF, OWL, RDFS, etc.) and hybrid documents that are meant for both machine and human consumption (RDFa, microformats and other such technologies that allow embedding of semantic markup in HTML/XHTML documents [7]). Our goal is to facilitate the retrieval of all the above three types of documents while fully exploiting semantic markup/descriptions, when available, to increase retrieval effectiveness.

We want to enable lay users to retrieve human-consumable documents (first and third types) using the traditional keyword-based query mechanism, enhanced using synonyms implicit in the context. For example, if the users are looking for information on William Gates, they should be able to access all the relevant information via both Bill Gates and William Gates, where the aliases are inferred from the corresponding anchor texts. Furthermore, we want to transparently use the available SW data to enhance the retrieval process. For example, if the users know that they are looking for Jaguar, the car, they should be able to communicate this disambiguating information in a query, to distinguish it from its animal sense and its football sense. Specifically, we introduce the notions of wordset and wordset pair queries for this purpose. A *wordset* is a set of words and phrases such as: <*"William Gates" "Bill Gates"*> that allows a user to search for nodes (URIs) about William Gates. Furthermore, a *wordset pair query* consists of two wordsets separated by the scope resolution operator: <*car*>*::*<*jaguar*> that enables the user to additionally provide the system with disambiguation information. The first of the pair refers to the class (resp. superclass) and the second of the pair to the instance (resp. subclass). So the wordset pair <*student*>*::*<*john*> (resp. <*car*>*::*<*jaguar*>) would match a node indexed by "john" (resp. "jaguar") *and* which is a direct or indirect instance (resp. subclass) of a node indexed by "student" (resp. "car").

For more informed users, we want to provide a light-weight, keyword-based hybrid query language to retrieve information from all three types of documents, even when the underlying schema (exact URIs) is not known. That is, the users should be able to query and retrieve *data* by posing questions such as *list all the elements in group 1 of the periodic table* [8], even in the absence of schema information.

To summarize: our goal is to *view* the two webs as a unified whole and devise hybrid techniques to retrieve data and documents from this Unified Web (UW).

Taking a unified view will enable us to utilize hyperlink connections from documents to SW data nodes resulting in improved document retrieval in terms of *precision*, *recall* and the type of documents that can be retrieved. It will also pave the way for keyword-based, hybrid querying of SW data obviating the need for the user to be intimately familiar with the schema information.

We first explain how certain hyperlinks can be viewed as semantic annotations, to improve retrieval of legacy HW documents using SW data in Section 2. We then describe the Unified Web model in Section 3, followed by the description of the Hybrid Query Language in Section 4. We briefly discuss the implementation of our system SITAR and present some results in Section 5. We discuss related research in Section 6 and conclude with suggestions for future work in Section 7. This chapter is a condensed version of our journal submission [36].

2 Hyperlinks as Semantic Markup

The SW is physically enclosed in web pages on the HW (as the RDF data is contained in documents (files) located on the Web). HTML markup tells the browser how to display a document (in spite of its idealistic beginnings as a means to structure a document). In contrast, semantic markup of content promotes its machine comprehension. Consider the following fragment from a document located at *http://www.one.com/A.html* that basically says that *B.html* is authored by John Smith.

```
<rdf:RDF…>
    <rdf:Description
              rdf:about="http://www.two.com/B.html">
      <mydomain:author> John Smith </mydomain:author>
    </rdf:Description>
</rdf:RDF>
```

The physical location of this fragment (that is, the document in which this triple resides) is irrelevant to the resource that it is describing. (We are deliberately ignoring provenance considerations for now.) It is metadata rather than "markup". A hyperlink from a HW document to a node on the SW, links the document to the node, and at the same time, annotates the document with the URI of the destination node. We propose that this valuable information be utilized to enhance document retrieval (especially meaningful in the context of legacy documents as explained later). For example, if a document contains a hyperlink to *mailto:bsmith@wright.edu*, and if there is a triple in the database that tells us that *<mailto:bsmith@wright.edu rdf:type univ:prof>* then this information can be used to enhance document retrieval. Specifically, a search for an instance of a *univ:prof* can uncover the document containing *mailto:bsmith@wright.edu.* Effectively, ISA relationship encoded in the SW can be used to broaden the search results.

Consider another example. On the web, it is not uncommon to see a document with hyperlinks from terms in the document to standard web pages (such as dictionary.com, Wikipedia, etc) that describe those terms.

"..The <*a href="http://dictionary.com/search?q=jaguar*"> *J*aguar </*a*> God of the Underworld …."

Here the hyperlink is from the term *Jaguar* to a webpage in an online dictionary that describes/defines the term. The dictionary webpage URL can be said to *annotate* the term *Jaguar*. In order to incorporate an existing Animal ontology in a *scalable* way, without modifying any extant legacy documents, or *dictionary.com* pages, or existing ontologies, we can simply add the following triple to the IR system's database.

<*http://dictionary.com/search?q=jaguar*
owl:Sameas *http://www.animalOnto.com/Jaguar* >

This information can be used to conclude that the (unmodified) web pages linking to *http://dictionary.com/search?q=jaguar* (which is also unmodified) are talking about Jaguar, the animal. This idea can be extended to create ontology websites where each web page corresponds to an entity in the ontology. For new documents, a user can annotate a document simply by adding a hyperlink to a Semantic Web data node or one of the pages in the web site that has explicit semantic annotations. Therefore, the existing hyperlink structure can be harnessed and used in conjunction with semantic descriptions to enhance document retrieval.

3 The Unified Web model

The Unified Web model aims to integrate the SW and the HW into a single unified whole by encoding the two webs and the connections between them. The UW model is a graph of nodes and edges (N, E). A node is an abstract entity that is uniquely identified by its URI. There are two categories of nodes: (i) Natural nodes (*NN*) and (ii) System defined nodes (*SN*). The natural nodes can be further classified as plain (or non-document) nodes (*PN*) and document nodes (*DN*) based on whether or not a node has an associated document. The system defined nodes can be further classified as literal nodes (*LN*), triple nodes (*TN*) and blank nodes (*BN*). The system creates a URI and assigns it to each blank node, triple and literal that it encounters on the Web.

There are two categories of edges: (i) User defined edges (*UE*) and (ii) System defined edges (*SE*). The user defined edges come from the triples in the (Semantic Web) documents while the system defined edges are defined to make explicit the interconnections between the HW and the SW. The system defined edges are the following. The *asserts* edge exists from a node (document) to each of the RDF statements found in the associated document. The RDF statement itself has a subject, a property and an object. There is no restriction as to how a triple is obtained from the document. The *hasDocument* edge exists from a node to a literal. The literal is the string representation of the document associated with the node. A *hyperlinksTo* edge exists from a node A to another node B if there is a hyperlink from the document of node A to the document of node B. The *linksTo* edge exists

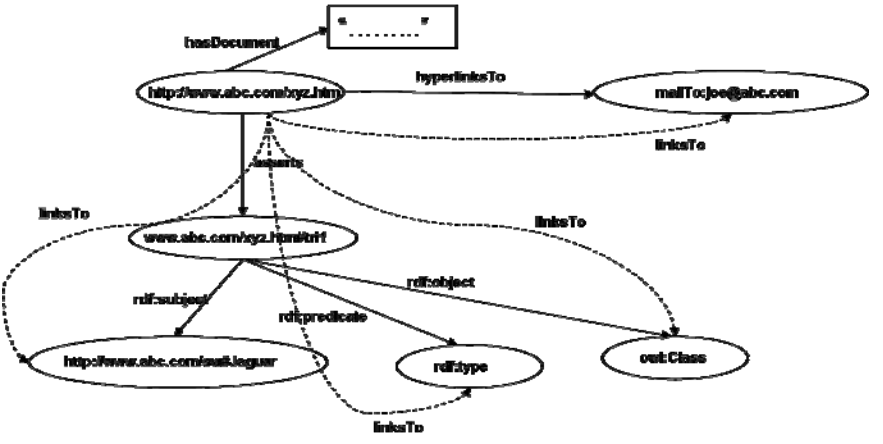


Fig. 1 Relationships

from node A to node B if a *hyperlinksTo* relationship exists from node A to node B, or node B occurs in any of the triples asserted by node A (see Figure 1).

More formally, these edges can be specified as functions/relations in terms of their signatures (domains and ranges), and include:

$$hyperlinksTo \subseteq DN \times NN$$
$$linksTo \subseteq DN \times NN$$
$$asserts \subseteq DN \times TN$$
$$hasDocument: DN \rightarrow LN$$

These relations are not independent and cannot be assigned arbitrarily. They must satisfy at least the following constraints.

$\forall n \in DN, \forall m \in NN$: if [n, *hyperlinksTo*, m] $\in SE$ then [n, *linksTo*, m] $\in SE$.

The Unified Web model is not a simple superposition of the SW graph over the hypertext graph. The Semantic Web can be thought of as a global RDF graph constructed by gathering all possible RDF triples from documents that reside on the Hypertext Web. The UW reifies each of the SW triples by explicitly encoding the *asserts* relationship between a document and the triple that is extracted from it (embodying provenance). The UW can be visualized as a meta – Semantic Web which in itself can be an RDF graph (one that subsumes all RDF graphs found on the web). In addition, this RDF graph also encodes the Hypertext Web (HTML documents and hypertext links between them). The aim is to encode the HW (*hyperlinksTo* and *hasDocument*), and the SW, and the connections between the two (such as *asserts*) while allowing easy mapping of data retrieval queries meant for the "conventional" SW to those for the UW [4, 5]. The *linksTo* tries to define a generic "connection" between two nodes. The *linksTo* edge is for the Unified Web what the hyperlink is for the HW and the property-link is for the SW. In our implementation, we use *linksTo* to view a document as being annotated by the URIs that it *linksTo* and use this information while retrieving documents.

The UW model can be specified and implemented using RDF [4, 5]. Since all the "user triples" are present (in reified from) in the model, query languages like SPARQL can be used to retrieve the data – all we need is a straight-forward mapping of the SPARQL query for the SW to the SPARQL query for the UW. For example, the following "conventional" query

```
SELECT ?title
  WHERE
  {
      <http://example.org/book/book1>
   http://purl.org/dc/elements/1.1/title  ?title .
  }
```

can easily be transformed into:

```
SELECT ?title
WHERE
{
?x <rdf:subject> <http://example.org/book/book1> .
?x <rdf:predicate>
       <http://purl.org/dc/elements/1.1/title> .
?x <rdf:object> ?title .
}
```

But since every triple is encoded as an *rdf:Statement*, it is necessary to implement the model in such a way as to minimize computational overheads.

4 Hybrid Query Language Specification

We now describe the Hybrid Query Language (HQL) that combines the Information and Data retrieval paradigms to enable hybrid retrieval of the data and documents from the Unified Web. One of the fundamental features of HQL is the concept of *wordset* that enables the *wordset pair* queries, which is a unique way of combining keyword-based search with inheritance reasoning. As explained later, these queries refer to the set of index strings (derived from the neighborhood) associated with a node.

4.1 Wordsets and Wordset Pairs

The wordsets allow users to search for nodes (URIs) based upon the words and phrases in their URI Index Words. A *wordset* is a set of words and phrases (multiple words enclosed in quotes) enclosed in angular brackets as shown: <*Microsoft research "William Smith"*>. Given a wordset, the system retrieves all the nodes in the UW such that *all* of the words/phrases in the wordset appear in the node's index words. For example, let the Home URI of a node be m*ailto:bsmith@microsoft.com.* Let this node be referenced from another HTML document as follows:

* Research Scientist *

Also, let the following triple be asserted by some node.

<mailto:bsmith@microsoft.com rdfs:label "William Smith">

Then the index words of the node will (perhaps) be: *{"bsmith" "microsoft" "Research Scientist" "Research" "Scientist" "William Smith" "William" "Smith"}*. This node will be retrieved by the query *<Smith Research>,* but *not* by *<Smith Research Bill>*. Thus multiple words inside angular brackets are conjoined.

The user can provide the system with disambiguation information using the novel wordset pair queries. A *wordset* pair consists of two wordsets separated by the scope resolution operator as shown below: *<animal>::<jaguar>*. The first of the pair refers to the class/superclass and the second of the pair to the instance/subclass. So the wordset pair *<student>::<john>*would match a node whose index words contains "john" *and* which is a direct or indirect instance of a URI whose index words contains "student".

4.2 Formal Description of HQL

The Hybrid Query Language [4, 5] enables convenient *navigation* and *extraction* of information from the UW. It enables formulation of precise queries involving URIs, and "approximate" word-based queries (e.g., wordset, wordset-pairs queries) that capture context and/or content (e.g., keyword queries). In other words, it enables access to both HW documents and SW data, *incorporating indexing information from the neighboring nodes*. Specifically, the wordset queries can use anchor text in the HW to retrieve SW nodes, and wordset pair queries can express disambiguation information using the ISA edges encoded in the SW for semantic search of HW documents.

Before we go into the details of the query language, let us first define some more utility functions/relations in addition to the four in the previous section:

homeURI: N –> Set(URI)
indexWords : NN –> PowerSet(STRINGS)
hasTriples: DN –> PowerSet(NN x NN x NN)
hasLiteral: LN –> STRINGS

URI denotes a string that must satisfy the URI syntax requirement (RFC 3986), while *STRINGS* denotes a set of words, phrases, and other fragments. (We represent set of words and phrases by simply listing them, separated by blanks.) *PowerSet* operator yields a set of all subsets. The members of *NN* x *NN* x *NN* are referred to as the triples (such as those found in RDF documents).

homeURI maps a node to its URI. *indexWords* maps a node to a set of strings that can serve as an index to it. These can be composed from the URI and the anchor text (associated with links) from the neighboring nodes among other things. *hasDocument* maps a document node to the associated document text string. *hyperlinksTo* relates a document node to a node that appears in a hyperlink in the corresponding document. *linksTo* relates a document node to a node that

appears in the corresponding document. This can be in the form of a hyperlink or embedded in a triple. *hasTriples* maps a document node to the set of 3-tuples of nodes that appear in the corresponding document. *asserts* relates a document node to a triple node that reifies the triple that appears in the corresponding document. *hasLiteral* maps a literal node to the string it is associated with. It is possible to have multiple literal nodes associated with the same string. *Note that a specific instantiation of the framework can be obtained by defining how these functions/relations* (such as *indexWords*) *are obtained from the node's neighborhood.*

The above functions must satisfy at least the following constraints.

$$\forall n \in NN, \forall [n1, n2, n3] \in NN \times NN \times NN:$$
$$[n1, n2, n3] \in hasTriples\ (n) \quad \text{only if}$$
$$[n, linksTo, n1] \in SE \ \wedge\ [n, linksTo, n2] \in SE \ \wedge\ [n, linksTo, n3] \in SE$$

$$\forall n \in N, [n1, n2, n3] \in NN \times NN \times NN:$$
$$[n1, n2, n3] \in hasTriples\ (n) \quad \text{if and only if}$$
$$\exists\ tn \in TN : [n, asserts, tn] \in SE \wedge [tn, rdf{:}subject, n1] \in SE$$
$$\wedge [tn, rdf{:}predicate, n2] \in SE \ \wedge [tn, rdf{:}object, n3] \in SE$$

For convenience, we abuse the language and say that n1, n2, and n3 *appear in* tn in the context of the reification constraint. We also abbreviate [n, *property*, m] ∈ *SE as* [n, *property*, m].

In what follows, we motivate and specify the abstract syntax of the queries using a context-free grammar, and the semantics of the queries in terms of the Unified Web model, in sufficient detail to enable prototyping. Our presentation focuses on queries that yield a set of nodes. The "domain information bearing" strings such as the document text, literal, etc. can be easily obtained from a URI by calling corresponding system functions such as *hasDocument*, *hasLiteral*, etc. and from triples using *rdf:subject*, *rdf:predicate*, and *rdf:object*, etc.

TopLevelQuery ::= Nodes-ref | *Triples-ref* | …

QUERY: *Nodes-ref ::=* u, where u ∈ *Set*(*URI*).

ANSWER: *Result*(*u*) = { n ∈ *N* | *HomeURI*(n) = u }

SEMANTICS: The *URI-query* returns the set containing the unique node whose `HomeURI` matches the given URI. Otherwise, it returns an error.

EXAMPLE: *http://www.aifb.uni-karlsruhe.de/Personen/viewPersonenglish?id_db=20*

QUERY: *Nodes-ref::=ss*, where *ss* ∈ *PowerSet*(*STRINGS*).

ANSWER: *Result*(*ss*) = { n in *N* | *ss* ⊆ *IndexWords*(n) }

SEMANTICS: The *wordset query*, *ss*, usually written as a set of strings delimited using angular brackets, returns the set of nodes whose *IndexWords* contain *ss*.

EXAMPLE: *<peter haase>*

QUERY: *Nodes-ref ::= pp::ss,* where *pp, ss* ∈ *PowerSet*(*STRINGS*).

ANSWER: *Result(pp::ss)* = { n ∈ *N* | *ss* ⊆ *IndexWords*(n) ∧ ∃m : n ISA m ∧ *pp* ⊆ *IndexWords*(m) }

SEMANTICS: The *wordset-pair query, pp::ss,* usually written as two wordsets delimited using colon, returns the set of nodes such that each node has *IndexWords* that contains *ss* and has an ISA ancestor whose *IndexWords* contains *pp*.

EXAMPLE: <*student*>::<*peter*>

QUERY: *Triples-ref* ::= u, where u ∈ *Set*(*URI*).

ANSWER: *Result*(u) = { n ∈ *TN* | *HomeURI*(n) = u }

SEMANTICS: The *triple node URI-query* returns the set containing the unique node whose *HomeURI* matches the given triple node URI. Otherwise, it returns an error. The triple nodes are system generated.

EXAMPLE: *http://www.aifb.uni-karlsruhe.de/Personen/viewPersonFOAF/foaf_80.rdf#tri52*

QUERY: *Triples -ref* ::= *Single-Var-Triples-ref*
Single-Var-Triples -ref ::= [?var *Nodes-ref Nodes-ref*]
Single-Var-Triples -ref ::= [*Nodes-ref* ?var *Nodes-ref*]
Single-Var-Triples –ref ::= [*Nodes-ref Nodes-ref* ?var]

where *?var* is a variable.

ANSWER: *Result*([?var *Nodes-ref1 Nodes-ref2*]) =
{ t ∈ *TN* | n1 ∈ *Result*(*Nodes-ref1*) ∧ n2 ∈ *Result*(*Nodes-ref2*)
∧ ∃m ∈ *N* : [m, *asserts*, t] ∧ [t, *rdf:predicate*, n1] ∧ [t, *rdf:object*, n2] }

Similarly, for the other two cases.

SEMANTICS: The *part triple query* [?var *Nodes-ref1 Nodes-ref2*] returns the set of (system generated) triple nodes that contain a node related by a binary predicate denoted by *Nodes-ref1* to some node denoted by *Nodes-ref2*. Similarly, for the other two cases. Note that this query characterizes a node using its neighborhood.

EXAMPLE*: [*<*silver*> <*atomic weight*> *?x]*

QUERY: *Triples-ref* ::= [*Nodes-ref*, *Nodes-ref*, *Nodes-ref*]

ANSWER: *Result*([*Nodes-ref1*, *Nodes-ref2*, *Nodes-ref3*]) =
{ t ∈ *TN* | n1 ∈ *Result*(*Nodes-ref1*)
∧ n2 ∈ *Result*(*Nodes-ref2*) ∧ n3 ∈ *Result*(*Nodes-ref3*)
∧ ∃m ∈ *N* : [m, *asserts*, t] ∧ [t, *rdf:subject*, n1]
∧ [t, *rdf:predicate*, n2] ∧ [t, *rdf:object*, n3] }

SEMANTICS: The *full triple query* [*Nodes-Ref*, *Nodes-Ref*, *Nodes-ref*] returns the set of (system generated) triple nodes matching the node references.

EXAMPLE: *[http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2062instance*
<*name*> <*peter*> *]*

QUERY: *Triples-ref* ::= *Double-Var-Triples-ref*
Double-Var-Triples-ref ::= [?var, ?var, *Nodes-ref*]
Double-Var-Triples-ref ::= [?var, *Nodes-ref*, ?var]
Double-Var-Triples-ref ::= [*Nodes-ref*, ?var, ?var]
where ?var is a variable.

ANSWER: $Result([?var, ?var, Nodes\text{-}ref]) = \{ t \in TN \mid m \in Result(Nodes\text{-}ref) \wedge [t, rdf{:}object, m] \wedge \exists n \in N : [n, asserts, t] \}$

Similarly, for the other two cases.

SEMANTICS: The *part triple to triples query* [?var ?var *Nodes-ref*] returns the set of (system generated) triple nodes that contain some node denoted by `Nodes-ref`. Similarly, for the other two cases. Note that this query characterizes the node neighborhood. Each variable occurrence is independent of the other occurrences.

EXAMPLE: *[?x <title> ?x]*

QUERY: *Nodes-ref* ::= *Nodes-ref AND Nodes-ref*
Nodes-ref ::= *Nodes-ref OR Nodes-ref*
Triples-ref ::= *Triples-ref AND Triples-ref*
Triples-ref ::= *Triples-ref OR Triples-ref*

SEMANTICS: "OR" and "AND" are interpreted as set-union and set-intersection respectively. Each variable occurrence is independent of the other occurrences.

4.2.1 Queries for Exploring the System-Generated Neighborhood of a Node

QUERY: *Nodes-ref* ::= *getAllTriples*(*Nodes-ref*)

ANSWER: $Result(\ getAllTriples(Nodes\text{-}ref)) = \{ t \in TN \mid n \in Result(Nodes\text{-}ref) \wedge \text{n appears in t} \wedge \exists m \in N: [m, asserts, t] \}$

SEMANTICS: This query retrieves the (system generated) triple nodes in which the queried node URI appears. (Recall that a node n is said to *appear in* triple node t, if n is a subject, a predicate, or an object associated with the triple corresponding to the triple node.)

EXAMPLE: *getAllTriples(http://www.daml.org/2003/01/periodictable/PeriodicTable#group_11)*

QUERY: *Nodes-ref* ::= *getLinkingNodes*(*Nodes-ref*)

ANSWER: $Result(\ getLinkingNodes(Nodes\text{-}ref)) = Result(Nodes\text{-}ref) \cup \{ m \in N \mid \exists n \in Result(Nodes\text{-}ref) : [m, linksTo, n] \}$

SEMANTICS: This query retrieves the nodes corresponding to *Nodes-ref* and the document nodes containing references to the nodes corresponding to *Nodes-ref*. Effectively, nodes and their neighborhoods (specifically, through in-links) are retrieved.

EXAMPLE: *getLinkingNodes(http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2023)*

QUERY: *Nodes-ref* ::= *getAssertingNodes*(*Triples-ref*)
ANSWER: *Result*(*getAssertingNodes*(*Triples-ref*)) =
{ m ∈ *DN* | ∃t ∈ *Results*(*Triples-ref*) : [m, *asserts*, t] }
SEMANTICS: This query retrieves document nodes containing the triples, a form of provenance information.
EXAMPLE: *getAssertingNodes([<peter haase> <publication> ?x])*

QUERY : *Nodes-ref* ::= *getDocsByKeywords*(*ss*), where *ss* ∈ *PowerSet(STRINGS).*
ANSWER: *Result*(*getDocsByKeywords*(*kws*)) =
{ m ∈ *DN* | *hasDocument*(m) = dt ∧ *match*(kws, dt) }

SEMANTICS: This query is analogous to the traditional keyword query that takes a set of keywords and retrieves document nodes that match the keywords. *match* embodies the criteria for determining when a document text is "relevant" to a keyword. It can be as simple as requiring verbatim occurrence, to as complex as requiring stemming, synonym generation, spelling correction, etc. *match* may be *compositional*, that is, *match* (kws, dt) = ∀w ∈ kws: *match*(w, dt), but it is not required.

QUERY : *Nodes-ref* ::= *getLiteralsByKeywords* (*ss*),
where *ss* ∈ *PowerSet*(*STRINGS*).
ANSWER: *Result*(*getLiteralsByKeywords*(kws)) =
{ m ∈ *LN* | *hasLiteral*(m) = dt ∧ *match*(kws, dt) }
SEMANTICS: This is analogous to the above query customized for literal nodes.
EXAMPLE: *getLiteralsByKeywords(semantic grid)*

4.2.2 Further Queries for Retrieving Documents

QUERY: *getDocsByContent*: *PowerSet*(*STRINGS*) -> PowerSet(DN)
ABBREVIATION FOR:
getDocsByContent(kws) = *getLinkingNodes*(*getDocsByKeywords*(kws))
where kws ∈ *PowerSet*(*STRINGS*).
SEMANTICS: This query retrieves the document nodes with content matching keywords in kws and the neighboring document nodes that reference such nodes. Intuitively, we want to pursue both the "authorities" and the "hubs" [9], assisting both navigational searches and research searches [10] in a novel way.

QUERY: *getDocsByIndexOrContent*: *PowerSet*(*STRINGS*) -> *PowerSet*(*DN*)
ABBREVIATION FOR: *getDocsByIndexOrContent* (kws) =
$$getDocsByKeywords(kws) \vee \bigvee_{kw \in kws} getLinkingNodes(kw)$$
where kws ∈ *PowerSet*(*STRINGS*).

SEMANTICS: This query retrieves the document nodes with content matching the keywords `kws` or in the neighborhood of nodes indexed by `kws`. Implicitly, the former captures syntactic retrieval and the latter enables semantic retrieval.
EXAMPLE: *getDocsByIndexOrContent(semantic web)*

QUERY: *getDocsByIndexAndContent*:
Nodes-ref x *PowerSet*(*STRINGS*) -> *PowerSet*(*DN*)
ABBREVIATION FOR: *getDocsByIndexAndContent* (nr, kws) =
getLinkingNodes(*Result*(nr)) ∧ *getDocsByKeywords*(kws)
where nr ∈ *Nodes-ref*, kws ∈ *PowerSet*(*STRINGS*).
SEMANTICS: This query retrieves the document nodes with content matching the keywords kws and in the neighborhood of nodes corresponding to nr. Implicitly, if `nr` is a URI of a document node containing the keywords kws, then the result will contain this document node. If nr is a URI and this URI and the keywords kws are contained in a document, then the result will contain the latter document node. Similarly, for nodes in *Result*(nr) when nr contains wordset and wordset-pairs.

QUERY: *getDocsByTriplesAndContent*:
Triples-ref x *PowerSet*(*STRINGS*) -> *PowerSet*(*DN*)
ABBREVIATION FOR: *getDocsByTriplesAndContent*(tr, kws) =
getAssertingNodes(tr) ∧ *getDocsByKeywords*(kws)
where tr ∈ *Triples-ref*, kws ∈ *PowerSet*(*STRINGS*).
SEMANTICS: This query retrieves (semantic web) document nodes that match the keywords and contain the referenced triples. (Cf. RDFa, Microformats)

QUERY: *Single-Var-Triples-list* ::= *Single-Var-Triples-ref*
Single-Var-Triples-list ::= *Single-Var-Triples-ref*
Single-Var-Triples-list
Nodes-ref ::= *getBindings*(*Single-Var-Triples-list*)

QUERY1: *Nodes-ref* ::= *getBindings*([?var *Nodes-ref Nodes-ref*])
ANSWER: *Result*(*getBindings*([?var *Nodes-ref1 Nodes-ref2*]))
= { n ∈ N | n1 ∈ *Result*(*Nodes-ref1*) ∧ n2 ∈ *Result*(*Nodes-ref2*)
∧ ∃m ∈ N : [m, *asserts*, t] ∧ [t, *rdf:subject*, n]
∧ [t, *rdf:predicate*, n1] ∧ [t, *rdf:object*, n2]}
Similarly, for the other two cases. Effectively, this query dereferences the triple node resulting from evaluating plain *Single-Var-Triples-ref* query.

QUERY2: *Nodes-ref* ::= *getBindings*(*Single-Var-Triples-ref*
Single-Var-Triples-list)

ANSWER: *Result*(*getBindings*(*Single-Var-Triples-ref Single-Var-Triples-list*)) =
Result(*getBindings*(*Single-Var-Triples-ref*)) ∩
Result(*getBindings*(*Single-Var-Triples-list*))

SEMANTICS: This query retrieves the bindings for the variables that satisfy all the triple references with single variable. All the variable occurrences are considered identical, that is, they must all be assigned the same value throughout the *getBindings*-argument.
EXAMPLE: *getBindings([<student>::<peter> <name> ?x])*
EXAMPLE: *getBindings([?x <group> <group 1>] [?x <color> <white>])*

QUERY: *getDocsByBindingsAndContent*:
Single-Var-Triples-list x *PowerSet*(*STRINGS*) -> *PowerSet*(*DN*)
ABBREVIATION FOR: *getDocsByBindingsAndContent*(vtl, kws) =
getBindings (vtl) ∧ *getDocsByKeywords*(kws)
where vtl ∈ *Single-Var-Triples-list*,
kws ∈ *PowerSet*(*STRINGS*).
SEMANTICS: This query retrieves document nodes that match the keywords and contain the matching triples.
EXAMPLE: *getDocsByBindingsAndContent([<student>::<peter> <homepage> ?x] "Semantic Grid")*

5 Implementation and Results

We have implemented an Apache Lucene [11] based retrieval system called SITAR (Semantic InformaTion Analysis and Retrieval system) based upon our model [6]. The system can currently index HTML and RDF/OWL files in addition to RDF data. (At present, the system does not support file formats such as PDF, MSWORD, etc., because only their URIs are indexed, but their content is not being analyzed).

Evaluating such a hybrid system is a tricky process. The system has DR components (triple matching) which render the precision and recall criteria irrelevant. But at the same time, the system also has IR components such as keyword based retrieval of documents in which case precision and recall become important. Furthermore, there are no suitable benchmarks that has documents, their semantic descriptions, some queries, and results of those queries (adjudged to be relevant by human experts). Here, we present results obtained by experimenting with the real-world AIFB SEAL [12] data that exercises some of the issues being addressed.

The general architecture of SITAR is shown in Figure 2. The crawler collects RDF/HTML/text documents from the Web and stores the documents in a local cache. The CyberNeko HTML parser [13] has been used to parse HTML documents and the Jena ARP [14] parser has been used to parse RDF documents. The output of the parsers are analyzed and indexed by Lucene. Every URI that is encountered (in HTML or RDF files) is analyzed using several heuristics to build

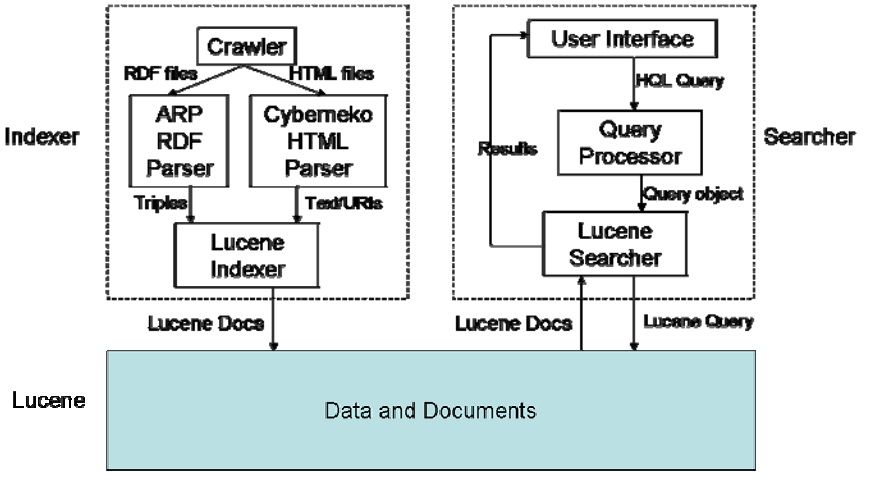


Fig. 2 SITAR Architecture

a set of index words for it. More importantly, if the URI occurs in a HTML document as a hyperlink, we use the anchor text to add to its index words set. A HTML document is indexed by the URIs that it *linksTo* (includes *hyperlinksTo*) as well as the words that are extracted from its body. (Recall that a HTML document *linksTo* an URI if the URI is present in the document.) In this sense, an HTML document can be seen as a bag of URIs and words. An RDF document is indexed by the URIs that it *linksTo* and by the triples (URIs) that it asserts. In this sense, an RDF document can be seen as a bag of URIs, triples, and words. Note that a hybrid document such as an RDFa document would be indexed by all of the above. The triples extracted from an RDF document are also stored in Lucene indexes.

Note that if all the documents in a given dataset are plain text documents (sans URIs and Triples), then SITAR behaves like a standard search engine (such as Lucene) for document retrieval. The HQL query *getDocByKeywords* is essentially a Lucene keyword search. However, in the presence of semantic information associated with the documents, the document retrieval can benefit as follows:

1. Users can query and retrieve RDF, HTML as well as hybrid documents (RDFa) by posing queries such as *retrieve Peter's homepage* or *retrieve the document asserting triples about Peter.*
2. The *precision* should improve because the users can convey their information need to the system unambiguously by refining queries by including facts such as *Peter, the student*.
3. Since a node is indexed by words obtained from its neighboring nodes (such as literals in triples in which it participates, anchor text, etc.) the *recall* can improve especially when synonyms or acronyms or aliases are involved.
4. Broadening searches such as *obtain pages about professors* should result in improved *recall*.

The AIFB SEAL website [12] has human-consumable XHTML documents (in English and German) along-side RDF/OWL documents. Every entity, such as person, research topic, project, publication, etc., has a URI and an RDF/OWL file associated with it. The URI also acts as the URL of the RDF/OWL page, which contains information about the entity. For example, if the entity is a person, the RDF file contains the person's name, designation, projects the person is working on, publications of the person and so on. The entity typically has an English and a German XHTML webpage (the entity's homepage) associated with it. For example, the following are the different "live" URIs associated with the entity (research project) called *AIFB SEAL*:

URI/OWL page:
http://www.aifb.uni-karlsruhe.de/Projekte/viewProjektOWL/id58instance
English homepage:
http://www.aifb.uni-karlsruhe.de/Projekte/viewProjektenglish?id_db=58
German homepage:
http://www.aifb.uni-karlsruhe.de/Projekte/viewProjekt?id_db=58

We crawled the SEAL website looking only for English versions of web pages and RDF/OWL files using heuristics. The crawler collected a total of 1665 files. Of these, we chose to deliberately ignore some large OWL files (multiple copies of the same file with each copy identified by a different URI) to simplify matters. The ARP parser could not parse some of the RDF documents - possibly because of problems with container elements. In the end, a total of 1455 files (610 RDF files and 845 XHTML files) were successfully parsed and indexed. A total of 193520 triples were parsed and indexed though there is no guarantee that the same triple was not asserted by two different documents. Note that all the index structures are persistently stored.

We now present some qualitative and quantitative results to give an idea about the dataset and about the kind of queries that can be formulated using HQL.

5.1 Qualitative Analysis

A user can use the HQL (Hybrid Query Language) described in the previous section to query for data and documents. A user searching for information about a person named *peter* can pose the query *<peter>*. This query, in effect, returns all nodes (URIs) that have been indexed using the word *peter*. A total of 52 URIs were retrieved in response to the above query including OWL files (instance data of people) and HTML files. The user can convey to the system that a Ph.D. student named *peter* is sought using the query *<phdstudent>::<peter>*. The following URIs were retrieved in response to this query:

http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2023instance
http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2119instance
http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2062instance

These are apparently URIs (of OWL files) representing individuals and containing information about them. In order to find out the names of these individuals, the user can use the query *getBindings([<phdstudent>::<peter> <name>*

?x]). This query returned 125 literal nodes gathered from different RDF files (apparently FOAF files). Note that the above queries are keyword-based, and hence easy to formulate, and enable transparent traversal of the semantic web. The system finds the bindings for the variable from triples such as those shown below:

uri : http://www.aifb.uni-karlsruhe.de/Personen/viewPersonFOAF/foaf_80.rdf#tri52
sub: http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2023instance
pre : http://xmlns.com/foaf/0.1/name
obj (Literal): Peter Haase
uri: http://www.aifb.uni-karlsruhe.de/Personen/viewPersonFOAF/foaf_2127.rdf#tri27
subj: http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2119instance
pred: http://xmlns.com/foaf/0.1/name
obj (Literal): Peter Bungert
uri: http://www.aifb.uni-karlsruhe.de/Personen/viewPersonFOAF/foaf_2069.rdf#tri132
sub: http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2062instance
pred: http://xmlns.com/foaf/0.1/name
obj (Literal): Peter Weiß

These triples repeated themselves in different files (with different URIs) and so a lot of duplicate data has been indexed by the system. The user can search for the homepages of Ph.D. students named *peter* by posing the query, *getBindings ([<phdstudent>::<peter> <homepage> ?x],* which returns the following results:

http://www.aifb.uni-karlsruhe.de/WBS/pha/
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/WBS
http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2023
http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2119
http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2062

The above URIs are, apparently, home pages of the above three individuals. The interesting thing is that all of the URIs except the first one points to a German page (whose content has not been indexed by our system). So, we cannot pose queries such as *get homepages of Ph.D. students named peter which talk about "semantic grid"* which translates into *getDocsByBindingsAndContent([<phdstudent>::<peter> <homepage> ?x] "semantic grid")* , unless we can convey to the system that the German version of the page should be treated "same as" the English version. Now that the user has the names, it can be used to query the system. The query *<peter haase>* retrieves the following URIs.

http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2023instance
http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2023
http://www.aifb.uni-karlsruhe.de/Personen/viewPersonenglish?id_db=2023
http://www.aifb.uni-karlsruhe.de/Publikationen/viewPublikationenPersonOWL/id2023.owl
http://www.aifb.uni-karlsruhe.de/Personen/viewPersonOWL/id2023.owl

These URIs are a mix of HTML (second and third URIs) and OWL documents. The second URI is the homepage of the individual named Peter Haase. It is almost synonymous with the individual [5] and so the pages that link to (*linksTo*) this page must be, arguably, relevant to the individual. The query *getLinkingNodes (http://www.aifb.uni-karlsruhe.de/Personen/viewPerson?id_db=2023)* retrieves a set of RDF and HTML documents most of which are pages of projects on which Peter Haase is working. Some of these results are shown below:

http://www.aifb.uni-karlsruhe.de/Personen/viewPersonDC/en/dc_2023.rdf
http://www.aifb.uni-karlsruhe.de/Personen/viewPersonFOAF/foaf_2023.rdf
http://www.aifb.uni-karlsruhe.de/Personen/Projekte/viewProjektenglish?id_db=78
http://www.aifb.uni-karlsruhe.de/Personen/Projekte/viewProjektenglish?id_db=80
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/Projekte/viewProjektenglish?id_db=51
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/Projekte/viewProjektenglish?id_db=71
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/Projekte/viewProjektenglish?id_db=81
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/Projekte/viewProjektenglish?id_db=42
http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/Projekte/viewProjektenglish?id_db=54

The user can query for publications by Peter Haase that have the word "semantic" in the title by composing the query:

getBindings([<peter haase> <publication> ?x] [?x <title> <semantic>])

which retrieves the following URIs.

http://www.aifb.uni-karlsruhe.de/Publikationen/viewPublikationOWL/id399instance
http://www.aifb.uni-karlsruhe.de/Publikationen/viewPublikationOWL/id449instance
http://www.aifb.uni-karlsruhe.de/Publikationen/viewPublikationOWL/id748instance
http://www.aifb.uni-karlsruhe.de/Publikationen/viewPublikationOWL/id1003instance

All of the above are OWL files corresponding to publications. The user can query for documents asserting the triples used to find the above bindings by using a query such as *getAssertingNodes([<peter haase> <publication> ?x]).* The query *getDocByKeywords* corresponds to straight-forward keyword search of HTML documents. The query *getDocByKeywords(peter haase)* retrieves 251 HTML documents. A Google search for "*peter haase*" retrieves 325 documents (with omitted results) on the AIFB website. But note that all of the AIFB web pages are not indexed and PDF documents and the likes are ignored. Moreover, SITAR indexes and retrieves RDF files too. SITAR allows users to simply enter a set of keywords which is then automatically plugged into the query *getDocsByIndexOrContent*. So, the query *peter haase* returns 299 documents which are a mix of HTML and OWL documents (it also retrieves a PDF document URI).

Note that in all of the above queries, the user is using intuitive keywords to explore the RDF data. The user is not aware of the underlying schema and hardly ever needs to know the exact URIs of the resources. The user however is required to have an idea of the underlying model. The idea is to retrieve data and document nodes from the same unified whole. As can be imagined, this will especially be useful when dealing with those documents that have both text and semantic markup. Such documents can be indexed using URIs, triples and text, and the *getLinkingNodes* and *getAssertingNodes* will play a major role in retrieving those documents.

These examples demonstrate that HQL can be used to query and retrieve RDF and HTML documents. Comparing RDF document retrieval with other search systems is pointless unless they provide a querying mechanism to retrieve RDF documents. Most of the search systems do not even handle XML properly let alone RDF.

5.2 Quantitative Analysis

SITAR allows informed users to use the wide range of queries that HQL offers to query and retrieve documents. We formulated ten different simple HQL queries and picked 50 XHTML documents from the SEAL dataset. The documents were picked semi-randomly in the sense that we picked documents using general keyword searches based upon each of the queries. We had to do this to make sure we had enough documents in the mix that are relevant to the queries. We then manually examined each of these documents and recorded which of the documents are relevant to each query. So in effect, we created a table which had the query on the left side and the number of the relevant documents on the right side.

We then ran each of the HQL queries against the dataset and, not surprisingly, found that both the *recall* and *precision* are 100% for each of the queries. The AIFB SEAL document set is such that a hyperlink is provided to the homepage of the entity whenever it is mentioned in a document. But what is interesting is the performance of plain text search for the same queries. We indexed the same set of documents using Lucene and tried to translate the queries into plain keyword queries. Table 1 shows a HQL query and related keyword queries and the resulting *recall* and *precision* values for them.

Table 1 Results

HQL Query	Keyword Query	Precision	Recall
<professor>:: <andreas>	Andreas	**0.42**	**0.91**
	Professor Andreas	**0.38**	**0.91**
	"Professor Andreas"	**0**	**0**
	Professor AND Andreas	**1**	**0.09**

Consider the first query where the user's information need is to retrieve documents about a professor whose name is Andreas. Note that there are documents talking about several different people named Andreas in the dataset. This need can be translated into plain keyword search in several different ways. The first keyword query *Andreas* is a very general one. It retrieves all the documents in which the term *Andreas* occurs and hence the *recall* value is very high (10/11 = **0.91**). But it suffers from poor *precision* (10/24 = **0.42**) since any document that mentions Andreas (irrespective of whether or not it is about the professor we want) is retrieved in response. The second query *Professor Andreas* has an implicit OR between Professor and Andreas. This query searches for documents that contain either the term *Professor* or the term *Andreas* and hence the number of returned documents is more than it was in the previous case. The *recall* in this case remains the same (10/11 = **0.91**) while the *precision* suffers further (10/26 = **0.38**). The third query, "*Professor Andreas",* encloses the previous query in

quotes. This essentially means that the system retrieves pages where the two terms occur in sequence. This turned out to be overly discriminating and returned zero documents. The fourth query *Professor AND Andreas* essentially retrieves documents that have *both* the keywords Professor and Andreas (compare it with query *Professor Andreas*). This again turned out to be overly discriminating. The *precision* was very high in this case but that's because only one document was returned which happened to be a relevant one. As a consequence the *recall* was very poor (1/11 = **0.09**). See [7] for additional queries.

The queries show the limitation of the standard keyword query language when it comes to expressing the user information need to the system. Even when the users know what they are looking for, they are unable to unambiguously convey it to the system due to the lack of expressive power. The above table also shows that the *precision* is usually very low. When *precision* is high, it is only because very few relevant documents were retrieved and hence *recall* suffers. The experiment also shows that documents can be annotated simply by adding hyperlinks from the document to a node which participates in a triple, and the *precision* improves compared to the standard keyword based search system if the user can express disambiguation information to the system. Note that the user can also combine wordset pairs with plain keywords and pose queries such as *getDocsByIndexAndContent* (*<professor>::<andreas> semantic grid*) which retrieves pages talking about professor Andreas and semantic grid. These hybrid queries combine the semantic search with plain keyword based search.

In order to illustrate how *recall* can improve in cases where synonymy is involved, we experimented with Wikipedia webpages involving the boxer Muhammad Ali alias *Cassius Clay*. See [7] for details.

The above queries are all the kind of queries that can be easily composed by laymen. Informed users can compose more advanced queries to Animal Kingdom ontology such as *getLinkingNodes(getBindings([?x <type> <mammal>]))* to do a broadening search resulting in higher *recall*. The result set of the above query will contain documents talking about cats, dogs, humans, horses, etc., even though these terms don't appear in the query or the document. In order to test this we ran the query *getLinkingNodes(getBindings([?x <type> <Person>::<Boxer>]))* which then returned all the documents talking about Muhammad Ali the boxer thereby resulting in 100% *recall*. In contrast, the keyword query *Boxer* was able to retrieve only seven documents out of twelve resulting in a *recall* of 58%. The user can pose a query such as *<Professor>::<?x>* to retrieve all entities of type Professor.

6 Related Research

Storing and retrieving RDF data is an area of research that has been well explored by researchers in the recent past [15, 16, 17, 18]. Well-known open source RDF storage systems include Jena/Jena2 [27], Redland [28], Sesame [29], KOAN/KOAN2 [30], rdfDB [31], RDFStore [32], Kowari [33], Boca [34], BRAHMS [35], etc., to name a few. Retrieving RDF data is typically viewed as a data retrieval problem and, not surprisingly, most of the query languages have the

SQL flavor [19]. When seen purely from the perspective of querying the RDF data, HQL is unique because it allows the users to explore the RDF graph even without any knowledge about the underlying schema (namespaces, exact URIs, ontologies, etc.). The user can use HQL to quickly get a feel for the underlying data.

Document retrieval systems retrieve documents based upon their annotations/descriptions [10, 19, 20, 21, 22, 23, 24], but none seems to aim at retrieving HTML, RDF *and* hybrid documents. We index a document based upon words, URIs, and triples that can be extracted from the document, and give the user a light-weight query language to retrieve documents based upon this information. The query language is *hybrid* in the sense that it has both "formal" and keyword components but what is *unique* is that the "formal" component itself is expressible using keywords.

Unlike our unified approach, Semantic Search [10] treats the SW and the HW as two separate repositories and aims at retrieving documents from the HW (in fact they use Google to search for documents). It lets the user communicate the disambiguation information using the user interface. Like SITAR, quizRDF[19] indexes a document (URL) using words obtained from its body as well as from the literals of triples in which its URL participates. Like Semantic Search, quizRDF too uses a GUI to let the user communicate disambiguation information.

SITAR views the SW and the HW as a unified whole (unlike Semantic Search). One benefit of this, compared to quizRDF, is that the URL of a document is also indexed by the anchor text words. Further, SITAR indexes a document using any URIs (*linksTo*) or triples (*asserts*) that can be extracted from the document. This allows it to index and retrieve RDF documents (and hybrid RDFa kind of documents in the future). Also, unlike the above two systems, the user can specify the disambiguation information (the "class") using word-set pairs, and use it in conjunction with *linksTo* information to retrieve documents. SITAR brings keyword-based searches to URIs.

Swoogle[21] specializes in retrieving ontology documents and URIs. It doesn't seem to index HTML documents or support triple search or keyword-based querying of the RDF graph.

OWLIR's[20] approach of treating a triple (that appears in the document) as an indexing term corresponds to what we are doing. But the way the indexing information is used and the nature of the query language is quite different. HQL is keyword-based and so the users can retrieve an "asserting" document even when the exact URIs are not known. Also, we index a document based upon the component URIs of the triples and the hyperlinks that appear in the document (*linksTo*).

There are several other systems [22, 23, 24, 25] that perform hybrid retrieval but our system is different due to the reasons discussed above and due to the fact that we view SW and HW as a single UW. In summary, we situate the SW data and the HW documents side by side and query the Unified Web using HQL which has both keyword and "formal" components. We also exploit existing hyperlink (*linksTo*) connections between HW documents and SW nodes while retrieving documents.

7 Conclusion and Future Work

Data retrieval aims at determining all objects that satisfy a semantically well-defined query, while information retrieval aims to decipher user information need and interpret document contents in order to satisfy a query. We have presented a distinctly unique method of retrieving Web documents and Semantic Web data in which Semantic Web data can be harnessed to enhance Web document retrieval and information from the Hypertext Web can be exploited to enable easy querying of the RDF graph. The Unified Web (UW) model conceptually unifies the Hypertext Web and the Semantic Web, and the Hybrid Query Language (HQL) combines Data and Information retrieval paradigms to access information from the Unified Web. In the absence of a benchmark to evaluate this novel approach, we have presented results obtained by experimenting with real-world AIFB SEAL data that adequately reflects our goals.

HQL is a light-weight, keyword-based query language that allows the users to query and explore the RDF graph even when no schema information is available. Specifically, the user need not know the exact URIs to retrieve RDF data. Thus, HQL combines the traditional keyword-based search with graph-based reasoning.

One of the fundamental ideas behind HQL is to index a URI using a set of keywords, which is a common notion in the literature. But because we position the RDF data in a web of hypertext documents, we have the freedom to exploit information from the hypertext documents (such as the anchor text) to enrich a URI's index words. At this level, we again see natural language induced problems such as synonymy, polysemy, etc (which only got pushed to a lower level). The resulting uncertainty necessitates ranking (not unlike what Swoogle [21] is doing). But, this is where the novel wordset pair queries enable disambiguation. This, in essence, is how ontologies can help in the retrieval of legacy documents. And this is where the "Semantic Web enabled Information Retrieval" starts deviating from traditional IR. Otherwise, we are simply pushing the problem of keyword-based document retrieval to the level of URIs (we have simply reduced the size of a typical term vector) and there is nothing "semantic" about it – *jaguar* will retrieve both the car and animal URIs in spite of "meaningful" label-literals. Another important contribution of this work is the idea of using hyperlinks as semantic annotations.

SITAR can be further extended by ranking URIs and documents. Even though Lucene does rank URIs (SITAR stores a URI in a Lucene document that is indexed by the index words), and of course, documents, we need a ranking algorithm that is based on *linksTo* relationship among others, in future.

In future, the hybrid retrieval system can be made scalable for larger datasets by reimplementing it using MapReduce ideas and Hadoop infrastructure [37, 38]. RDF can be gleaned from RDFa or microformat-enriched documents using GRDDL [39, 40]. Heuristics for extracting meaningful keywords from URLs can be improved. To make the result sets more manageable, better techniques for ranking and organizing URLs should take into account the link semantics and their relationships respectively [41].

References

1. Semantic Web Activity page, http://www.w3.org/2001/sw/
2. Prud'hommeaux, E., Seaborne, A. (eds.): SPARQL Query Language for RDF, [W3C WD] (October 2006), http://www.w3.org/TR/rdf-sparql-query/
3. Adida, B., Birbeck, M. (eds.): "RDFa," [W3C WD] (2006), http://www.w3.org/TR/xhtml-rdfa-primer/
4. Immaneni, T., Thirunarayan, K.: Hybrid Retrieval from the Unified Web. In: Proceedings of the 22nd ACM Symposium on Applied Computing, Semantic Web and Applications Track (ACM SAC 2007), pp. 1376–1380 (March 2007)
5. Immaneni, T., Thirunarayan, K.: A Unified approach To Retrieving Web Documents and Semantic Web Data. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 579–593. Springer, Heidelberg (2007)
6. Immaneni, T.: A Hybrid Approach to Retrieving Web Documents and Semantic Web Data. Doctoral Dissertation, Department of Computer Science and Engineering, Wright State University, Dayton, OH (October 2007)
7. Thirunarayan, K.: On Embedding Machine-Processable Semantics into Documents. IEEE Transactions on Knowledge and Data Engineering 17(7), 1014–1018 (2005)
8. Periodic Table in OWL, http://www.daml.org/2003/01/periodictable/
9. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In: Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms (1998)
10. Guha, R., McCool, R., Miller, E.: "Semantic Search". In: Proceedings of the 12th International Conference on World Wide Web, May 2003, pp. 700–709. ACM Press, New York (2003)
11. Apache Lucene, http://lucene.apache.org/
12. Hartmann, J., Sure, Y.: An Infrastructure for Scalable, Reliable Semantic Portals. IEEE Intelligent Systems 19(3), 58–65 (2004)
13. CyberNeko HTML Parser, http://people.apache.org/~andyc/neko/doc/html/
14. Jena ARP, http://www.hpl.hp.com/personal/jjc/arp/
15. Beckett, D.: SWAD-E Deliverable 10.2: Mapping Semantic Web Data with RDBMSes (2003), http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/
16. Beckett, D.: SWAD-Europe Deliverable 10.1: Scalability and Storage: Survey of Free Software / Open Source RDF storage systems (2002), http://www.w3.org/2001/sw/Europe/reports/rdf_scalable_storage_report/
17. Bailey, J., Bry, F., Furche, T., Schaffert, S.: Web and Semantic Web Query Languages: A Survey. In: Eisinger, N., Maluszynski, J. (eds.) Reasoning Web. LNCS, vol. 3564, pp. 35–133. Springer, Heidelberg (2005)
18. Haase, P., Broekstra, J., Egerhart, A., Volz, R.: A Comparison of RDF Query Langauges. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 502–517. Springer, Heidelberg (2004)
19. Davies, J., Weeks, R., Krohn, U.: QuizRDF: Search Technology for the Semantic Web. In: Workshop on Real World RDF and Semantic Web Applications, Proceedings of WWW 2002, Hawaii, USA (2002)
20. Mayfield, J., Finin, T.: Information Retrieval on the Semantic Web: Integrating Inference and Retrieval. In: Proceedings of the SIGIR 2003 Semantic Web Workshop, pp. 461–468 (2003)

21. Ding, L., et al.: Finding and Ranking Knowledge on the Semantic Web. In: Proceedings of the 4th International Semantic Web Conference, November 2005, pp. 156–170 (2005)
22. Rocha, C., Schwabe, D., Aragao, M.P.: A Hybrid Approach for Searching in the Semantic Web. In: Proceedings of the 13th International World Wide Web Conference, New York, May 2004, pp. 374–383 (2004)
23. Zhang, L., Yu, Y., Zhou, J., Lin, C., Yang, Y.: An Enhanced Model for Searching in Semantic Portals. In: Proceedings of the 14th International World Wide Web Conference, May 2005, pp. 453–462. ACM Press, Chiba (2005)
24. Vallet, D., Fernández, M., Castells, P.: An Ontology-Based Information Retrieval Model. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 455–470. Springer, Heidelberg (2005)
25. Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., Petrelli, D.: Hybrid Search: Effectively Combining Keywords and Semantic Searches. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 554–568. Springer, Heidelberg (2008)
26. Hausenblas, M., Herman, I., Adida, B.: RDFa—Bridging the Web of Documents and the Web of Data. Tutorial given at: The 7th International Semantic Web Conference, Karlsruhe, Germany (October 2008)
27. Jena/Jena2, http://jena.sourceforge.net/
28. Redland, http://librdf.org/
29. Sesame, http://www.openrdf.org/
30. KOAN/KOAN2, http://kaon2.semanticweb.org/
31. rdfDB, http://www.guha.com/rdfdb/
32. RDFStore, http://rdfstore.sourceforge.net/
33. Kowari, http://www.kowari.org/
34. Boca, http://ibm-slrp.sourceforge.net/
35. BRAHMS, http://lsdis.cs.uga.edu/projects/semdis/brahms/
36. Thirunarayan, K., Immaneni, T.: Hybrid Retrieval of Hypertext Web Documents and Semantic Web Data (submitted to Journal)
37. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM 51(1), 107–113 (2008)
38. Hadoop, http://hadoop.apache.org/core/
39. RDFa vs Microformats, http://evan.prodromou.name/RDFa_vs_microformats
40. GRDDL, http://www.w3.org/TR/grddl/
41. Thirunarayan, K., Verma, R.: A Framework for Trust and Distrust Networks. In: Proceedings of Web 2.0 Trust Workshop (W2Trust) (June 2008)

Bipolar Queries: A Way to Enhance the Flexibility of Database Queries

Sławomir Zadrożny and Janusz Kacprzyk

Abstract. In many real life scenarios the use of standard query languages may be ineffective due to the difficulty to express the real user requirements (information needs). The use of fuzzy logic helps to fight this ineffectiveness making it possible to model and properly process linguistic terms in queries. This way a user may express his or her requirements in a more intuitive and flexible way. Recently another dimension of such a flexibility attracted the attention of many researchers. Namely, it is now widely advocated that by specifying his or her requirements the user is usually having in mind both negative and positive preferences. Thus, a combination of an intuitive appeal of natural language terms in queries with a bipolar nature of preferences seems to be a next promising step in enhancing the flexibility of queries. We look at various ways of how to understand bipolarity in database querying, propose fuzzy counterparts of some crisp approaches and study their properties.

1 Introduction

Databases are indispensable for the functioning of modern societies. They have been in use for many years now, but only relatively recently they have found a widespread use among novice users having a limited computer literacy. In particular, the Internet provides a platform for many applications such as hotel booking systems that rely on the database access. The effectiveness and efficiency of these applications may be considered in two dimensions. First, traditionally considered aspects such as response time are of utmost importance. They are addressed using better data storage and retrieval techniques, faster hardware, etc. Another dimension is related

Sławomir Zadrożny
Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warszawa, Poland
e-mail: Slawomir.Zadrozny@ibspan.waw.pl

Janusz Kacprzyk
Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warszawa, Poland
e-mail: Janusz.Kacprzyk@ibspan.waw.pl

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 49–66.
springerlink.com

to the ease of expressing the requirements of a user searching for information in a database. In order to save the user from intricacies of the traditional database querying languages, some user friendly interfaces are usually provided. Such interfaces make it *technically* easier to form a query by the use of typical controls and toolbars. However, behind the scene, such a query have to be translated into a query language supported by a database in use, most often to SQL. This may be a sufficient solution in case of search criteria that are clear-cut as, e.g., in case of the search for name of a specific product at the Internet shopping site. On the other hand in many applications, such as related to hotel booking or real estate property search, the criteria of the user may be rather vague. This is mostly due to their original form in the mind of the user which is a natural language expression as in the query: "Find *cheap* houses located *near* a railway station". In order to support such queries a proper modelling of natural language terms, such as "cheap" and "near" in the example above mentioned, have to be provided and an essential extension to the very database querying has to be implemented.

Fuzzy logic seems to offer an excellent solution in this respect and research in this area has a long history. The reader is referred to a recently published handbook (cf. Galindo [14]) and, in particular, to a survey of the research on the flexible, fuzzy querying of databases therein (cf. Zadrożny, De Tré, De Caluwe and Kacprzyk [24]). Thus, the user does not have to translate his or her requirements from the "internal" natural language form but can express them more directly via a flexible fuzzy query. This way a kind of a *conceptual* ease of use is achieved.

Recently, in addition to vagueness also the *bipolarity* of information and a need to take it into account are getting more and more advocated. We are here interested in the bipolarity of requirements (preferences) of the user looking for information in a database. This may be understood in a few slightly different ways which we will briefly discuss later on. Now let us adopt a general, database query oriented view on this feature of information.

Basically, a database query may be identified with a *condition* that the data sought should satisfy. Such a condition may often have a complex structure and comprise of some atomic conditions combined using logical connectives. Various extensions has been proposed in the literature in the framework of flexible querying. As mentioned above, the use of *fuzzy predicates* modelling *linguistic terms* in conditions has been advocated (cf., e.g., Zadrożny, De Tré, De Caluwe and Kacprzyk [24], Bosc [5], Kacprzyk and Zadrożny [15]). Also the assignment of the *importance weights* to particular parts of the condition has been proposed and studied (cf., e.g., Dubois and Prade [9]).

In case of bipolarity it is assumed, which is well founded on results of psychological research, that the user has in mind in fact two types of conditions:

- *hard constraints* which have to be met by the data sought, and
- just *preferences* making it possible to differentiate among the data items meeting the above mentioned hard constraints.

Such conditions, if they are crisp, define therefore two sets of data items:

- *rejected*, *infeasible*, etc., or, equivalently, taking a complement of the former, *acceptable*, *satisfactory*, *feasible* etc., and
- *preferred*, *desired*, etc.

Thus the former conditions provide the *negative* information indicating what should be avoided, while the latter provide the *positive* information indicating what is really preferred. This is the motivation for the use of the term "bipolar" to characterize queries comprising both types of conditions.

The bipolar queries (or, more generally, bipolar preferences) may be studied from different perspectives – some of them are presented in Section 2. For example, one may primarily look for a very important aspect of their proper theoretical modelling (cf., e.g., Dubois [11], Benferhat [1]) or, more specifically, for the ways to properly combine several preferred (positive) conditions (cf., e.g., Bosc and Pivert [3, 4]). On the other hand, one may focus on the question of a proper combination of both types of conditions in case they are fuzzy rather than crisp.

The following general interpretation of the bipolar queries may better explain the problem. Both the negative (required) and positive (preferred) conditions are treated as atomic. The data items sought have to satisfy the former unconditionally, while the latter is of somehow secondary importance. The relation between these two types of conditions may be meant in various ways and leads to different interpretations of the bipolar queries. The simplest approach is to use the positive conditions just to order the data items which satisfy the negative conditions. However as soon as the negative conditions are fuzzy, i.e., may be satisfied *to a degree*, it is not at all obvious what their satisfaction should mean. Thus, we focus on the question how the satisfaction degrees of both the negative and positive conditions may be combined.

From the adopted perspective, the way these conditions are jointly taken into account may be treated as the question of a definition/selection of an appropriate aggregation operator to be applied. In the literature this problem has been studied under different names, and sometimes in slightly different contexts, by many authors. In the framework of database querying the paper by Lacroix and Lavency [16] was first to propose such type of queries and triggered the interest of other researchers. This has led to the development of a more general concept of a *query with preferences* and a corresponding new relational algebra operator, *winnow*, introduced by Chomicki [6, 7]. Both Lacroix and Lavency's as well as Chomicki's approach deal with crisp conditions only. In Zadrożny [23] we propose a direct "fuzzification" of the approach by Lacroix and Lavency and in Zadrożny and Kacprzyk [26] we study some properties of this solution. In Zadrożny and Kacprzyk [25] we discuss a similar approach with respect to Chomicki's winnow operator. Here we further develop this line of research and present it in a unified framework.

2 On Some Related Works

Here we briefly review research on the bipolar queries of databases. Various approaches mentioned bear different names but they may be easily shown to deal with the bipolar information, as discussed in Section 1. The name "bipolar queries" seems

to be proposed and used for the first time by Dubois and Prade [10]. However we will start with a much earlier approach which triggered the interest of the database community in this type of queries.

We will consider in a usual way a relation as a data structure to be searched through. Let $T = \{t_j\}$ denote a set of tuples of this relation.

Lacroix and Lavency [16] were the first to propose the use of a query comprising two categories of conditions: one which is mandatory (C) and another which expresses just mere preferences (desires) (P). The bipolarity of these conditions becomes evident when one adopts the following interpretation. The former condition C may be seen as expressing the *negative* preferences: the tuples which do not satisfy it are definitely not matching the whole query. The latter condition P, on the other hand, expresses the *positive* preferences: a tuple satisfying it is preferred over another tuple not satisfying it, provided both tuples satisfy the mandatory condition C. These conditions will be referred to as a positive and negative condition, for short.

We will identify the negative and positive condition of a bipolar query with the predicates that represent them and denote them as C and P, respectively. For a tuple $t \in T$, $C(t)$ and $P(t)$ will denote that the tuple t satisfies the respective condition. Then, a bipolar query may be expressed in natural language as follows:

"Find tuples t satisfying C *and possibly* P"

The bipolar queries may be exemplified by:

"Find a house cheaper than USD 250,000 *and possibly* located not more than two blocks from a railway station" (1)

Here the negative condition excludes houses more expensive than USD 250,000 and the positive condition favors houses located closer to a railway station. Such a query may be more formally written as

$$C \text{ and possibly } P$$

or, equivalently, an answer to a bipolar query may be defined as the following set of tuples:

$$\{t : C(t) \textit{ and possibly } P(t)\} \tag{2}$$

The above form puts emphasis on the question of a proper modelling of the aggregation of both types of conditions, which is expressed here with the use of the "and possibly" operator. This operator was used in a different context by Yager [21, 20], and Bordogna and Pasi [2].

According to the original (crisp) approach by Lacroix and Lavency [16] such an operator has an important property: the aggregation result depends not only on the explicit arguments, i.e., $C(t)$ and $P(t)$, but also on the content of the database. If there are no tuples meeting both conditions then the result of the aggregation is determined by the negative condition C alone. Otherwise the aggregation becomes a regular conjunction of both conditions. This dependence is best expressed by the following logical formula [16] (showing, by the way, that in case of crisp conditions

C and P, a bipolar query is easily expressible in the classical querying formalism of the relational data model, i.e., in the relational calculus):

$$C(t) \textit{ and possibly } P(t) \equiv C(t) \wedge \exists s(C(s) \wedge P(s)) \Rightarrow P(t) \tag{3}$$

Lacroix and Lavency [16] consider only the case of crisp conditions C and P. Then this important property is preserved if the "first select using C and then order using P" understanding of the bipolar query is adopted, i.e., the answer to the bipolar query (C,P) is generated as follows:

- find tuples satisfying C,
- order them according to their satisfaction degree of P.

This understanding is predominant in the literature dealing with fuzzy extensions of the original concept of Lacroix and Lavency. Both, direct extensions proposed by Bosc and Pivert [3, 4] as well as a more sophisticated possibility theory based interpretation of this concept by Dubois and Prade [11] focus, in fact, on the proper treatment of *multiple* required and preferred conditions, basically assuming the above strategy as the way of combining the negative and positive conditions.

The research on such fuzzy compound conditions may be well illustrated with the papers by Bosc and Pivert [3, 4] and Dubois and Prade [10]. First, let us briefly recall an approach to the aggregation of multiple positive conditions proposed for the crisp case by Lacroix and Lavency [16]. They consider the case in which there is a set $\{P_i\}$ of preferred (positive) conditions rather than just one, which may be formally written as

$$C \text{ and possibly } \{P_i\} \tag{4}$$

The conditions P_i are meant to be combined in a non-standard way, i.e., are not treated as a Boolean combination. Lacroix and Lavency [16] proposed a few ways to aggregate them which are interesting from a practical point of view. Basically two types of such aggregation operators are considered: one based on the cardinality of the set of satisfied conditions P_i, and one based on a varying importance of these conditions. In the former case, a tuple satisfies a query (4) if:

- it satisfies the required condition C, and
- there is no tuple s satisfying C and more conditions P_i that t satisfies.

In the latter case the positive conditions are assumed to be linearly ordered and a tuple t satisfies a query (4) if:

- it satisfies the required condition C, and
- there is no tuple s satisfying C and a condition P_i, while $\neg P_i(t)$ and $P_j(t) \equiv P_j(s)$ for all $j < i$.

For both types of such compound positive conditions Lacroix and Lavency define an equivalent query in the relational calculus, in the spirit of (3).

Bosc and Pivert [3] study fuzzy counterparts of these types of compound positive conditions. For the cardinality based combination they consider a fuzzy set H_t of positive conditions P_i satisfied by a given tuple t, where H_t's membership function

is defined as $\mu_{H_t}(P_i) = P_i(t)$; remember that P_i is now a fuzzy condition, $P_i(t) \in [0,1]$ denotes its satisfaction degree by the tuple t and a tuple satisfies (matches) the whole bipolar query to *a degree*. Then, the scalar cardinality (the so-called ΣCount) of the fuzzy set H_t is used as the matching of degree of the tuple t with respect to the combination of the positive conditions P_i (after a proper normalization). Bosc and Pivert also propose a fuzzy counterpart of the importance based combination of the positive conditions introducing a Hierarchical Combination Operator; for details see [4]. Anyway, in their approach to the evaluation of the overall bipolar query with fuzzy negative and positive conditions Bosc and Pivert follow the already mentioned rule "first select using C and then order using P".

Dubois and Prade [10] define a bipolar query as a set of pairs (C_i, P_i) of, respectively, negative and positive conditions imposed on values of selected attributes $\{A_i\}_{i=1,k}$. These conditions may be identified with fuzzy sets defined in the domain of a given attribute. Some of the conditions/sets C_i may be equal to the domain of an attribute A_i, i.e., for a given attribute there is no negative condition. On the other hand, some of the conditions/sets P_i may be empty, i.e., there is no positive condition for a given attribute. These pairs of conditions are combined to yield overall conditions C and P as follows:

$$(C,P) = (\times_i C_i, +_i P_i)$$

where $\times_i C_i = C_1 \times C_2 \times \ldots \times C_k$, $+_i P_i = (P_1^c \times P_2^c \times \ldots \times P_k^c)^c$ and X^c is a complement of the set X. Thus, the overall negative condition is obtained via the conjunction of all negative conditions concerning particular attributes while the overall positive condition is obtained via the disjunction of all positive conditions concerning particular attributes. In case of this, a more evident, equivalent formula for the pair of overall conditions is:

$$(C(t), P(t)) = (\min_i C_i(t), \max_i P_i(t))$$

A pair of the combined conditions (C,P) is then used according to the aforementioned principle "first select using C and then order using P". This principle is implemented via the use of the lexicographic order $\preceq$ of tuples against the bipolar query, i.e., $t_1 \preceq t_2 \iff (C(t_1) < C(t_2)) \vee ((C(t_1) = C(t_2)) \wedge (P(t_1) \leq P(t_2)))$.

Dubois and Prade [10] consider also non-Boolean combinations of the set of positive conditions P_i. An evaluation of a bipolar query proceeds then as follows. Each tuple t is represented by a vector: $(C(t), P_{\sigma(1)}(t), \ldots, P_{\sigma(n)}(t))$ where σ is a permutation of the positive conditions P_i such that $P_{\sigma(1)}(t) \geq \ldots \geq P_{\sigma(n)}(t)$. Then, the lexicographic order of these vectors is used to rank-order the tuples. Thus, the *leximax* operator (cf., e.g., Dubois, Fargier and Prade [8]) is used here with respect to the positive conditions.

There is also another line of research pursued by Dubois and Prade [12, 11] which aims at providing a formal, possibility theory based framework for dealing with bipolar queries. Namely, two possibility distributions π and δ are assumed to represent query conditions (the user's preferences). The former corresponds to the

negative condition, i.e., $\pi(t) = 1$ and $\pi(t) = 0$ mean, respectively, that a tuple t is totally acceptable and totally unacceptable, while the intermediate values of $\pi(t)$ express an intermediate degree of acceptability. The latter possibility distribution δ represents the positive condition: $\delta(t) = 1$ denotes the maximum degree of preference (desirability) of t but $\delta(t) = 0$ means merely that t is not specifically preferred. Both types of conditions are then syntactically represented in the framework of *possibilistic logic* (cf. Dubois [11]). The possibility distributions π and δ are generated by sets of formulas defining the negative and positive conditions, accompanied by the constraints on the minimum value of the *necessity* and *guaranteed possibility* measures, respectively.

The bipolar queries may be also seen a special case of queries which employ the concept of a *non-dominance relation*, which has been deeply studied in the context of decision making for many years. Such a general class of queries has been proposed recently, for the crisp case, by Chomicki [6] under the name of *queries with preferences*. In this approach a new relational algebra operator, called *winnow*, is introduced. This unary operator selects from a set of tuples those which are *non-dominated* with respect to a given *preference relation*, a binary relation defined on the set of tuples. A bipolar query may then be obtained using a proper combination of the *select* operator with the winnow operator. The negative conditions define the select operator while the positive conditions are expressed by the preference relation. Thus the new winnow operator may be easily combined with the traditional relational algebra operators. In the crisp case, similarly to the case of queries studied by Lacroix and Lavency [16], this combination is quite straightforward, somehow still corresponding to the "first select and then order" strategy, mentioned earlier. However in the fuzzy case such an approach becomes problematic and some special measures are needed. A possible approach, both in the case of bipolar queries in the sense of Lacroix and Lavency and in a more general context of Chomicki's query with preferences, is presented in the next section.

Recently, some attempts were undertaken to include the support for bipolar queries in the well-known fuzzy querying languages. For example, in Lietard [17] a preliminary approach related to the SQLf language (cf. Bosc [5]) is reported.

3 Fuzziness and Bipolar Queries: An Approach

Here we follow the Lacroix and Lavency [16] original approach to bipolar queries and extend it to the case of fuzzy conditions. Moreover, we also propose a fuzzy version of the winnow operator and show its relation to "fuzzy" bipolar queries.

3.1 Fuzzification of the Lacroix and Lavency Approach

We start with the concept of a bipolar query exemplified by (1) and formalized by (2) and (3). In a more realistic case the user will prefer to express the conditions in a query like (1) using fuzzy predicates what may result in the following query:

"Find a *cheap* house *and possibly* located *near* a railway station (5)

Let us remind that, according to the original interpretation of Lacroix and Lavency, (5) should be understood in such a way that we are looking for a house that:

– has to be *cheap*,
– if there is a cheap house near the railway station then other, just cheap houses are of no interest.

Notice that now the strategy of the bipolar query evaluation "first select using negative condition (here: cheap) and then order using the positive condition (here: near the station)" cannot be directly applied. Namely, it is not any longer clear what it should mean that a tuple (house) satisfies the negative condition (is cheap) as the satisfaction of this condition is now a matter of the degree. Let us illustrate the problem on a simple example. Let there be a house $H1$ definitely cheap (to a degree 1), but rather away from the station (near to a degree 0.2) and another house $H2$, still cheap but not that much as house $H1$ (to a degree 0.9), but located quite close to the station (to a degree 0.9). Now there is a question which of these two houses should belong to the answer to the query (5)? The "first select then order" strategy could be now implemented by the lexicographic order on the vectors of the satisfaction degrees representing both houses. For house $H1$ we have a vector $[1.0, 0.2]$ and for $H2$ a vector $[0.9, 0.9]$. Thus the lexicographic order indicates $H1$ as better than $H2$ what may be questionable, at least in certain scenarios. Moreover, the lexicographic order (nor any other order) does not give us any scalar measure of query matching by particular tuples, thus it does not help much in deciding which tuples should really form the answer to the query in question, what may be important in some applications.

Looking for a consistent solution to this problem we start with the formula (3) and interpret it in terms of fuzzy logic. Firstly, let us rewrite (3) using standard fuzzy counterparts of the logical connectives appearing there. Moreover, we will write it as a formula of the membership function of the resulting fuzzy set $ans(C,P,T)$ of tuples forming the answer to the bipolar query (C,P) against a set of tuples T as:

$$\mu_{ans(C,P,T)}(t) = \min(C(t), \max(1 - \max_{s \in T} \min(C(s), P(s)), P(t))) \tag{6}$$

Symbol T appears in $ans(C,P,T)$ to emphasize that the membership degree (matching degree) of a tuple t depends not only on this tuple itself and on the conditions C and P but also on the current whole set of tuples T – according to the semantics of bipolar queries in the sense of Lacroix and Lavency.

The *matching degree* of a tuple against a bipolar query is meant as the truth value of the formula (3), computed in the framework of fuzzy (multivalued) logic using right-hand side of the formula (6). Thus, the evaluation of a bipolar query produces a fuzzy set of tuples, where the membership function value for a tuple t corresponds to the matching degree of this tuple against the query. The answer to a bipolar query is then a list of the tuples, non-increasingly ordered according to their membership degree.

In the formula (6) the min, max and $1-x$ operators are used to model the connectives of conjunction, disjunction and negation, respectively. Moreover, the implication connective $\Rightarrow$ is modelled by the Kleene-Dienes implication operator (cf., e.g., [13]) and the existential quantifier $\exists$ is modelled via the maximum operator. We discuss the alternative choices of these operator later in this section.

The formula (6) has been proposed by Yager [21, 20, 22] for an aggregation operator in the context of the multicriteria decision making for the case of so-called *possibilistically qualified criteria*. Yager [22] intuitively characterizes a possibilistically qualified criterion as such which should be satisfied unless it interferes with satisfaction of other criteria. This is in fact the essence of bipolar queries in the sense advocated here. This concept was also applied by Bordogna and Pasi [2] for the information retrieval task.

In fact Dubois and Prade [10] considered a similar formula too. However their version of (6) employs an arbitrary parameter (instead of $\max_{s\in T}\min(C(s),P(s))$ in (6)) what makes the results obtained for a certain specific range of values $(C(t),P(t))$ difficult to justify. In the current approach this expression has a meaningful interpretation providing some justification for this behavior; cf. Zadrożny [23] for details.

Now let us look at the formula (6) again. This is definitely only one of possible ways to "fuzzify" the original formula (3) proposed by Lacroix and Lavency [16]. In particular, different interpretations of the conjunction and implication connectives may be employed. Moreover, also the disjunction connective may be seen as related to the existential quantifier and its different possible forms also should be taken into account. Basically, various t-norms, t-conorms and implication operators (cf., e.g., [13]) may be assumed to model corresponding logical connectives in the framework of fuzzy logic based interpretation of formula (3).

In particular one may consider so-called De Morgan Triplets $(\wedge,\vee,\neg)$ that comprise a t-norm operator $\wedge$, a t-conorm operator $\vee$ and a negation operator $\neg$, where $\neg(x\vee y)=\neg x\wedge\neg y$ holds. The following three De Morgan Triplets play the most important role in fuzzy logic (cf., e.g., Fodor and Roubens [13] for a justification) $(\wedge_{min},\vee_{max},\neg)$, $(\wedge_{\Pi},\vee_{\Pi},\neg)$, $(\wedge_{W},\vee_{W},\neg)$, where the particular t-norms and t-conorms are defined as follows

$$
\begin{array}{lll}
x\ \wedge_{min}\ y = \min(x,y) & & \textit{minimum}\\
x\ \wedge_{\Pi}\ y = x\cdot y & & \textit{product}\\
x\ \wedge_{W}\ y = \max(0,x+y-1) & & \textit{Łukasiewicz t-norm}\\
x\ \vee_{max}\ y = \max(x,y) & & \textit{maximum}\\
x\ \vee_{\Pi}\ y = x+y-x\cdot y & & \textit{probabilistic sum}\\
x\ \vee_{W}\ y = \min(1,x+y) & & \textit{Łukasiewicz t-conorm}
\end{array}
$$

The negation operator $\neg$ in case of all the above De Morgan Triplets is defined as: $\neg x=1-x$.

Due to their associativity, both t-norms and t-conorms may be employed as the m-ary operators, i.e., it is well defined what $x\wedge y\wedge\ldots$ and $x\vee y\vee\ldots$ mean.

Usually the general and existential quantifiers are identified in fuzzy logic, for the case of a finite universe, with the maximum and minimum operators,

respectively. Namely, the following identities are employed: $\text{truth}(\forall x A(x)) = \min_x \mu_A(x)$ and $\text{truth}(\exists x A(x)) = \max_x \mu_A(x)$. As soon as we consider the use of other t-norms and t-conorms to "fuzzify" formula (3) it is reasonable to look for a consistency via the use of an appropriate *t-quantifiers* and *s-quantifiers*; cf., e.g., [18]. Thus we adopt the following definitions:

$$\text{truth}(\forall x A(x)) = \mu_A(a_1) \wedge \mu_A(a_2) \wedge \ldots \wedge \mu_A(a_m) \tag{7}$$

$$\text{truth}(\exists x A(x)) = \mu_A(a_1) \vee \mu_A(a_2) \vee \ldots \vee \mu_A(a_m) \tag{8}$$

The particular t- and s-quantifiers will be denoted by the $\forall$ and $\exists$ symbol with a subscript indicating the underlying t- or s-norm. For example, $\exists_{max}$ denotes the "standard" fuzzy existential quantifier which is obtained when the maximum t-conorm is used.

There are two most popular ways of deriving an implication operator with respect to a given De Morgan Triple $(\wedge, \vee, \neg)$, namely so-called S-implications and R-implications defined as follows:

$$R - \text{implication:} \quad x \rightarrow y = \sup\{z : x \wedge z \leq y\} \tag{9}$$

$$S - \text{implication:} \quad x \rightarrow y = \neg x \vee y \tag{10}$$

Thus, for the particular De Morgan Triplets one obtains the following R-implication operators:

$$\text{Gödel's implication} \qquad x \rightarrow_{R-min} y = \begin{cases} 1 & \text{for } x \leq y \\ y & \text{for } x > y \end{cases}$$

$$\text{Goguen's implication} \qquad x \rightarrow_{R-\Pi} y = \begin{cases} 1 & \text{for } x = 0 \\ \min\{1, \frac{y}{x}\} & \text{for } x \neq 0 \end{cases}$$

$$\text{Łukasiewicz' implication} \qquad x \rightarrow_{R-W} y = \min(1 - x + y, 1)$$

and the following S-implication operators:

$$\text{Kleene–Dienes' implication} \qquad x \rightarrow_{S-max} y = \max(1 - x, y)$$

$$\text{Reichenbach's implication} \qquad x \rightarrow_{S-\Pi} y = 1 - x + x \cdot y$$

The S-implication operator $\rightarrow_{S-W}$ is identical with $\rightarrow_{R-W}$.

Let us write a fuzzy version of (3) in a more general form than (6), where a specific t-norm and other related operators were assumed:

$$\mu_{ans(C,P,T)}(t) = C(t) \wedge (\exists_{\vee} s \in T \ (C(s) \wedge P(s)) \rightarrow P(t)) \tag{11}$$

In order to simplify the notation let us fix C, P and T in the formula (6) and denote its version for a given De Morgan Triple, its related R or S implication and a corresponding existential quantifier as, respectively, $\gamma_{\wedge,R}$ and $\gamma_{\wedge,S}$. Thus, for example, $\gamma_{\min,S}(t) = \mu_{ans(C,P,T)}(t)$ denotes the original version of the formula (6).

In Table 1 various emerging interpretations of the formula (6) are shown.

Table 1 Right-hand side of formula (6) for different interpretations of the logical connectives

$\gamma_{\wedge,\cdot}$	**Resulting form of the formula (6)**
$\gamma_{min,S}$	$\min(C(t), \max(1 - \max_{s\in T} \min(C(s), P(s)), P(t)))$
$\gamma_{min,R}$	$\begin{cases} C(t) & \text{if } \max_{s\in T} \min(C(s), P(s)) \leq P(t) \\ \min(C(t), P(t)) & \text{otherwise} \end{cases}$
$\gamma_{\Pi,S}$	$C(x) \cdot (\prod_i (1 - C(y_i) \cdot P(y_i)) \cdot (1 - P(x)) + P(x))$
$\gamma_{\Pi,R}$	$\begin{cases} C(t) & \text{if } \exists_{\Pi}(C(s_i) \cdot P(s_i)) = 0 \\ C(t) \cdot \min(\frac{P(t)}{\exists_{\Pi}(C(s_i) \cdot P(s_i))}, 1) & \text{otherwise} \end{cases}$
γ_W	$C(t) \wedge_W (\exists_W (C(s) \wedge_W P(s)) \rightarrow_W P(t))$

The issue of how to appropriately model the logical connectives (including the existential quantifier) in formula (3) may be basically approached in two ways. First, one may look for some axioms that should characterize the behavior of this formula and try to check which operators modelling the connectives provide for the expected behavior (in what follows we will refer to the operators modelling the logical connectives as *logical operators*). The second, more modest, approach consists in studying the properties of (11) under different choices of logical operators. In the framework of the former approach we discuss below a property which should seem to be reasonable and which eliminates a class of the logical operators. We also show a few properties in the vein of the second approach.

The property which we impose on any bipolar query evaluation scheme, implied by a choice of logical operators in (11), may be – in a bit informal way – expressed as follows. It should not be the case that a tuple t satisfying very well the negative condition (let $C(t) = 1$) and not satisfying at all the positive condition (i.e., $P(t) = 0$) is indicated by the evaluation scheme as inferior to any tuple s satisfying both conditions to an infinitely small, but greater than 0, extent (i.e., $C(s) = P(s) = \varepsilon$, $\varepsilon \in [0, 1]$ and ε is a very small number).

It may be easily shown that for a t-norm without zero divisors (exemplified by the minimum and product t-norms, $\wedge_{min}$ and $\wedge_{\Pi}$) and related R-implication operator this property does not hold. This may be shown as follows. Let us assume that

$$\exists_{\vee} s(C(s) \wedge P(s)) > 0$$

Then $\exists_{\vee} u(C(u) \wedge P(u)) \rightarrow_{\wedge,R} P(t)$ is equal 0 for t such that $C(t) = 1$ and $P(t) = 0$, while it is greater than 0 for s such that $C(s) = P(s) = \varepsilon$. Then the right-hand side of the formula (11) is equal 0 for t and greater than 0 for S (the t-norm is assumed to have no zero divisors) and thus a tuple s emerges as preferred to t.

In the framework of the second approach to the choice of the logical operators, mentioned above, we have shown some properties in Zadrożny and Kacprzyk [26]. First, we note two general properties, valid for any combination of a t-norm,

t-conorm and S-implication or R-implication. Namely, if there exists a tuple t such that $C(t) = 1$ and $P(t) = 1$, then (11) turns into $C(t) \wedge P(t)$, where $\wedge$ is represented by a given t-norm. Thus, whatever the choice of the logical operators is, a characteristic feature of bipolar queries is preserved: if there is a tuple satisfying both the required and preferred conditions, then (11) turns into a conjunction. On the other hand, if for a tuple $t \in T$ $P(t) = 1$, then the formula (11) turns into $C(t)$. This property is implied by the characteristic features of the t-norm and implication operators: $x \rightarrow 1 = 1$ and $x \wedge 1 = x$, for any choice of operators $\wedge$ and $\rightarrow$ (cf., e.g., [13]). Thus, if a tuple fully satisfies the positive condition P, then its overall matching degree is equal to its satisfaction of the negative condition C.

The most important question concerning the choice of the logical operators is the following: does this choice influence the resulting order of the tuples in the answer to a bipolar query ?

Contributing to an answer to this question we notice the following properties. First, as to the property concerning the choice of the existential quantifier model, the usual fuzzy existential quantifier $\exists_{max}$: (A) yields greater or equal matching degrees, and (B) may change the resulting ordering of the tuples, when used instead of the $\exists_{\Pi}$ or $\exists_{W}$ quantifiers. Property A is a direct consequence of the fact that the maximum operator is the smallest of all t-conorms and that implication operators and t-norms are monotonic. Property B may be shown on an example; cf. Zadrożny and Kacprzyk [26].

Second, we have shown that, in general, the choice between an S-implication and an R-implication, keeping all other logical operators fixed, may change the order of the tuples.

On the other hand, we have pointed out a specific case where such a change does not occur. Namely, for the (t_{min}, s_{max}, N) De Morgan Triplet and for tuples t verifying the conditions: $(P(t) \geq \exists CP)$ or $((P(t) \leq \exists CP)$ and $(P(t) \geq 1 - \exists CP))$ it holds that: (A) $\gamma_{min,R}(C,P,t,T) \geq \gamma_{min,S}(C,P,t,T)$, and (B) replacing the R-implication with S-implication or vice-versa preserves the resulting order of the tuples ($\exists CP$ denotes here the truth value of the formula $\exists_{\vee} s \in T\ C(s) \wedge P(s)$, which is a part of (11)).

Thus for the tuples t satisfying the specified condition their resulting order does not depend on the choice between the S-implication and the R-implication. The "troublesome" tuples may appear both for high and low values of $\exists CP$. However, in the former case these are less interesting as for them $P(t)$ is small while there are tuples well satisfying both C and P (as $\exists CP$ is high).

4 Queries with Preferences and Bipolar Queries

The concept of a bipolar query we adopt here may be interpreted as an extension of the original idea of Lacroix and Lavency [16] in the framework of *fuzzy logic*. Here, we study its relation to the concept of a query with preferences, introduced by Chomicki [6, 7]. This concept may be conveniently presented in terms of a new operator of the relational algebra, called *winnow*, which is proposed in Chomicki [6]. This is an unary operator which selects from a set of tuples those which are

non-dominated with respect to a given preference relation. Chomicki defines this operator for the crisp case only, i.e., where preference relations and sets of tuples are crisp sets. We propose a fuzzy version of the *winnow* operator and show its relation to bipolar queries.

The *winnow* operator is defined with respect to a *preference relation*. In [6, 7] this is any binary relation R defined on the set of tuples T: $R \subseteq T \times T$. If two tuples $t,s \in T$ are in relation R, i.e., $R(t,s)$, then it is said that tuple t *dominates* tuple s with respect to the relation R.

Let T be a set of tuples and R a preference relation defined on T. Then the *winnow* operator ω_R is defined as follows

$$\omega_R(T) = \{t \in T : \neg \exists_{s \in T} R(s,t)\} \tag{12}$$

Thus, for a given set of tuples it yields a subset of the *non-dominated* tuples with respect to R.

A relational algebra query employing the *winnow* operator is referred to as a *query with preferences*. It may be easily shown (cf, Chomicki [6]) that the *winnow* operator may be expressed as a combination of the standard classical relational algebra operators. However, distinguishing the *winnow* operator makes it possible to study its behavior. Furthermore, some specialized methods of its execution may be conceived taking into account the optimal plans of the execution of the whole query.

The concept of the *winnow* operator may be illustrated with the following simple example. Let us consider a database of a real-estate agency with a table `HOUSES` describing the details of particular real-estate properties offered by the agency (each house is represented by a tuple). The schema of the relation `HOUSES` contains, among other, the attributes `city` and `price`. Let us assume that we are interested in the list of the *cheapest* houses in each city. Then the preference relation should be defined as follows

$$R(t,s) \Leftrightarrow (t.\texttt{city} = s.\texttt{city}) \wedge (t.\texttt{price} < s.\texttt{price})$$

where $t.A$ denotes the value of attribute A (e.g., `price`) at a tuple t. Then the *winnow* operator $\omega_R(\texttt{HOUSES})$ will select the houses that are sought (here a database table, such as `HOUSES`, is treated as a set of tuples). Indeed, according to the definition of the *winnow* operator, we will get as an answer a set of houses, which are non-dominated with respect to R, i.e., for which there is no other house in the same city which has a lower price.

A bipolar query with *crisp* conditions: the negative C and positive P may be expressed using the *winnow* operator in the following way. Let us show this first on an example of a bipolar query given in (1), i.e. "Find a house cheaper than USD 250,000 *and possibly* located not more than two blocks from a railway station". The preference R relation should be now defined as follows:

$$R(t,s) \Leftrightarrow (t.\texttt{to\_station} \leq 2) \wedge (s.\texttt{to\_station} > 2)$$

assuming that the `to_station` attribute characterizes each house, indicating how many blocks away it is located from a closest railway station. Then, the following relational algebra query with the *winnow* operator yields the required results

$$\omega_R(\sigma_{\texttt{price}\leq 250000}(\texttt{HOUSES}))$$

where σ_ϕ is the classical *selection* operator that selects from a set of tuples those for which condition ϕ holds.

This query preserves the characteristic property of bipolar queries, discussed earlier, i.e., if there are houses cheaper than USD 250,000 and located closer than two blocks from the station then only them will be selected (houses satisfying only the negative condition will be ignored). Otherwise, all houses satisfying the negative condition will be selected, if such exist.

A general scheme for translating a bipolar query characterized by a pair of negative and positive conditions (C,P) into its corresponding query with preferences is the following. The preference relation R is defined as

$$R(t,s) \Leftrightarrow P(t) \wedge \neg P(s) \tag{13}$$

and then the overall query with preferences takes the form:

$$\omega_R(\sigma_C(T))$$

Now we propose a fuzzy counterpart of the *winnow* operator, which also will make it possible to express (fuzzy) bipolar queries. We have to take into account that:

- R should be assumed to be a *fuzzy preference relation*,
- a fuzzy counterpart of the *non-dominance* concept has to be employed,
- the set of tuples T should also be assumed to be a *fuzzy set*.

In order to address the above requirements it is convenient to use the concept of a *fuzzy choice function* (cf. Świtalski [19]). In this approach the set of non-dominated elements with respect to a fuzzy preference relation may be conveniently expressed. Let us start with a concept of a crisp set $R^-(s)$, defined as follows:

$$R^-(s) = \{u \in T : R(s,u)\} \tag{14}$$

and gathering all tuples dominated by a tuple s with respect to a crisp preference relation R. Then $N(T,R)$, defined as follows:

$$N(T,R) = T \cap \bigcap_{s\in T} \overline{R^-(s)} \tag{15}$$

denotes the set of all non-dominated tuples of a crisp set T with respect to a crisp preference relation R, while $\overline{A}$ denotes the complement of the set A. The above definition is a bit more complicated than necessary (cf. the intersection with the set T),

however this is useful for deriving its fuzzy counterpart. For a further "fuzzification" it is convenient to rewrite (15) as a predicate calculus formula

$$N(T,R)(t) \Leftrightarrow T(t) \wedge \forall_{s \in T} \neg R^{-}(s)(t) \tag{16}$$

where particular predicates are denoted with the same symbols as corresponding sets (in particular, $R^{-}(s)$ denotes a predicate corresponding to a set (14) defined for a tuple s).

Using (15) we can define the *winnow* operator in the following way, equivalent to (12)

$$\omega_R(T) = N(T,R) \tag{17}$$

Now let us adapt (17) to the case of fuzzy preference relation R. A *fuzzy preference relation* on a crisp set of tuples T is any *fuzzy* binary relation $\tilde{R}$, $\tilde{R} \in \mathscr{F}(T \times T)$, where $\mathscr{F}(X)$ denotes the set of all fuzzy sets defined over the universe X. It will be identified with its membership function $\mu_{\tilde{R}}$.

As soon as the preference relation becomes fuzzy, then also the dominance (and non-dominance) naturally becomes a matter of degree. Thus we define a fuzzy set of tuples non-dominated with respect to a fuzzy preference relation $\tilde{R}$, using the formulas (14)-(15) and interpreting the set operations of the intersection and complement appearing thereof as standard operations on fuzzy sets. We start with a fuzzy counterpart of the set (14), defining the membership function of the fuzzy set $\tilde{R}^{-}(s)$ of tuples dominated (to a degree) by a tuple s with respect to the fuzzy preference relation $\tilde{R}$:

$$\mu_{\tilde{R}^{-}(s)}(u) = \mu_{\tilde{R}}(s,u) \tag{18}$$

Next let us rewrite (16), replacing a preference relation R with a fuzzy preference relation $\tilde{R}$ and replacing R^{-} with $\tilde{R}^{-}$, according to (18):

$$N(T,\tilde{R})(t) \Leftrightarrow T(t) \wedge \forall_{s \in T} \neg \tilde{R}(s,t) \tag{19}$$

We still have to take into account that the set T (and corresponding to it predicate) is, in general, fuzzy. Thus, we denote it as $\tilde{T}$ and replace the restricted quantifier $\forall_{s \in T}$ in (19) with an equivalent non-restricted form obtaining:

$$N(\tilde{T},\tilde{R})(t) \Leftrightarrow \tilde{T}(t) \wedge \forall_{s} \left(\tilde{T}(s) \rightarrow \neg \tilde{R}(s,t)\right) \tag{20}$$

Finally, we can define a fuzzy counterpart of the *winnow* operator in the following way. Let $\tilde{T}$ be a fuzzy set of tuples and $\tilde{R}$ a fuzzy preference relation, both defined on the same set of tuples T. Then, the *fuzzy winnow operator* $\omega_{\tilde{R}}$ is defined as:

$$\omega_{\tilde{R}}(\tilde{T})(t) = N(\tilde{T},\tilde{R})(t) \qquad t \in T \tag{21}$$

where the fuzzy predicate $N(\tilde{T},\tilde{R})$ is determined by (20), and $\omega_{\tilde{R}}(\tilde{T})(t)$ denotes the value of the fuzzy membership function of the set of tuples defined by $\omega_{\tilde{R}}(\tilde{T})$ for a tuple t.

Again, as in the case of fuzzy bipolar queries, one may study the effect of the choice of various logical operators to model logical connectives in the formula (20).

We leave a general study of this issue for further research. Now we will just show how a bipolar query may be expressed using the concept of the fuzzy *winnow* operator.

Let us consider a bipolar query defined by a pair of fuzzy conditions (C,P). These conditions will be identified with fuzzy predicates, denoted with the same symbols. Let $\tilde{R}$ be a fuzzy preference relation of the following form (cf. (13))

$$\tilde{R}(t,s) \Leftrightarrow P(t) \wedge \neg P(s) \tag{22}$$

Then, the bipolar query may be expressed as the following combination of the selection and fuzzy *winnow* operators:

$$\omega_{\tilde{R}}(\sigma_C(T)) = N(C(T),\tilde{R}) \tag{23}$$

where $C(T)$ is a fuzzy set of the elements of T satisfying (to a degree) the condition C, i.e., $\mu_{C(T)}(t) = C(t)$. Using (20) we can define the predicate (set) $N(C(T),\tilde{R})$ in (23) as follows

$$N(C(T),\tilde{R})(t) \Leftrightarrow C(t) \wedge \forall_s \left(C(s) \rightarrow \neg(P(s) \wedge \neg P(t))\right) \tag{24}$$

Note that the selection operator σ_C in (23) may also be applied to a fuzzy set of tuples T, which may be convenient if the set of tuples T is a result of another fuzzy query.

In Zadrożny and Kacprzyk [25] we show that for the conjunction, negation and implication connectives in (24) modelled by the operators of the minimum, $n(x) = 1 - x$ and the Kleene-Dienes implication, respectively, the fuzzy set of tuples obtained using (23) is identical with the fuzzy set defined by (6).

5 Concluding Remarks

We discuss the concept of bipolar queries. We show its origin in the work of Lacroix and Lavency [16] and briefly review some selected relevant approaches recently proposed in the literature. In particular we point out two main lines of research. One focuses on the formal representation within some well established theories and the question of a meaningful combinations of multiple conditions. Another one is concerned first of all with the study of some appropriate ways to aggregate negative (required) and positive (desired) conditions. We follow the second line of research and show the relation of this concept, from this point of view, with other approaches, both concerning database querying (exemplified by Chomicki [6]) as well as other domains (exemplified by Yager [21]). In the former case we offer a fuzzy counterpart of a new relational algebra operator *winnow* .

As in many cases of fuzzy logic applications the resulting concepts exhibit some arbitrariness with respect to a choice of the way logical connectives should be modelled. We contribute with some preliminary conclusions here but a further research is definitely needed.

References

1. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar possibility theory in preference modeling: Representation, fusion and optimal solutions. Information Fusion 7(1), 135–150 (2006)
2. Bordogna, G., Pasi, G.: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. International Journal of Intelligent Systems 10(2), 233–248 (1995)
3. Bosc, P., Pivert, O.: Discriminated answers and databases: fuzzy sets as a unifying expression means. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), San Diego, USA, pp. 745–752 (1992)
4. Bosc, P., Pivert, O.: An approach for a hierarchical aggregation of fuzzy predicates. In: Proceedings of the Second IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 1993), San Francisco, USA, pp. 1231–1236 (1993)
5. Bosc, P., Pivert, O.: SQLf: A relational database language for fuzzy querying. IEEE Transactions on Fuzzy Systems 3(1), 1–17 (1995)
6. Chomicki, J.: Querying with intrinsic preferences. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 34–51. Springer, Heidelberg (2002)
7. Chomicki, J.: Preference formulas in relational queries. ACM Transactions on Database Systems 28(4), 427–466 (2003)
8. Dubois, D., Fargier, H., Prade, H.: Refinement of the maximin approach to decision-making in fuzzy environment. Fuzzy Sets and Systems (81), 103–122 (1996)
9. Dubois, D., Prade, H.: Using fuzzy sets in flexible querying: why and how? In: Andreasen, T., Christiansen, H., Larsen, H. (eds.) Flexible Query Answering Systems, pp. 45–60. Kluwer Academic Publishers, Dordrecht (1997)
10. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS (LNAI), vol. 2522, pp. 174–182. Springer, Heidelberg (2002)
11. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: Galindo [14], pp. 97–114
12. Dubois, D., Prade, H.: An introduction to bipolar representations of information and preference. International Journal of Intelligent Systems 23(8), 866–877 (2008)
13. Fodor, J., Roubens, M.: Fuzzy Preference Modelling and Multicriteria Decision Support. Series D: System Theory, Knowledge Engineering and Problem Solving. Kluwer Academic Publishers, Dordrecht (1994)
14. Galindo, J. (ed.): Handbook of Research on Fuzzy Information Processing in Databases. Information Science Reference, New York (2008)
15. Kacprzyk, J., Zadrożny, S.: Computing with words in intelligent database querying: standalone and internet-based applications. Information Sciences 134(1-4), 71–109 (2001)
16. Lacroix, M., Lavency, P.: Preferences: Putting more knowledge into queries. In: Proceedings of the 13 International Conference on Very Large Databases, Brighton, UK, pp. 217–225 (1987)
17. Lietard, L., Rocacher, D., Tbahriti, S.E.: Towards an extended SQLf: Bipolar query language with preferences. International Journal of Applied Mathematics and Computer Sciences 4(1), 58–63 (2008)
18. Mesiar, R., Thiele, H.: On T-Quantifiers and S-Quantifiers. In: Novak, V., Perfilieva, I. (eds.) Discovering the World with Fuzzy Logic, pp. 310–326. Physica-Verlag, Heidelberg (2000)

19. Świtalski, Z.: Choice functions associated with fuzzy preference relations. In: Kacprzyk, J., Roubens, M. (eds.) Non-Conventional Preference Relations in Decision Making, pp. 106–118. Springer, Berlin (1988)
20. Yager, R.: Fuzzy sets and approximate reasoning in decision and control. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), San Diego, USA, pp. 415–428 (1992)
21. Yager, R.: Higher structures in multi-criteria decision making. International Journal of Man-Machine Studies 36, 553–570 (1992)
22. Yager, R.: Fuzzy logic in the formulation of decision functions from linguistic specifications. Kybernetes 25(4), 119–130 (1996)
23. Zadrożny, S.: Bipolar queries revisited. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 387–398. Springer, Heidelberg (2005)
24. Zadrożny, S., De Tre, G., De Caluwe, R., Kacprzyk, J.: An overview of fuzzy approaches to flexible database querying. In: Galindo [14], pp. 34–53
25. Zadrożny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA 2006), pp. 415–419. IEEE Computer Society, Krakow (2006)
26. Zadrożny, S., Kacprzyk, J.: Bipolar queries using various interpretations of logical connectives. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS, vol. 4529, pp. 181–190. Springer, Heidelberg (2007)

On Deriving Data Summarization through Ontologies to Meet User Preferences

Troels Andreasen and Henrik Bulskov

Abstract. A summary is a comprehensive description that grasps the essence of a subject. A text, a collection of text documents, a query answer can be summarized by simple means such as an automatically generated list of the most frequent words or ”advanced” by a meaningful textual description of the subject. In between these two extremes are summaries by means of selected concepts exploiting background knowledge providing selected key concepts. We address in this paper an approach where conceptual summaries are provided through a conceptualization as given by an ontology. The idea is to restrict a background ontology to the set of concepts that appears in the text to be summarized and therebyl provide a structure, a so-called instantiated ontology, that is specific to the domain of the text and can be used to condense to a summary not only quantitatively but also conceptually covers the subject of the text. In this chapter we introduce different approaches to summarization. We consider a strictly ontologly based approach where summaries are derived solely from the instantiated ontology, a conceptual clustering over the instantiated concepts based on a semantic similarity measure, and an approach based on probabilities.

1 Introduction

The purpose of a summary is to provide a simplification to highlight the major points from the subject, e.g. a text or a set of texts such as a query answer. The aim is to provide a summary that grasps the essence of the subject.

Most common are summaries as those provided manually by readers or authors as a result of intellectual interpretation. Summaries can however also be provided automatically. One approach, in the Question Answering style, such as this is investigated in for instance the DUC and TREC conferences (see for instance [6],[5],[7]),

Troels Andreasen
Roskilde University, Building 43-2, 4000 Roskilde, Denmark
e-mail: troels@ruc.dk

Henrik Bulskov
Roskilde University, Building 43-2, 4000 Roskilde, Denmark
e-mail: bulskov@ruc.dk

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 67–87.
springerlink.com

is to provide a full natural language generation based summary construction while a less ambitious, in the same tradition, is rather to perform a sentence selection from the text to be summarized.

In the other end the most simple approach is to select a reasonable short list of words among the most frequent and/or the most characteristic words from the set of words found in the text to be summarized. So rather than a coherent text the summary is simple a set of items.

Summaries in the approach presented here are also sets of items, but involves improvements over the simple set of words approach in two respects. First, we go beyond the level of keywords and aim to provide conceptual descriptions from concepts identified and extracted from the text. Second, we involve background knowledge in the form of an ontology. Strictly these two aspects are closely related – to use the conceptualization in the ontology we need means to map from words and phrases in the text to concepts in the ontology.

Summarization is a process of transforming sets of similar low level objects into more abstract conceptual representations [19] and more specifically a summary for a set of concepts is an easy to grasp and short description – in the form of a smaller set of concepts. For instance $\{car, house\}$ as summary for $\{convertible, van, cottage, estate\}$ or $\{dog\}$ as summary for $\{poodle, alsatian, golden\ retriever, bulldog\}$.

In this paper we present different directions to conceptual summaries as answers to queries. In these cases an ontology plays a key role as reference for the conceptualization. The general idea is from a world knowledge ontology to form a so-called "instantiated ontology" by restricting to a set of instantiated concepts.

First, we consider a strictly ontology based approach where summaries are derived solely from the instantiated ontology. Second, we consider conceptual clustering over the instantiated concepts based on a semantic similarity measure. Finally, we present an approach based on probabilities.

The general idea, in the approach presented here, is to restrict a general world knowledge ontology to the given set of concepts extending this with relations and related concepts and thereby providing a structure for navigation and further investigation of the concepts. Conceptual investigation of a set of documents can be performed by extracting the set of concepts appearing in the documents and by providing means for navigation and retrieval within the set of extracted concepts.

The paper is organized as follows. First, we introduce the general ontology, extraction of conceptual descriptions, and the instantiated ontology. Second, we describe the various approaches to conceptual summaries. And finally we present a conclusion and give some pointers to future research.

2 Representing Background Knowledge – Ontology

Background knowledge is knowledge that complements the primary target data (the text or text collection / database) that is subject of the summarization with information that is essential to the understanding of this. Background knowledge can take different forms varying from simple lists of words to formal representations. To

provide, in the Question Answering style, a full natural language generation based summary, means for reasoning within the domain as well as means for processing language expressions are needed so background knowledge should include axiomatic formalization of essential domain knowledge as well as knowledge to guide the natural language synthesis process. In this context, however, our goal is conceptual summaries provided as sets of words or concepts so background knowledge to support this can range from unstructured lists of words to ontologies.

A simple list of words can be applied as a filter, mapping from a text to the subset of the word list that appears in the text. Such a controlled list of keywords or a vocabulary of topics can by obvious means be improved to capture also morphology by stemming or inflection patterns. However, for summary purposes we will have to rely on course-grained principles as statistics on frequencies to reduce the number of items of a list, to obtain an easy to grasp summary. What is needed to obtain significant improvement is a structure that relates individual words and thereby supports fusion into commonly related items in the contraction towards sufficiently brief summaries. In addition to this the presence of relations introduce the element of definition by related items and thus justifies the notion as a structure of concepts rather than a list of words. So taxonomies, partonomies, semantic networks and ontologies are structures that potentially contribute also to knowledge-based summarization. Our main focus here is on ontologies ordered around taxonomic relationship. Rather than the common description logic based approach we choose here a simpler concept algebraic approach to ontologies.

One important rationale for this is that our goal here is not ontological reasoning in general but rather extraction of sets of mapped concepts and manipulation of such sets (e.g. contraction). Another is that the concept algebraic approach has an inherent and very significant notion of generativity, where the ontology includes also compound concepts that can be formed by means of other concepts.

2.1 An Algebraic Approach to Ontologies

Assume that a basis taxonomy that situates a set of atomic term concepts $\mathscr{A}$ in a multiple inheritance hierarchy is given. Based on this we define a generative ontology by generalization of the hierarchy to a lattice and by introducing a (lattice-algebraic) concept language (description language) that defines an extended set of well-formed concepts, including both atomic and compound term concepts.

The concept language used here, ONTOLOG[9], has as basic elements concepts and binary relations between concepts. The algebra introduces two closed operations *sum* and *product* on concept expressions φ and ψ, where $(\varphi + \psi)$ denotes the concept being either φ or ψ and $(\varphi \times \psi)$ denotes the concept being φ and ψ (also called *join* and *meet* respectively).

Relationships r are introduced algebraically by means of a binary operator (:), known as the Peirce product $(r : \varphi)$, which combines a relation r with an expression φ. The Peirce product is used as a factor in conceptual products, as in $x \times (r : y)$,

which can be rewritten to form the feature structure $x[r\colon y]$, where $[r\colon y]$ is an *attribution* of the concept x. Thus we can form compound concepts by attribution.

Given atomic concepts $\mathscr{A}$ and semantic relations $\mathscr{R}$, the set of well-formed terms $\mathscr{L}$ is:

$$\mathscr{L} = \{\mathscr{A}\} \cup \{x[r_1 : y_1, ..., r_n : y_n] | x \in \mathscr{A}, r_i \in \mathscr{R}, y_i \in \mathscr{L}\} \tag{1}$$

Compound concepts can thus have multiple as well as nested attributions. For instance with $\mathscr{R}$= {WRT, CHR, CBY, TMP, LOC, ...}[1] and $\mathscr{A}$ ={*entity, physical_entity, abstract_entity, location, town, cathedral, old*} we get:

$$\begin{aligned}&\mathscr{L} = \\ &\{entity, physical\_entity, abstract\_entity,\\ &location, town, cathedral, old,\\ &\ldots, cathedral[\text{LOC}\colon town, \text{CHR}\colon old],\\ &cathedral[\text{LOC}\colon town[\text{CHR}\colon old]], \ldots\}\end{aligned}$$

2.2 Modelling Ontologies

Obviously modelling ontologies from scratch will often be the best way to ensure that the result will be correct and consistent. However, for many applications the effort it takes is simply not at disposal and manual modeling have to be limited to narrow subdomains and complemented with extracts derived from relevant general sources. Sources that may contribute to modeling of ontologies may have various forms. A taxonomy is an obvious choice and it may be supplemented with, for instance, word and term lists as well as dictionaries for definition of vocabularies and for handling of morphology. Among the obviously useful resources are the semantic network WordNet [11] and the Unified Medical Language System (UMLS) [4] that unifies several resources in the biomedical science area.

To go from a resource to an ontology is not necessarily straightforward, but if the goal is a generative ontology and the given resource is a taxonomy, one option is to proceed as follows. Let $\mathscr{T}$ be a taxonomy over the set of atomic concepts $\mathscr{A}$ and let $\mathscr{L}$ be the language of well-formed terms over $\mathscr{A}$ for a given set of relations $\mathscr{R}$ according to (1) above. $\hat{\mathscr{T}}$ is the transitive closure of $\mathscr{T}$. $\hat{\mathscr{T}}$ can be generalized to an inclusion relation "$\leq$" over all well-formed terms of the language $\mathscr{L}$ by the following

$$\begin{aligned}\text{“}\leq\text{“} = \hat{\mathscr{T}}\\ &\cup\{\langle x[\ldots, r\colon z], y[\ldots]\rangle | \langle x[\ldots], y[\ldots]\rangle \in \hat{\mathscr{T}}\}\\ &\cup\{\langle x[\ldots, r\colon z], y[\ldots, r\colon z]\rangle | \langle x[\ldots], y[\ldots]\rangle \in \hat{\mathscr{T}}\}\\ &\cup\{\langle z[\ldots, r\colon x], z[\ldots, r\colon y]\rangle | \langle x, y\rangle \in \hat{\mathscr{T}}\}\end{aligned} \tag{2}$$

where repeated ... denote zero or more attributes of the form $r_i\colon w_i$.

[1] for *with respect to, characterized by, caused by, temporal, location*, respectively.

The general ontology $\mathcal{O} = (\mathcal{L}, \leq, \mathcal{R})$ thus encompasses a set of well-formed expressions $\mathcal{L}$ derived in the concept language from a set of atomic concepts $\mathcal{A}$, an inclusion relation generalized from the taxonomy relation in $\mathcal{T}$, and a supplementary set of semantic relations $\mathcal{R}$. For $r \varepsilon \mathcal{R}$, we obviously have $x[r: y] \leq x$, and that $x[r: y]$ is in relation r to y. Observe that $\mathcal{O}$ is generative and that $\mathcal{L}$ therefore is potentially infinite.

An example is given in figure 2 showing a segment of a generative ontology build with WordNet as resource.

2.3 Deriving Similarity

An ontology, that covers a document collection, may provide an excellent means to survey and give perspective to the collection. However as far as access to documents is concerned, ontology reasoning is not the most obvious evaluation strategy as it may well entail scaling problems. Applying measures of similarity derived from the ontology is a way to replace reasoning with simple computation still influenced by the ontology.

One obvious way to measure similarity in ontologies, given the graphical representation, is to evaluate the distance between the concepts being compared, where a shorter distance implies higher similarity and vice versa.

A number of different ontological similarity measures along this line have been proposed over the years. *Shortest Path Length* [12] forms the basis of a group of measures classified as path length approaches. The *Weighted Shortest Path* [15] is a generalization of *Shortest Path Length* where weights are assigned to relations in the ontology. Two different alternatives are *Information Content* [16] and *Weighted Shared Nodes* [17], where the former uses the probability of encountering concepts in a corpus to define the similarity between concepts, and the latter uses the density of concepts shared by the concepts being compared to measure the similarity.

3 Referencing the Background Knowledge – Providing Descriptions

As already indicated the approach involves surveying text through the ontology provided and delivering summaries on top of the conceptualization of the ontology. For this purpose we need to provide a description of the text to be summarized in terms of the concepts in the ontology. So words and/or phrases must be extracted from the text and mapped into the ontology. This is a knowledge extraction problem, and obviously such knowledge extraction can span from full deep natural language processing (NLP) to simplified shallow processing methods.

Here we will consider the latter due to the counterbalance between the need for a full interpretation and the computational complexity of getting it. A very simple solution would match words in text with labels of concepts in the ontology, hence make a many-to-many relation between words in text and labels in the ontology that

just accepts the ambiguity of natural language. Improvements can easily be obtained through pattern based information extraction / text mining and through methods in natural language processing.

First, a heuristic part of speech tagging can be performed on the text, and provided that word classes are assigned to the concepts given in the ontology this enables a word class based disambiguation.

Second, a stemming or, provided lexical information is available, a transformation to a standardized inflectional form can significantly improve the matching.

Third, given part of speech tagged input , simple syntactic natural language grammars can be used to chunk words together forming utterances or phrases [3], that can be used as the basis for matching against compound concepts in the ontology. Obviously the matching of chunks from the text and concepts in the ontology is in principle the same complex NLP problem over again, but the chunks identified will often correspond to meaningful concepts and therefore lead to a more refined and better result of the matching and, in addition, allow for a simple pattern-based approaches. We refer to [18] and [1] for more refined approaches. Here we will only cover a simple pattern-based approach.

Finally, some kind of word sense disambiguation [22] can be introduced in order to narrow down the possible readings of words, hence ideally mapping words of phrases to exactly one concept in the ontology.

A very simple approach along these lines is the following. Given a part-of-speech tagged and NP-chunked input a grammar for interpretation of the chunks is the following:

$$\begin{aligned} \text{Head} &::= \text{N} \\ \text{NP} &::= \text{A* N* Head} \mid \text{NP P NP} \end{aligned} \tag{3}$$

where *A*, *N* and *P* as placeholders for adjective, noun and preposition respectively. A very course-grained mapping strategy on top of this interpretation can be formed using the following transition rules, where premodifying adjectives relates to the head through *characterized by* (CHR) while premodifying composite nouns and prepositions both relates through *with respect to* (WRT):

$$\begin{aligned} &\text{A}_1, \ldots, \text{A}_n \text{N}_1, \ldots, \text{N}_m \text{Head} \mapsto \\ &\qquad \text{Head}[\text{CHR}: \text{A}_1, \ldots, \text{CHR}: \text{A}_n, \text{WRT}: \text{N}_1, \ldots, \text{WRT}: \text{N}_m] \\ &\text{NP (P NP)}_1, \ldots, \text{(P NP)}_n \mapsto \\ &\qquad \text{NP}[\text{WRT}: \text{NP}_1, \ldots, \text{WRT}: \text{NP}_n] \end{aligned} \tag{4}$$

To test this approach we consider the Metathesaurus in the Unified Medical Language System (UMLS) [13] as resource and build a generative ontology from this. For part of speech tagging and phrase chunking we use the MetaMap application [2].

Consider the following utterance[2] as an example:

> [...] *the plasma patterns of estrogen and progesterone under gonadotropic stimulation simulating early pregnancy* [...]

The first part of the analysis leads to part of speech tagging and phrase recognition as follows:

Phrase	Type	Word	POS
Noun Phrase	det	the	det
	mod	plasma	noun
	head	patterns	noun
Preposition	prep	of	prep
	head	estrogen	noun
	conj	and	conj
Noun Phrase	head	progesterone	noun
Preposition	prep	under	prep
	mod	gonadotropic	adj
	head	stimulation	noun
	verb	simulating	verb
Noun Phrase	mod	early	adj
	head	pregnancy	noun

By applying the grammar (3) this can be transformed into the following three noun phrases:

plasma/N patterns/N of/P estrogen/N

progesterone/N under/P gonadotropic/A stimulation/N

early/A pregnancy/N

and by using the transition rules (4) we can produce the following compound expressions:

patterns[WRT: plasma, WRT: estrogen]

progesterone[WRT:stimulation[CHR:gonadotropic]]

pregnancy[CHR:early]

then we can attach the mapping from words in these expressions to node identifiers in the Metathesaurus given by MetaMap:

[2] This utterance is from a small 50K abstracts fraction of MEDLINE [14] having both *Homones* and *Reproduction* as major topic keywords.

patterns{C0449774}
[WRT: plasma{C0032105, C1546740}, WRT: estrogen{C0014939}]

progesterone{C0033308}
[WRT:stimulation{C1948023,C1292856}[CHR:gonadotropic{C1708248}]]

pregnancy{C0425965,C0032961}[CHR:early{C1279919}]

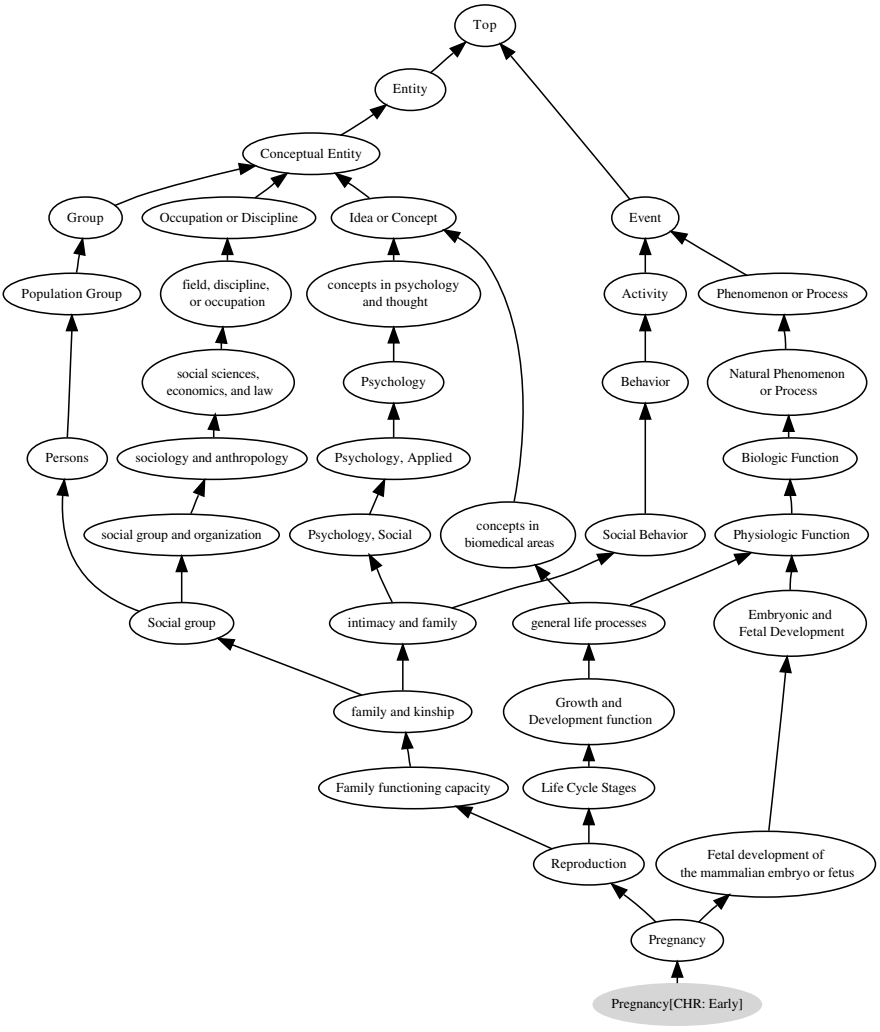


Fig. 1 Mapping of the concept pregnancy{C0032961}[CHR:early{C1279919}] into UMLS (slightly modified, i.e. some paths are removed, due to considerations of space)

Naturally, since natural language is ambiguous, but also due to the fact that the Metathesaurus is built by merging different knowledge sources together, MetaMap is not able to disambiguate all parts of the expressions, for instance, here *plasma*, *stimulation*, and *pregnancy* all are ambiguous. A simple solution to this problem is just to accept the ambiguity in the generation of descriptions, hence produce all possible interpretations of the expressions, for instance, the two readings of *early pregnancy*

$$\text{pregnancy}\{\text{C0425965}\}[\text{CHR:early}\{\text{C1279919}\}]$$

$$\text{pregnancy}\{\text{C0032961}\}[\text{CHR:early}\{\text{C1279919}\}]$$

but more advanced solutions could introduce additional methods for disambiguation of descriptions, for instance, try to include context analysis in order to further reduce the ambiguity, see e.g. [22] for a survey of word sense disambiguation approaches. An example of the mapping of the concept

$$\text{pregnancy}\{\text{C0032961}\}[\text{CHR:early}\{\text{C1279919}\}]$$

into the UMLS is given in figure 1.

Regardless of whether rules to combine into compound concepts are applied or not the result of a mapping from a piece of text T to an ontology O will be a set of concepts. This set of concepts $d_O(T)$ we call the *description* of T (with respect to O) and the elements of d are called descriptors. $d_O(T)$ is, so to say, T viewed through the ontology O. $d_O(T)$ may be used as the content of an ontology-based indexing, for instance on the level of sentences. Here our main focus is on summarization and thus we will also be concerned with descriptions covering larger texts and collections of texts. So all in all, no matter the size, form or structure of a given text T, the basic description is a set of descriptors.

3.1 Instantiated Ontology

The description $d_O(T)$ of a text T given the ontology O comprise a set of concepts in O and as indicated the purpose here is to summarize based on relations in the ontology. Now given the set of concepts (the description) $d_O(T)$ an obviously relevant subontology is a subontology that covers all elements of $d_O(T)$. Such a subontology can be considered an instantiation of the text T (or the set of concepts $d_O(T)$). A very simple example of such is the ontology for the concept pregnancy[CHR:early] given in figure 1.

Given an ontology $\mathscr{O} = (\mathscr{L}, \leq, \mathscr{R})$ and a set of concepts C we define the instantiated ontology $\mathscr{O}_C = (\mathscr{L}_C, \leq_C, \mathscr{R})$ as a restriction of $\mathscr{O}$ to cover only the concepts in C, that is, C and every concept from $\mathscr{L}$ that subsumes concepts in C or attributes for concepts in C. $\mathscr{L}_C$ can be considered an "upper expansion" of C in $\mathscr{O}$. More specifically, with C^+ being C extended with every concept related by attribution from a concept in C:

$$\mathscr{L}_C = C \cup \{x | y \in C^+, x \in \mathscr{L}, y \leq x\}$$
$$\text{“} \leq_C \text{“} = \{(x,y) | x,y \in \mathscr{L}_C, x \leq y\} \quad (5)$$

Thus $\mathscr{O}_C$ is not generative. ”$\leq_C$” may be represented by a minimal set ”$\leq'_C$”$\subseteq$”$\leq_C$” such that ”$\leq_C$” is derivable from ”$\leq'_C$” by means of transitivity of ”$\leq$” and monotonicity of attribution:

$$transitivity : x \leq y, y \leq z \Rightarrow x \leq z$$
$$monotonicity : x \leq y \Rightarrow z[r\colon x] \leq z[r\colon y]$$

Figure 2 shows an example of an instantiated ontology. The general ontology is based on (and includes) WordNet and the ontology shown is ”instantiated” wrt. the following set of concepts:

$$C = \{cathedral[\text{LOC}\colon town[\text{CHR}\colon old]], abbey,$$
$$fortification[\text{CHR}\colon large, \text{CHR}\colon old], stockade,$$
$$fortress[\text{CHR}\colon big]\}$$

3.2 Weighted Descriptors

As described above descriptions are crisp sets. However, since there are obvious grounds to attach weights to descriptors, we can naturally extend the notion to capture also weighted descriptors and to collect these into *fuzzyfied descriptions*.

First of all weighting of descriptors can be based on frequencies. We can define term frequency $tf(d)$ of a descriptor d as the number of different places in T the descriptor d can be extracted. Furthermore if the text at hand is a collection of documents we can also take into account the document frequency $df(d)$ (the number of different documents d can be extracted from). However, it should be noted that while the significance of the (inverse) df on the tfidf term-weighting-measure is well-understood and established in conventional information retrieval, it is less clear what role it should play in connection with weighting for summarization. In the tfidf-measure the (inverse) df diminishes the weight of terms that occur very frequently in the collection and increases the weight of terms that occur rarely, so terms that tend to select most documents in the collection will have reduced weights. However, when summarizing over collections of documents we are not interested in distinguishing on the document level since the summary should be characteristic for the collection as a whole.

Weighting can also be introduced to reflect degrees of membership due to overlap and relatedness. A compound descriptor, as $disease[\text{CHR} : serious]$, covers the two concepts disease and serious but even so these two latter could also be added to the description eventually with a reduced weight due to the overlap. Given a similarity measure sim a description can also be expanded with related / similar concepts with membership degrees corresponding to the similarity.

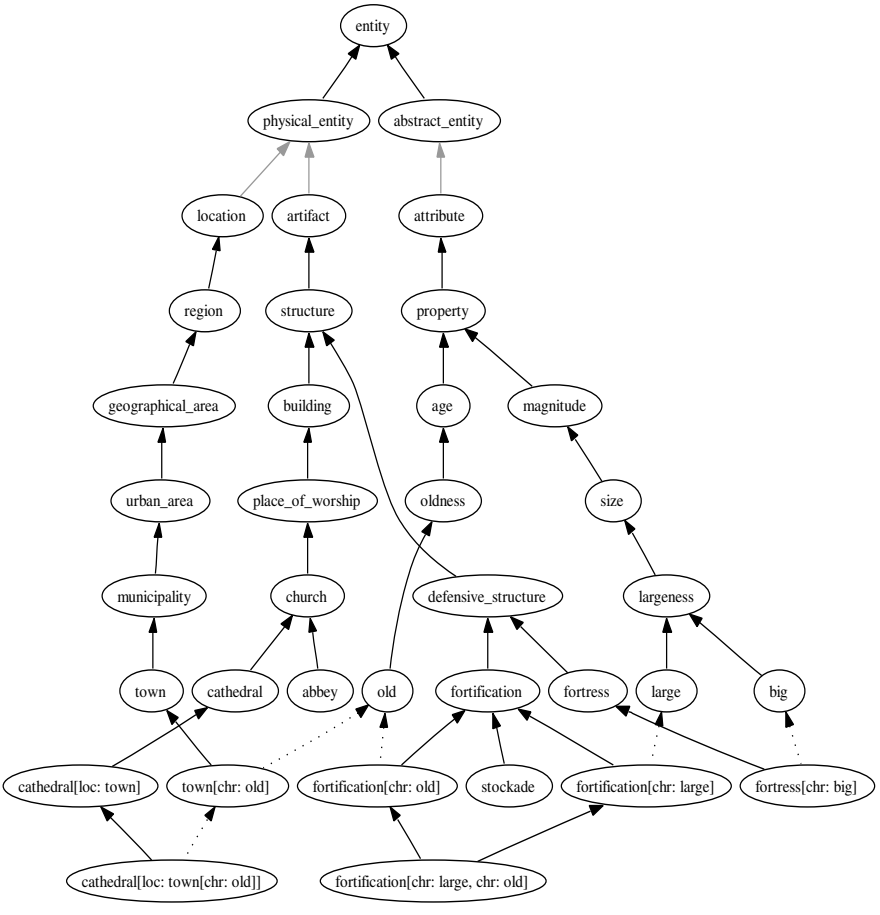


Fig. 2 A segment of an ontology based on Wordnet. Does also correspond to an instantiated ontology for the set of instantiated concepts $\{$*cathedral*[LOC: *town*[CHR: *old*]], abbey, *fortification*[CHR: *large*, CHR: *old*], stockade, *fortress*[CHR: *big*]$\}$

4 Data Summarization through Background Knowledge

The general idea here is to exploit background knowledge through conceptual summaries, that are to provide a means to survey textual data, for instance a query result. A set of concepts from the background knowledge is first identified in the text and then contracted into smaller set of, in principle, most representable concepts.

This can be seen as one direction in a more general conceptual querying approach where queries can be posed an or answers be presented by means of concepts. For a general discussion on other means, except from conceptual summaries, of conceptual querying, where also a dedicated language constructs for this purpose is presented we refer to [21]. Here we discuss summaries only.

In the approach to summarization described here we assume an ontology to guide the summarization and, for the text to be summarized, an initial extraction of concepts as described in the previous section. Thus, we can assume an initial set of concepts C and we a facing a challenge to provide a smaller set of representative concepts covering C, that is, an appropriate summary that grasps what´s most characteristic about C. For computation of the summary we restrict to the subontology $\mathscr{O}_C = (\mathscr{L}_C, \leq_C, \mathscr{R})$ corresponding to the instantiated ontology for C.

Below we will use the example ontology shown in figure 3, which is derived from the following small fragment of text taken from SEMCOR[10]:

> *Greases*, *stains*, and miscellaneous *soils* are usually sorbed onto the *soiled surface*. In most cases, these *soils* are taken up as *liquids* through *capillary action*. In an essentially static *system*, an *oil* cannot be replaced by *water* on a *surface* unless the *interfacial tensions* of the *water phase* are reduced by a *surface-active agent*.

Words in italics indicate the initial set of concepts, in this case nouns that are mapped into WordNet[11], from which the instantiated ontology is created[3]. SEMCOR is a subset of the documents in the Brown corpus which has the advantage of being semantically tagged with senses from WordNet.

We introduce three different directions for deriving summaries below: one based directly on connectivity in the ontology, one drawing on statistical clustering applying similarity measures, and one that uses corpus statistics and thus is based on probabilities. Finally, we will consider approaches that join these three different directions.

4.1 Connectivity Clustering

Connectivity Clustering is clustering based solely on connectivity in an instantiated ontology. More specifically the idea is to cluster a given set of concepts based on their connections to common ancestors, for instance grouping two siblings due to their common parent, and in addition to replace the group by the common ancestor. Thus rather than, when taking a bottom-up hierarchical clustering view, moving towards a smaller number of larger clusters, connectivity clustering is about moving towards a smaller number of more general concepts.

For a set of concepts $C = \{c_1, \ldots, c_n\}$ we can consider as *generalizing description* a new set of concepts $\delta(C) = \{\widehat{c}_1, \ldots, \widehat{c}_k\}$, where $\widehat{c}_i$ is either a concept generalizing concepts in C or an element from C. Each generalizer in $\delta(C)$ is a *least upper bound* (*lub*) of a subset of C, $\widehat{c}_i = lub(C_i)$, where $\{C_1, \ldots, C_k\}$ is a division (clustering) of C. Notice that the *lub* of a singleton set is the single element in this.

We define *most specific generalizing description* $\delta(C)$ for a given $C = \{\widehat{c}_1, \ldots, \widehat{c}_k\}$ as a description restricted by the following properties:

[3] Notice that due to the use of SEMCOR there is no attribution in the initial set of concepts.

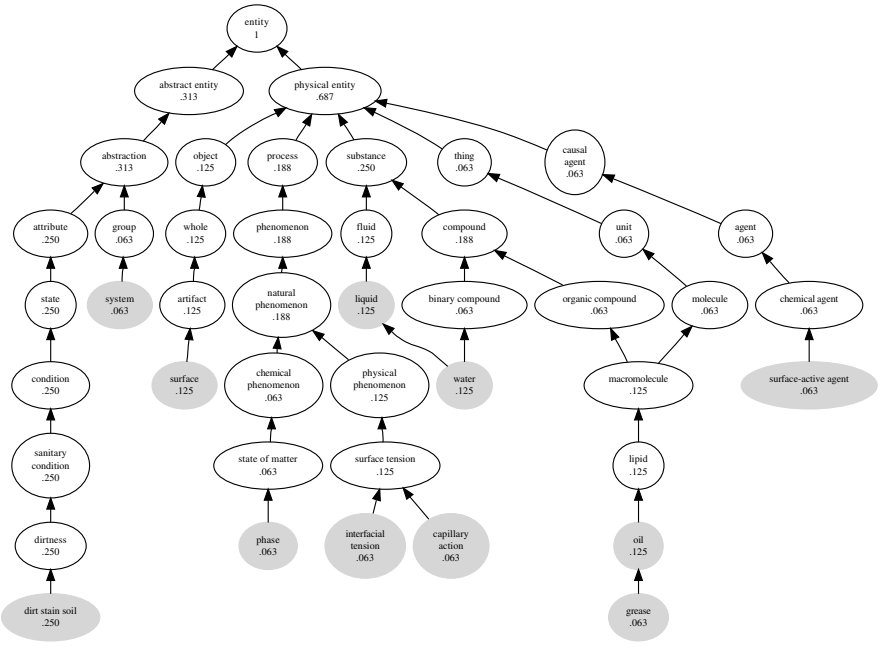


Fig. 3 An instantiated ontology based on a paragraph from SEMCOR. Probabilities of encountering the concepts in the corpus are given

$$\forall \widehat{c} \in \delta(C) : \widehat{c} \in C \vee \exists c', c'' \in C \wedge c' \neq c'' \wedge c' < \widehat{c} \wedge c'' < \widehat{c} \tag{6}$$

$$\forall \widehat{c}', \widehat{c}'' \in \delta(C) : \widehat{c}' \not< \widehat{c}'' \tag{7}$$

$$\forall c', c'' \in C, \widehat{c}' \in \delta(C), \neg \exists x \in L_C : c' \leq x \wedge c'' \leq x \wedge x \leq \widehat{c}' \tag{8}$$

where (6) restricts $\delta(C)$ to elements that either originate from C or generalize two or more concepts from C. (7) restricts $\delta(C)$ to be without redundance (no element of $\delta(C)$ may be subsumed by another element), and (8) reduces to the most specific in the sense that no subsumer for two elements of C may be subsumed by an element of $\delta(C)$.

Observe, that $\delta(C)$, like C, is a subset of $\mathscr{L}_C$, and that we therefore can refer to an m'th order summarizer $\delta^m(C)$. Obviously, to obtain an appropriate description of C we will in most cases need to consider higher orders of δ. At some point m we will in most cases have that $\delta^m(C) = Top$, where Top is the topmost element in the ontology. An exception is when a more specific single summarizer is reached in the ontology.

The most specific generalizing description $\delta(C)$ for a given C is obviously not unique and there are several different sequences of most specific generalizing descriptions of C from C towards Top. However, a reasonable approach would be to

go for the largest possible steps obeying the restrictions for δ above, as done in the algorithm below.

For a poset S we define $min(S)$ as the subset of minimal elements of S. $min(S) = \{s|s \in S, \forall s' \in S : s' \not< s\}$

ALGORITHM – Connectivity summary

INPUT: Set of concepts $C = \{c_1, \ldots, c_n\}$

OUTPUT: A most specific generalizing description $\delta(C)$ for C.

1) Let the instantiated ontology for C be $\mathscr{O}_C = (\mathscr{L}_C, \leq_C, \mathscr{R})$
2) $U = min(\{u|u \in \mathscr{L}_C \wedge \exists c_i, c_j \in C : c_i < u \wedge c_j < u\})$,
3) $L = \{c|c \in \mathscr{L}_C \wedge \exists u \in U : c < u\}$
4) $M = min(\{m|m \in \mathscr{L}_C \backslash L \wedge \exists c \in L : c < m\})$,
5) set $\delta(C) = C \cup U \cup M/L$

In 2) all most specific concepts U that generalize two or more concepts from C are derived. Notice that these may include concepts from C when C contains concepts subsuming other concepts. In 3) L defines the set of concepts in C that specializes the generalizers in U. In 4) additional parents for (multiple inheriting) concepts covered by generalizations in U are added. 5) derives $\delta(C)$ from C by adding the most specific generalizers and subtracting concepts specializing these.

Notice especially that 4) is needed in case of multiple inheritance. If a concept c has multiple parents and is replaced by a more general concept due to one of its senses (parents) we need to add parents corresponding to the other senses of c – otherwise we loose information corresponding to these other senses. For instance in figure 4 we have that $\delta(\{a,b\}) = \{c,d\}$ because a and b will be replaced by c, and d will be added to specify the second sense of b.

As a more elaborate example consider again figure 3. Summarization of C by connectivity will proceed as follows.

C = {*system, dirt, phase, capillary action, interfacial tension, grease, oil, water, liquid, surface-active agent, surface*}
$\delta^1(C)$ = {*abstraction, binary compound, liquid, oil, phase, surface, surface-active agent, surface tension*}
$\delta^2(C)$ = {*abstraction, compound, liquid, molecule, natural phenomenon, surface, surface-active agent*}
$\delta^3(C)$ = {*abstraction, molecule, natural phenomenon, substance, surface, surface-active agent*}
$\delta^4(C)$ = {*abstraction, physical entity*}
$\delta^5(C)$ = {*entity*}

The chosen approach, taking the largest possible steps where everything that can, will be grouped, is of course not the only possible. If we alternatively want to form only some of the possible clusters complying with the restrictions some kind of priority mechanism for selection is needed.

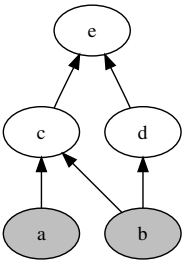


Fig. 4 Ontology fragment with multiple inheritance

Among important properties that might contribute to priority are *deepness*, *redundancy* and *support*. The deepest concepts, those with the largest depth in the ontology, are structurally and thereby often also conceptually the most specific concepts, thus collecting these first would probably lead to a better balance with regard to how specific the participating concepts are in candidate summaries. Redundancy, where participating concepts include (subsumes) others, is avoided as regards more general concepts introduced (step 3 in the algorithm). However redundancy in the input set may still survive so priority could also be given to remove this first. In addition we could consider support for candidate summarizers. One option is simply to measure support in terms of number of subsumed concepts in the input set while more refinement could be obtained by also taking frequencies of concepts as well as their distribution in documents[4] in the original text into consideration. Support may guide the clustering in several ways. It indicates for a concept how much it covers in the input and can thus be considered as an importance weight for the concept as summarizer for the input. High importance should probably infer more reluctance as regards further generalization.

4.2 Similarity Clustering

Given a similarity measure, summaries can be derived from a clustering of concepts applying this measure. Obviously, if the measure is derived from an ontology, and thereby do reflect this, then so will the clustering. We will below assume an ontology-based similarity measure *sim*. A simple example of such a measure can be derived from the path lenght in the ontology graph (Rada' *Shortest Path Length* [12]). More refined approaches are *Information Content* [16] and *Weighted Shared Nodes* [17].

A Hierarchical Similarity-Based Approach

With a given path-length dependent similarity measure derived from the ontology a *lub*-centered, agglomerative, hierarchical clustering can be performed as follows.

Initially each "cluster" corresponds to an individual element of the set to be summarized. At each particular stage the two clusters which are most similar are

[4] Corresponding to term and document frequencies in Information Retrieval.

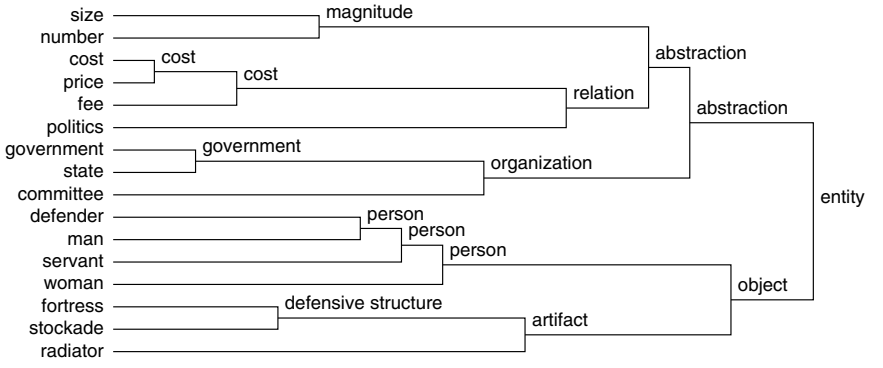


Fig. 5 An illustration of the hierarchical clustering summary. The merging of two clusters are shown with their *lub*

joined together. This is the principle of conventional hierarchical clustering. However rather than replacing the two joined clusters with their union as in the conventional approach they are replaced by their *lub*. Thus given a set of concepts $C = \{c_1, \ldots, c_n\}$ summarizers can be derived as follows.

ALGORITHM – Hierarchical clustering summary
INPUT: Set of concepts $C = \{c_1, \ldots, c_n\}$
OUTPUT: Generalizing description $\delta(C)$ for C.

1) Let the instantiated ontology for C be $\mathscr{O}_C = (\mathscr{L}_C, \leq_C \mathscr{R})$
2) Let $T = \{\langle x,y\rangle | sim(x,y) = max_{z,w \in C}(sim(z,w))\}$
3) Let $U = min(\{u | u \in \mathscr{L}_C \wedge \exists x,y \in \mathscr{L}_C : x < u \wedge y < u\})$
4) $L = \{x | \langle x,y\rangle \in T \vee \langle y,x\rangle \in T\}$
5) set $\delta(C) = C \cup U/L$

As was also the case with the connectivity clustering, to obtain an appropriate description of C we might have to apply δ several times and at some point m we have that $\delta^m(C) = Top$.

Figure 5 illustrates the application of δ a total of 15 times to the set of concepts from the previous example, where we might have (depending on the exact similarity values) for instance:

C = {*number, size, committee, government, state, defender, man, servant, woman, bribe, cost, price, fee, fortification, fortress, stockade*}
$\delta^1(C)$ = {*number, size, committee, government, state, defender, man, servant, woman, bribe,* ***cost****, fee, fortification, fortress, stockade*}
$\delta^2(C)$ = {*number, size, committee,* ***government****, defender, man, servant, woman, bribe, cost, price, fee, fortification, fortress, stockade*}

etc.

Thus summaries are generated iteratively and at each step the two closest concepts are clustered and the result is replaced by the corresponding *lub*.

Simple Least Upper Bound-Based Approach

The principle of replacing clusters by their least upper bound can by applied on top of, in principle, any clustering approach. A straightforward similarity based approach is simply to apply a crisp clustering to the set of concepts $C = \{c_1, \ldots, c_n\}$ leading to $\{C_1, \ldots, C_k\}$ and then provide the set of *lub*'s $\{\widehat{c}_1, \ldots, \widehat{c}_k\} = \{lub(C_1), \ldots, lub(C_k)\}$ for the division of C as summary. However to take into account also the importance of clusters in terms of their sizes, the summary can be modified by the support of the generalizing concepts, $support(x,C)$, that for a given concept specifies the fraction of elements from the set C covered:

$$support(x,C) = \frac{|\{y|y \in C, y \leq x\}|}{|C|} \tag{9}$$

leading to a fuzzyfied (weighted) summary, based on the division (crisp clustering) of C into $\{C_1, \ldots, C_k\}$:

$$\Sigma_i support\,(lub(C_i), C)\,/lub(C_i) \tag{10}$$

To illustrate this lub-based approach consider table 1. Five groups are given that are derived as clusters of synsets in Wordnet.[5]

Table 1 A set of clusters and their least upper bounds from WordNet

cluster	lub
{number, size}	magnitude
{committee, government, state}	organization
{defender, man, servant, woman}	person
{bribe, cost, fee, price}	cost
{fortification, fortress, stockade}	defensive structure

From these clusters the fuzzyfied summary {*.13/magnitude + .19/organization + .25/person + .25/cost + .19/defensive structure*} can be generated.

We may expect a pattern similar to hierarchical clustering in derivation of summaries in an approach based on similarity when the similarity measure reflects simple shortest path in the ontology.

This approach to summarization can be generalized using a fuzzyfied notion in place of the least upper bound as candidate representative. The generalization reduces the sensitivity against noise in the groups resulting from the initial clustering. This approach is described in [20, 21].

[5] The first four are set of clusters and their least upper bounds in where $C_1, \ldots, C_4$ are from SEMCOR and C_5 is from the example ontology in figure 2.

4.3 A Probability-Based Approach

We consider above a summary of textual input based on the concepts that appears in the text and how these are related in a background ontology. In the hierarchical approach candidate summarizers are chosen regardless of their coverage of the input, while the *lub*-based approached is introduced with a support that measures the degree to which all occurring input concepts are summarized. An obvious extension in this direction is to also consider frequencies of terms in the input text (as described in section 3.2) and thereby measure the probability of encountering an instance of a concept in the text.

Probabilities provide a means for selecting summarizers without taking other measures into account and thus allows for a straightforward approach as follows.

Assume that a set of concepts $C = \{c_1, \ldots, c_n\}$ is given and let $\mathscr{O}_C = (\mathscr{L}_C, \leq_C, \mathscr{R})$ be the instantiated ontology. Let further $child(c)$ denotes the set of immediate children and $parent(c)$ the set of immediate parents for any concept $c \in \mathscr{L}_C$. The principle is to accumulate the frequencies to more general concepts but only so that a child c contributes with $\frac{1}{|parent(c)|}$ to each parent. A summary can be derived as follows.

ALGORITHM – Probability-based summary

INPUT: Set of concepts $C = \{c_1, \ldots, c_n\}$, their relative frequencies $\{freq(c_1), \ldots, freq(c_n)\}$ and a threshold α

OUTPUT: Generalizing description $D(C, \alpha)$ for C.

1) Let the instantiated ontology for C be $\mathscr{O}_C = (\mathscr{L}_C, \leq_C, \mathscr{R})$
2) Accumulate the frequencies in correspondence with the ontology such that $\forall c \in \mathscr{L}_C \backslash C : freq(c) = \sum_{c' \in child(c)} \frac{1}{|parent(c')|} freq(c')$
3) Let $N = |C|$ and $p(c) = freq(c)/N$ be the probability of encountering c.
4) Let $\mathscr{O}'_C = \left(\mathscr{L}'_C, \leq'_C, \mathscr{R}'\right)$ be the restriction of $\mathscr{O}_C$ to the concepts that appear in $\{c \in \mathscr{L}_C | p(c) \geq \alpha\}$
5) Set the α-level summary of C to the most specific concepts appearing in $\mathscr{O}'_C$, that is $D(C, \alpha) = \{c \in \mathscr{L}'_C | \nexists c' \in \mathscr{L}'_C : c' < c\}$

As an example consider again the SEMCOR instantiated ontology in figure 3. Among the recognized concepts most appear only once, while the frequency of *surface* and *water* is two and the frequency of *soils* is 4 (includes *stains*). We have N =16 and $C = \{$*system, dirt stain soil, phase, capillary action, interfacial tension, grease, oil, water, liquid, surface-active agent, surface*$\}$ and thus get for instance

$D(C, 0.1)$=$\{$*dirt stain soil*, *surface*, *surface tension*, *water*, *oil*$\}$

$D(C, 0.15)$=$\{$*dirt stain, soil*, *natural phenomenon*, *compound*$\}$

5 Concluding Remarks

In this paper we have considered how to use ontologies to provide data summaries with a special focus on textual data. Such summaries can be used in a querying approach where concepts describing documents, rather than documents directly, are retrieved as query answer. The summaries presented are conceptual due to fact that they exploit concepts from the text to be summarized and ontology-based because these concepts are drawn from a reference ontology.

We have presented three summary principles. Two based on similarity and clustering and the third on probabilities derived from frequencies in the text to be summarized. Obviously a ”meaningful” clustering may lead to good summaries if characterizing subsuming concepts can be found in the ontology. However, as indicated also counting occurrences, rather than only recognizing presence, of concepts may contribute to encircling essential concepts.

The connectivity and similarity based approaches as presented in this chapter are iterative in that they are defined by summarization functions to be applied repeatedly on previous results until a satisfactory contraction is obtained. The probability approach, on the other hand, is here defined with a threshold function to be applied only once but of course with the possibility to try again if the threshold given lead to too much or too little contraction of the input. Basically this difference is not decisive since initially deciding a number of iterations would correspond to specifying a threshold and likewise an iterative approach could be obtained from an appropriate repeated regulation of the threshold.

A more crucial difference between the approaches presented relates to on what ground summarizing concepts are chosen. Connectivity and similarity approaches apply background knowledge only and the main differentiation is wether a concept appears in the corpus or not although a weighting scheme based on counting subsumed concepts is also introduced with the support measure. The probability approach, on the other hand, is quite different and introduces a far more dominating weighting scheme based on corpus statistics. Intuitively this should lead to more refined results. Concepts that are very frequent in the data that we want to summarize should also play a central role in the summarization.

However the probability approach as presented ignores similarity and does not take into account whether a candidate summarizers subsumed concepts are similar or not. Obviously a summarizer would be more characteristic if its subsumed concepts are similar and less similar if they do not have much in common.

So an approach that combines similarity and corpus statistics appears to be an obvious next step. One solution is an extended similarity approach based on weighted descriptors as briefly introduced above. If the weights reflect frequencies in the corpus then we obviously have such a combined approach. This way the selection of which clusters to merge next not only is based on similarity but also on frequencies, hence generalization of highly frequent concepts can be delayed or prevented.

Alternatively we can consider an extended probability approach where for each concept, in addition to the probability, is derived a measure reflecting a ”deviation”, that is, a measure of the extend to which the given concept’s subsumed concepts are

similar. One possibility for this measure is to aggregate the similarity between all pairs of subsumed concepts. A node with only little deviation is more representative than one with high. Rather than an ontology only with probabilities, as in figure 3, we can provide the ontology with node weights, that is, for instance a weighted average between the probabilities and the representational value. This can be used for a more refined selection of concepts in a summary.

Independent of the summarization principle in play one major challenge we also need to attack in the future is the evaluation problem, that is, how to measure the quality of conceptual summarization. Good characteristics for summaries are not obvious, but for a summarization principle to work in practise users clearly need some kind of guidance on how many times to iterate or how to set a threshold. Initial considerations on the quality of summaries can be found in [19] but the issue is also an obvious direction for further work in continuation of what has been described here.

References

1. Jensen, P.A., Nilsson, J.F.: Ontology-based Semantics for Prepositions, in Syntax and Semantics of Prepositions. In: Paint-Dizier, P. (ed.) Text, Speech and Language Technology, vol. 29. Springer, Heidelberg (2006)
2. Aronson, S.R.: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proc. AMIA Symp., pp. 17–21 (2001)
3. Abney, S.: Partial parsing via finite-state cascades. In: Proceedings of the ESSLLI 1996 Robust Parsing Workshop (1996)
4. Bodenreider, O.: The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research 32, D267–D270 (2004)
5. Hahn, U., Mani, I.: The Challenges of Automatic Summarization Computer (November 2000)
6. Melli, G., Wang, Y., Liu, Y., Kashani, M.M., Shi, Z., Gu, B., Sarkar, A., Popowich, F.: Description of SQUASH, the SFU Question Answering Summary Handler for the DUC 2005 Summarization Task. In: Proceedings of DUC 2005, Vancouver, Canada, pp. 103–110 (2005)
7. Shi, Z., Melli, G., Wang, Y., Liu, Y., Gu, B., Kashani, M.M., Sarkar, A., Popowich, F.: Question Answering Summarization of Multiple Biomedical Documents. In: Kobti, Z., Wu, D. (eds.) Canadian AI 2007. LNCS, vol. 4509, pp. 284–295. Springer, Heidelberg (2007)
8. Andreasen, T., Bulskov, H.: Conceptual Querying Through Ontologies. Fuzzy Sets and Systems (2008) (to appear)
9. Nilsson, J.F.: A logico-algebraic framework for ontologies – ONTOLOG. In: Jensen, P.A., Skadhauge, P. (eds.) First International OntoQuery Workshop, University of Southern Denmark (2001)
10. Miller, G.A., Chodorow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a semantic concordance for sense identification. In: Proc. of the ARPA Human Language Technology Workshop, pp. 240–243 (1994)
11. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM 38(11), 39–41 (1995)

12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics 19(1), 17–30 (1989)
13. Unified Medical Language System U.S. National Library of Medicine, http://www.nlm.nih.gov/research/umls/
14. Medical Literature Analysis and Retrieval System Online U.S. National Library of Medicine, http://www.ncbi.nlm.nih.gov/pubmed/
15. Bulskov, H., Knappe, R., Andreasen, T.: On measuring similarity for conceptual querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS, vol. 2522, pp. 100–111. Springer, Heidelberg (2002)
16. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language (1999)
17. Andreasen, T., Knappe, R., Bulskov, H.: Domain-specific similarity and retrieval. In: Proceedings IFSA 2005, pp. 496–502. Tsinghua University Press (2005)
18. Andreasen, T., Jensen, P.A., Nilsson, J.F., Paggio, P., Pedersen, B.S., Thomsen, H.E.: Content-based text querying with ontological descriptors. Data Knowledge Engineering 48(2), 199–219
19. Yager, R.R., Petry, F.E.: A Multicriteria Approach to Data Summarization Using Concept Hierarchies. IEEE Trans. on Fuzzy Sys. 14(6) (2006)
20. Bulskov, H., Andreasen, T., Terney, T.V.: Conceptual Summaries as Query Answers. In: Fuzzy Information Processing Society, 2007. NAFIPS apos 2007. Annual Meeting of the North American, June 24-27, pp. 458–462 (2007)
21. Andreasen, T., Bulskov, H.: Conceptual Querying Through Ontologies. In: Fuzzy Sets and Systems (2008) (to appear)
22. Zhou, X., Han, H.: Survey of word sense disambiguation approaches. In: 18th FLAIRS Conference (2005)

Granular Computing for Web Intelligence

Yiyu Yao and Ning Zhong

Abstract. The World Wide Web, or simply the Web, is a large-scale and complex system that humans created in recent years. The Web brings opportunities and challenges for academic and industry communities and almost everyone on this planet as well. Due to its huge scale and complexity, one may find that it is impossible to search for simple theories and models for explaining the Web. Instead, more complicated theories and methodologies are needed, so that the Web can be examined from various perspectives. There are two purposes of the this chapter. One is to present an overview of the triarchic theory of granular computing, and the other is to examine granular computing perspectives on Web Intelligence (WI).

1 Introduction

The Web has evolved quickly into a huge and complex system with great social and technological impacts. The investigation and research of such a large-scale system requires a full exploration of existing theories and tools and calls for new ones. Many proposals have been made and extensively explored, including Web Intelligence (WI) [1, 2, 3, 4, 5, 6], Web Engineering (WE), Web Technology (WT), Semantics Web (SW) [7] and Web Science (WS) [8]. WI research represents one of the frontier efforts in making the Web towards the intelligent Web and eventually the Wisdom Web [3, 9, 10, 11]. The scientific exploration of the Web, leading by WI, would bring us to new horizons. Results, lessons, and experiences from existing disciplines can be applied to the study of the Web. The Web may also introduce new problems and challenges to the established disciplines [2].

Yiyu Yao
Department of Computer Science, University of Regina,
Regina, Saskatchewan, Canada, S4S 0A2

The International WIC Institute/BJUT
e-mail: yyao@cs.uregina.ca

Ning Zhong
Department of Life Science and Informatics, Maebashi Institute of Technology, Japan,
460-1 Kamisadori-Cho, Maebashi-City 371-0816, Japan

The International WIC Institute/BJUT
e-mail: zhong@maebashi-it.ac.jp

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 89–102.
springerlink.com

The emerging field of study, known as granular computing [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24], may provide the necessary theories that support WI research. The philosophy, methodology, and computation paradigm of granular computing offer multiple levels of explanation of the Web and guides the design and implementation of new types of Web-based information processing systems. Granular computing provides an effective means for building conceptual models of the Web, studying the organizations and structures of the Web, analyzing the contents and knowledge of the Web, and discovering useful knowledge from the usage of the Web. The essential notions of granular computing, namely, multilevel and multiview, will provide new insights into WI research.

2 The Triarchic Theory of Granular Computing

Granular computing may be viewed as a new way of thinking and computation that exploits varying sized data, information, knowledge, and wisdom granules. It is nature-inspired computing that is applicable to machine-centric computing. Its fundamental issues, principles, methodologies, scopes, and goals can be studied in a triarchic theory of granular computing [17, 18, 19, 21]. The theory integrates three perspectives, namely, the philosophy, the methodology and the computation, based on granular structures. The three perspectives are corners of the granular computing triangle.

2.1 Multilevel View and Multiview Understanding

Results from cognitive science and psychology on human guessing, knowing, thinking and languages provide evidence to support the view that humans perceive, understand, and represent the real world in multiple levels of granularity and abstraction [22]. To formally represent this view of human-inspired computing, granular computing studies the fundamental notions of granules and associated granular structures. It is the stress on structures that makes granular computing unique and potentially useful.

Intuitively speaking, granules are parts of a whole. They are the focal points of the current interest or the units used to obtain a description or a representation. The meaning of granules would become clearer as soon as a particular problem or application is considered. For example, in computer programming granules may be interpreted as modules or subprograms. In scientific writing granules may be words, sentences, paragraphs, etc., depending on the focus of attention. In the study of organizational structures, granules represent various levels of division. The interpretation of granules as parts of whole is simple and yet sufficient for building a useful model of granular computing.

A granule normally serves dual roles. It is a single un-dividable unit when it is considered as part of another granule; it is a whole consisting of interconnected and interacting granules when some other granules are viewed as its parts. Properties of

granules may be grouped into three types. The internal properties of a granule reflect its organizational structures and the relationships and interaction of its element granules. The external properties of a granule reveal its interaction with other granules. The contextual properties of a granule show its relative existence in a particular environment. The three types of properties provide us a full picture of the notion of granules.

Relationships among granules provide a base for building granular structures. A family of granules of a similar type may be collected together in order to study their collective properties. This leads to the notion of levels. While each granule provides a local view, a level provides a global view. It is desirable that granules in the same level should be as independent as possible, and granules in different levels are related to each other based on their external and contextual properties. That is, the internal properties of a granule are not considered when investigating its relationships to other granules.

An important property of granules and levels is their granularity, which results in a partial ordering of levels. Formally, this can be described by a hierarchical structure (i.e., a hierarchy). The term hierarchy is used loosely to define a structure that is weaker than a tree or a lattice. Only a minimum requirement is imposed on the hierarchical structure, namely, different levels consist of granules with different-sized granules so that the ordering of levels is meaningful. That is, granular structure is made up by a family of partially ordered levels, and each level is made up by a family of granules.

Building a hierarchical granular structure relies on a vertical separation of levels and a horizontal separation of granules at the same level [25]. Since these separations normally come with information loss, a granular structure may be viewed as an approximate model of the reality from a particular angle. In order to obtain a more realistic modeling, it may be necessary to consider multiple hierarchies. By doing so, a multiview framework is obtained [26].

In a nutshell, granular structures may be described as a multilevel view given by a single hierarchy and a multiview understanding given by many hierarchies.

2.2 Structured Thinking

The philosophy of granular computing is a world-view that promotes an understanding and interpretation of the reality by focusing on multiple levels of granularity. This philosophy requires structured thinking in stating and solving real-world problem. The philosophy of granular computing has been influenced by the traditional reductionist thinking and the more recent systems thinking. Reductionist thinking models a complex system or problem by dividing it into simpler and more fundamental parts, and further dividing these parts. It is assumed that an understanding of the system can be reduced to the understanding of its parts. In other words, it is possible to deduce fully the properties of a system based solely on the properties of its parts.

Systems thinking focuses on the connectedness, relationships, and context of parts and whole [27, 28, 29]. A complex system is viewed as an integrated whole consisting of a web of interconnected, interacting, and highly organized parts. The properties of the whole are not present in any of its parts, but emerge from the interactions and relationships of the parts. In other words, it is impossible to deduce the properties of a system based solely on the properties of its parts.

The use of nested structures for modeling is common to both the reductionist thinking and the systems thinking. The adoption of hierarchical granular structures of the last section is, in fact, motivated by this commonality. In some sense, granular computing attempts to unify complementary reductionist thinking and systems thinking into structured thinking. It combines analytical thinking for decomposing a whole into parts and synthetic thinking for integrating parts into a whole. One may switch between the two at different stages in problem solving.

2.3 Structured Problem Solving

From the methodology perspective, granular computing may recall results from different disciplines. For example, the effective methodology of structured programming [30], characterized by top-down design and step-wise refinement, is generally applicable to other types of problem solving, and hence may provide a methodological foundation for granular computing. Similarly, problem solving heuristics, methods, and strategies well studied in cognitive science [31] and artificial intelligence [32] may offer their contributions to the methodology of granular computing. The solid formulation and successful applications of the rough set theory [14, 15, 33, 34, 35] not only gives a concrete model but also convinces many researchers about the potential values of granular computing.

Granular computing promotes systematic approaches, effective principles, and practical heuristics and strategies that have been used effectively by humans for solving real-world problems. A central issue is the exploration of granular structures. This involves three basic tasks: constructing granular structures, working within a particular level of the structures, and switching between levels. The methodology of granular computing is inspired by human problem solving. Thus, granular computing is related to natural-inspired computing [36].

A good way to describe the methodology perspective of granular computing is to construct a set of principles. A few examples of such principles are [21, 22]:

- the principle of multilevel view;
- the principle of multiview understanding;
- the principle of focused efforts;
- the principle of granularity conversion;
- the principle of view switching.

The first two principles emphasizes the use of a hierarchical structure as well as many hierarchies. It is necessary to consider multiple representations at different levels of granularity in a hierarchy and to consider multiple hierarchies at the same time. From the consideration of multiple levels one obtains a view with many levels

of abstraction; from the consideration of multiple hierarchies one obtains multiple versions of the same world. They together offer many angles and perspectives for the same problem. Once granular structures are obtained, other principles come into play. The principle of focused efforts calls for attention on the focal point at a particular stage of problem solving. While the principle of granularity conversion links the different stages in this process, the principle of view switching allows us to change views and to compare different views.

Many more principles may be formulated and articulated. It is hoped that the study of granular computing will lead to a set of well accepted principles and laws in the near future. In this regard, one can draw results from structured programming and systems theory.

2.4 Structured Information Processing

As a paradigm of structured information processing [12], granular computing focuses on computations based on granular structures. This exploration of a pyramid of different-sized information granules is essential to any knowledge-intensive system.

For the computational perspective, two related basic issues to be considered are representation and process. In general, a representation method is a formal system that describes explicitly certain entities or types of information. The result is called a description of the entity in the representation [37]. A process may be interpreted as actions or procedures for carrying out information processing tasks [37]. A representation method determines the effectiveness of processes. In the context of computer programming, a representation method may specify a certain type of data structures, and processes are allowable operations on the data structures. For granular computing, a representation method concerns descriptions of granular structures and processes concerns operations on such structures.

Any representation method used in granular computing must capture the essential features of granules, levels, and hierarchies. The results are formal descriptions of these notions on which it is possible to carry out information processing tasks. In order to implement the principles discussed earlier for the methodological perspective, a representation needs to deal with three aspects. For individual granules, a representation method concerns their internal, external and contextual properties. For a hierarchy, the representation concerns the collective properties of a family of granules in different levels, as well as partial ordering of levels. For many hierarchies, the representation concerns relationships between distinct hierarchies. Consequently, it is possible to switch between levels in a single hierarchy and to switch between many hierarchies.

Processes of granular computing may be broadly divided into the two classes: granulation and computation with granules. Granulation processes involve the construction of the building blocks and structures, namely, granules, levels, and

hierarchies. Computation processes systematically explore the granular structures. This involves two-way communications up and down in a hierarchy, as well as switching between levels.

Structured information processing may be viewed a stepwise refinement process. At a higher level, one may produce an approximate, a partial, or a schematic solution. The latter is to be made more precise, complete, and detailed at a lower level. The process stops when a desirable (approximate) solution is obtained.

3 Web Intelligence (WI)

WI is both an interdisciplinary and a transfdisciplinary study that focuses on systematic investigations of advanced Web related theories, methodologies, technologies, and tools, as well as the design and implementation of Intelligent Web Information Systems (IWIS) [2]. It explores the fundamental roles as well as practical impacts of Artificial Intelligence (AI) and advanced Information Technology (IT) on the next generation of Web-empowered products, systems, services, and activities [1, 3, 4, 5, 9, 10, 11, 38, 39]. A practical goal of WI research is the design and implementation of Intelligent Web Information Systems [2].

On the one hand, WI research is based on, and draws results from, many fields such as Artificial Intelligence (AI), Information Technology (IT), cognitive science and human problem solving, data mining and knowledge discovery, information storage and retrieval, and many more. On the other hand, WI research contributes in a novel way to these related areas. For example, WI explores the notion of intelligence based on the new platform of the Web, which is massively distributed and self-organizing and evolving. The Web log data provide new challenges for machine learning and data mining. The Web search totally changes the agenda of conventional information retrieval systems. The Web also provides a new media for collaboration and co-creation among researchers on a different scale. WI is the key and the most urgent research field of IT in the era of Web and agent intelligence.

Research on WI can be broadly divided into the following groups [4, 5]:

- WI Foundations;
- World Wide Wisdom Web (W4);
- Social Networks and Social Intelligence;
- Knowledge Grids and Grid Intelligence;
- Web Mining and Farming;
- Semantics and Ontology Engineering;
- Web Agents;
- Web Services;
- Web Information Retrieval and Filtering;
- Intelligent Human-Web Interaction;
- Web Support Systems;
- Intelligent e-Technologies.

They clearly focus more on the perspectives of the *form*, *matter*, and *process*, with less emphasis on the *meaning*. Future research may be carried out on the *meaning* perspective, namely, the social perspective of the Web [40, 41].

From the current research on the Web, one can observe two broad classes with distinctive goals [42]. One class of research concentrates on the *exploration and utilization of the Web*. It may be considered as the Web user's view. The goal is to make full and effective use of the existing Web for problem solving in many diverse domains. Studies in this class include application of existing tools and methodologies to explore the rich information available on the Web, adaptation of existing systems and technologies to the Web, integration with legacy systems, integration of different intelligent information systems, and so on.

The other class focuses on the *enhancement and extension of the Web*. It may be considered as the Web developer's view. The goal is to build new theories and tools for the next generation of the Web. Studies in this class include foundations of the Web, new Web infrastructures, new Web functionality, and so on. The Wisdom Web, Semantic Web, Web 2.0 and new waves of the Web are typical examples from this class of research.

There does not exist a clear cut between the two classes of research. The applications of the Web raise questions and additional requirements of the Web; a new generation of the Web leads to improved applications, as well as new applications. By embracing both classes of research, WI attempts to fully explore the structures, semantics, and knowledge of the Web. This brings in granular computing as one of its key theories.

4 Granular Computing Perspectives on Web Intelligence

Granular computing is relevant to the WI research in several ways. The ideas of multilevel and multiview of granular computing may guide the investigation of the Web, including the structures, semantics and knowledge of the Web and on the Web. The methodology and paradigm of granular computing may be helpful in the design and implementation of Web-based intelligent information processing systems.

4.1 The Relevance of Granular Computing

In an attempt to arrive at a systemic understanding of life, Capra [27] suggests a conceptual framework represented by a tetrahedron. It integrates the four perspectives of life, namely, form, matter, process, and meaning. If the Web is viewed as an evolving system, Capra's model is immediately applicable to its understanding. In this context, the *form* may be viewed as a network of people, machines, documents, and so on. Such a form is embodied in the underlying physical computer network and information network, namely, the *matter*. The communications and interactions in the Web may be viewed as the *process*. Finally, the values, namely, the *meaning*, of the Web are created by its social and technological impacts. Integrating the four perspectives may lead to a holistic understanding of the Web.

The acceptance of the Web as an evolving system makes the large reservoir of ideas, methods and tools from general systems theory [28] at our disposal for the study of the Web. One may compare the recent rise of granular computing with the rise of systems theory a few decades earlier [22]. There is compounding evidence supporting that granular computing may benefit from a study of systems theory. In spite of their different contexts, systems theory and granular computing share high-level similarity with respect to their ideas, philosophy, scope and goals. The general systems theory attempts to discover and investigate structures and underlying principles common to most natural and artificial systems. Granular computing attempts to derive a unified framework of human-inspired computing characterized by an effective use of multiple levels of granularity. More specifically, general systems architectures are relevant to granular structures, systems hierarchy is relevant to multiple levels of granularity, and the systems models are relevant to granule based computational models.

With the general framework of Capra, one may focus on exploring particular perspectives of the Web. From the viewpoint of a computer scientist, the form, matter and process of the Web are of immediate concerns. They can be investigated by using ideas from granular computing.

4.2 Multiview Understanding of the Web

According to the principles of granular computing, one can study the Web from multiple views and at multiple levels in each view. Some perspectives of the Web and WI research are given as follows [40]:

- Computer Science Perspectives. The Web is considered as an infrastructure to support information, knowledge, services and resources sharing, realized by many types of intelligent systems on the Web. Some of the main tasks are: to study its theoretical foundations, to establish its technical foundations, to build physical infrastructures and to develop software systems that support the Web, and to develop various applications that fully realize the potentials of the Web in many different domains. Research efforts can be broadly summarized into three categories: theoretical studies or the logical view of the Web, implementations or the physical view of the Web, and the application view of the Web.
- Information Science and Knowledge Management Perspectives. The data, information, knowledge and wisdom hierarchy is a well studied notion in information science and knowledge management. The hierarchy represents increasing levels of complexity that require increasing levels of understanding. The generations of intelligent information systems are determined by the hierarchy. The hierarchy immediately offers a natural multilevel study of the Web. It is possible to lay out a general evolution trend of the Web, namely, from the Data Web, to the Information Web, to the Knowledge Web, eventually to the Wisdom Web.
- Social Intelligence Perspectives. The Web is both a social creation and a technical one [41]. It provides a means for people to collaborate and interact better. Web-based community and society may be formed either explicitly or implicitly

through associations. Such associations create the social networks. The connectivity of the Web leads to the connectivity of people, which is an essential component of a web-like and virtual digital society. At multiple levels, WI research aims at analyzing the social network intelligence, in order to better support interaction and collaboration.

- Application Perspectives. The Web supports a wide range of applications, ranging from simple information sharing to complicated e-market place. Applications drive the evolution of the Web and evolve with the Web. The principles of multilevel and multiview of granular computing promote an organization of applications into multiple levels according some criteria. Such organizations enable the Web to provide better supports.

One may easily consider other perspectives. The integrations of these perspectives leads to a holistic understanding of the Web and the WI research.

Each class of perspectives can be further divided. Within each view, one may have a more detailed study based on multiple levels. For example, from computer science perspectives, we may consider the following four conceptual levels [3]:

- Internet-level communication, infrastructure, and security protocols.
- Interface-level multimedia presentation standards.
- Knowledge-level information processing and management tools.
- Application-level ubiquitous computing and social intelligence environments.

According to the evolution of the Web, one may consider the following levels [40]:

- Web of Machines.
- Web of Pages (Websites).
- Web of Dynamic/Adaptive Pages (Websites).
- Web of Agents.
- Web of Services.
- Web of Resources.

Each hierarchy explores a particular multilevel view and many such views reflect the diversity of WI research, as well as different application domains and goals.

Studying the Web from the multiple perspectives serves as an example to demonstrate the effectiveness of granular computing in organizing and guiding research in a particular field. The ideas of granular computing can be equally applicable to other fields. For example, the principles of granular computing can be used to study its own research. As shown in the last section, the results are also a multilevel and multiview understanding.

4.3 Web-Based Information Retrieval Support Systems

A significant feature of the Web is that it carries a huge amount of information. The usefulness of the Web depends, to a large extent, on many search engines for searching and browsing the Web. Information retrieval support systems (IRSS) are the next generation of search systems in their evolution from data retrieval to

information retrieval, and from information retrieval to information retrieval support [43, 44, 45, 46].

Most search engines are designed based on the principle of information retrieval (IR) [47, 48]. They inherit many of the disadvantages of traditional IR systems. For example, IR systems focus mainly on the retrieval functionality, namely, the selection of a subset of documents from a large collection. There is little support for other activities. IR systems use simple document and query representation schemes. A document is typically represented as a list of keywords and a query is represented as either a list of keywords or a Boolean expression. There is little consideration of the relationships between different documents and between different portions of the same document. Semantic and structure information in each document is not used. IR systems use simple pattern based matching methods to identify relevant documents. It becomes evident that today's search engines with yesterday's IR ideas are inadequate to support the new waves of the Web.

The study of information retrieval support systems attempts to extend and modify traditional IR systems to meet the new challenges brought by the Web. These systems should have the following features [43, 44, 45, 46]:

- A new design philosophy. IRSS is based on a design philosophy that is different from the traditional information retrieval. In addition to support searching and browsing, an information retrieval support system provides the necessary models, languages, utilities, and tools to assist a user in investigating, analyzing, understanding, and organizing a document collection and search results. These tools allow the user to explore both semantic and structural information of each individual document, as well as the entire collection. In the process of finding useful information, a user plays an active role by using the utilities, tools, and languages provided by the system.
- An active support role. An IRSS actively supports a user based on user profiles, usage history, and other relevant information. In some sense, such a system may serve as a personal agent that actively advices a user when new information is available on the Web. In particular, on behalf of the user, the system may browse and analyze the Web on a periodical basis.
- An emphasis on effectiveness. An IRSS emphasizes effective support rather than the efficiency required by online search engines. Instead of providing a list of references, the system must provide reports covering multiple levels of details. A user will save time by browsing the organized results from the systems, instead of browsing the Web.
- Knowledge-based and personalized support. An IRSS must provide personalized support based on its domain-specific knowledge bases and user profiles. In other words, the system is adaptive to individual users. In contrast to the current search engines, an IRSS will provide better support by exploring domain knowledge and semantics information on the Web.

These features require that an IRSS must employ multiple representations for Web documents, multiple strategies for retrieval, and multiple profiles for users. The principles and ideas of granular computing again can be used to achieve such goals.

Three related types of models need to be considered in IRSS. Documents in a collection serve as the raw data. The document models deal with representations and interpretations of documents and the document collection. The retrieval models deals with the search. The presentation models deal with the representation and interpretations of results from the search. A single document model, a retrieval model, or presentation model may not be suitable for different types of users. Therefore, IRSS must support multi-model and provide tools for users to manage various models.

An important tool used in these models is multilevel granular structures introduced in granular computing. Specifically, four types of granulations are considered. They are:

- Term space granulations. In the context of information retrieval, index terms may be viewed as granules, and granular structures are typically given by term hierarchies. A term hierarchy serves as an effective tool to summarize knowledge about a specific domain. As documents, queries, and retrieval results are normally represented as terms, term space granulation will naturally lead the granulations of others.
- Document space granulation. Documents may be granulated in several ways, such as content based, query based, and citation based approaches. The most commonly way is content or topic based. Documents with similar content or topic are put into the same cluster. A clustering of documents provide a granulated view of the document collection. A hierarchical clustering of documents is produced by decomposing large clusters into smaller ones. The large clusters offer a rough representation of the document. The representation becomes more precise as one moves towards the smaller clusters. A document is described by different representations at various levels. This leads to multi-representation of documents. Document space granulations may also be derived and explained by term space granulations. An added advantage of document space granulation is that different retrieval methods may be employed at different levels.
- Query (User) space granulations. One can construct granulated views of query (user) space in several ways, such as content based, document based, and usage based approaches. Similar queries are grouped together to represent the needs of a group of users. Query space granulation is useful in providing personalized and domain-specific support.
- Retrieval results granulations. By clustering retrieval results, one can organize the results and provide coarse-grained summarization to users. An important issue in query specific document clustering is to obtain a meaningful description of the derived clusters to be presented to the user. One may extract some important sentences from the documents in a cluster as a description of the cluster.

Based on these granulations, ideas and principles of granular computing can be easily applied. With granular structures, the document, retrieval and presentation models work together to provide many support functionalities to a user.

5 Conclusion

The advances of the Web require new theories, methodology and tools. WI represents one of the frontier research efforts in making the Web evolve towards an Intelligent Web and eventually the Wisdom Web. Granular computing, as human-inspired computing, seems to be a potentially useful theory for WI research.

Regarding the relevance of granular computing to WI research, a few directions are pointed out. First, the ideas of multilevel and multiview of granular computing may guide WI research. Second, hierarchical granular structures may be helpful to the investigation of the data, information and knowledge on the Web, as well as the physical structures of the Web and its embodied virtual structures. Third, granular computing is useful to the design and implementation of Web-based intelligent systems, of which Web-based information retrieval support systems are a special class.

According to the principles of granular computing, an understanding of a field of study involves explanations at many levels of abstraction. Although any new field of study always starts with scattered and concrete ideas, a high level explanation may be valuable to its healthy growth. It would be less effective to work on a lower level, if a higher-level understanding is not reached and available. This chapter serves exactly such a purpose for both WI and granular computing, as well as their integration.

According to principles of granular computing, it is necessary to switch to different levels. Future research will therefore aim at obtaining an understanding of the same problem at lower levels. Guided by a higher level understanding, future research will be directed at further articulation, elaboration, and concretization of ideas briefly mentioned in this chapter.

References

1. Zhong, N., Liu, J., Yao, Y.Y., Ohsuga, S.: Web Intelligence (WI). In: Proceedings of the 24th IEEE Computer Society International Computer Software and Applications Conference, pp. 469–470 (2000)
2. Yao, Y.Y., Zhong, N., Liu, J., Ohsuga, S.: Web Intelligence (WI): research challenges and trends in the new information age. In: Zhong, N., Yao, Y., Liu, J., Ohsuga, S. (eds.) WI 2001. LNCS (LNAI), vol. 2198, pp. 1–17. Springer, Heidelberg (2001)
3. Zhong, N., Liu, J., Yao, Y.Y.: In search of the Wisdom Web. IEEE Computer 35, 27–31 (2002)
4. Zhong, N.: Toward Web Intelligence. In: Menasalvas, E., Segovia, J., Szczepaniak, P.S. (eds.) AWIC 2003. LNCS (LNAI), vol. 2663, pp. 1–14. Springer, Heidelberg (2003)
5. Zhong, N., Liu, J., Yao, Y.Y. (eds.): Web Intelligence. Springer, Berlin (2003)
6. Zhong, N., Liu, J., Yao, Y.Y.: Envisioning intelligent information technologies through the prism of web intelligence. Communications of the ACM 50, 89–94 (2007)
7. Berners-Lee, T., Hendler, J., Lassila, O.: Semantic Web, a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American 248, 34–43 (2001)
8. Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K., Shadbolt, N., Weitzner, D.J.: A framework for web science. Foundations and Trends in Web Science 1, 1–130 (2006)

9. Liu, J.: Web Intelligence (WI): What makes Wisdom Web? In: Proceedings of the International Joint Conferences on Artificial Intelligence, pp. 1596–1701 (2003)
10. Liu, J.: New challenges in the World Wide Wisdom Web (W4) research. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2003. LNCS (LNAI), vol. 2871, pp. 1–6. Springer, Heidelberg (2003)
11. Liu, J.: The World Wide Wisdom Web (W4). In: Bianchi-Berthouze, N. (ed.) DNIS 2003. LNCS (LNAI), vol. 2822, pp. 1–4. Springer, Heidelberg (2003)
12. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston (2002)
13. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centred information processing. IEEE Transactions On Fuzzy Systems 16, 320–330 (2008)
14. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Data Mining, Rough Sets and granular Computing. Physica-Verlag, Heidelberg (2002)
15. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin (2003)
16. Yao, J.T.: A ten-year review of granular computing. In: Proceedings of the 3rd IEEE Internationational Conference on Granular Computing, pp. 734–739 (2007)
17. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)
18. Yao, Y.Y.: A partition model of granular computing. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 232–253. Springer, Heidelberg (2004)
19. Yao, Y.Y.: Perspectives of granular computing. In: Proceedings of the IEEE International Conference on Granular Computing, pp. 85–90 (2005)
20. Yao, Y.Y.: Three perspectives of granular computing. Journal of Nanchang Institute of Technology 25, 16–21 (2006)
21. Yao, Y.Y.: The art of granular computing. In: Kryszkiewicz, M., Peters, J.F., Rybinski, H., Skowron, A. (eds.) RSEISP 2007. LNCS (LNAI), vol. 4585, pp. 101–112. Springer, Heidelberg (2007)
22. Yao, Y.Y.: The rise of granular computing. Journal of Chongqing University of Posts and Telecommunication 20, 299–308 (2008)
23. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 19, 111–127 (1997)
24. Zadeh, L.A.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computing 2, 23–25 (1998)
25. Simon, H.A.: The Sciences of the Artificial. The MIT Press, Massachusetts (1969)
26. Chen, Y.H., Yao, Y.Y.: A multiview approach for intelligent data analysis based on data operators. Information Sciences 178, 1–20 (2008)
27. Capra, F.: The Hidden Connections: A Science for Sustainable Living. Anchor Books, New York (2002)
28. Skyttner, L.: General Systems Theory, Ideas & Applications. World Scientific, Singapore (2001)
29. Capra, F.: The Web of Life. Anchor Books, New York (1997)
30. Ledgard, H.F., Gueras, J.F., Nagin, P.A.: PASCAL with Style: Programming Proverbs. Hayden Book Company, Rechelle Park (1979)
31. Solso, R.L., MacLin, M.K., MacLin, O.H.: Cognitive Psychology, 7th edn. Allyn and Bacon, New York (2005)
32. Zhang, L., Zhang, B.: The quotient space theory of problem solving. Fundamenta Informatcae 59, 287–298 (2004)

33. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 106–110 (1998)
34. Nguyen, H.S., Skowron, A., Stepaniuk, J.: Granular computing: a rough set approach. Computational Intelligence 17, 514–544 (2001)
35. Polkowski, L., Semeniuk-Polkowska, M.: On foundations and applications of the paradigm of granular rough computing. International Journal of Cognitive Informatics and Natural Intelligence 2, 80–94 (2008)
36. Liu, J., Tsui, K.C.: Toward nature-inspired computing. Communications of the ACM 49, 59–64 (2006)
37. Marr, D.: Vision, A Computational Investigation into Human Representation and Processing of Visual Information. W.H. Freeman and Company, San Francisco (1982)
38. Zhong, N.: Representation and construction of ontologies for Web Intelligence. International Journal of Foundations of Computer Science 13, 555–570 (2002)
39. Liu, J., Zhong, N., Yao, Y.Y., Ras, Z.W.: The Wisdom Web: new challenges for Web Intelligence (WI). Journal of Intelligence Information Systems 20, 5–9 (2003)
40. Yao, Y.Y.: Web intelligence: new frontiers of exploration. In: Proceedings of the International Conference on Active Media Technology, pp. 3–8 (2005)
41. Berners-Lee, T., Fischetti, M.: Weaving the Web: the Original Design and Ultimate Destiny of the World Wide Web by its Inventor. Harper, San Francisco (1999)
42. Yao, Y.Y., Zhong, N., Liu, J., Ohsuga, S.: Web Intelligence: exploring structures, semantics, and knowledge of the Web. Knowledge-Based Systems, 175–177 (2004)
43. Yao, Y.Y.: Information retrieval support systems. In: Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 773–778 (2002)
44. Yao, Y.Y.: Granular computing for the design of information retrieval support systems. In: Wu, W., Xiong, H., Shekhar, S. (eds.) Clustering and Information Retrieval, pp. 299–329. Kluwer Academic Publishers, Dordrecht (2003)
45. Yao, J.T., Yao, Y.Y.: Web-based information retrieval support systems: building research tools for scientists in the new information age. In: Proceedings of the IEEE/WIC International Conference on Web Intelligence, pp. 570–573 (2003)
46. Yao, Y.Y., Zeng, Y., Zhong, N.: Supporting literature exploration with granular knowledge structures. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS (LNAI), vol. 4482, pp. 182–189. Springer, Heidelberg (2007)
47. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)
48. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)

Part II
Knowledge Discovery

Visualizing High Dimensional Classifier Performance Data

Rocio Alaiz-Rodríguez, Nathalie Japkowicz, and Peter Tischer

Abstract. Classifier performance evaluation, which typically yields a vast number of results, may be approached as a problem of analyzing high dimensional data. Conducting an exploratory analysis of visual representations of this evaluation data enables us to exploit the advantages of the powerful human visual capabilities. This allows us to gain insight into the performance data, interact with it and draw meaningful conclusions about the classifiers and domains under study. We illustrate how visual techniques, based on a projection from a high dimensional space to a lower dimensional one, enable such an exploratory process. Moreover, this approach can be viewed as a generalization of conventional evaluation procedures based on point metrics that necessarily imply a higher loss of information. Finally, we show that within this framework, the user is able to study the evaluation data from a classifier point of view and from a domain point of view, which is infeasible with traditional evaluation methods.

1 Introduction

In supervised learning, a classifier is a function which, given a set of independent attributes that correspond to an instance, produces an output indicative of this instance's class labels. The classifier's output may be either a hard output (i.e., a strict

Rocío Alaiz-Rodríguez
Dpto. de Ingeniería Eléctrica y de Sistemas y Automatica,
Universidad de León, Campus de Vegazana,
24071 León, Spain
e-mail: rocio.alaiz@unileon.es

Nathalie Japkowicz
School of Information Technology and Engineering, University of Ottawa,
150 Louis Pasteur, P.O. Box 450 Stn. A Ottawa, Ontario, Canada
e-mail: nat@site.uottawa.ca

Peter Tischer
Clayton School of Information Technology, Monash University, Australia
e-mail: Peter.Tischer@infotech.monash.edu.au

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 105–129.
springerlink.com

class label) or a soft output (for example, a vector of probabilities) that can be turned into a hard output by selecting an appropriate threshold.

Classifier performance evaluation is a key issue in the pattern recognition field. It is important both in the process of developing a new classification technique and in selecting or building a particular classifier for a practical application domain.

Classifier evaluation typically leads to a multitude of results. The evaluation data, in its raw format, contains the information about the class label and the output given by the classifier for each test instance (coming from several domains or a single one), and based on these high dimensional data, the researcher or practitioner should be able to answer questions and draw conclusions about the classifiers' performance.

There are many aspects of performance we may be interested in analyzing [4]. With some of them, there is a natural reference point and we can refer to this comparison as absolute performance analysis. For instance, if we are measuring the accuracy of the classifier's prediction, we can compare against the ideal classifier. In other cases, we are interested in comparisons involving two entities. This can be regarded as relative performance analysis. This is the case, for example, when we want to compare a particular classifier with respect to the trivial one.

In supervised learning, performance evaluation is traditionally carried out by considering a performance vector together with the true class label vector (or its derived confusion matrix) from the test runs of several classifiers on various domains. Then, each vector (or confusion matrix) is collapsed into a scalar value and these values are compared to each other.

The natural thing to assume is that, by classifier performance, we mean the likelihood that the classifier will predict the class label correctly. For this reason, accuracy appears to have been the most widely spread criterion (i.e., scalar value) for the past two decades. Its limitations when used in isolation, however, have already been pointed out by many authors (see [9] and the references therein). Another important aspect of performance is related to the reliability of the classifier and to the extent to which a classifier's output can be trusted. The classifier's capability to estimate posterior probabilities can be measured by metrics like the Root Mean Square Error (RMSE) or the Cross Entropy (CE). Additionally, in binary classification domains, ranking the test instances becomes interesting in applications where the user may need to select the best n examples. This is commonly measured by the ROC (Receiver Operating Characteristics) curve which is usually summarized by the Area Under the Curve (AUC). There are many more metrics that can be related in some way with any of the previous ones [4].

Apart from the issues related to prediction, classifiers may also be compared in terms of the space and time resources they need to train and classify new instances. These considerations might be extremely important for real-time applications. This chapter, however, will not be concerned with time and space issues.

One limitation of the traditional evaluation approach is that, by the time the classifiers' performance are compared on a given domain the details of the raw high dimensional performance data have been lost. At this moment, the comparison only involves a single number, be it the error rate, or the RMSE. This problem becomes

worse if the comparison is conducted on several domains, when dealing with multi-class problems or when assessing supervised multi-label classification techniques.

For instance, the authors in [4] trained models using seven learning algorithms with many variants and parameter settings. There were 140,000 values of performance metrics and 1,761,018,000 results of models on test instances!

We consider that the classifier evaluation problem can be viewed as a problem of analyzing high dimensional data and therefore, it should be conducted following an exploratory data analysis. In particular, we assume that classifier evaluation requires two general steps. In the first stage, the researcher computes the results obtained by the various classifiers creating a considerable amount of data, which will, in turn, need to be analyzed, in a second stage, in order to draw valid and useful conclusions about the algorithms under study. We can say that this second stage is a data mining process in and of itself.

Note that we can see the performance metrics currently used by the machine learning community as one class of projections that can be applied to these evaluation data. Scalar metrics can be viewed as a projection from the original high dimensional space to a one dimensional space. Such projections, however, only allow us to analyze where a classifier stands in relation to *one* other classifier, which usually is the ideal classifier. If we think of our current evaluation metrics as specialized projection methods, we can generalize the procedure by extending it to (a) any projection method (standard or not), (b) any distance measure, (c) different data grouping formats of the original space and (d) any dimension of the final space.

Note that if we move from a projection to a one-dimensional space to one into two dimensions or more, we get a visual representation that allows to discover patterns in performance data, establishing both rankings with respect to the ideal classifier as well as comparisons of each classifier to the others. This approach considers the core step of evaluation as a data mining process aided by Scientific Visualization. This is also in the line with more recent evaluation methods such as ROC analysis [13] or cost curves [7] which also suggest a move towards visual approaches.

The remainder of this chapter is organized as follows. The foundations of Scientific Visualization are presented in Section 2 and its application to Performance Evaluation in Section 3. Section 4 gives an illustration of the framework and its formal description is given in Section 5. Finally, performance analysis experiments are discussed in Section 6 and conclusions in Section 7.

2 Scientific Visualization

Perhaps, we can define Scientific Visualization as representing information in visual form so that some observer could discover useful information. There are two main ways to think about Scientific Visualization. Let us call these two approaches the descriptive approach and the exploratory approach.

With the descriptive approach, visual means are chosen to convey information in such a way that the observer will readily recognize certain properties of the data. These properties are determined in advance of the visualization. This is what

happens in the use of graphs and bar charts to convey numerical data. The descriptive approach might be applied to classifier performance evaluation to highlight how classifiers which have similar accuracy can be different with respect to other aspects of performance.

For example, we might discover that techniques A, B and C tend to have similar performance while techniques D and E, which also have similar performance to each other, do not come close, in performance, to the first three techniques. We might find that classifiers generated using classification technique A tend to exhibit less variation in performance than classifiers generated using classification technique B.

With the exploratory approach the evaluation data is reduced, information is discarded, and the reduced data is displayed to the observer who may be able to recognize informative patterns in the data by using the pattern recognition abilities of the human visual system. Scientific Visualization can therefore be used in an exploratory approach to discover important properties about the data and, in this way, can be seen as a form of machine learning / data mining.

We can also think of the descriptive approach as a supervised data mining process (since the user guides or supervises the visualization) whereas the exploratory approach is unsupervised (since the user is left to discover interesting patterns in the evaluation data).

2.1 Scientific Visualization as More Than Projection

When visualizing a data set, the object of interest may be represented with different conceptual techniques depending on the nature of the data and the goal of the analysis [3]: (a) complex icons (glyphs) with features that depend on the data values (e.g., Chernoff faces or stars), (b) traces where the objects are represented by functions (Andrew curves or parallel coordinates are some representative examples) or (c) scatterplots where the objects are represented by points. Some comprehensive surveys of visualization techniques are available in [11, 6].

Often, Scientific Visualization is employed for data sets with high dimensionality. In this case, both icons and traces may lead to blurred representations, while the scatterplot based approach does not suffer from this problem. For this reason, hereafter, we will focus on this technique. A scatterplot representation requires that the dimensionality be reduced to the extent the reduced data can be displayed using two or three spatial dimensions (X,Y) or (X,Y,Z), three colour dimensions, e.g. (Y,U,V) or (R,G,B), and, by using animation techniques, possibly using a time dimension, (t). A function which maps some data in a h-dimensional space to a lower l-dimensional space, where $h > l$, is called a projection.

There is an extensive literature on projections [10, 8] both linear and non-linear. When Scientific Visualization is applied to high dimensional data, projection is needed. However, a visualization of the data might involve more than projection.

Projections map points in the higher dimensional space to points in a lower dimensional space. However, in the visualization we might convey additional information in a number of ways. For conveying information about a specific point

in higher dimensional space we can allow the representation of the point in the visualization space to vary so as to carry point-specific information. For example, in showing cities on a map, we may represent the points as circles with different radii, such that the greater the population, the larger the radius. We could convey more than one characteristics of a point by using the radius of the circle to convey one attribute, such as population size, and then splitting the circle into different coloured sectors. For example, a sector could show the proportion of people in that city who spoke a particular language.

If we are trying to convey information about the relationship between two points in the original space, a natural way to achieve this is to link the images of the points by a line segment and convey information about the relationship in the way the line segment is represented. If we want to represent the relationship between three points in the original space, we can link the representations of the points by line segments to form a triangle.

We have been investigating a projection approach where it is possible to guarantee that the distances between a pair of projected points is exactly the same as the distance between the original points (Section 5.3). In the graphs in this chapter, some points may appear to be close, but we can only be sure that the distance between projected points equals the distance between actual points if there is a line segment connecting the projected points.

2.2 Scientific Visualization as Projection Plus Visualization

A general structure for Scientific Visualization can be as follows: *Data reduction*, *Projection* to a lower dimensional space and *Visualization* of lower dimensional data.

The *Data Reduction* step removes information which is not relevant to the purpose of the Scientific Visualization. For example, if we are interested in the performance on a special class of problems, e.g. problems where items have very large numbers of attributes, we exclude information from problems that are not in that special class. If we believe that results on each test case are equally important, we can throw away information about results on specific test cases by aggregating results into counts of true positives, false positives, true negatives and false positives.

The *Projection* stage involves a mapping from the original high-dimensional space to the visualization space. The visualization space may range in dimensionality from 2, e.g. (X,Y), to 7, e.g. (X,Y,Z,R,G,B,t). By definition, projection methods map from a higher-dimensional space to a lower-dimensional space. No assumptions are made about the special properties of the two spaces. For example, the projection method may only assume that the same distance measure can be used in both spaces, e.g., an L_2-norm.

In the case of Scientific Visualization, the second space will be used to create a visualization which presents information to the Human Visual System (HVS). The HVS perceives spatial distances, colour and intensity differences and time distances differently. Thus, if we want to relate distances or differences in the visualization

space to distances or differences in the original higher-dimensional space, we need to take into account the special properties of the HVS.

When looking at a *Visualization,* a human observer will assume that large separations in the spatial dimensions will tend to correspond to large separations in the original space. Items which are projected in such a manner that they tend to occupy the same region of the visualization will be assumed to represent items which are close in the original space. A perceptual distance should weight differences in the spatial dimensions as being more important than differences in the intensity and colour dimensions. A human observer will also be less sensitive to changes in colour than to changes in intensity. Thus, after the projection method has mapped the reduced data to a generic lower-dimensional space a further transformation which preserves the dimensionality can then be carried out so that the perceived distances in the transformed lower dimensional space correspond more closely with the actual distances between items in the original space.

Thus, we propose that Projection be regarded as consisting of two steps. The first involves a projection that reduces the dimensionality of the data. The second involves a dimensionality-preserving transformation that maps the lower-dimensional space to a space which takes into account properties of perceptual distances for the HVS.

In the final *Visualization* stage, information from the original data might be reintroduced into the visualization. The mathematical concept of a projection is a function which maps points to points but the human visual system treats images as consisting of two-dimensional, spatially-connected regions separated by lines or curves. As discussed previously, points can be expanded to symbols and so convey more information from the original space. The relationships between numbers of points can be indicated by adding features such as lines to the visualization. Relationships between three points can be indicated by introducing polygons and relationships between four points by introducing polyhedra.

3 Scientific Visualizing Applied to Performance Data Visualization

In general terms, classifier performance evaluation involves generating large amounts of performance data and trying to reduce this data to meaningful descriptors of performance. Thus, classifier performance evaluation implies discarding information and data reduction. In this sense, performance evaluation can be approached as a problem of how to project the large amounts of data to a lower dimensional space. Note that in this process, it is desirable to retain as much of the information as possible, discarding only what may be regarded as irrelevant.

In the extreme case, all the performance data get turned into a single number (projection to one dimension) and the classifiers get compared on the basis of a single quantity, i.e., a scalar metric. However, this involves the maximum amount of information loss and single value indicators of classifier performance are most likely

to be unsatisfactory in conveying information about classifier performance. This is one of the reasons why several single metrics are required to describe different aspects of performance.

In general, the volume of data we need to retain is such that listing numerical values in tables is inadequate and presenting the remaining data in visual form is desirable. Scientific Visualization may be a great aid in this process: (a) to carry out data reduction and therefore, communicate what we believe is significant about the performance results and (b) to allow a human observer to easily discover meaningful patterns in the performance results.

In order to compare classifiers on an exploratory basis rather than through standard evaluation, different tools may be useful depending of the amount of data available. They vary from simple approaches to plotting the results in a convenient way (such as histograms, spider graphs) to dimensionality reduction techniques such as Multidimensional Dimensional Scaling (MDS) [5] or Self Organizing Maps [14]. Although simple graphs are helpful for the analysis, they have limitations as the number of dimensions increases. In this case, a dimensionality reduction technique that preserves the original data structure as much as possible, seems more convenient. Next, we use a simple example to illustrate our proposal.

4 An Illustrative Example

Consider an empirical study that seeks to assess 20 binary classifiers comparing them to each other as well as to the trivial classifier and the ideal one. The two classes are labeled as *class 0* (negative class) and *class 1* (positive class). The classifier has been tested on a domain with 100 examples from *class 1* ($N_1 = 100$) and the same number from *class 0* ($N_0 = 100$). The confusion matrices for each of these hypothetical classifiers are shown in Table 1 where the confusion matrices have the format shown at the bottom of the table.

Next, the results are recorded in a table with 20 rows (one for each classifier) and 200 columns (one for each classifier output). In other words, our object of interest (each classifier) is defined in a 200 dimensional space.

4.1 Traditional Classifier Evaluation

Traditional evaluation techniques could approach this problem, for example, by computing the classifier's accuracy and then, comparing the classifiers based on it. That is, performing a projection to a 1-dimensional space. However, as we will see, this implies a high loss of information. Table 2 shows the classifier ranking according to accuracy, as well as their position with regard to the ideal classifier and the trivial classifiers T_0 and T_1 that assign all the test instances to *class 1* and *class 0*, respectively. Next, the visual analysis is presented and finally both are compared.

Table 1 Confusion Matrices for different binary classifiers. Test set with $N_1 = 100\ N_0 = 100$

Characteristics	Classifier Confusion matrices			
	a1 $\begin{pmatrix} 95 & 5 \\ 5 & 95 \end{pmatrix}$	a2 $\begin{pmatrix} 90 & 10 \\ 10 & 90 \end{pmatrix}$	a3 $\begin{pmatrix} 80 & 20 \\ 20 & 80 \end{pmatrix}$	a4 $\begin{pmatrix} 70 & 30 \\ 30 & 70 \end{pmatrix}$
	b1 $\begin{pmatrix} 95 & 5 \\ 10 & 90 \end{pmatrix}$	b2 $\begin{pmatrix} 90 & 10 \\ 20 & 80 \end{pmatrix}$	b3 $\begin{pmatrix} 85 & 15 \\ 30 & 70 \end{pmatrix}$	b4 $\begin{pmatrix} 80 & 20 \\ 40 & 60 \end{pmatrix}$
	c1 $\begin{pmatrix} 90 & 10 \\ 5 & 95 \end{pmatrix}$	c2 $\begin{pmatrix} 80 & 20 \\ 10 & 90 \end{pmatrix}$	c3 $\begin{pmatrix} 70 & 30 \\ 15 & 85 \end{pmatrix}$	c4 $\begin{pmatrix} 60 & 40 \\ 20 & 80 \end{pmatrix}$
	d1 $\begin{pmatrix} 92 & 8 \\ 2 & 98 \end{pmatrix}$	d2 $\begin{pmatrix} 80 & 20 \\ 5 & 95 \end{pmatrix}$	d3 $\begin{pmatrix} 60 & 40 \\ 10 & 90 \end{pmatrix}$	d4 $\begin{pmatrix} 40 & 60 \\ 15 & 85 \end{pmatrix}$
	e1 $\begin{pmatrix} 98 & 2 \\ 8 & 92 \end{pmatrix}$	e2 $\begin{pmatrix} 95 & 5 \\ 20 & 80 \end{pmatrix}$	e3 $\begin{pmatrix} 90 & 10 \\ 40 & 60 \end{pmatrix}$	e4 $\begin{pmatrix} 85 & 15 \\ 60 & 40 \end{pmatrix}$
Trivial classifiers	T0 $\begin{pmatrix} 0 & 100 \\ 0 & 100 \end{pmatrix}$	T1 $\begin{pmatrix} 100 & 0 \\ 100 & 0 \end{pmatrix}$		
Ideal classifier	0 $\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$		Confusion matrix format	$\begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$

Table 2 Classifier Ranking according to the accuracy metric for a toy example

Accuracy	1	.95	.925	.90	.875	.85	.80	.775	.75	.70	0.625	0.50
Classifier	Ideal	e1	c1	a2	e2	c2	a3	c3	e3	a4	e4	T0
		d1	b1		d2	b2		b3	d3	b4	d4	T1
		a1								c4		

4.2 Visualization-Based Evaluation

Within a scientific visualization based framework, we approach this problem, as explained in Section 2.2, by carrying out a projection from a h-dimensional space ($h = 200$) to a lower l-dimensional space ($l = 2$) followed by a representation of the resultant information in a 2D graph.

Projection. The projection to a two dimensional space is conducted by the classical MultiDimensional Scaling (MDS) with the Manhattan distance as distance metric. Note that this metric, defined over the individual classifier outputs which are 0 or 1 for a binary case, counts the number of mismatches between two given classifiers.

Visualization. Fig. 1 depicts the projected classifiers' performance in two dimensions. It can be seen that classifier distance to 0 (the ideal classifier) is a function of the error rate, but we also have more information.

Classifiers are placed in different regions due to the mismatch that appears among them. Note, for example, that the classifiers d4 and e4 appear with the same rank in Table 2, whereas Fig. 1 actually shows they are very different indeed. Going back to the data in the high dimensional space, we could have corroborated this fact. Nonetheless, analyzing the data in the high dimensional space ($l = 200$) becomes difficult, even when the dimension is not very high.

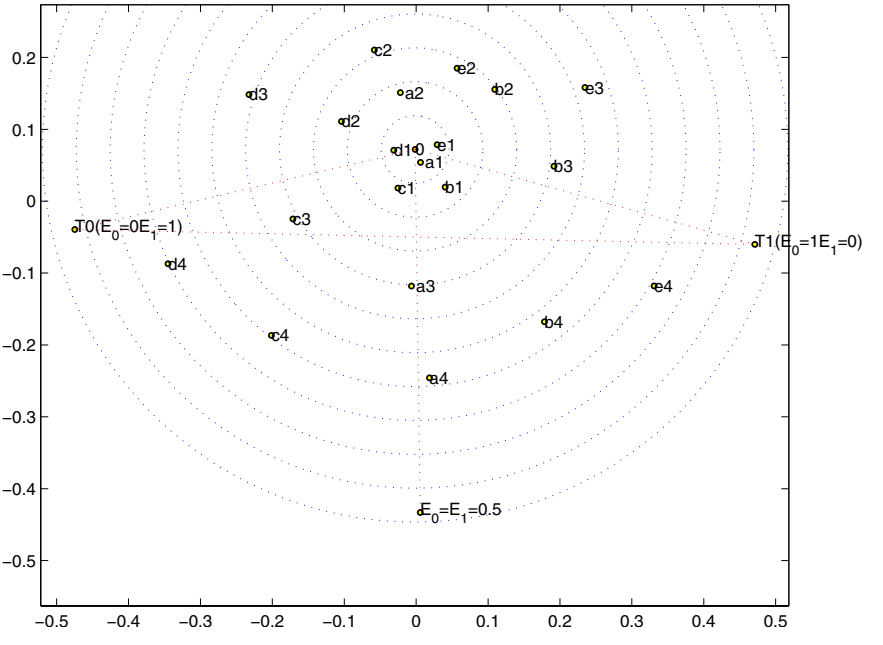


Fig. 1 Classical MDS Projection from 200 dimensions to 2 dimensions based on the classifier outcomes recorded for a toy binary domain

Also, looking at Fig. 1 we can find out whether the classifiers are closer to the trivial classifier T1 than to T0, or vice versa. That is, we can easily observe whether it makes more mistakes on class 0 or on class 1. In contrast, extracting that information from Table 2 is not possible. Even, getting that information from the conventional confusion matrices shown in Table 1 is not as straightforward as in a visual way.

Comparing and evaluating classifiers based on this graphical representation would allow us to find, besides the ranking with respect to the ideal classifier, true similarities between classifiers, classifier candidates for an ensemble, as well as establishing comparisons between each classifier.

5 The General Framework

As discussed in the previous sections, current evaluation metrics such as Accuracy, F-measure or AUC can be viewed as particular projections from a high-dimensional space to a 1-dimensional space. In the same way, it can be seen as two projections to a 1-dimensional space in the case of Precision/Recall or Sensitivity/Specificity. In this work, we generalize this idea by proposing that the techniques proposed in the field of visualization can be used for classifier performance evaluation, as well. More specifically, we propose to use the projection techniques and distance measures used in that field for our purpose. Next, we will discuss the methodology we propose, and then clarify the details of the implementation.

5.1 Methodology

Consider a standard empirical study where L classifiers are evaluated on D domains. The visualization tool we propose works following this general scenario:

1. *Performance Data Generation*
2. *Data Selection*
3. *Data Formatting*
4. *Data Projection*
5. *Data Visualization*

Next, we describe, in more detail, each of the steps.

1. *Performance Data Generation*
   All the classifiers involved in the study are run on all the domains considered and the outcome on each testing point saved. With a set of N_{ij} classifier outcomes recorded for each pair (classifier-i, domain-j), we get a total of $N_i = \sum_{j=1}^{D} N_{ij}$ records for each classifier. Note that the true class label has to be saved as well.

2. *Data Selection*
   From the performance data available, we might choose to look only at some subset of the data and focus our study on the classifiers/domains of interest (be they multiclass domains, a subset of classifiers, binary domains, imbalanced classification problems,...)

3. *Data Formatting*
   In this step, we might form vectors whose points represent the objects of interest in the analysis. By default, we describe each classifier in the original space by its outcomes on the test instances. However, we might also describe each classifier with attributes that have been computed from the original data.

   For example, we can replace the N_{ij} results of classifier-i on classification problem j by the elements of the confusion matrix that can be computed from them (4 numbers in a binary classification problem and $p \times p$ in a case where we have p classes).

If we have been using 10-fold cross validation, we will have results for ten classifiers, which we may choose to reduce to their average. We might also choose to replace a set of results by a pair of values (mean and standard deviation or minimum and maximum).

This approach is also compatible with evaluation based on typical performance metrics. Thus, we may choose to analyze the classifiers according to their ranking properties and their capabilities to estimate posterior probabilities. In this case, we could replace the set of results on a given domain by a couple of metrics: AUC and RMSE.

Note also that this approach enables the researcher to focus his or her analysis on other objects of interest, besides the classifier. In particular, we may describe a problem domain with the classifiers' performance as attributes.

At the end of this step we should have grouped all the data into vectors in a high h-dimensional space. It is worth highlighting that there is a pairwise correspondence between the vector components of all the objects that allows for a more reasonable comparison than a comparison based solely on straight averaging.

4. *Data Projection*
In this stage, data is projected into a l-dimensional space (l equal to two or more dimensions, in case that is considered helpful) that can be visualized.

We must choose the distance measure to represent the distance between two vectors in the high dimensional space as well as the projection method used to project the vectors into the l-dimensional space.

5. *Data Visualization*
In the final step, the data is plotted in l dimensions. Additional features can be added to ease comparisons between classifiers and to allow the user to discover data performance patterns.

5.2 Implementation

There are some details about the implementation of the current system that deserve further explanation. In particular, the distance metric and projection method selected to implement the method and their implications.

The distance measures can take several forms, each with different properties. The Euclidean distance (L_2 norm), for example, considers all the performance data equally, although it penalizes more for the presence of a few extreme differences than for the presence of several small differences. The Manhattan distance (L_1 norm) assigns less importance to large differences. The matching metric describes the agreements between two classifiers's hard outputs. Other distance measures can weigh different components differently. For example, true positives can be given more importance than true negatives (similarly to precision) or false positives can be given different weigh to false negatives (in order to assess cost sensitive problems). In a multi-class problem, a distance measure can focus on different errors in a different way or focus on the performance of one class, grouping all the other

classes, and so on. In fact, all the biases provided by the traditional measures (accuracy, precision, recall, F-measure) can be reproduced in our framework.

The choice of a projection method will determine important characteristics of the plot. It is worth highlighting some desirable properties in the projections for Scientific Visualizations.

Firstly, there should be the guarantee that two distinct points in the original space will not get projected to the same point. Additionally, one observer will tend to assume that points which are close to each other in the projected space are also close to each other in the original space. Thus, one desirable property would be that points which are close to each other in the projected space also be close to each other in the original space. Suppose we denote points in the original space by $\mathbf{x}$ and $\mathbf{y}$ and the distance between them as $d(\mathbf{x},\mathbf{y})$. These points are projected to $P(\mathbf{x})$, $P(\mathbf{y})$ and the distance between them in the l dimensional space given by $d_l(P(\mathbf{x}),P(\mathbf{y}))$. Then, we would like the following property to be true for all $\mathbf{x}$ and $\mathbf{y}$:

$$d(\mathbf{x},\mathbf{y}) \leq d_l(P(\mathbf{x}),P(\mathbf{y})) \tag{1}$$

Note that points which are close together in the higher dimensional space might not have projections which are close together. This is because in a higher dimensional space we can pack points closer together than in a lower dimensional space.

Another desirable property of a projection is that structure which is present in the original points should also be present in the projections of those points. There are many possible kinds of structure but one kind of structure that is interesting to our purpose is the cluster structure. If points form natural groups in the higher dimensional space, we would hope to be able to recognize those clusters in the projected space.

In this work, we considered two methods: Multidimensional Scaling (in its classical and metric version)[5], and the Minimal Cost Spanning Tree Projection (MCST), recently proposed by [12, 16].

MDS is a set of methods intended for analyzing and visualizing proximity data, i.e data characterized by dissimilarity measures for all pairs of objects under study. Basically, the main idea of MDS techniques is to map objects into points in a low dimensional space in such a way that the original dissimilarities are well approximated by distances in the plot. The different versions of MDS techniques may be categorized as follows: While the *classical* older MDS version is based on inner products, the leading *nonclassical* MDS method is defined in terms of iteratively optimizing an objective function called *stress*. The nonclassical MDS technique comes in two flavors: *metric* MDS where the stress function measures how well the interpoint distances for a given configuration of points in the low dimensional space approximate the dissimilarities in the high dimensional space. The *nonmetric* MDS relaxes the metric MDS requirements by preserving the ranks or order of the dissimilarities instead of their exact values. In this work on performance evaluation, we focus on the metric version of the MDS methods. In order to be safe enough that our MDS plots are not misleading, the stress criterion can be assessed. That is, measuring the mismatch between the distances in the original high dimensional space and the new

low dimensional space, we can get a confidence level of how much information has been lost in the projection. Moreover, a Shepard plot , which is a scatterplot of the dissimilarities against distances in the low dimensional space, provides an intuitive qualitative assessment of the goodness of the projection.

The second approach, (MCST), has the advantage of guaranteeing theoretically that the distance from each point to at least one of its nearest neighbours is preserved. Having plotted a number of graphs using MDS (classical and metric) and MCST and compared their results, we found that in most cases, the three approaches yield similar information. There were a few situations, where this was not the case for the classical MDS. We found that when projecting with classical MDS the outcome of some classifiers in a three class domain, points that were not close to each other were projected to the same point.

For the reason mentioned before, along the remainder of the chapter we will use the metric MDS and the MCST projection methods. MDS is a well known technique, which description can be found in many books (we refer the interested reader to [5, 2], for example). The detailed description of the MCST technique and how it has been adapted to this evaluation problem is included in next Section.

5.3 MCST: A Distance Preserving Projection Approach

Our approach is a slight variation on an approach by [12, 16]. It is described as follows and summarized by Algorithm 1:

Let $d(\mathbf{x},\mathbf{y})$ represent the distance between $\mathbf{x}$ and $\mathbf{y}$ in the original h high dimensional space; Let $P(\mathbf{x})$, $P(\mathbf{y})$ be the projections of $\mathbf{x}$ and $\mathbf{y}$ into the l dimensional space with distances given by $d_l(P(\mathbf{x}),P(\mathbf{y}))$ (in our examples, points are projected into a 2-dimensional space). Let us consider a general analysis where the objects of interest are the classifiers $\mathbf{c}_i$, with $i = 1,2,\ldots,n$ grouped in a matrix $\mathbf{M}$ with n rows and h columns. We also introduce the ideal classifier $\mathbf{p}_0$ that is mapped to the origin.

Firstly, we find the classifier which is closest to the ideal, we label it as $\mathbf{p}_1$ and put this on the y-axis at $(0, d_2(\mathbf{p}_0,\mathbf{p}_1))$

For the remaining classifiers, at each stage we find the classifier $\mathbf{p}_i$ which is closest to the one that has just been plotted, $\mathbf{p}_{i-1}$. When we plot $\mathbf{p}_i$ we want to preserve two constraints:

$$d_2(P(\mathbf{p}_i),P(\mathbf{p}_{i-1})) = d(\mathbf{p}_i,\mathbf{p}_{i-1}) \tag{2}$$

and

$$d_2(P(\mathbf{p}_i),P(\mathbf{p}_0)) = d(\mathbf{p}_i,\mathbf{p}_0) \tag{3}$$

In other words, we want the projections of $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ to be the same distance apart as $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ and the same for the projections of $\mathbf{p}_i$ and $\mathbf{p}_0$ and the original points $\mathbf{p}_i$ and $\mathbf{p}_0$. This means that in the projected space the distance to the origin is a measure of how close the classifier is to the ideal one. The better the classifier, the closer its projection will be to the origin.

Sometimes, there may be two possible positions for $P(\mathbf{p}_i)$ which satisfy both constraints. When this is the case, the solution will be chosen to satisfy a third constraint as closely as possible:

$$d_2(P(\mathbf{p}_i), P(\mathbf{p}_{i-2})) = d(\mathbf{p}_i, \mathbf{p}_{i-2}) \tag{4}$$

Our implementation differs to [12, 16] in the fact that we choose $\mathbf{p}_i$ to be the point which has not yet been projected which is closest to the most recently projected point, whereas the original algorithm chooses $\mathbf{p}_i$ to be the point which has not yet been projected which is closest to any of the points which have already been projected. The original approach projects the points in the same order as Prim's algorithm would add the points to a Minimal Cost Spanning Tree. Both approaches were tried, but we preferred the results produced by the modified approach because it seemed to separate clusters more.

Please, note that in our graphs we have found it useful to draw lines between pairs of projected points to show that the distance between the projected points is equal to the distance between the points in the original, higher dimensional space. Dotted lines connect projected points to the origin and indicate the exact distance in the high dimensional space from the classifier to the ideal classifier. Unbroken lines connect a point to the point that was projected immediately before it in the projection order. The distance between these projected points is also identical to the distance between the points in the original space.

When looking at the projected points, it is useful to remember that the triangle formed by $P(\mathbf{p}_i)$, $P(\mathbf{p}_{i-1})$, $P(\mathbf{p}_0)$ is congruent to the one formed by $\mathbf{p}_i$, $\mathbf{p}_{i-1}$, $\mathbf{p}_0$.

6 Performance Analysis Illustration

In this section, we illustrate the use of the visual approach presented here on several representative experimental studies. These include performance evaluation experiments from the classifier point of view (on single multiclass domains, on several binary domains) as well as an analysis from a domain point of view.

Our approach is illustrated with the MCST and the metric MDS projection methods and the Euclidean distance as metric.

The experiments are conducted in order to assess eight classifiers (1-Nearest neighbor (Ib1), Naive Bayes, C4.5 Decision Tree, Bagged Decision Trees, Boosted Decision Trees, Random Forest, SVM and JRip) based on the confusion matrices or representative metrics extracted from them. Evaluation is carried out by 10-fold cross-validation in the WEKA environment [15] with parameters set as default.

6.1 Experiments on Several Binary Domains

Consider a standard empirical study where L classifiers are evaluated on D domains. Consider also that, in this analysis, from the classifier outcomes, a set of K representative metrics are extracted and these metrics recorded for each pair of the *values of*

Algorithm 1. MCST Projection

Inputs:
Performance Data Matrix $\mathbf{M}$ (n row vectors that represent objects $\mathbf{c}_i$ and h columns the attributes)
Ideal object Performance Vector $\mathbf{c}_0$
Initialize:
$\mathbf{p}_0 = \mathbf{c}_0$
Map $\mathbf{p}_0$ to the origin of the 2-dimensional space.
for $i = 1$ to n **do**
$\mathbf{p}_i = \arg\min_{\mathbf{c}_j} d(\mathbf{p}_{i-1}, \mathbf{c}_j)$ with $\mathbf{c}_j \in \mathbf{M}$
Remove $\mathbf{p}_i$ from $\mathbf{M}$
Plot $\mathbf{p}_i$ with the constrains

$$d_2(P(\mathbf{p}_i), P(\mathbf{p}_{i-1})) = d(\mathbf{p}_i, \mathbf{p}_{i-1}) \tag{5}$$
$$d_2(P(\mathbf{p}_i), P(\mathbf{p}_0)) = d(\mathbf{p}_i, \mathbf{p}_0) \tag{6}$$
$$\tag{7}$$

and the additional constraint when necessary

$$d_2(P(\mathbf{p}_i), P(\mathbf{p}_{i-2})) = d(\mathbf{p}_i, \mathbf{p}_{i-2}) \tag{8}$$

end for
Outputs:
Projected points $P(\mathbf{p}_i)$ with $i = 1, \ldots, n$

these metrics recorded for each pair of domain-classifier. The results, then, can be organized in K tables with elements $m_{ij}^{(k)}$ where k is the metric evaluated, $i = 1, \ldots, L$ and $j = 1, \ldots, D$.

The selected eight classifiers are evaluated on fifteen binary classification problems from the UCI repository (Sonar, Heart-v, Heart-c, Breast-y, Voting, Breast-w, Credits-g, Heart-s, Sick, Hepatitis, Credits-a, Horse-colic, Heart-h, Labor and Krkp). In the following, D1 will refer to Sonar, D2 to Heart-v, and so on.

Different metrics reflect different properties that may be desirable for a classifier. From the three categories established in [4], we chose the most representative ones: RMSE that reflects the classifier's ability to estimate posterior probabilities, AUC with information about its ranking capabilities and the Error Rate metric as a threshold metric. Tables 3, 4 and 5 show the Error rate, RMSE and AUC, respectively for the 15 UCI domains evaluated here.

After the classifier evaluation analysis is performed, typical questions we would like to answer are related to similarities/dissimilarities between classifiers: (a) Which classifiers perform similarly enough so that they can be considered equivalent? (b) Which classifiers could be worth combining? (c) Does the relative performance of the classifiers change as a function of data dimensionality? (d) Does it change for different task difficulties?

Table 3 Error rate for different classifiers on several domains

	ERROR RATE														
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.1342	0.2957	0.2378	0.2757	0.0986	0.0486	0.2800	0.2481	0.0381	0.1937	0.1884	0.1873	0.2317	0.1733	0.0372
NB	0.3211	0.2360	0.1652	0.2830	0.1284	0.0400	0.2460	0.1629	0.0739	0.1554	0.2231	0.2200	0.1629	0.1000	0.1210
C4.5	0.2883	0.2663	0.2248	0.2445	0.0917	0.0544	0.2950	0.2333	0.0119	0.1620	0.1391	0.1470	0.1893	0.2633	0.0056
Bagging	0.2545	0.2513	0.2080	0.2656	0.0895	0.0415	0.2600	0.2000	0.0127	0.1683	0.1463	0.1442	0.2105	0.1533	0.0056
Boosting	0.2219	0.2965	0.1786	0.3035	0.1010	0.0429	0.3040	0.1963	0.0082	0.1420	0.1579	0.1659	0.2142	0.1000	0.0050
RF	0.1926	0.2460	0.1850	0.3144	0.0965	0.0372	0.2730	0.2185	0.0188	0.2008	0.1492	0.1524	0.2177	0.1200	0.0122
SVM	0.2404	0.2463	0.1588	0.3036	0.0827	0.0300	0.2490	0.1592	0.0615	0.1483	0.1507	0.1740	0.1726	0.1033	0.0456
JRip	0.2692	0.2660	0.1854	0.2905	0.0986	0.0457	0.2830	0.2111	0.0177	0.2200	0.1420	0.1306	0.2104	0.2300	0.0081

Table 4 RMSE for different classifiers on several domains

	RMSE														
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.3512	0.5342	0.3045	0.5042	0.2956	0.1860	0.5278	0.4848	0.1936	0.4252	0.4295	0.4261	0.2950	0.3197	0.1936
NB	0.5263	0.4164	0.2256	0.4480	0.3310	0.1945	0.4186	0.3542	0.2285	0.3409	0.4346	0.4179	0.2238	0.1997	0.3018
C4.5	0.5172	0.4531	0.2689	0.4311	0.2760	0.2105	0.4790	0.4526	0.1035	0.3565	0.3290	0.3521	0.2461	0.4209	0.0638
Bagging	0.3926	0.4177	0.2359	0.4335	0.2564	0.1769	0.4201	0.3768	0.0902	0.3388	0.3186	0.3440	0.2290	0.3412	0.0634
Boosting	0.4366	0.4700	0.2497	0.5105	0.2875	0.1864	0.5054	0.4294	0.0757	0.3507	0.3671	0.3690	0.2579	0.2281	0.0603
RF	0.3530	0.4166	0.2295	0.4686	0.2607	0.1615	0.4223	0.3912	0.1156	0.3512	0.3323	0.3376	0.2405	0.2962	0.1116
SVM	0.4837	0.4942	0.2872	0.5470	0.2667	0.1520	0.4979	0.3934	0.2479	0.3606	0.3837	0.4105	0.2885	0.2249	0.2110
JRip	0.4647	0.4360	0.2385	0.4475	0.2828	0.1932	0.44637	0.40846	0.1189	0.4075	0.3419	0.336	0.2574	0.3776	0.0782

Table 5 AUC* (1-AUC) for different classifiers on several domains

	AUC* (1-AUC)														
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
Ideal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ib1	0.1361	0.4635	0.2403	0.3687	0.0622	0.0256	0.3400	0.2500	0.1912	0.3362	0.1917	0.2035	0.2512	0.1750	0.0105
NB	0.2000	0.2826	0.0955	0.2845	0.0483	0.0120	0.2122	0.0994	0.0747	0.1408	0.1040	0.1501	0.1009	0.0125	0.0479
C4.5	0.2653	0.3983	0.2032	0.3719	0.0629	0.0515	0.3534	0.2450	0.0505	0.3034	0.1064	0.1507	0.2341	0.2666	0.0012
Bagging	0.1478	0.2869	0.1296	0.3518	0.0362	0.0105	0.2469	0.1291	0.0050	0.1769	0.0771	0.1237	0.1178	0.1583	0.0007
Boosting	0.0938	0.3055	0.1187	0.3569	0.0370	0.0176	0.2770	0.1166	0.0123	0.2003	0.0945	0.1118	0.1389	0.0625	0.0007
RF	0.0889	0.2914	0.1215	0.3537	0.0376	0.0137	0.2499	0.1386	0.0072	0.1599	0.0886	0.1023	0.1444	0.0916	0.0012
SVM	0.2418	0.4335	0.1639	0.4072	0.0869	0.0316	0.3292	0.1633	0.5001	0.2487	0.1434	0.1912	0.2033	0.1250	0.0457
JRip	0.2631	0.4366	0.1591	0.3877	0.0839	0.0368	0.3871	0.2041	0.0579	0.3960	0.1285	0.1562	0.2427	0.2416	0.0055

A first attempt at answering these questions could be to analyze directly the data gathered in the three tables. However, it does not seem straightforward given the quantity of results recorded.

As an alternative, metrics like SAR try to summarize all the gathered information with a point estimation. Thus, SAR carries out the projection $SAR^* = (1 - SAR) = RMSE + Error + AUC^*$ where $AUC^* = (1 - AUC)$. The closer to zero the SAR values (and all its components) are, the better the classifier performs. Table 6 shows the classifiers' performance values and ranking according to the SAR metric. We consider, however, that combining metrics uniformly may be dangerous. Instead we argue that we should select the information that is relevant to our purpose and concentrate on it to conduct the performance analysis.

Table 6 Classifier Ranking according to SAR

Ideal	RF	Bagging	Boosting	NB	JRip	C4.5	SVM	Ib1
0	.1958	.1965	.2037	.2126	.2362	.2365	.2420	.2530

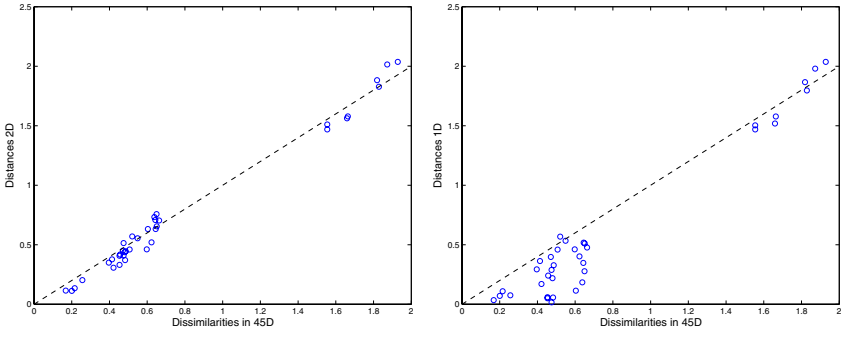


Fig. 2 Shepard plot for the metric MDS projection: (a) from 45 dimensions to 2 dimensions. (b) from 45 dimensions to 1 dimension

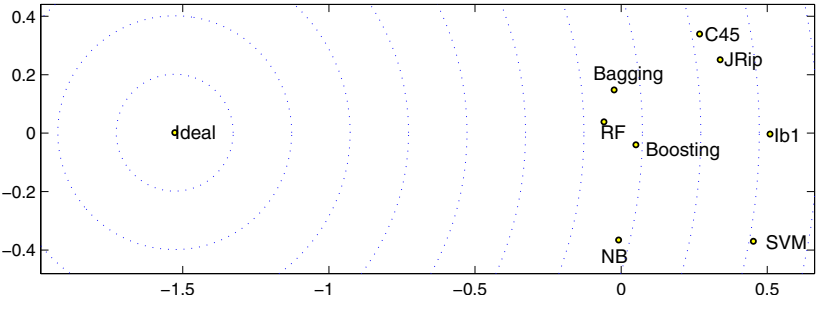


Fig. 3 Metric MDS projection from 45 dimensions to 2 dimensions based on the RMSE, AUC* and Error rate gathered over 15 domains

Visual data mining allows to easily discover data patterns, a task that may be difficult by simply looking at the results organized in tables and inaccurate when summarized by a SAR-like measure. In this section, firstly we demonstrate the use of MDS to visualize the classifiers in a graph and secondly, the representation provided by the MCST projection. Both methods conduct the projection so that interpoint distances in the high dimensional (metric/domain) space are preserved as much as possible in the 2D space.

Let us now study what information may be extracted from a graphic where the information provided in Tables 3, 4 and 5 is not simply averaged (over domains and over different metrics) but is projected using MDS. The distance between two points is calculated as the Euclidean distance and the stress criterion (see below) is normalized by the sum of squares of the interpoint distances.

Before starting to explore the graphical representation, it is interesting to assess the stress criterion. It is important to know how much of the original data structure is preserved after projecting the data to two dimensions. We can also get an idea of the information gained when moving from a one dimensional representation to a two dimensional one. In our example the stress becomes 0.08 for two dimensions (not much loss of information), but it increases to 0.31 when considering only one dimension. This is supported by the Shepard plot in Fig. 2 that shows the reproduced distances in the new projected space (y axis) versus the dissimilarities in the original space (x axis). It can be seen that a projection to 2D leads to a narrow scatter around the ideal fit, while the scatter with a projection to 1D becomes larger and indicates a higher loss of information.

Now we focus on the whole information (Error rate, RMSE and AUC*) reflected in Fig. 3. In this particular case, we analyze eight classifiers described in the original high dimensional space by 45 dimensions (3 metrics $\times$ 15 domains) and then, projected to a 2-dimensional space. Dotted lines represent points that are at the same distance to the ideal classifier (iso-performance lines). The ideal classifier is also introduced, to allow us to compare classifiers by their projected distance to the ideal classifier as well as to their relative position with respect to the other classifiers. Note that this second type of information is lost when a one-dimensional projection is used. Indeed, scalar performance measures, can only aim to convey one kind of information, usually the distance to ideal.[1]

Fig. 4 shows the representation obtained with MCST. Dashed lines represent distance to the ideal classifier, while continuous ones represent the link of a given classifier with the closest one (it is selected from the set of classifiers that have not yet been plotted). The figure's legend shows additional information such as, the classifier's name, its distance to the ideal classifier and to the previous classifier.

From the point metric SAR (see Table 6), we can easily draw the conclusion that the C4.5 and SVM performances are very similar. The same applies to C4.5 and JRip as well as to Boosting and NB.

However, in the projection that SAR represents (from 45 dimensions to 1), we lose a lot of information about the similarities between classifiers. Keeping more information (projecting to 2 dimensions) allow us to identify several clusters of classifiers whose performance are very close or equivalent across the fifteen domains in terms of the three metrics considered (RMSE, AUC and Error rate). With the aid of Fig. 3 (or equivalently, Fig.4) the two following classifier clusters with equivalent performance can be identified: {C4.5, JRip} (or {(9), (8)} in the MCST figure) and {Bagging, RF, Boosting} (or {(2), (3), (4)} in the MCST figure). Note that in this case, the practitioner can extract similar information from both, the MCST and the MDS plot.

If we now go back to Tables 3, 4 and 5, we would be able to confirm the similarities among the classifiers within the cluster. Nonetheless, finding the similarities directly from the information gathered in these tables does not seem straightforward.

[1] This is not the only type of information that gets lost, by the way, since, once in two dimensions, a lot more flexibility is possible, especially if we consider colours, motion pictures, and potentially more.

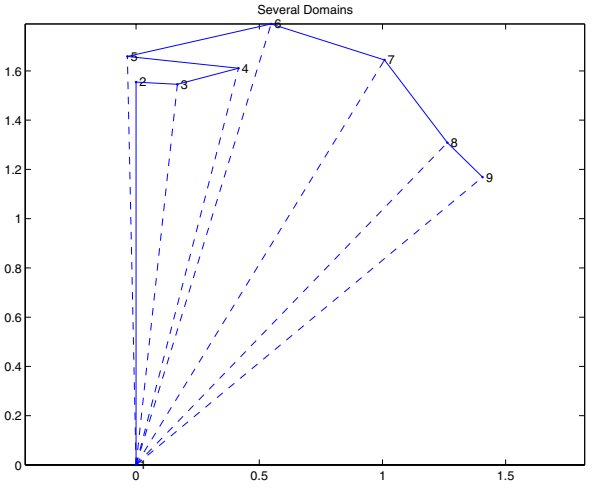


Fig. 4 MCST Projection from 45 dimensions to 2D based on three metrics gathered over 15 domains. Information: Classifier name, classifier rank, distance to the origin (ideal) and distance from the previous classifier. Ideal 1
RF 2 1.55
Bagging 3 1.55 0.17
Boosting 4 1.66 0.26
NB 5 1.66 0.45
SVM 6 1.87 0.60
Ib1 7 1.93 0.48
JRip 8 1.82 0.42
C45 9 1.83 0.20. MCST - Stress is 5.910467e+000

Recalling the similarities found by analyzing the SAR metric, we are able to conclude that: (i) C4.5 and JRip's performance are very close (this corroborates SAR-based analysis) , (ii) C4.5 and SVM's performance are divergent (although their difference to the ideal classifier seems to be approximately equal) and (iii) Boosting and NB's behaviours are not as close as the information in Table 6 suggests. While this clarifies the results, it also suggests a whole series of new questions: In which way are these classifiers different? Where do these differences among classifiers come from? Do they arise, for example, because of different capabilities to estimate posterior probabilities? Can we impute them to the domain characteristics? Next, we could further explore the evaluation data to get insight into the classifier dissimilarities by looking at the metrics in an individual way (we refer the interested reader to [1] for a more detailed exploration).

6.2 Experiments on Single Multiclass Domains

Consider now an experimental study where L classifiers are evaluated on a single multiclass domain with p classes. Consider also that for each classifier i, a set of m_{ij} values with $j = 1, \ldots, h$) are extracted from the classifier's outcomes on the test instances and organized in a table with L rows and h columns.

In this section, we evaluate the eight classifiers mentioned previously on the Page-blocks problem from the UCI repository and the performance data are described by the confusion matrix elements ($h = p \times p$). This 5-class data set is quite imbalanced since the classes contain 4913, 329, 28, 88 and 115 test cases, respectively. Evaluation data is presented in a table with 8 rows (one per classifier) and 25 columns. The ideal classifier is also included for comparison purposes.

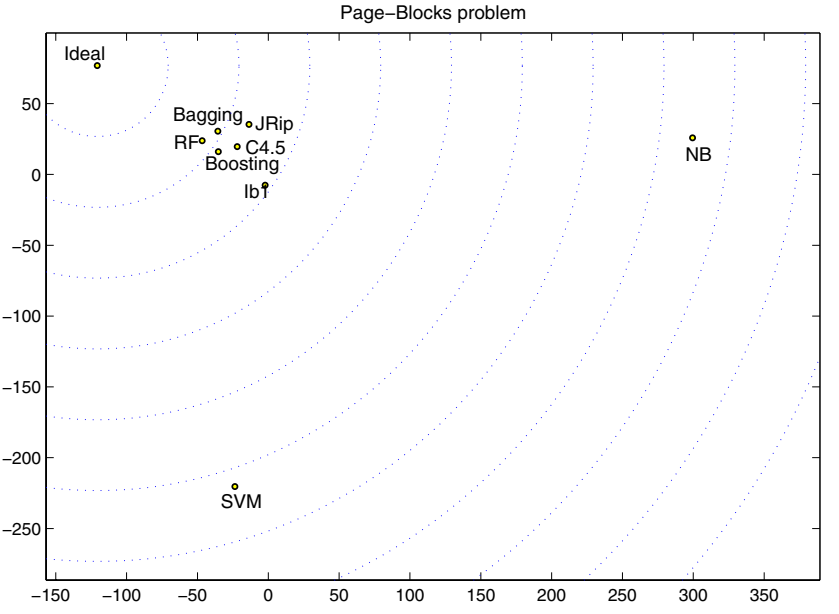


Fig. 5 Metric MDS Projection of the results for the Page-blocks multiclass domain from 25 to 2 dimensions

Table 7 Page-blocks multiclass domain: classifier ranking according to Accuracy

Ideal	RF	Bagging	Boosting	C4.5	JRip	Ib1	SVM	NB
1.000	.974	.0.973	.970	.969	.968	.959	.929	.909

Projection results on this domain (from 25 dimensions to 2) are displayed in Fig. 5 and Fig. 6 according to the MDS and MCST projection methods, respectively. Both visual tools highlight how poor the performance of SVM and NB is. Moreover, it is also shown that these two classifiers behave quite differently.

Let us compare the results we get form the visual approach with the accuracy results obtained on this domain shown in Table 7. While the analysis based on accuracy (a specific projection to one dimension) suggests that NB and SVM do not classify the data as well as the other classifiers, it does not inform us of the fact that these two classifiers approach the problem differently. Indeed, while it is true that NB's accuracy of 90.9% is different from SVM's accuracy of 92.9%, this 2.0% difference is too small to be deemed significant. This is quite different from what we get in Figs. 5 and 6. In fact, these two classifiers are approximately at the same distance from one another as they are from the ideal classifier.

In order to interpret the results, it is important to remember that the Page-blocks problem is very imbalanced. The effects of the imbalance are clearly seen in the confusion matrices of NB and SVM in Table 8. We see that SVM fails at

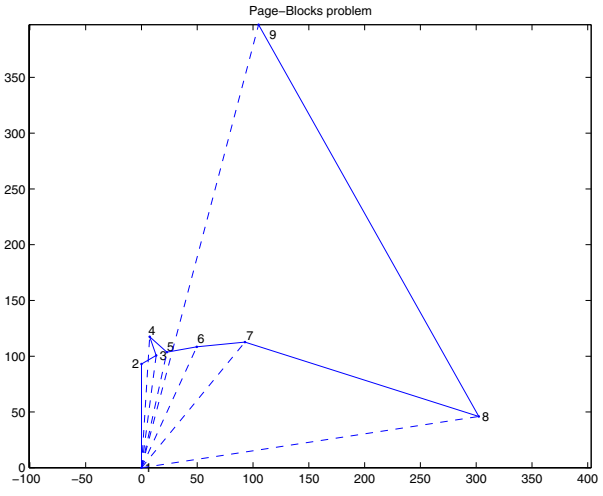


Fig. 6 MCST Projection of the results for the Page-blocks multiclass domain from 25 to 2 dimensions. Information: Classifier name, classifier number, distance to the origin (ideal) and distance from the previous classifier.
Ideal 1
RF 2 93.01
Bagging 3 101.60 15.30
C4.5 4 117.65 17.66
Boosting 5 106.11 20.69
JRip 6 119.21 27.20
Ib1 7 145.99 43.41
SVM 8 305.86 219.96
NB 9 410.86 402.94

Table 8 Confusion Matrices for NB and SVM for the Page-blocks multiclass problem

Characteristics	NB	SVM
Confusion matrix	$\begin{pmatrix} 4607 & 29 & 12 & 207 & 58 \\ 84 & 217 & 1 & 24 & 3 \\ 8 & 0 & 18 & 0 & 2 \\ 1 & 1 & 0 & 84 & 2 \\ 52 & 1 & 9 & 7 & 46 \end{pmatrix}$	$\begin{pmatrix} 4902 & 11 & 0 & 0 & 0 \\ 166 & 162 & 1 & 0 & 0 \\ 16 & 0 & 12 & 0 & 0 \\ 84 & 1 & 0 & 2 & 1 \\ 107 & 0 & 0 & 0 & 8 \end{pmatrix}$
Accuracy	0.909	0.929

classifying instances from class-4 and class-5, and tends to assign examples to the majority class. NB, however, performs better on these two classes, but badly on class-1. The other classifiers, however, are able to deal with the imbalance problem (confusion matrices have been omitted due to space constraints). To sum up, the visual tool gives important information regarding evaluation, while a point-metric like accuracy, does not warn us about the severity of the mistakes not does it differentiate between them.

6.3 Experiments on Domain Difficulties

Apart from a general study about classifiers, the visual approach provides a simple way to study a number of other questions that cannot be answered with traditional evaluation procedures. These include questions for which performance data is analyzed according to the domain characteristics.

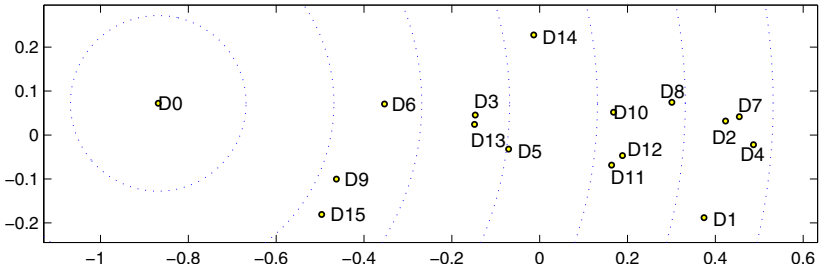


Fig. 7 Metric MDS projection from 8 dimensions to 2 dimensions based on RMSE metric gathered for eight classifiers

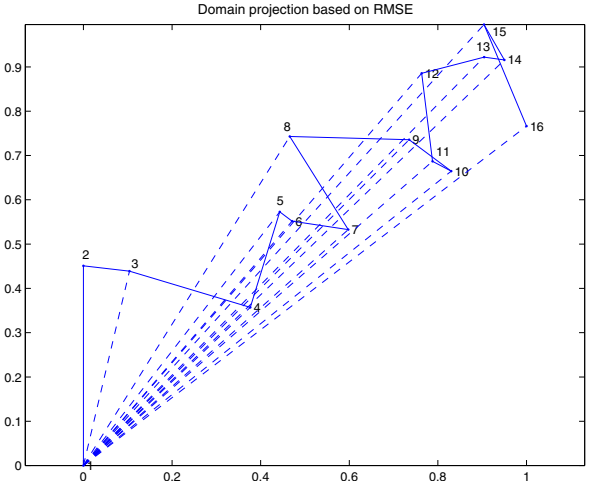


Fig. 8 MCST projection from 8 dimensions to 2 dimensions based on the RMSE metric.
D0 1
D9 2 0.45
D15 3 0.45 0.10
D6 4 0.52 0.28
D13 5 0.72 0.23
D3 6 0.73 0.03
D5 7 0.80 0.13
D14 8 0.88 0.25
D10 9 1.04 0.27
D12 10 1.06 0.12
D11 11 1.05 0.05
D8 12 1.17 0.20
D2 13 1.29 0.15
D7 14 1.32 0.05
D4 15 1.34 0.09
D1 16 1.26 0.25

In this case, we can regard each domain as an object with attributes representing a measure of how several classifiers have performed on that domain. Note that the objects of interest are the domains, the attributes are classifier performance measures and the classifier dimensions are the ones reduced when projecting.

Based on this analysis, we are able to address questions about the domains, such as: Can domains be organized into equivalence classes within which various classes of classifiers behave predictably?. What domain characteristics influence the behaviour of different domains (e.g., domain difficulty, dimensionality, etc.)?

Consider a study where D domains are analyzed based on how L classifiers have performed on them. In this analysis, each object-j, is a domain defined by a vector with components $m_{ij}^{(k_l)}$ with metric $k = k_l$ and classifiers $i = 1, \ldots, L$.

For example, let us assume that we concentrate on the posterior probability capabilities measured by the RMSE metric and the fifteen binary domains used

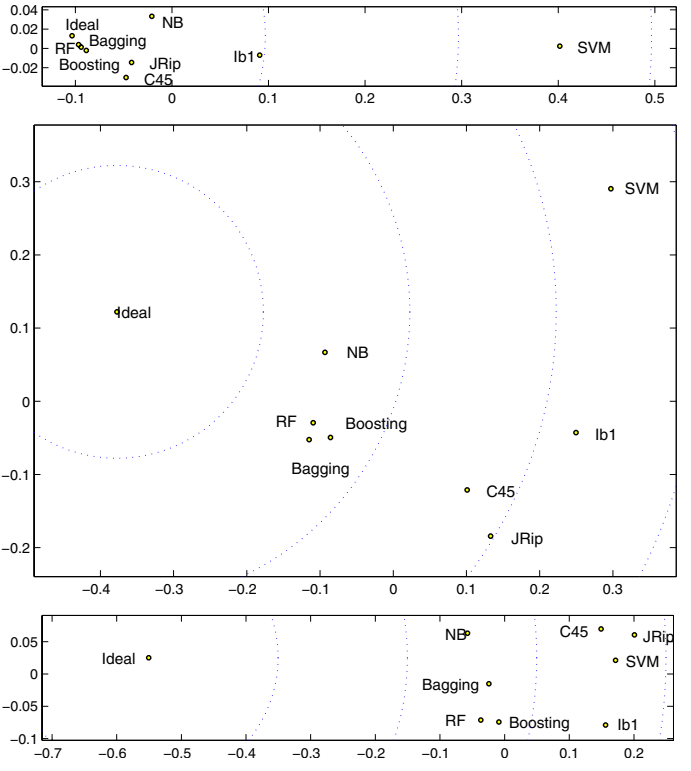


Fig. 9 MDS projection based on the AUC metric (from 15 to 2 dimensions): (a) Simple domains (9 6 15). (b) Domains (3 13 5 6 9 15 10 11 12 8). (c) Difficult Domains (7 2 4 1)

previously. The ideal domain D0, for which the estimation is perfect, is also included as a reference. Our original space has 16 objects (15 domains plus an ideal one, D0, for which all classifiers get the minimum RMSE) and has 8 dimensions (as many as classifiers; L=8).

Fig. 7 and Fig. 8 show (with the aid of MDS and MCST, respectively) the similarities/dissimilarities among domains in terms of the difficulty for the classifiers to estimate posterior probabilities. It is now feasible to identify groups of domains (e.g., {D3, D13, D5} or {D2, D7, D4}) for which the task of estimating posterior probabilities has similar complexity. Moreover, there are domains for which ranking becomes easier and others for which it is a more difficult task. We argue that the classifier performance may differ according to the task difficulty. Both graphs give us the same information, but the graph resulting from the MDS projection is easier to read than that resulting from the MCST projection.

Fig.9(a), Fig.9(b) and Fig.9(c) show the classifier relation based on RMSE for low, medium and high difficulty domains, respectively. These graphs may allow us to extract information with regard to domain difficulty and draw conclusions

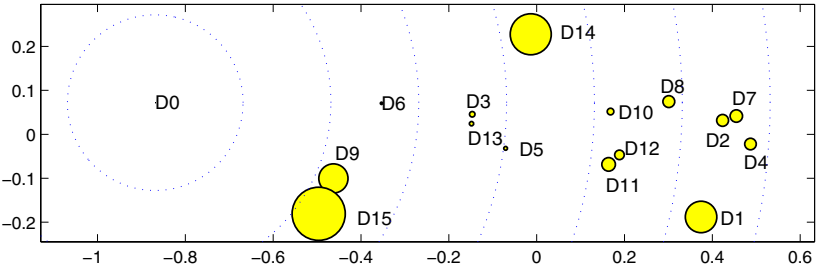


Fig. 10 Metric MDS projection from 8 dimensions to 2 dimensions based on the RMSE metric gathered for eight classifiers

such as the following: "As domain difficulty increases, classifier A becomes less competitive than classifier B..."

For example, from Fig.9 we can reach the following conclusions in terms of the classifiers' capabilities to estimate posterior probabilities:

1. C4.5 and JRip performance compared to the other classifiers' tend to decrease as the task difficulty increases.
2. NB is less competitive when difficulty is low.
3. Bagging, Boosting and RF relative performance hold across different domain difficulties.

Finally, note that the plots may also show additional information about the evaluation process. For example, Fig.10 displays the size point that represents each domain, according to the variance of classifier performance on that domain.

We can identify some domains like D15, D9 (low difficulty), D14 (medium) and D1(high difficulty), for which classifier performance shows great variance. Therefore, these are domains that can take more advantage from classifier combination than domains with very low variance of classifier performance (D6, D3 , D13,...). Additionally, the correlation between task difficulty and variance in the classifiers' response could be easily observed with this visual approach. Note that in this empirical study no correlation is noticed when looking at Fig.10.

7 Concluding Remarks

In this work, we take the view that classifier comparison can be stated as a problem of analyzing high dimensional data and it should be done on an exploratory basis rather than through standard evaluation. This means that as long as the performance analysis progresses, we will discover tendencies, similarities, dissimilarities or outliers, but there is no need to know what we want to find in advance.

We provide a visualization-based technique that takes this very general view and transforms it into a practical endeavour. This approach, rather than aggregating the performance results into a single metric, represents each object of interest (be it a

classifier or a domain) as a point in a high dimensional space. Next, a projection to a low dimensional space (2D,3D) makes feasible to easily discover data patterns and draw meaningful conclusions about the performance results, a task that is quite difficult when simply looking at the results organized in tables, and inaccurate when summarized by point metrics. There are many avenues to explore in tasks like model selection and combination which may be conducted the aid of an evaluation tool we are designing and implementing.

References

1. Alaiz-Rodriguez, R., Japkowicz, N., Tischer, P.: Visualizing classifier performance on different domains. In: Proceedings of the 20th IEEE International Conference on Tools for Artificial Intelligence, ICTAI 2008 (2008)
2. Borg, I., Groenen, P.: Modern Multidimensional Scaling: Theory and Applications. Springer, Heidelberg (2005)
3. Buja, A., Cook, D., Swayne, D.F.: Interactive high-dimensional data visualization. Journal of Computational and Graphical Statistics 5, 78–99 (1996)
4. Caruana, R., Niculescu-Mizil, A.: Data mining in metric space: An empirical analysis of suppervised learning performance criteria. In: Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining, KDD 2004 (2004)
5. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall, Boca Raton (1994)
6. Ferreira de Oliveira, M.C., Levkowitz, H.: From visual data exploration to visual data mining: A survey. IEEE Transactions on Visualization and Computer Graphics 9(3), 378–394 (2003)
7. Drummond, C., Holte, R.C.: Cost curves: An improved method for visualizing classifier performance. Machine Learning 65(1), 95–130 (2006)
8. Fodor, I.K.: A survey of dimension reduction techniques. Technical report (2002)
9. Hand, D.J.: Classifier technology and the illusion of progress. Statistical Sciences 21(1), 1–15 (2006)
10. Holmström, L.: Nonlinear dimensionality reduction by john a. lee, michel verleysen. International Statistical Review 76(2), 308–309 (2008)
11. Keim, D.A.: Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics 8(1), 1–8 (2002)
12. Lee, R.C.T., Slagle, J.R., Blum, H.: A triangulation method for the sequential mapping of points from n-space to two-space. IEEE Trans. Comput. 26(3), 288–292 (1977)
13. Provost, F., Fawcett, T.: Robust classification systems for imprecise environments. Machine Learning 42(3), 203–231 (2001)
14. Soukup, T., Davidson, I.: Visual Data Mining: Techniques and Tools for Data Visualization and Mining. Wiley, Chichester (2002)
15. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (1999)
16. Yang, L.: Distance-preserving projection of high-dimensional data for nonlinear dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1243–1246 (2004)

Extending Rule-Based Classifiers to Improve Recognition of Imbalanced Classes

Jerzy Stefanowski and Szymon Wilk

1 Introduction

Knowledge discovery in general, and data mining in particular, have received a growing interest both from research and industry in recent years. Its main aim is to look for previously unknown relationships or patterns representing knowledge hidden in real-life data sets [16]. The typical representations of knowledge discovered from data are: associations, trees or rules, relational logic clauses, functions, clusters or taxonomies, or characteristic descriptions of concepts [16, 29, 21]. In this paper we focus on the rule-based representation. More precisely, we are interested in *decision* or *classification rules* that are considered in *classification* problems. In data mining other types of rules are also considered, e.g., association rules or action rules [16, 29, 34], however, in the text hereafter we will use the general term "rules" to refer specifically to decision rules.

Rules represent functions mapping examples (objects), described by a set of *attributes* (features) to *decision classes* (concepts) and they are expressed in the form: **if** P **then** Q, where P is the *condition part* formed as a conjunction of elementary conditions – tests on values of attributes, and Q is the *decision part* of the rule, which indicates the assignment of an example satisfying the condition part to a specific decision class.

The rule-based representation due to its symbolic expressiveness is considered to be more comprehensible and human-readable than other representations (see discussions in [28, 29, 33]). Although, such characteristic is shared by the tree-based representation, a set of rules may be more compact than a decision tree [33]. Moreover, rules constitute "blocks" of knowledge, and experts can more easily analyze individual rules [28, 39]. Finally, rules were successfully used in several applications as demonstrated in a review paper by Simon and Langley [23].

Induction of rules has been intensively studied in machine learning [29, 31] and many algorithms have been proposed, for reviews see [10, 31, 29]. The majority of them try to generate rules following a *sequential covering* strategy. They are focused

Jerzy Stefanowski and Szymon Wilk
Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 2, 60–965 Poznań, Poland
e-mail: jerzy.stefanowski@cs.put.poznan.pl,
szymon.wilk@cs.put.poznan.pl

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 131–154.
springerlink.com

on creating a *minimal set of rules*, which means that learning examples are covered by the smallest number of non-redundant rules.

Sets of rules induced from learning examples are usually applied to *predict class labels* for new examples. In such a classification-oriented perspective, rules and a strategy of using them constitute a *classifier*.

Although sequential covering algorithms have been shown to be quite effective from this perspective, there are also other algorithms which provide "richer" sets of rules. Such rules are also often characterized by better descriptive properties, e.g., they are supported by a larger number of learning examples (see [42, 16]). In general they could better characterize some regularities hidden in data, what corresponds to so-called descriptive perspective of knowledge discovery [50].

On the other hand, such rules could be also useful for handling more difficult classification problems. One of the main reasons for difficulty is *class imbalance* in learning data, i.e., a situation when one class (further called the *minority class*) includes much smaller number of examples comparing to other *majority* classes. The minority class is usually of primary interest in a given problem and it is required to recognize its members as accurately as possible. The imbalanced distribution of classes constitutes a difficulty for standard learning algorithms because they are biased toward the majority classes. As a result examples from the majority classes are more likely to be classified correctly by created classifiers, whereas examples from the minority class tend to be misclassified.

The problem of dealing with the class imbalance receives a growing research interest in machine learning and data mining communities (for a review see [3]). Several methods have been proposed to improve performance of various types of classifiers, not only rule-based ones. In general, one can distinguish two kinds of approaches [20]. The first approach, which is classifier-independent, relies on transforming an original data set to change the balance between classes, e.g., by re-sampling . The second approach involves modifying classifiers in order to improve their sensitivity to the minority class.

In the paper we discuss how rule-based classifiers can be adopted to deal with imbalance in the learning set. We consider two ways of doing it – either by modifying a classification strategy, or by using a different approach to induce rules for the minority class. The main aim of this study is to present a new method of extending the structure of a rule-based classifier in order to improve its sensitivity to the minority class. The main principle of the proposed method is that a minimal set of rules for the minority class is replaced by a new set of stronger rules. Such rules are discovered by a special algorithm, called EXPLORE, which was previously introduced by Stefanowski and Vaderpooten in [42]. Thus, using such rules for the minority class, while preserving the original minimal set of rules for the majority classes, improves the chance that an example from the minority class is correctly recognized.

Within our approach we do not only preserve the comprehensible representation of decision knowledge for the minority class, but we even make it more comprehensive by discovering additional rules, still hidden in data, which have not been revealed in a minimal set. Discussing the usefulness of the EXPLORE algorithm

for providing such comprehensible patterns is the second goal of this paper. Maintaining comprehensible representation of knowledge is consistent with arguments behind rule paradigms. It further distinguishes our proposal from other known approaches, which also aim at improving the minority class prediction, however, at the cost of making extensive changes in data.

Finally, we present results of experiments where the performance of our approach is compared against LEM2 – a typical sequential covering algorithm, and its modification for handling imbalanced data on several benchmark data sets.

The paper is organized as follows. We begin with a brief review of two main categories of rule induction algorithms and classification strategies. Then, in Section 3 we describe the EXPLORE algorithm, which is employed by our approach. Our previous experience with using this algorithm is also reported. The next section contains a short review of methods for handling imbalanced data. Then, we present our approach to changing the structure of a rule-based classifier. In Section 6 we experimentally evaluate its usefulness in a comparative study. Final remarks and discussion on future research are provided in the last section.

We would like to note that this paper is a summary, which includes partial results from other papers by Stefanowski and coauthors on the EXPLORE algorithms [42] and from our joint research with Grzymala-Busse on handling imbalanced data by modifying rule-based classifiers [15].

2 Approaches to Rule Induction

This chapter gives basic information on rule induction and classification strategies which are necessary for presenting our method. More comprehensive descriptions can be found in [10, 12, 21, 39]

2.1 Basic Notation

For classification problems data sets include examples described by attributes and assigned to decision classes. We assume that these examples are represented in a *decision table* $DT = (U, A \cup \{d\})$, where U is a set of examples, A is a set of condition attributes describing them, and $d \notin A$ is a decision categorical attribute that partitions examples into a set of disjoint decision classes $\{K_j : j = 1,...,k\}$. A decision rule r assigning examples to a class K_j is represented in the following form:

$$\textbf{if } P \textbf{ then } Q,$$

where $P = p_1 \wedge p_2 \wedge \ldots \wedge p_n$ is the *condition part* of r, and Q is the *decision part* of r indicating that an example should be assigned to a class K_j. The condition part is a conjunction of elementary conditions p_i. Each condition represents a test on a value of a corresponding attribute. For a symbolic attribute the test compares its value to a constant, and for a numerical attribute other relations (e.g., greater than) are possible.

A decision table DT contains *learning* examples for inducing rules, therefore, it is called a *learning* set. For a given class K_j, learning examples from this class are called its *positive* examples, while examples belonging to the remaining classes are called *negative examples* of K_j .

Using these terms we briefly present some definitions of basic rule properties. $[P]$ is a *cover* of the condition part of a rule r in DT, i.e., it is a set of examples, which descriptions (values of condition attributes) satisfy elementary conditions in P. Let $[K_j]$ be a set of positive examples of a class K_j. A rule r is *discriminant* (also called *certain* or *consistent*) if it distinguishes positive examples of K_j from its negative examples, i.e., $[P] \subseteq [K_j]$. Moreover, P should be a minimal conjunction of elementary conditions satisfying this requirement.

A set of decision rules R *completely* covers (describes) all positive examples of a class K_j, if each positive example is covered by at least one decision rule from R. Moreover, if there is no other $R' \subset R$ that covers all positive examples of K_j, we say that R is the minimal cover of K_j. In other words it completely describes positive examples of this class by the smallest number of rules.

If the learning set contains noisy or inconsistent examples, also so-called *partially discriminant* or *possible* rules can be constructed. Besides positive examples such rules cover a limited number of negative examples.

2.2 Perspectives of Rule Induction and Evaluation

In general induction of decision rules can be performed according to different perspectives. The most common ones are [39, 50]:

- classification-oriented induction,
- descriptive-oriented induction.

The aim of the *classification-oriented induction* is to create from learning examples a set of rules which will be further used to *classify* new objects. Rules are then combined with a strategy defining how to use them to produce the final prediction for a new object – such a combination constitutes a *classifier*. This perspective has been extensively studied in machine learning and several approaches for deriving *rule-based classifiers* have been proposed.

The aim of the *descriptive-oriented induction* is to *extract* from learning examples information patterns (regularities or sometimes exceptions or anomalies) which may be *interesting* and *useful* for different users [16]. These patterns (represented as *rules*) aim at clarifying dependencies between values of attributes and decision classes [42] and usually are much more comprehensive than rules created following the classification-oriented perspective. The descriptive-oriented induction has been conceived and considered within the field of knowledge discovery, however, there has been successful research on building classifiers using rules constructed according to this approach [16, 39].

The two perspectives of rule induction do not only have different goals – there are other profound differences between them. One of the main distinctions consists in different evaluation criteria [42] of constructed rules. In the classification-oriented induction, a *complete set of rules is evaluated* as a classifier. An evaluation criterion is usually single and defined as the classification (predictive) accuracy or similar prediction measure (e.g., based on a confusion matrix – see Section 4) of a rule-based classifier using these rules. This criterion is evaluated in an experimental and automatic way.

In the descriptive-oriented induction, *each rule is evaluated individually and independently* as possible representation of an interesting pattern, which is definitely a more difficult task. Depending on a rule induction algorithm, the user may obtain quite a large number of rules to interpret. Selecting some of them is a non-trivial issue, it is also partly subjective as it generally depends on the problem at hand and on interests and expertise of users. To support the selection, several *quantitative measures* (also called *interestingness* measures) have been proposed and studied, each capturing different characteristic of rules. Many of these measures characterize relationships between the condition and the decision parts of a rule and a data set, from which the rule has been discovered. Generality, support, confidence, logical sufficiency or necessity are examples of widely approved and used measures. Their systematic review is available, e.g., in [17]. Below we present two of the most commonly used measures, i.e. *support* and *confidence* of a rule.

The support of the condition part P, denoted as $sup(P)$, is equal to the number of examples in U satisfying P, i.e., its equal to $|\,[P]\,|$, where where $|\,.\,|$ denotes the cardinality of a set. In a similar way we define the support of the decision part Q and denote it as $sup(Q) = |\,[Q]\,|$.

The support of a rule r denoted as $sup(r)$, is equal to the number of objects in U satisfying the condition and the decision parts (P and Q respectively), i.e., $sup(r) = |\,[P \cap Q]\,|$. The support could be given in relation to the number of examples in U as

$$sup(r) = \frac{|\,[P \cap Q]\,|}{|\,[U]\,|}.$$

The confidence of a rule r shows the degree to which P implies Q and it is defined as

$$conf(r) = \frac{|\,[P \cap Q]\,|}{|\,[P]\,|}.$$

This measure is also known as *certainty factor*, *accuracy* or *discrimination level*.

Let us notice that both these measures characterize two different properties of a rule – support corresponds to the generality of a pattern represented by the rule in data, while confidence estimates the certainty of assignment to a decision class indicated by the rule.

Another measure of rule generality is called *coverage* or rule strength, and it is used in the description of the EXPLORE algorithm. The coverage of a rule r is defined as

$$cov(r) = \frac{|\,[P \cap Q]\,|}{|\,[Q]\,|}.$$

The other major distinction between the classification and descriptive perspectives corresponds to different rule induction algorithms. The former perspective employs algorithms inducing minimal sets of rules, while the latter requires different methods (producing non-minimal sets of rules). These two groups of algorithms are briefly discussed in the following subsections.

2.3 Induction of Minimal Sets of Rules

The majority of rule induction algorithms employed by the classification-oriented perspective follow the sequential covering strategy, which historically comes from the early Michalski's works on the family of the AQ algorithms. It is also known as the *separate-and-conquer* strategy and used in several inductive logic programs [10].

Figure 1 shows the basic idea of the sequential covering strategy. It sequentially generates a *minimal set* of decision rules for each decision class.[1] In each run it accepts as input a set of positive and negative examples of a class K_j and provides as output a set of rules R covering all positive examples of this class and not covering any of its negative examples (if the learning set does not contain any inconsistent examples). The strategy iteratively creates the best possible rule based on the "best" conjunction of elementary conditions according to selected criteria (see the function *find_single_best_rule*). Then, it stores the rule and excludes from consideration all positive examples that match this rule. This process is repeated if at least one positive example of the decision concept remains uncovered.

The function *find_single_best_rule* produces a candidate for a rule, which in general should cover as many positive examples of the target class as possible and no negative ones (for consistent data), or a limited number of negative examples (for inconsistent or noisy data). This function can be formulated in different ways depending on a particular version of the algorithm. In majority of them the condition part of a candidate rule is constructed by successively adding new elementary conditions to the conjunction (the process starts with an empty condition part). This process is repeated until a selected acceptance criterion has been fulfilled, e.g., the current condition part does not cover any of the negative examples (e.g., see the description of AQ [29] or LEM2 [12]).

The search for the best elementary condition to be added to the conjunction is driven by specific evaluation criteria. The number of proposals is quite large, for a review see [10]. For instance, in the LEM2 algorithm Grzymala-Busse proposed to select conditions in the following way [12]:

[1] The are also some versions of this strategy, which do not sequentially go through classes but attempt to consider all classes together, however, still maintaining the principle of recursively learning the best rule, removing covered examples, etc.

```
procedure sequential_covering(input K_j: class;
E : its positive learning examples; N : and its negative examples
output R: set of rules)
begin
    R ← ∅    { initialize rules };
    while E ≠ ∅ do
    begin
        r ← find_single_best_rule(K_j, E, N)
        [r]_E ← set of positive examples covered by r
        if rule_stopping_conditions(r) then exit;
        E ← E \ [r]_E;
        R ← R ∪ r
    end
    R ← post-process(R)
end
```

Fig. 1 Sequential covering strategy

1. Choose a condition that results in the maximum support of a candidate rule,
2. If a tie occurs, choose a condition that results in the largest confidence of a candidate rule.

Several other criteria are considered, the most common choices are: entropy-based measures calculated over the distribution in the examined cover [6, 37], Laplace estimate or more flexible m-estimate [11]. Weighted formulas are useful as well, e.g., weighted information gain used by Quinlan in FOIL, J-measures and many others (for a review see again [10]).

The basic covering strategy presented above reveals drawbacks if data is noisy. Rules for noisy examples may be too complicated (*overfitted to noise*) and lead to low predictive accuracy while classifying new examples. In general, there are several solutions to overcome the overfitting, which usually rely on *pruning*. They allow induced rules not to cover all positive examples or to cover some negative ones. Efficient techniques for rule post-pruning were employed in the RIPPER algorithm [7], which is one of the most popular techniques of rule induction. Different rule pruning techniques are summarized in [9].

Finally, we would like to note that rules are also successfully applied inside multiple classifiers (ensembles). For instance, basic concepts of RIPPER were adopted inside SLIPPER [8], similarly, MODLEM was used inside extended bagging and pairwise coupling [40].

2.4 Induction of Non-minimal Sets of Rules

Minimal sets of rules usually contain only a *limited number* of interesting rules, they may also include some rules of very little or no interest, which is undesirable from the discovery-oriented perspective. These shortcomings result directly from the

sequential covering strategy described in the previous section. This strategy excludes from consideration learning examples that have been already covered by generated rules, thus, some interesting rules cannot be discovered. This happens especially when different patterns are shared by a large number of examples. Moreover, the sequential covering strategy aims at covering all positive learning examples, therefore, in last iterations it may produce very specific rules, consisting of many elementary conditions, which refer only to one or very few learning examples that have been left uncovered. More detailed discussion on this problem is given in [42].

In order to overcome the above limitations other induction strategies and algorithms have been proposed. The most radical solution is to produce a so-called *exhaustive* set of rules, which contains *all* rules that can be induced on the basis of positive examples of a class. Examples of such approach include the *dropping condition* technique described in [12] or *Boolean reasoning* approach to looking for local object reducts [36] (specific for rough set theory). However, time complexity for the latter technique is exponential and using it may be not practical for larger data sets, so approximate algorithms are employed. Moreover, the data analyst could be "overloaded" by getting too many rules to be considered. In fact, only a small number of them is usually interesting (these approaches may generate many specific rules supported by few learning examples).

Another category of induction algorithms employs "more efficient" search strategy leading to less numerous sets of rules, which should be also characterized by better values of evaluation measures. A good example is the BRUTE algorithm introduced in [35]. The name comes from authors' motivation to perform a massive, brute-force search for accurate rules in place of the greedy hill-climbing search typical for the iterative sequential covering. Briefly speaking, BRUTE conducts an exhaustive depth-bounded search for the most accurate and shortest rules. It optimizes the search by introducing canonical order in possible conditions. Moreover, it limits the search to rules not exceeding the maximum number of conditions in their condition parts. Finally, it outputs only a limited number of rules that have been most accurate on a learning set. Experimental results showed that a classifier using the top 50 rules outperformed CART and C4 trees [35]. A similar idea is present *Data Surveyor system* described in [18].

Finally, the last group of approaches includes adaptation algorithms for association rule mining. Such rule are transformed into a form where the right hand side of a rule contains the decision class. Rules should also satisfy predefined requirements for the minimum support (such rules are called frequent ones) and the minimum confidence.[2] The key issue is to adopt in a proper way search strategies derived from algorithms for mining frequent items, e.g., to construct an iterative sequential extension approach similar to Apriori, which efficiently prunes candidates. In [16] there is a short review of some proposals, e.g., a method of associative classification by Liu et al. [26].

[2] These requirements are similar to the ones presented in EXPLORE - see Section 3.

2.5 Classification Strategies

In the classification-oriented perspective a set of rules is used to classify new examples, i.e., examples unseen in the learning phase, by matching them to the condition parts of rules. Sets of rules can be either ordered or unordered. In the first case rules are organized into a priority list. The matching is done starting from the first rule. The first matched rule from the list is used to classify a new example and the remaining rules are skipped. The last rule is a default rule and it is used if no other rule has been matched.

For unordered sets of rules matching a new object may lead to three situations. The first one is a unique match to one or more rules from the same class. The two other situations are *matching multiple rules* indicating different classes or *not matching* any rules at all. In both situations a suggestion is ambiguous, thus, proper resolution strategy is necessary. Review of different strategies is given in [39]. Below we briefly summarize a classification strategy introduced by Grzymala-Busse in LERS [13] as it is employed in our experiments. In case of ambiguous multiple matching the decision how to classify an example e is made on the basis of voting and e is assigned to the strongest class (i.e., the class that has received most votes). For each matched rule its absolute support is considered as a basic score. The total *support* for a class K_i and an example e is defined with the following expression:

$$sup(K_i, e) = \sum_{rules\, for\, K_i\, matching\, e} sup(r).$$

The class K_j with the largest support is the winner and the example e is assigned to it.

If complete matching is impossible, all *partially matching rules* are identified. These are rules with at least one elementary condition matching an example e. The total support is then calculated from the support of identified rules, and from their matching factors, defined as a ratio of conditions matched by e to all conditions in a rule (or to the length of a rule):

$$sup(K_i, e) = \sum_{rules\, for\, K_i\, partially\, matching\, e} sup(r) \times match(r, e).$$

Again, the class K_j with the largest support is the winner and an example e is classified as its member.

3 EXPLORE Algorithm

In this chapter we present the EXPLORE algorithm that extracts from data all rules that satisfy requirements defined by the user. Thus, EXPLORE is able to generate rules which are general, simple, accurate and relevant. This makes it very useful not only from the descriptive-oriented perspective but also also from the classification-oriented one, especially when imbalanced data has to be dealt with.

3.1 Presentation of the Algorithm

The *EXPLORE* algorithm, first presented in [30], is a procedure that extracts from data all decision rules that satisfy certain requirements. In this study we focus on the following ones:

- *support* or *coverage* of a rule: the user can expect that the general and strong rule should cover a large enough number of positive examples,
- *consistency* of a rule represented by its *confidence*: the rule should cover no or very few negative examples,
- *simplicity* of a rule represented by its *length*: generally the rule should be short,
- *total number of rules* for all decision classes: the resulting set of rules should be limited in size for cognitive reasons.

These requirements are used to impose restrictions on the rule space explored during induction. The algorithm can handle inconsistent examples either by using rough set theory to define approximations of decision classes, or by determining appropriate threshold for confidence of induced rules to be used in pre-pruning.

Exploration of the rule space is performed using a procedure, which is *repeated* for each class K_j to be described. The main part of the algorithm is based on a *breadth-first search*, which amounts to generating rules of increasing size, starting from one-condition rules. Exploration of a specific branch is stopped as soon as a rule satisfying the requirements is obtained or any of stopping conditions *SC*, reflecting the impossibility to fulfill the requirements, has been met. EXPLORE is formally presented in Figure 2. A short description of the algorithm is given below, more enhanced discussion is provided in [42], and its implementation details can be found in [30]).

An initial list *LS* representing elementary conditions is created by analyzing positive examples provided as input to EXPLORE (for more precise description see [38]). Obviously, conditions in *LS* must cover at least one example from K_j; they may also be subject to specific constrains on their syntax. This initial list is first pruned to discard conditions, which directly correspond to rules, as well as those which already satisfy *SC*, and thus cannot give rise to rules (procedure *good_candidates*). Conditions remaining in *LS* are then combined to form *complexes* (i.e., conjunctions of elementary conditions), which are candidates for the condition parts of rules. This is achieved by procedure *extend*, which at iteration k creates conjunctions of size $k+1$ by extending candidate conjunctions of size k with conditions from *LS*. While extending the conjunctions we can use the monotonicity principle known from the Apriori algorithm, stating that all subsets of a candidate conjunction must also be sufficiently strong [16]. The resulting conjunctions are then tested by procedure *good_candidates*.

In general, stopping conditions *SC* can be defined according to requirements expressing various expectations of the user, e.g., imposed on coverage, length, number of rules, etc. In our experiments presented in Section 6 we mainly consider requirements referring to the minimal coverage l. The corresponding stopping condition for a conjunction C currently examined is thus simply: $cov(C) < l$. Let us remark

```
procedure EXPLORE
(input LS: list of valid elementary conditions; SC: stopping conditions;
output R: set of rules)
begin{Main search procedure}
    R ← ∅
    good_candidates(LS, R);   {LS is a list of valid elementary conditions s1, s2, ..., sn
            ordered according to decreasing coverage}
    Q ← LS; {Copy current LS to a queue Q}
    while Q ≠ ∅ do
    begin
        select the first conjunction C in Q;
        Q ← Q \ {C}; {remove it from the queue}
        extend(C,LC); {generate LC – a list of extended conjunctions}
        good_candidates(LC,R);
        Q ← Q ∪ LC {place all conjunctions from LC at the end of Q}
    end
end;

procedure extend(input C: complex;
    output L: list of conjunctions)
{ This procedure puts in list L extensions of conjunctions C that are potential candidates for rules.}
begin
    Let k be the size of C and h be the highest index of the elementary condition involved in C;
    L ← {C ∧ s_{h+i} where s_{h+i} ∈ LS and such that all the k subconjuctions of C ∧ s_{h+i}
            of size k and involving s_{h+i} belong to Q (i = 1, ..., n − h)}
end;

procedure good_candidates(input L: list of conjunctions;
    output R: set of rules)
{ This procedure prunes list L, discarding:
    - conjunctions, which cannot give rise to rules due to SC,
    - conjunctions corresponding to rules, which are stored into R. }
begin
    for each C ∈ L do
    begin
        if C satisfies SC then L ← L \ {C}
        else
            if conf(C) ≥ α then      {α = 1 for totally discriminant rules}
            begin
                R ← R ∪ {C};
                L ← L \ {C}
            end
    end
end;
```

Fig. 2 The main procedure of the EXPLORE algorithm

that stopping conditions restrict exploration space and reduce computational costs. If the user does not define any requirements, the algorithm will produce all rules, which is at the risk of exponential complexity (see the evaluation of complexity in [39]). Examples of using EXPLORE and tuning *SC* are presented in the next two subsections.

Besides basic requirements represented by the stopping conditions, EXPLORE can be easily adopted to handle additional expectations of the user with regard to the syntax of the condition parts of rules. For instance, the user can express her preferences for some specific elementary conditions or attributes to be used in a rule (or to be excluded from a rule). It is also possible to focus the search on some specific subsets of examples. Such an approach is typical for interactive knowledge discovery tools and it has been described in [41, 39].

3.2 Example: Using EXPLORE for Technical Diagnostics

To demonstrate the benefits of a non-minimal set of rules induced by the EXPLORE algorithm we describe a real life problem of technical diagnostics of a homogeneous fleet of buses [52]. 76 buses were described by 8 diagnostic symptoms (attributes) and divided into two classes depending on their technical conditions (good or bad). The following symptoms were chosen: *s1* – maximum speed, *s2* – compression pressure, *s3* – blacking components in exhaust gas, *s4* – torque, *s5* – summer fuel consumption, *s6* – winter fuel consumption, *s7* – oil consumption and *s8* – maximum horsepower of the engine. All these attributes were numeric.

We started with inducing a minimal set of rules using the MODLEM algorithm (this algorithm follows the sequential covering strategy and is well suited for numerical data [37]). The generated set contained the three following rules covering all learning examples (numbers in brackets correspond to positive learning examples covered by each rule) :

1. if ($s2 \geq 2.4$ MPa) & ($s7 < 2.1$ *l*/1000km) then (technical state=good) [46]
2. if ($s2 < 2.4$ MPa) then (technical state=bad) [29]
3. if ($s7 \geq 2.1$ *l*/1000km) then (technical state=bad) [24]

Prediction accuracy of a classifier using these rules was evaluated using the leaving-one-out technique, and it was equal to 98.7%. Although it was a very accurate predictor of the technical condition, the analysis of the syntax of these three rules showed that only two symptoms were important. In particular, the compression level was crucial as it nearly perfectly discriminated buses from the two considered classes. On the other hand, practical measurements of this symptom were the most difficult at the diagnostic stand. Thus, the experts were interested in discovering other rules, formulated using symptoms that were easier to collect. Therefore, they decided to discover strong rules covering more than 50% of buses in each class. EXPLORE found 11 rules satisfying the above requirements, they are listed below.

1. if ($s1 > 85$ km/h) then (technical state=good) [34]
2. if ($s8 > 134$ KM) then (technical state=good) [26]
3. if ($s2 \geq 2.4$ MPa) & ($s3 < 61$ %) then (technical state=good) [44]
4. if ($s2 \geq 2.4$ MPa) & ($s4 > 444$ Nm) then (technical state=good) [44]
5. if ($s2 \geq 2.4$ MPa) & ($s7 < 2.1$ *l*/1000km) then (technical state=good) [46]
6. if ($s3 < 61$ %) & ($s4 > 444$ Nm) then (technical state=good) [42]
7. if ($s1 \leq 77$ km/h) then (technical state=bad) [25]

8. if ($s2$<2.4 MPa) then (technical state=bad) [29]
9. if ($s7$≥2.1 l/1000km) then (technical state=bad) [24]
10. if ($s3$≥61 %) & ($s4$≤444 Nm) then (technical state=bad) [28]
11. if ($s3$≥61 %) & ($s8$<120 KM) then (technical state=bad) [27]

These rules provided much more information about values of symptoms than the previous minimal set of rules. We used them to construct a new classifier and estimated its accuracy again in the leaving-one-out test. The classification accuracy was exactly the same as for the classifier with the minimal set of rules.

3.3 Other Experience with EXPLORE

Let us remind that setting proper thresholds for the stopping conditions *SC* is crucial for the EXPLORE algorithm. An iterative procedure based on stepwise changes of the *rule coverage threshold* and observing its influence on the set of rules was presented in [42]. Experiments on several data sets from the UCI repository [1] showed that it was possible to determine a range of values for this threshold, which led to good sets of rules in terms of their classification accuracy, the average coverage, the average length and their number. In Table 1 we present sample results obtained for the *congress voting* data. The last line lists results for the minimal set of rules generated by LEM2. One can notice that threshold values between 20% and 30% led to sets of rules, which had significantly better descriptive properties (e.g., the average support was twice as high as for rules in the minimal set) and not worse classification properties at the same time.[3]

Table 1 Characteristics of rules induced by EXPLORE vs. the minimal set or rules induced by LEM2 for voting data (*SC* – rule support threshold, N_R – number of rules, *cov* – average rule coverage (absolute number of examples), *len* – average rule length (number of elementary conditions), *acc* – overall classification accuracy [%])

SC	N_R	*Cov*	*Len*	*Acc*
5%	231	45.86	3.36	97.91
10%	138	66.96	3.19	97.67
15%	125	75.46	3.71	96.98
20%	103	82.75	3.81	96.07
25%	80	86.95	3.95	95.38
30%	63	95.16	3.75	92.61
40%	21	133.00	2.76	80.23
LEM2	26	43.77	3.69	95.87

[3] We would like to clarify that the main aim of these experiments was not to get the most accurate classifier. The classification accuracy was just an additional criterion to evaluate the "quality" of a set of rules.

Continuing our discussion, we would like to stress that for large values of the coverage threshold EXPLORE may induce a set of rules covering only a subset of learning examples. Some "difficult" examples (e.g., located in sparse subregions of classes) may not be covered by any strong rule. In [43] we proposed a solution to this problem and introduced a *hybrid approach*, where the first level of representation is constituted by rules and the second level is a set of learning examples not covered by these rules. This first level can be obtained either by rule pruning or by using EXPLORE with large values of the coverage threshold (the stepwise tuning mentioned above could be used to establish the threshold – for more details see [39, 43]). The classification strategy for new examples is a two stage approach. A new example is first classified by rules. If there is no match or matching is ambiguous, then the example is classified according to the k-nearest neighbor principle on the basis of stored examples.

This idea was verified in the problem of evaluating business loans [43]. The interesting observation was that the hybrid approach led to the highest classification accuracy of 81%, while the rule level itself gave 77% and other classifiers (e.g., a decision tree) around 74%. Furthermore, we noticed that this approach slightly increased the sensitivity for the minority class, which corresponded to the most risky loans leading to questionable or lost liabilities. Similar improvements were observed in a medical case study.

Such an impact of the threshold tuning procedure on the sensitivity for the minority class has been a direct inspiration for our current research on dealing with imbalanced data, which is presented in the following sections.

4 Handling Imbalanced Data

Many learning algorithms are formulated with an explicit or implicit assumption that learning sets are balanced. However, this is not always the case. Imbalanced data sets are quite common as many processes produce certain observations with different frequencies. A good example is medicine, where databases regarding a rare but dangerous disease usually contain a smaller group of patients requiring special attention, while there is much larger number of members of other classes – patients who do not require special treatment. Similar situations occur in other domains, e.g., in technical diagnostics or continuous fault-monitoring tasks, where non-faulty examples may heavily outnumber faulty ones. Survey papers [49, 3] report other real technical or engineering problems, e.g., detection of oil spills in satellite radar images, detection of fraudulent telephone calls or credit card transactions, prediction of telecommunication equipment failures, and information retrieval and filtering.

If the imbalance in the class distribution is extensive, i.e., some classes are *heavily under-represented*, these learning methods do not work properly. They are "somehow biased" to focus searching on the more frequent classes while "missing" examples from the minority class. As a result constructed classifiers are also biased toward recognition of the majority classes and they usually have difficulties (or even are unable) to classify correctly new examples from the minority class. In [25]

authors described an information retrieval system, where the minority class (being of primary importance) contained only 0.2% of examples. Although the classifiers achieved accuracy close to 100%, they were useless because they failed to deliver requested documents from this class. Similar degradation of classifier's performance for the minority class was reported for other imbalanced problems [4, 14, 20, 22, 49].

The class imbalance also affects rule-based classifiers – especially classifiers using minimal sets of rules are biased toward the majority classes. Rules induced for the majority classes are more general and cover more learning examples, while rules for the minority class are usually more specific and "weaker" in terms of their cover. As a result new examples from the minority class tend to be misclassified.

Imbalanced data constitutes a problem not only when inducing rules to be used in a classifier, but also when evaluating its performance. Indeed, overall classification accuracy is not the only and the best criterion characterizing performance of a classifier. Satisfactory recognition of the minority class may be often more preferred, thus, a classifier should be characterized rather by its *sensitivity* and *specificity* for the minority class. Sensitivity (also called a true-positive rate) is defined as the ratio of correctly recognized examples from the minority class and specificity is the ratio of correctly excluded examples from the majority classes. More attention is usually given to sensitivity than to specificity [14]. However, in general there is trade-off between these two measures, i.e., improving sensitivity may lead to deterioration of specificity – see experimental results in [45]. Thus, some measures summarizing both points of view are considered. One of them is *G-means* [22], calculated as a geometric mean of sensitivity and specificity.

Several authors also use the *ROC (Receiver Operating Characteristics) curve* analysis. A ROC curve is a graphical plot of a true positive rate (sensitivity) as a function of a false positive rate (1 − specificity) along different threshold values characterizing performance of a studied classifier. The quality of the classifier performance is reflected by the area under a ROC curve (so-called AUC measure) [3, 49].

A small number of examples in the minority class ("the lack of data") is not the only source of difficulties in inducing classifiers. Several researchers claim that besides the size of this class it is necessary to go deeper into its other characteristics. Quite often the minority class overlaps heavily the majority classes. In particular, boundaries between classes are ambiguous. Both boundaries and the inside of the minority class may be affected by noisy examples from other classes, which cause incorrect classification of many examples from the minority class. Their influence is more critical for this class than for the majority ones – see [22, 24] for experiments and discussion. Japkowicz in her experimental study [19] also showed that the class imbalance becomes even more difficult problem particularly when the minority class contains very small subclusters, which are difficult to be learned (so-called, a small disjunct problem).

Several methods have been proposed to improve performance of classifiers learned from imbalanced data, for a review see [20, 49]. In general, one can distinguish two types of approaches. The first category includes pre-processing techniques that change the distribution of examples among classes by appropriate

sampling. Simple random *over-sampling*, which replicates examples from the minority class, or random *under-sampling* that randomly eliminates examples from the majority classes until a required degree of balance between classes is reached are not the best solutions. Focused methods like SMOTE, one-side-sampling, NCR or selective filtering attempt to take into account internal characteristics of regions around examples from the minority class. Thus, they modify only these examples from majority classes, which most likely lead to misclassifying their minority class neighbors and in a more sophisticated way over-sample or introduce synthetic examples in local sub-regions of the minority class. These methods and their combinations were experimentally shown to be quite good [2, 22, 45, 47].

Other approaches proposed in the literature modify either induction or classification strategy, assign weights to examples, and use boosting or other combined classifiers [48]. Some researchers transform the problem of learning from imbalanced data to the problem of cost learning (although it is not the same and misclassification costs are unequal and unknown) and use techniques from the ROC curve analysis.

Considering the approach we propose later in this paper, the most related research is the work by Grzymala-Busse [14] on increasing sensitivity of LEM2 rule classifiers by changing the LERS classification strategy – as described in Section 2.5. Necessary changes of the strategy are limited to formulas for calculating support for a given class that are presented in Section 2.5. The main idea of this approach is to multiply the support of all minority class rules by the same real number, called a *support multiplier*, while not changing the support of rules from the majority classes. This support multiplier is a positive number greater or equal to 1 – for the majority classes it should be equal to 1, while for the minority class it should be greater than 1. As a result, during classification of a new example, such minority class rules have a better chance to influence the voting, so the minority class is finally predicted for the new object.

Another problem is selecting a value for the support multiplier. In general, the sensitivity of a classifier increases with increase of the support multiplier. However, at the same time specificity decreases, thus, it is important to identify a proper value of this parameter. In [14] Grzymala-Busse proposed to maximize a measure called $gain = sensitivity + specificity - 1$. Following this proposal, a value of the support multiplier was established experimentally in a loop, where in each iteration the support multiplier was increased, the classifier was evaluated on extra validation examples, and the loop stopped as soon as a value of the gain measure decreased. The value of the support multiplier resulting in the best gain was used in the final classifier. Experimental results confirmed that this approach outperformed the standard LEM2 classifier for many imbalanced medical data sets [14, 15].

Some other researchers tried to develop a *less greedy search* strategy while looking for rules (an example is a version of the BRUTE algorithm described in [35], or a specific genetic algorithm [49]), or to change the *inductive bias of the algorithm*, e.g., Holte at al. modified the rule induction algorithm CN2 to improve its performance for small disjuncts corresponding to rare examples from the minority class. Moreover, Weiss describes hybrid and two-phase rule induction [49], where

one phase focuses on optimizing sensitivity, while the other optimizes specificity. Other approaches may use knowledge about prior distribution of probabilities or transforming the task to cost sensitivity learning [49].

5 Replacing a Set of Rules for the Minority Class

In this section we briefly describe our classifier-specific approach to handle imbalanced data – we evaluate it experimentally in Section 6 together with Grzymala-Busse's proposal of modifying the rule support for the minority class. Let us remark that both approaches assume an initial classifier uses a minimal set of rules. In general, it can be induced by any sequential covering algorithm, but in this paper we have chosen the LEM2 algorithm [13] and the classification strategy described in Section 2.5.

The new approach is inspired by the observation that in a minimal set of rules the average support of rules pointing at the majority classes is greater than the average support of rules for the minority class, so when classifying a new example the minority class may be easily outvoted. Such situation results in deteriorated sensitivity of a classifier.

The new approach, called *Replacing Rules for the Minority Class*, has been sketched for the first time in [43]. Generally speaking, unlike the Grzymala-Busse's approach, which addresses this issue by artificially increasing the support of rules for the minority class, it improves sensitivity for this class by replacing the minimal set of rules by a non-minimal set of stronger rules generated by the EXPLORE algorithm. Since these rules have better (greater) support than the original ones and are shorter (i.e. easier to be matched by a new example), there is no need for any modification of the classification strategy.

When inducing rules with EXPLORE, the stopping condition *SC* specifies the minimum required coverage or the support for constructed rules (rules with coverage below a given threshold are discarded). Setting the right values of this threshold is crucial. If the threshold is very low, EXPLORE may generate a very large set of rules for the minority class, that easily outvote rules for the majority classes what leads to high sensitivity at a cost of low specificity. On the other hand, if the threshold is very high, EXPLORE generates a very small set of very strong rules. Such rules well describe most common learning examples, however, fail to capture less frequent ones, thus many new examples are classified using partially matched rules. Then, the rules for the majority classes by the virtue of their number have better chance to win in the voting, what results in higher specificity and lower sensitivity.

We establish the range for the coverage threshold by checking the minimum and maximum coverage of the initial minimal set of rules for the minority class. The maximum coverage of rules generated by LEM2 and EXPLORE should be the same, thus, there is no sense in examining larger values (EXPLORE would generate no rules in such case). Moreover, the minimum coverage indicates the prevalence of the least frequent pattern in learning data, so it is not necessary to check smaller thresholds. We iteratively examine possible coverage thresholds within the

procedure replace_rules (**input** K_{min}: the minority class;
R : initial minimal set of rules;
L : learning examples; T : validation examples;
output R^{final}: resulting set of rules)
begin
    $min\_sup \leftarrow$ minimum coverage in R for K_{min}
    $max\_sup \leftarrow$ maximum coverage in R for K_{min}
    $R_{maj} \leftarrow$ rules from R pointing at the majority classes
    $R_{min}^{min\_sup} \leftarrow$ use EXPLORE to induce rules from L for K_{min}
        with minimum required coverage set to min_sup
  for $sup = min\_sup$ **to** max_sup **do**
  begin
    $R_{min}^{sup} \leftarrow$ select these rules from $R_{min}^{min\_sup}$ for which coverage $\geq sup$
    $R^{sup} \leftarrow R_{min}^{sup} \cup R_{maj}$
    $gain \leftarrow$ evaluate R^{sup} on T
    memorize $gain$ and R^{sup}
  end
  $R^{final} \leftarrow R^{sup}$ corresponding to the best observed $gain$
end

Fig. 3 Replacing rules for the minority class

identified range. In each iteration of the loop we use EXPLORE to generate rule for the minority class with the minimum coverage equal to the current threshold. Then, we combine these rules with the minimal rules for the majority classes and evaluate the resulting classifier on extra validation examples using the gain measure as in the Grzymala-Busse's approach. Finally, we select the set of rules that resulted in the highest gain to be embedded in the final classifier. In order to avoid repeating induction for various coverage thresholds, it is sufficient to create a set of rules for the minimal threshold and filter it appropriately in subsequent iterations of the loop. Figure 3 illustrates a basic version of our approach.

6 Experimental Evaluation

To evaluate the usefulness of our approach we experimentally compared it to a baseline classifier using a minimal set of rules generated by LEM2 [12]. Moreover, we considered a variant of such a classifier with a modified classification strategy expanded with the support multiplier (this modification proposed by Grzymala-Busse is particularly suited to deal with imbalanced data – see its description in Section 4).

We decided to examine the three measures: sensitivity, specificity and G-mean because they are more intuitive than AUC measure and they correspond to the fully deterministic algorithms (which is a case of our rule-based classifiers). Additionally, we report overall classification accuracy. Values of all these measures are presented as percentages. They are estimated as means in the k-fold cross validation. Moreover, to minimize the influence of splitting data sets on the classification results

Table 2 Characteristics of data sets used for experiments (N – number of examples, N_{Pos} – number of examples in the minority class, N_{Oth} – number of examples in the majority classes, $R_{Pos} = N_{Pos}/N$ – ratio of examples in the minority class)

Data set	N	N_{Pos}	N_{Oth}	R_{Pos}
Abdominal Pain	723	202	521	27.9%
Breast Slovenia	294	89	205	30.3%
Breast Wisconsin	625	112	513	17.9%
Bupa	345	145	200	42.0%
German	666	209	457	31.4%
Hepatitis	155	32	123	20.6%
Pima	768	268	500	34.9%
Scrotal Pain	201	59	142	29.4%
Urology	498	155	343	31.1%

Table 3 Results for the original LEM2 algorithm (*sens* – sensitivity, *spec* – specificity, *GM* – G-mean, *acc* – overall accuracy, N_R – number of rules)

Data set	*Sens*	*Spec*	*GM*	*Acc*	N_R
Abdominal Pain	58.42	92.90	73.67	83.26	20.0
Breast Slovenia	36.47	88.56	56.83	73.08	20.5
Breast Wisconsin	31.25	92.59	53.79	81.60	29.5
Bupa	32.41	74.00	48.97	56.52	42.0
German	30.14	84.68	50.51	67.57	42.5
Hepatitis	43.75	95.12	64.51	84.52	6.5
Pima	39.18	82.60	56.89	67.45	66.0
Scrotal Pain	54.24	83.10	67.14	74.63	12.0
Urology	12.18	82.27	31.65	60.40	28.0

obtained for all approaches, the division into folds was performed only once and the same subsets of examples were used to construct all three variants of classifiers.[4]

Experiments were conducted on 9 imbalanced data sets, which are coming from UCI repository except two data sets *abdominal pain* and *scrotal pain* – these are coming from our practical case studies [51, 27]. Let us notice that nearly all data sets, except *German credit*, come from a medical domain. Data sets, which originally included more than two classes, were transformed to binary ones, by collapsing all the majority classes into one. Moreover, some of the original data sets contained numerical attributes, which was a disadvantage for LEM2, thus, these attributes were discretized by a Grzymala-Busse's method based on clustering with merging intervals [5]. Table 2 lists the data sets along with their basic characteristics.

[4] In our previous joint research with Grzymala-Busse [15] we conducted some experiments, thus, some of results for LEM2 come from that paper.

Table 4 Best results of increasing rule support by multipliers (*mult* – support multiplier, *sens* – sensitivity, *spec* – specificity, *GM* – G-mean, *acc* – overall accuracy)

Data set	*Mult*	*Sens*	*Spec*	*GM*	*Acc*
Abdominal Pain	5	80.69	84.84	82.74	83.68
Breast Slovenia	1	36.47	88.56	56.83	73,08
Breast Wisconsin	5	57.14	86.74	70.41	81.44
Bupa	3	55.86	58.50	57.17	57.39
German	4	57.89	64.11	60.92	62.16
Hepatitis	18	84.38	77.24	80.73	78.71
Pima	3.5	59.33	76.40	67.32	70.44
Scrotal Pain	3	67.80	80.99	74.10	77.11
Urology	14	51.92	49.42	50.65	50.52

Table 5 Results for the *Replacing Rules* approach (*SC* – coverage threshold, *sens* – sensitivity, *spec* – specificity, *GM* – G-mean, *acc* – overall accuracy, N_R – number of rules)

Data set	*SC*	*Sens*	*Spec*	*GM*	*Acc*	N_R
Abdominal Pain	8.0	83.14	83.68	83.41	83.54	88.0
Breast Slovenia	3.0	47.09	84.11	62.93	73.08	37.0
Breast Wisconsin	2.0	63.85	81.60	72.18	78.57	158.5
Bupa	2.0	42.75	63.00	51.90	54.50	61.5
German	5.0	62.71	72.65	67.50	69.50	73.5
Hepatitis	4.0	75.30	81.56	78.37	80.02	76.5
Pima	2.0	68.78	67.89	68.33	68.10	341.5
Scrotal Pain	4.0	68.87	87.24	77.51	81.56	12.5
Urology	4.0	71.73	43.20	55.67	51.61	691.5

Results for all compared approaches are presented in Tables 3 – 5. The approach to extend the classification strategy with the support multiplier is consistent with the Grzymala-Busse's proposal [14] of optimizing the gain measure (see also its description in Section 4) – the best values of the multiplier are listed in Table 3. For our approach we additionally present values of the rule support (coverage) threshold for the minority class and the number of rules generated by EXPLORE to replace the initial minimal set for the minority class. We compare results of both these approaches to the standard LEM2 rule-based classifier using the Wilcoxon Signed Ranks test (with $\alpha = 0.05$). Considering sensitivity and G-mean both approaches (based on the multiplier and replace techniques) outperform it, and the difference between them is significant for sensitivity, what emphasizes superiority of our approach. Moreover our approach significantly outperforms the multiplier approach with respect to G-mean. On the other hand, we should be aware of the fact that number of rules for the minority class significantly increases (especially for

abdominal pain, hepatitis, pima and urology). According to discussion in [15] it may be possible to impose an upper limit on the number of replaced rules so it is closer to the number in the original minimal set.

7 Conclusions

Our study focuses on using rule induction algorithms on imbalanced data to create improved rule-based classifiers. Following a comprehensive discussion of the most common rule induction algorithms and classification strategies we have shown they are too biased towards the majority classes – both during learning and classification phases. This is attributed mainly to a greedy search strategy of the sequential covering employed by many rule induction algorithms. However, this bias can be avoided either by changing the classification strategy or by using less greedy search for rules for the minority class.

The main research contribution of our study involves introducing a new approach to constructing rules for a rule-based classifier, where minimal sets of rules are induced for the majority classes, while for the minority class we create a non-minimal set of rules (more numerous, and characterized by higher average coverage) that improves a chance of a classification strategy to recognize the minority class. We have proposed to use the EXPLORE algorithm to generate rules for this class. As opposed to algorithms based on sequential covering, EXPLORE performs less greedy search and induces all rules that satisfy specific requirements (e.g., coverage greater than a given threshold).

In a series of experiments we have compared our approach to a baseline classifier with the minimal set of rules and the basic classification strategy, and a classifier with the classification strategy expanded with the strength multiplier. Experimental results have shown that both our approach and the approach with the support multiplier have increased sensitivity in comparison to the baseline classifier. However, let us notice that the multiplier approach is similar to over-sampling of learning data and in some cases it may lead to quite extensive changes in balance between classes (see Table 4). On the other hand, our approach does not modify learning data, rules discovered by EXPLORE correspond to really existing patterns and they are still comprehensible for human experts.

Further directions for our research include expanding our approach by post-pruning rules for the majority classes and manipulating learning examples in an "intelligent" way. The latter is a subject of our current work and we have already introduced a new approach to selective pre-processing of imbalanced data that aims at improving sensitivity of an induced classifier, while keeping overall accuracy at an acceptable level [45]. Briefly speaking, it combines selective filtering of difficult examples from the majority classes (either by removing examples, which may contribute to misclassification of examples from the minority class, or by relabeling some of them) with limited over-sampling of the minority class. In the first experimental study presented in [45] this approach was successfully combined with MODLEM. The more advanced research [46] also involved the use of C4.5 decision

trees [33] and RIPPER rules [7]. We conducted comprehensive experiments, where this approach was compared against other pre-processing methods, such as SMOTE, NCR, or simple random under- and over-sampling, showing its advantages. Unfortunately, more elaborated presentation is beyond the scope and limit of this paper.

Acknowledgements. This research is supported by the grant N N519 3505 33. We also acknowledge cooperation of J.Stefanowski with D. Vanderpooten from University Paris Dauphine on introducing the EXPLORE algorithm and of both of us with J.Grzymala-Busse from the University of Kansas on comparative experiments with LEM2.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6(1), 20–29 (2004)
3. Chawla, N.: Data mining for imbalanced datasets: an overview. In: Maimon, O., Rokach, L. (eds.) The Data Mining and Knowledge Discovery Handbook, pp. 853–867. Springer, Heidelberg (2005)
4. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16, 341–378 (2002)
5. Chmielewski, M.R., Grzymala-Busse, J.W.: Global discretization of continuous attributes as preprocessing for machine learning. In: Lin, T.Y., Wildberger, A. (eds.) Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery, pp. 294–297. Simulation Councils Inc. (1995)
6. Clark, P., Niblett, T.: The CN2 induction algorithm. Machine Learning 3, 261–283 (1989)
7. Cohen, W.: Fast effective rule induction. In: Proc. of the 12th International Conference on Machine Learning (ICML 1995), pp. 115–123 (1995)
8. Cohen, W., Singer, Y.: A simple, fast and effective rule learner. In: Proc. of the 16th National Conference on Artificial Intelligence (AAAI 1999), pp. 335–342. AAAI Press, Menlo Park (1999)
9. Furnkranz, J.: Pruning algorithms for rule learning. Machine Learning 27(2), 139–171 (1997)
10. Furnkranz, J.: Separate and conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)
11. Dzeroski, S., Cestnik, B., Petrovski, I.: Using the m-estimate in rule induction. Journal of Computing and Information Technology 1, 37–46 (1993)
12. Grzymala-Busse, J.W.: LERS - a system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, pp. 3–18. Kluwer, Dordrecht (1992)
13. Grzymala-Busse, J.W.: Managing uncertainty in machine learning from examples. In: Proc. of the 3rd International Symposium in Intelligent Systems, Wigry, Poland, pp. 70–84. IPI PAN Press (1994)
14. Grzymala-Busse, J.W., Goodwin, L.K., Grzymala-Busse, W.J., Zheng, X.: An approach to imbalanced data sets based on changing rule strength. In: AAAI Workshop at the 17th Conference on AI, AAAI 2000, Learning from Imbalanced Data Sets, Austin, TX, July 30–31, pp. 69–74 (2000)

15. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS, vol. 3213, pp. 757–763. Springer, Heidelberg (2004)
16. Han, J., Kamber, M.: Data mining: Concepts and techniques. Morgan Kaufmann, San Francisco (2000)
17. Hilderman, R.J., Hamilton, H.J.: Knowledge Discovery and Measures of Interest. Kluwer Academic, Boston (2002)
18. Holsheimer, M., Kersten, M.L., Siebes, A.: Data Surveyor: searching the nuggets in parallel. In: Fayyad, U.M., et al. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 447–467. AAAI/MIT Press, Cambridge (1996)
19. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis 6(5), 429–450 (2002)
20. Japkowicz, N.: Learning from imbalanced data sets: a comparison of various strategies. In: AAAI Workshop at the 17th Conference on AI, AAAI 2000, Learning from Imbalanced Data Sets, Austin, TX, July 30–31, pp. 10–17 (2000)
21. Klosgen, W., Żytkow, J.M.: Handbook of Data Mining and Knowledge Discovery. Oxford Press, Oxford (2002)
22. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-side selection. In: Proc. of the 14th International Conference on Machine Learning (ICML 1997), pp. 179–186 (1997)
23. Langley, P., Simon, H.A.: Fielded applications of machine learning. In: Michalski, R.S., Bratko, I., Kubat, M. (eds.) Machine learning and data mining, pp. 113–129. John Wiley & Sons, Chichester (1998)
24. Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. Technical Report A-2001-2, University of Tampere (2001)
25. Lewis, D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Proc. of 11th International Conference on Machine Learning (ICML 1994), pp. 148–156 (1994)
26. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proc. of the 4th International Conference on Knowledge Discovery and Data Mining, KDD 1998 (1998)
27. Michalowski, W., Wilk, S., Farion, K., Pike, J., Rubin, S., Slowinski, R.: Development of a decision algorithm to support emergency triage of scrotal pain and its implementation in the MET system. INFOR 43(4), 287–301 (2005)
28. Michalski, R.S.: A theory and methodology of inductive learning. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach, pp. 83–134. Morgan Kaufman, San Francisco (1983)
29. Michalski, R.S., Bratko, I., Kubat, M. (eds.): Machine learning and data mining. John Wiley & Sons, Chichester (1998)
30. Mienko, R., Stefanowski, J., Toumi, K., Vanderpooten, D.: Discovery-oriented induction of decision rules. Cahier du Lamsade no. 141, Paris, Université Paris Dauphine (September 1996)
31. Mitchell, T.: Machine learning. McGraw-Hill, New York (1997)
32. Pawlak, Z.: Rough sets. In: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
33. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1992)
34. Ras, Z., Wieczorkowska, A.: Action rules: how to increase profit of a company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)

35. Riddle, P., Segal, R., Etzioni, O.: Representation design and brute-force induction in a Boening manufacturing fomain. Applied Artificial Intelligence Journal 8, 125–147 (1994)
36. Skowron, A.: Boolean reasoning for decision rules generation. In: Komorowski, J., Raś, Z.W. (eds.) ISMIS 1993. LNCS (LNAI), vol. 689, pp. 295–305. Springer, Heidelberg (1993)
37. Stefanowski, J.: The rough set based rule induction technique for classification problems. In: Proc. of the 6th European Conference on Intelligent Techniques and Soft Computing EUFIT 1998, Aachen, pp. 109–113 (1998)
38. Stefanowski, J.: Handling continuous attributes in discovery of strong decision rules. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 394–401. Springer, Heidelberg (1998)
39. Stefanowski, J.: Algorithims of rule induction for knowledge discovery. Habilitation Thesis published as Series Rozprawy no. 361. Poznan Univeristy of Technology Press, Poznan (2001) (in Polish)
40. Stefanowski, J.: On combined classifiers, rule induction and rough sets. In: Peters, J., et al. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 329–350. Springer, Heidelberg (2007)
41. Stefanowski, J., Borkiewicz, R.: Interactive rule discovery of decision rules. In: Proc. of the VIIIth Intelligent Information Systems, June 1999, pp. 112–116. Wyd. Instytutu Podstaw Informatyki PAN, Warszawa (1999)
42. Stefanowski, J., Vanderpooten, D.: Induction of decision rules in classification and discovery-oriented perspectives. International Journal of Intelligent Systems 16(1), 13–28 (2001)
43. Stefanowski, J., Wilk, S.: Evaluating business credit risk by means of approach integrating decision rules and case based learning. International Journal of Intelligent Systems in Accounting, Finance and Management 10, 97–114 (2001)
44. Stefanowski, J., Wilk, S.: Rough sets for handling imbalanced data: combining filtering and rule-based classifiers. Fundamenta Informaticae 72, 379–391 (2006)
45. Stefanowski, J., Wilk S.: Improving rule based classifiers induced by MODLEM by selective pre-processing of imbalanced data. In: Proc. of the RSKD Workshop at ECML/PKDD, Warsaw, pp. 54–65 (2007)
46. Stefanowski, J., Wilk, S.: Selective pre-processing of imbalanced data for improving classification performance. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 283–292. Springer, Heidelberg (2008)
47. Van Hulse, J., Khoshgoftarr, T., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: Proc. of the 24th International Conference on Machine Learning (ICML 2007), pp. 935–942 (2007)
48. Wang, B., Japkowicz, N.: Boosting support vector machines for imbalanced data sets. In: An, A., Matwin, S., Raś, Z.W., Ślezak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 38–47. Springer, Heidelberg (2008)
49. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter 6(1), 7–19 (2004)
50. Weiss, S.M., Indurkhya, N.: Predicitive Data Mining. Morgan Kaufmann, San Francisco (1999)
51. Wilk, S., Slowinski, R., Michalowski, W., Greco, S.: Supporting triage of children with abdominal pain in the emergency room. European Journal of Operational Research 160(3), 696–709 (2005)
52. Zak, J., Stefanowski, J.: Determining maintenance activities of motor vehicles using rough sets approach. In: Proc. of Euromaintenance 1994 Conference, Amsterdam, pp. 39–42 (1994)

Converting between Various Sequence Representations

Gilbert Ritschard, Alexis Gabadinho, Matthias Studer, and Nicolas S. Müller

Abstract. This chapter is concerned with the organization of categorical sequence data. We first build a typology of sequences distinguishing for example between chronological sequences and sequences without time content. This permits to identify the kind of information that the data organization should preserve. Focusing then mainly on chronological sequences, we discuss the advantages and limits of different ways of representing time stamped event and state sequence data and present solutions for automatically converting between various formats, e.g., between horizontal and vertical presentations but also from state sequences into event sequences and reciprocally. Special attention is also drawn to the handling of missing values in these conversion processes.

Keywords: Sequence data organization, State sequence, Event sequence, Transition, Converting between sequence formats.

1 Introduction

Categorical sequence data appear in many different fields. We encounter for instance word or letter sequences in text mining, protein or DNA sequences in biology, functioning state sequences in device control, sequences of successively visited web pages in web log analysis and biographical data describing life trajectories in social sciences. There are also multiple ways of analysing such data: Markov chain models and their extensions for analysing transitions between states, data-mining-based methods for discovering regular patterns in sequences, aligning techniques for finding component similarities, edit based distances for measuring proximities between sequences, survival analysis for studying time-to-event distributions to mention just a few of them. Now, depending on how data were collected, longitudinal and more generally sequential data may be organized in many different manners. On the other hand, when it comes to analysis, each software and method requires data

Gilbert Ritschard, Alexis Gabadinho, Matthias Studer, and Nicolas S. Müller
Department of Econometrics and Laboratory of Demography,
University of Geneva, Switzerland
e-mail: gilbert.ritschard@unige.ch

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 155–175.
springerlink.com

to be inputted in some specific form. For instance, the mining of frequent sequential pattern is usually intended for event sequences, Markov models and edit-distance-based methods work on state sequences, while discrete survival models need data in person-period form. The end-user, and especially the end-user who wants to combine different types of analyses thus faces the difficult and often discouraging task of transforming his data in the right form.

The aim of this chapter is to help the analyst in this data preparation task. We propose a systematic description of the different ways sequential data can be organized, which should help to identify the nature of the data at hand. We then discuss issues raised by the transformation of one type of organization into another one. We explain why some of these transformations can be done straightforwardly in an automatic way, while others may require the user to define some rules to ensure that the outcome best suits her/his needs. We describe the automatic and semi-automatic procedures for switching from one type of organization to the other.

Our primary interest is in sequential data describing life courses, that is in sequences with order determined by the time. Hence, we shall indeed also pay attention to the time content, that is to the different ways of accounting for time, essentially calendars for defining time stamps and clocks for measuring time spans and spell durations. We adopt however on this aspect a practical standpoint as opposed to the logical definition of time concepts that can be found for instance in Hobbs and Pan [6].

This chapter is, as far as we know, the first attempt to present a general systematic view of the different ways of organizing and reorganizing time sequenced data. Karweit and Kertzer [7] discussed some aspects of the organization and conceptualization of life course data, but they mainly focused on data storage and access issues and the characterization of the units of analysis in terms of case and time. Sequence data organization issues have indeed been considered in the literature, but most often for a specific task only as Zaki [11] who describes in details an efficient way of organizing data for mining frequent sequence patterns or Blossfeld et al. [2] who discuss data organization for event history analysis.

The remainder of the chapter is organized as follows. In section 2 we define the different kinds of discrete sequences. A comprehensive list of sequence formats is presented in Section 3 and Section 4 is devoted to the handling of missing values. Then, in Section 5 we discuss the conversion between formats proposing among others rough basic solutions for automatically converting between state and event sequence data. Section 6 shortly comments on the implementation of the proposed solutions in our TraMineR package for R and Section 7 presents a few concluding remarks.

2 Sequence Concepts

We consider sequences of discrete or categorical data. Formally, we define thus a sequence of length k as an ordered list of k elements successively chosen from a finite set A. The set A is called the *alphabet*. A natural representation of a sequence x is by

listing the successive elements that form the sequence, for example as a sorted list $x = (x_1, x_2, \ldots, x_k)$, with $x_j \in A$. When there is no ambiguity, we can just concatenate the successive values into a string, $x = x_1 x_2 \ldots x_k$. A separator would indeed be necessary when the alphabet includes any non-single symbol, which happens if we use for instance *M* for *married* and *MC* for *married with a child*.

Now, the nature of the sequence, that is its information content, depends on

1. what the position j in the sequence refers to;
2. the nature of the elements that compose the alphabet.

Regarding the position of each element in a sequence, it is important to distinguish between sequences with a time dimension and those without any reference to time. An occupational trajectory, a buyer's history, or a record of device control signals typically contain chronologically sorted elements and, hence, have a time dimension. On the other hand, the order in texts and DNA sequences does not refer to time. In the latter case, i indicates simply the rank position, while j may bear more information when time matters. For instance, when data are collected at periodic dates as with panel data, the positions correspond to pre-specified dates (or periods). In that case, the position j informs about the date and a difference between positions can be interpreted as a duration.

Concerning the second point, namely the nature of the elements in the alphabet, an important distinction to make is whether the elements are states or whether they represent events. A *state* lasts as long as nothing happens, i.e. during some interval time, while an event happens at a given time point and may cause a change of state. For instance, consider a device turned on at 9 for 3 hours, and turned off after that. We may either report the sequence of states at each hour, e.g., "off at 8, on at 9, on at 10, on at 11, off at 12", or alternatively report the sequence of on-off events, that is "turns on at 9, turns off at 12". This can indeed also be done for non-chronological sequences such as the sequence of nucleotides "AGGC". Here we could say that we start with 'A', switch to 'G' at position 2 and then to 'C' at position 4.

This preliminary discussion leads us to distinguish the four types of sequences depicted in Table 1. Notice that though an event can just be a transition between two successive states of a chronological sequence, a transition such as from 'single' to 'married with a child' for example, may be the result of more than one event, namely here 'marriage' and 'childbirth'. We discuss this issue in more details in Section 5.

For sequences with a time dimension, it is important to preserve the time information in any attempt to change the sequence representation. Hence it is essential to know the kind of time information the sequence holds. There exist different concepts of time: instant time ('I started a new job the 1st of December'), time interval

Table 1 Types of sequences

	Time Dimension	
Alphabet	No	Yes
States or objects	sequence of labels	**state sequence**
Events or transitions	sequence of transitions between labels	**event sequence**

('I had a job during the whole last year'), absolute time (birth date), relative time (age). For instance, assume we face a sequence of annual occupational statuses such as (full time, full time, unemployed, ...). What does 'unemployed in 2008' mean? Does it mean that the concerned person was continuously unemployed during the whole year (interval time), experienced unemployment in 2008 (interval time), or that he was unemployed at the time of observation say in December 2008 (instant time)? In the first case, the state change at the beginning of a sequence of unemployment states clearly corresponds to a 'falling in unemployment' event, while in the two latter situations, there could be alternating employed-unemployed sequences during the same spell. Time granularity is also an issue, Data collected with a year granularity are hardly comparable with monthly based sequences. Turning data into finer granularity can only be done through rough approximations such as by assuming that the state reported for the year remains valid for all months, while the reverse raises time aggregation issues such as how can I transform monthly sequences into yearly sequences?

Another important point for characterizing a sequence is whether or not it can admit multiple elements at a same position. This is clearly a concern for event

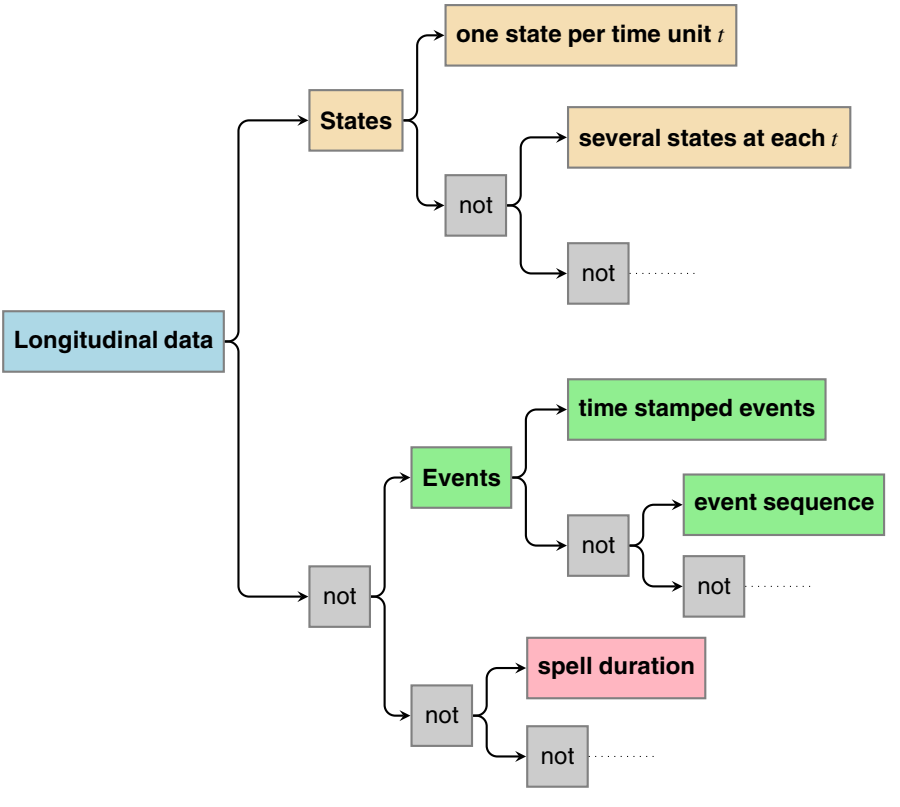


Fig. 1 Ontology of types of longitudinal data

sequences, where multiple events (e.g., leaving home and getting married) may occur at a same time. For state sequences, we would get multiple states at a same position when we deal simultaneously with different dimensions such as the cohabitational and occupational statuses, i.e., when we have non exclusive states. The latter kind of sequence is sometimes referred to as multichanel sequence [5]. The Aristotelean tree of Figure 1 can be seen as a tentative ontology of longitudinal data types, that is types of sequential data with time content.

3 Basic Sequence Formats

This Section discusses different ways of representing sequence data. We explicit in each case the nature of the represented sequence elements as well as how time is accounted for when it matters.

3.1 Horizontal Sequence Organizations

We consider first representations in which sequences are organized in row (record) form, that is with the position in the sequence determined by a column index.

State sequence (STS)

We defined a sequence as an ordered list of k elements. A natural way of representing this list is as a (row) vector of k elements. For instance, by considering the states single, S, married, M, married with a child MC and divorced D, a sequence of length 10 would look as

$$\big(S, S, S, M, M, MC, MC, MC, MC, D\big) \quad . \tag{1}$$

Using this representation, a set of n sequences would be stored in a $n \times k$ matrix in which each column j collects the states at position j for all cases.

We get a somewhat more compact form of exactly the same information by concatenating the elements into a single character string using for instance the '-' separator

"*S-S-S-M-M-MC-MC-MC-MC-D*" .

This concatenated form should be more economical from the storage space standpoint since it requires a single character variable.

Time information is here accounted for by assigning absolute — date — or relative — age, process duration — times to each position. For our example, assuming that the sequence reports yearly states between ages 18 to 27 years, we assign age labels to the position indexes:

Age	18	19	20	21	22	23	24	25	26	27
State	*S*	*S*	*S*	*M*	*M*	*MC*	*MC*	*MC*	*MC*	*D*

.

Notice that when we have a set of sequences, the time labeling of the position index needs just to be done once for the whole set of sequences, which is also economical in terms of required storage space.

Distinct-state, State-permanence and State-start sequence

In our previous example, we observe the same state in consecutive positions. We have for instance *S* in the three first positions. This suggests a simplified presentation in which we cite only one of several same consecutive states. This *distinct-state-sequence* (DSS) representation of our sequence is

$$\big(S,\, M,\, MC,\, D\big) \quad .$$

It preserves the state sequencing, but clearly all time and more generally alignment information is lost.

We can, however, easily reintroduce it by assigning a time or duration stamp to each element of a DSS sequence. Aassve et al. [1], for instance, stamp each state with the number of times it is repeated and call the resulting format *State-permanence-sequence*. We denote it as SPS. For our example, the SPS form is:

$$\big((S,3)\ (M,2)\ (MC,4)\ (D,1)\big) \quad .$$

We can indeed use any other notation for representing the (state, duration) couples, such as $\big(S/3,\, M/2,\, MC/4,\, D/1\big)$.

An alternative possibility, that we call *state-start-sequence* (SSS), consists in stamping states with start time instead of duration.

$$\big((S,18)\ (M,21)\ (MC,23)\ (D,27)\big) \quad .$$

Notice that strictly speaking SPS and SSS formats do not reproduce exactly the content of the corresponding STS form. With SPS data, we would need also the start time of the sequence, while for data in the SSS form we should specify either the end time or the duration of the last state.

Shifted-replicated-sequence

This data presentation is intended for analyses where the concern is the transition from the states observed at previous time points, $t-1, t-2, \ldots$, to the one observed at time t. Consider for example the sequence A, A, C, D, D where the first element in the sequence corresponds to year 2000 and the last one to year 2004, that is

2000	2001	2002	2003	2004
A	*A*	*C*	*D*	*D*

.

The *shifted-replicated-sequence* representation of this sequence is obtained by repeating each sequence $k-2$ times, shifting it each time one step on the right and dropping at each *i*th step the *i* right most elements out:

Table 2 Sequence data representations

Code	Data type	(S)tates or (E)vents	Several rows for a same case	Usage examples
STS	State-sequence	S	No	Markov modeling, OM
SPS	State-permanence	S	No	Markov modeling, OM
SSS	State-start	S	No	Markov modeling, OM
SRS	Shifted-replicated-sequence	S	Yes	Mobility tree
DSS	Distinct-state-sequence	S	No	OM without time reference
SPELL	Spell	S	Yes	Survival analysis
PPER	Person-period	S	Yes	Discrete survival analysis
FCE	Fixed-column-event	E	No	Survival analysis
HTSE	Horizontal time-stamped-event	E	No	Event sequence mining
TSE	Vertical time-stamped-event	E	Yes	Event sequence mining

OM stands for optimal matching and other analyses based on dissimilarities between pairs of sequences.

$t-4$	$t-3$	$t-2$	$t-1$	t
A	A	C	D	D
.	A	A	C	D
.	.	A	A	C
.	.	.	A	A

Finally, we relabel the columns with the relative time labels $t-k+1$ to t.

In this SRS form we collect for instance in the columns named '$t-1$' and 't' all consecutive subsequences of length two, and hence all observed transitions between two successive positions. The column $t-i$ gives the state found i positions before the state reported in the last column t. The column reference is no longer a given date or age, but is relative. This organization of the data is for example required for growing mobility trees [9].

3.2 Vertical Sequence Organizations

We now consider representations in which the elements of the sequence are given in successive rows. Such data representation is for instance especially useful for discrete time survival analysis [10].

Person-period data

The *Person-period* (PPER) data form is obtained by defining a separate record for each period lived by each individual. The person-period representation of a single

sequence can be seen as the transpose of its STS form. In PPER form however, the set of sequences is arranged in a single column by laying the sequences on top of each other. One advantage of this representation is that it allows to handle time varying covariates in a straightforward manner by simply completing the data with columns giving the values of the covariates for each considered time (row). A second advantage is that, unlike the STS form, it does not require all sequences to have the same start and end times. Periods not observed for a given case can be simply omitted. The price to pay for these advantages, especially the last one, is that in the PPER format the concerned period must be explicitly specified for each record. Here is the PPER format of our earlier example (1):

Id	*Index*	*Age*	*State*
101	1	18	Single (S)
101	2	19	Single (S)
101	3	20	Single (S)
101	4	21	Married (M)
101	5	22	Married (M)
101	6	23	Married w Child (MC)
101	7	24	Married w Child (MC)
101	8	25	Married w Child (MC)
101	9	26	Married w Child (MC)
101	10	27	Divorced (D)

.

Spell data

The *Spell* data (SPELL) organization is a compacted person-format form that uses a single record for representing successive periods with unchanged state. For a given sequence, it can be seen as the transpose of either the SPS (state-permanence) or SSS (state-start) format. As with the PPER format, however, in Spell form the sequences are stacked and not laid one beneath each other. Each record should indeed specify either the start and end times of the spell, or equivalently its start time and duration. Notice that if we can assume, what is not too restrictive, that each spell ends when the next one starts, then it would be sufficient to give the start time only (or the duration only) of each spell. The SPELL format of example (1) looks as follows

Id	*Index*	*From*	*To*	*State*
101	1	18	20	Single (S)
101	2	21	22	Married (M)
101	3	23	26	Married w Children (MC)
101	4	27	27	Divorced (D)

.

3.3 Event Sequences

The format discussed so far are primarily intended for state sequences. States are supposed to last and are naturally associated with interval time. Here we consider events, that is phenomenons that occur at given time-points and do not last. For

instance, starting a new job, getting married and switching a device off are events. They may result in a lasting new state, but events do not persist themselves. Hence, it is in time reference that the representation of a sequence of events will differ from state sequences.

As long as we are only interested in the sequencing of the events, we may rely on STS like or DSS representations. State-permanence has indeed no sense for non lasting events. Likewise, spell representation are not suited for event data. The most common way of representing event sequences is as *time-stamped-event* either horizontally or vertically.

Horizontal time stamped events

There are two possibilities for presenting *time-stamped-event* data horizontally. The first is similar to the STS form, with the dates at which events occur as column headings. This may be justified when events occur at the same regular dates for each case. Most often, however, a sequence of time stamped events is represented by a sequence of (event, time stamp) couples. We call this format *horizontal-time-stamped-event* (HTSE). For instance, if we consider the events defined by the state transitions of our state sequence example plus a second childbirth at 26, we would write down the data as

$$\big((\text{starts as single}, 18)\ (\text{marriage}, 21)\ (\text{childbirth}, 23)\ (\text{childbirth}, 26)\ (\text{divorce}, 27)\big)\ . \quad (2)$$

Notice that it is most often necessary to specify the state at the start of the observed period if we want to retrieve the whole state information from the sole knowledge of the events.

When events are repeatable, such as "starting a new job", "childbirth" or "turning a device on", it may be useful to know the rank of the event. In such cases, a sometime more convenient representation consists in grouping the events and reporting the number of events of a certain type, let us say the number of childbirths, and then list the successive time stamps of the events in fixed columns. We call this form *fixed-column-event* (FCE). For instance, the childbirth information of our previous example would look as follows in FCE format

id	*Number of childbirths*	*Age at 1st childbirth*	*Age at 2nd childbirth*	*Age at 3rd childbirth*	$\cdots$
101	2	23	26	*NA*	$\cdots$

.

Biographical data bases are often presented this way with for instance dates of changes in living arrangement, marital status, number of children, education and professional careers. It is convenient for survival methods such as for instance the estimation of a Kaplan-Meier curve, since it permits to easily compute the required duration from a start event until the event of interest. The disadvantage is that it may result in a very scarce table with plenty of empty entries.

Vertical time stamped events

As for state sequences, event sequences can indeed also be organized vertically by reporting each (event, time stamp) couple in a new line. We designate this *vertical-time-stamped-event* simply as TSE. Here is how our example looks out in this TSE format

id	*index*	*time*	*event*
101	1	18	Start as single
101	2	21	Marriage
101	3	23	Childbirth
101	4	26	Childbirth
101	5	27	Divorce

.

In this format, two simultaneous events, that is events with same time stamp such as (Marriage, 25) and (Childbirth, 25) would be represented by two lines. A variant sometime considered consists in giving the time stamp with the list of events in a same single line

id	*index*	*time*	*event*
102	$1-2$	25	Marriage, Childbirth

.

4 Missing Values

Before discussing how we can convert between formats, a few words are worth about how to record missing elements in sequences. Depending on where they appear in sequence, we distinguish the following types of missing values:

- *left-missing-values*, that is missing values appearing before the first valid entry in the sequence;
- *inner-missing-values* or *gaps*, that is missing values appearing somewhere between the first and last valid entries;
- *right-missing values*, that is missing values appearing after the last valid entry.

The distinction is important for time referenced sequences, where each type may result from different reasons and may require different handling. For instance, *left-missing-values* may occur with sequences that do not start at the same time, *right-missing-values* when sequences are of different lengths, and gaps when we missed the observation at some date. In the first case we may want to align the beginning of the sequences, preferring for instance an age or process time to a calendar time. On the other hand, right-missing-values corresponding to truncation could be simply ignored, while for gaps the treatment may depend on whether or not it is important to preserve the time alignment across sequences. These are indeed just examples, the specific treatment of each type of missing values will indeed depend of whether the analysis method we envisage to use supports missing values, and if yes which kind it supports. For instance, a missing element will have less dramatic consequences for running methods for event sequences than methods for state sequences.

Beside radical list wise deletion solutions, basic handling of missing values include:

- deleting them from the sequence, which implies shifting one position to the left all elements appearing on the right side of the missing value;
- maintaining missing values at their place, and using special treatments for them during the analysis stage;
- completing the alphabet with a "missing" term, i.e., treat missing values as if they were normal elements of the sequence;
- missing data imputation using for instance techniques for microarrays [3] or for repeated measures [8].

It is behind the scope of this chapter to discuss the handling of missing values for specific types of analysis. Nevertheless, it is important to have the distinctions made above into the mind when we want convert data into a new format.

5 Transforming between Formats

When converting from one format into the other we usually want to preserve the information. This should not be a problem when turning a state sequence format into another state sequence format, as well as when converting between event sequence formats. The transformation of states into events and vice-versa is more complicated and requires additional information from the user. This section addresses some of the issues raised by format conversions.

5.1 Converting between State Sequence Formats

Conversion between STS (state-sequence), SPS (state-permanence), SSS (state-start) and SRS (shifted-replicated) horizontal formats of state sequences is straightforward as shown from the examples in Table 3. Such conversions can easily been automatized with a few lines of code and functions doing the job are for instance proposed in our TraMineR package [4]. However, to get exactly the same information, that is to make the transformation reversible, we should store separately the start time with both SRS and duration stamped SPS formats, and either the duration or end time of the last spell with the SSS format. For instance, from STS to SRS we just repeatedly replicate and shift each sequence. The relabeling of the columns with the appropriate lag length needs no further information. For the converse, that is from SRS to STS, we just have to retain the left most aligned sequence for each case. The relabeling of the columns with time stamps requires however here the externally stored start time of each sequence.

Retained missing values are dealt with as other state values and need no special attention. Conversion can be done as well when there are dropped out missing values. In case of dropped out gaps, however, the lags used in SRS and the duration

Table 3 Sequence data representations: Examples (Code explanation in Table 2)

Code	Example
STS	*Id* 18 19 20 21 22 23 24 25 26 27 101 S S S M M MC MC MC MC D 102 S S S MC MC MC MC MC MC MC
SPS	*Id* 1 2 3 4 101 (S,3) (M,2) (MC,4) (D,1) 102 (S,3) (MC,7)
SSS	*Id* 1 2 3 4 101 (S,18) (M,21) (MC,23) (D,27) 102 (S,18) (MC,21)
SRS	*Id* $t-9$ $t-8$ $t-7$ $t-6$ $t-5$ $t-4$ $t-3$ $t-2$ $t-1$ t 101 S S S M M MC MC MC MC D 101 . S S S M M MC MC MC MC MC 101 . . S S S M M MC MC MC ⋮ 101 . . . . S S 102 S S S MC MC MC MC MC MC MC 102 . S S S MC MC MC MC MC MC ⋮
DSS	*Id* 1 2 3 4 101 S M MC D 102 S MC
SPELL	*Id* *Index* *From* *To* *State* 101 1 18 20 Single (S) 101 2 21 22 Married (M) 101 3 23 26 Married w Children (MC) 101 4 27 27 Divorced (D) 102 1 18 20 Single (S) 102 2 21 27 Married w Children (MC)
PPER	*Id* *Index* *Age* *State* 101 1 18 Single (S) 101 2 19 Single (S) 101 3 20 Single (S) 101 4 21 Married (M) ⋮ ⋮ ⋮ 101 10 27 Divorced (D) 102 1 18 Single (S) ⋮ ⋮ ⋮
FCE	*Id* *#marr.* 1*st marr.* 2*nd marr.* ⋯ *#child.* 1*st child* 2*nd child* ⋯ 101 1 21 . . 2 23 26 . 102 1 21 . . 1 21 . .
HTSE	*Id* 1 2 3 ⋯ 101 (marriage, 21) (childbirth, 23) (childbirth, 26) (divorce, 27) 102 (marriage, 21) (childbirth, 21)
TSE	*Id* *Time* *Event* 101 21 Marriage 101 23 Childbirth 101 26 Childbirth 101 27 Divorce 102 21 Marriage 102 21 Childbirth

stamps in SPS will lose their meaning. To be able to restore true time or duration stamps requires then to hold somewhere the original positions of these dropped out missing values. Information about left and right missing values is less important, except for the true observation start time when the duration since this start time matters.

Table 4 Feasible automatic and semi-automatic conversions

From/To	STS	SPS	SRS	PPER	SPELL	DSS	TSE	HTSE	FCE
STS	.	A	A	A	A	A	A/U	A/U	A/U
SPS	A	.	A	A	A	A	A/U	A/U	A/U
SRS	A	A	.	A	A	A	A/U	A/U	A/U
PPER	A	A	A	.	A	A	A/U	A/U	A/U
SPELL	A	A	A	A	.	A	A/U	A/U	A/U
DSS	N	N	N	N	N	.	N	N	N
TSE	A/U	A/U	A/U	A/U	A/U	?	.	A	A
HTSE	A/U	A/U	A/U	A/U	A/U	?	A	.	A
FCE	A/U	A/U	A/U	A/U	A/U	?	A	A	.

A: automatic, U: needs user intervention, N: not possible
A/U: automatic under some conditions, otherwise needs user intervention.

Transforming from an horizontal to a vertical format is almost as straightforward. PPER and SPELL are the vertical equivalents of respectively the STS and SPS forms. Be aware, however, that left and right missing values are typically ignored in vertical formats, i.e. they are dropped out. Hence, when we want explicitly account for the existence of such left and right missing values, the information should be held separately.

The distinct-states format DSS is indeed the SPS form without the time stamp information. Hence it can automatically be obtained from any of STS, SPS, SRS, PPER or SPELL format. The transformation is clearly not reversible however since DSS holds no time stamped information.

5.2 Converting between Event Sequence Formats

Conversion between event sequence formats, that is between FCE, HTSE and TSE can also be automatized. The FCE (fixed-column-event) form requires either to determine in advance the maximal number of each kind of events known by each subject (number of marriages, number of childbirths, etc.), or to be able to insert columns when necessary. Conversion to HTSE and TSE has no such requirement and is therefore easier to implement.

In any of the event sequence formats, it may be useful, especially for a later transformation into a state sequence format, to consider a “start of observation” event together with the original state of the subject at this start time.

Table 5 Example of a transition-definition matrix for state sequence (1)

From\To	*S*	*M*	*MC*	*D*
S	*starts as S*	Marriage	Marriage, Childbirth	Marriage, Divorce
M	*impossible*	*starts as M*	Childbirth	Divorce
MC	*impossible*	Child leaving	*starts as MC*	Divorce, Child leaving
D	*impossible*	Marriage	Marriage, Childbirth	*starts as D*

5.3 Conversion from State to Event Sequences

Transforming state data into event data as well as the converse, that is event data into state data is less straightforward. It is easy to derive a sequence of transitions between states from a state sequence and reciprocally states from a the transitions between them. Transitions, however, are not equivalent to events. Let us first clarify the distinction between them.

We define a *transition* as the change between two consecutive states in the sequence. This definition holds whether or not the sequence includes time information. An *event*, on the other hand, is something that happens at a given time point and hence makes sense for chronological sequences only. Though a transition in a chronological sequence can be considered as an event, the two concepts are not equivalent. The event 'Marriage' for instance characterizes the transition $S \rightarrow M$, but participates also to the characterization of the transition $S \rightarrow MC$, that is'from 'single' to 'married with a child'. Assuming we have yearly data, the latter transition results when both events 'Marriage' and 'Childbirth' happen the same year, and hence requires that the event 'Marriage' occurs. This example illustrates also that a transition may reflect the conjunction of several events. Another example is the transition $S \rightarrow D$ (single to divorced) which makes only sense when divorce follows a marriage within the same year.

Converting state sequences into event sequences requires to specify the mapping between transitions and events, that is to specify the events that must necessarily happen to generate each given transition. This can be formalized by a *transition-definition matrix* where we give the 'from' states as row labels, the 'to' states as column labels, and list the joint events that characterize each transition in the corresponding cell.

Table 5 shows how this matrix could look out for the transitions between the states considered in sequence example (1). We would expect also the conversion process to account for the state in which the sequence starts. This requires to assign one of the events 'starts as *S*', 'starts as *M*', 'starts as *MC*' or 'starts as D' to the beginning of the sequence. For convenience, we could just insert these start events on the otherwise unused diagonal of the matrix. Using this transition-definition matrix we can then automatically convert state sequences by replacing all encountered transitions by the associated events and stamping them with the time at which the transition occurs. A convention must however be adopted for the time stamp depending on whether we assume the state reported for time unit t, say year t, is the

Table 6 Transition-definition matrix for state sequence (1) generated by the 'Transition' method

From\To	S	M	MC	D
S	starts as S	$S \to M$	$S \to MC$	$S \to D$
M	$M \to S$	starts as M	$M \to MC$	$M \to D$
MC	$MC \to S$	$MC \to M$	starts as MC	$MC \to D$
D	$D \to S$	$D \to M$	$D \to MC$	starts as D

Table 7 Transition-definition matrix for state sequence (1) generated by the 'End-Begin' method

From\To	S	M	MC	D
S	bgn_S	end_S, bgn_M	end_S, bgn_MC	end_S, bgn_D
M	end_M, bgn_S	bgn_M	end_M, bgn_MC	end_M, bgn_D
MC	end_MC, bgn_S	end_MC, bgn_M	bgn_MC	end_MC, bgn_D
D	end_D, bgn_S	end_D, bgn_M	end_D, bgn_MC	bgn_D

state at the beginning of this observed time unit or at the end of it. In the first case, we would stamp events with the time of the 'from' state of the transition and otherwise with that of the 'to' state. Adopting this latter convention for converting our example sequence (1) we get the following event time stamped sequence

$$\big((\text{starts as } S, 18)\ (\text{marriage}, 21)\ (\text{childbirth}, 23)\ (\text{divorce}, 27)\big)\ . \tag{3}$$

Notice that this sequence differs from that of example (2), which mentions an additional childbirth that cannot be accounted for with the four sole states considered. A state 'MC2', married with two children, would be necessary for that. This illustrates the tight relationships that should exist between states and events when we want to get state and event representations holding exactly the same information.

The designing of the transition-definition matrix belongs to the user, which prevents the conversion process to be fully automatized. As shown by the above small example, it may also be an awkward task. It may therefore be useful to be able to automatically generate some rough transition-definition matrix. Even when not completely relevant, such a matrix could serve as a start point for the design process. It could also be used as is for applying quickly tentatively methods of event sequence analyses on data presented in state sequence formats. We propose hereafter two such rough automatic methods.

A first method that we call 'Transition', consists in considering each observed transition as a simple event. The transition-definition matrix generated from our example sequence is given in Table 6.

The second method named 'End-Begin' assigns two events to each transition, namely the end of the 'from' state and the beginning of the 'to' state. We use the

prefixes 'end_' and 'bgn_' to denote these events in Table 7 that reports the matrix obtained this way from our example.

The diagonal terms of the matrices should indeed not be interpreted as the other entries. They do not stand for the transition of the corresponding state to itself, but indicate the event that initiates sequences starting in the corresponding column state. Remember also that the automatically generated transition-definition matrices are just rough solutions that will most often require adjustments to suite the user's research objectives.

5.4 Conversion from Event to State Sequences

The reverse conversion from event to state sequences needs again a definition of transitions from the events. However, we face now a different problem, the goal being to find the a priori unknown states resulting from the successive known events. To do that, we assume that a state transition can only occur when an event happens and that the state at each time position t depends uniquely of the events that occurred before t, including indeed the sequence initiating event. Under these hypotheses the successive states can be determined recursively from the event sequence. The first state is defined by the initiating event, which means that we have to know the starting state of the sequence. This state is then replicated for each time until the time at which the next event happens. At this point we switch to the new state caused by the event. We then repeat the process until we get the state generated by the last event. When necessary, we can repeat the last state until a fixed sequence end time.

The only difficulty in implementing the process is the determination of the state in which we fall after each event. Note that, as in a Markov chain, the ancestor state at t summarizes all the information we need from the sequence of previous events. The new state can be determined from the joint knowledge of the event and this previous state. Thus, what we need is a *state-definition* matrix giving the resulting new state for each (previous state, event) couple. Table 8 shows one possible matrix for the events considered in sequences (2) and (3). To keep the example small, the design of this matrix assumes that we are only interested in the four states S, M, MC and D, i.e., we are not interested to distinguish among single (S) or divorced (D) people those who live with children from those who live without children, that is the 'child' distinction is supposed relevant only for married people.

Once we have defined the state-definition matrix, we can proceed with converting the event sequence into a state sequence. At each event we switch to the state found

Table 8 Example of a state-definition matrix for event sequences (2) and (3)

From\Event	Marriage	Childbirth	Divorce
S	*M*	*S*	*impossible*
M	*impossible*	*MC*	*D*
D	*M*	*D*	*impossible*
MC	*impossible*	*MC*	*D*

Table 9 State-definition matrix for generated from the events in sequence (2) and (3)

From\Event	Marriage	Childbirth	Divorce
none	{marr}	{child}	{div}
{marr}	{marr}	{marr,child}	{marr,div}
{child}	{marr,child}	{child}	{child,div}
{div}	{marr,div}	{child,div}	{div}
{marr,child}	{marr,child}	{marr,child}	{marr,child,div}
{marr,div}	{marr,div}	{marr,div,child}	{marr,div}
{child,div}	{marr,child,div}	{child,div}	{child,div}
{marr,child,div}	{marr,child,div}	{marr,child,div}	{marr,child,div}

at the intersection of the row corresponding to the ancestor state and the column associated with the event.

As for the state to event conversion, we may imagine some automatic process for generating a basic state-definition matrix. One possibility consists in defining the state by the set of experienced events without accounting for their order, i.e., by associating a state to each possible combination of events. For c different events we would thus generate 2^c possible states, including a *none* state that remains valid as long as no event is experienced. Table 9 shows the state-definition matrix automatically generated from the three events considered in sequences (2) and (3). The matrix contains a row for each of the 2^3 combinations of the three states. In each row, we read the state in which we fall when the corresponding column event happens.

This automatically generated state-definition matrix is just a rough basic solution. It may be worth to make some adjustments before using it for making the conversion. First, it can happen that the automatic process generates some theoretically unattainable states. In our example, for instance, it makes no sense to consider states where we have divorced without getting married, which suggests to exclude the states {div} and {child,div}. Maintaining them would nevertheless have no consequences, since we should never fall in such unattainable states. Secondly, the number of automatically generated states, which raises exponentially with the number of events, may become too large for an efficient state sequence analysis. For example, with $c = 5$ events we get 32 states, and $c = 10$ leads to 1024 states. The user may then want to reduce the number of states by selecting only the more relevant of them. A possible empirical solution — or at least an empirical guide line — could be here to consider only states that exceed some threshold frequency for the whole sequence data set. This would indeed also exclude unfeasible states.

An important limitation of the just described method is that a new state can only be obtained by augmenting the set of events that defines the ancestor state. This precludes any return to a previously visited state. We can overcome this limitation by combining the process with an event dropping out mechanism. We define such a mechanism by means of a binary $c \times c$ *event-drop-out-matrix* in which a 1 is set in cell (i, j) to indicate that event i should be dropped out when event j happens. For our example, we could define the matrices shown in Table 10. The left side matrix states that a divorce cancels a previous marriage, but also that any previous divorce

Table 10 Two possible event-drop-out-matrices

Element to drop out	Occurring event		
	Marriage	Childbirth	Divorce
marr	0	0	1
child	0	0	0
div	1	0	0

Element to drop out	Occurring event		
	Marriage	Childbirth	Divorce
marr	0	0	1
child	0	0	1
div	1	0	0

Table 11 State-definition matrix generated with a drop-out mechanism

From\Event	Marriage	Childbirth	Divorce
none	{marr}	{child}	{div}
{marr}	{marr}	{marr,child}	{div}
{child}	{marr,child}	{child}	{div}
{div}	{marr}	{child,div}	{div}
{marr,child}	{marr,child}	{marr,child}	{div}
{child,div}	{marr,child}	{child,div}	{div}

Table 12 Cancel-event-matrix for preventing transitions after childbirths for non married people

Element of state definition	Canceled event		
	Marriage	Childbirth	Divorce
none	0	1	0
marr	0	0	0
child	0	0	0
div	0	1	0

will be ignored after a new marriage. In the right side matrix, we state in addition that we should forget about any preceding childbirth when a divorce happens.

Using the right side matrix for the drop out mechanism in the automatic design of the state-definition-matrix, we get Table 11. This matrix differs from the one we defined by hand in Table 8 in that it defines specific states for people that have a child while they are not married, namely states {child} and {child,div}.

Similarly to the event-drop-out-mechanism, we can implement a 'cancel event effect' mechanism that would prevent any transition after events occurring in specified states. This requires to specify a binary $(c+1) \times c$ *cancel-event-matrix* in which a 1 in cell (i, j) indicates that event j should be ignored when it occurs while we are in any state containing event i. Considering again our example, we would define the matrix as shown in Table 12 for ignoring childbirths experienced by unmarried people. Notice that we have here the *none* row for accounting for the state that prevails while no event occurs.

Generating the state-definition matrix with both the drop-out and cancel-event-effect mechanisms we get Table 13. If we relabel $S = none$, $M =$ marr, $C =$ child

and $D = div$, this matrix appears to be equivalent to our first state-definition matrix of Table 8 except for the cells labeled as *impossible*. The latter have, however, no importance since they correspond to states that will never be reached.

The last solutions proposed are not wholly automatic since they require the user to specify the event-drop-out and cancel-event matrices. In our experiences, the specification of these matrices was however much simpler than the complete design of the state-definition matrix and proved thus to be a valuable help in the conversion process.

Table 13 State-definition matrix generated with drop-out and cancel-event mechanisms

From\Event	Marriage	Childbirth	Divorce
none	{marr}	*none*	{div}
{marr}	{marr}	{marr,child}	{div}
{div}	{marr}	{div}	{div}
{marr,child}	{marr,child}	{marr,child}	{div}

6 Implementation in the R Environment

Most of the sequence formats discussed in this chapter are already supported by our TraMineR package for rendering, mining and analysing sequence data in R [4]. The package offers functions that do the automatic conversion between either state sequence formats or between event sequence formats, as well as the conversion between state and event sequences from a user provided definition matrix. The building of rough basic transition-definition and state-definition matrices were also implemented in the latest — currently in testing stage — release of the package.

TraMineR uses state-sequence objects and event-sequence objects. The former store the data internally in STS form and the latter in TSE form. To avoid the multiplication of the conversion procedures, the conversion between any two state sequence formats, say SPS to SSS, is done by converting first to the internal STS and then to the destination format. Likewise, the conversion between formats of event sequences is done by passing through the TSE form. This remains indeed transparent for the user. The transformation between state and event sequences is implemented for a conversion between the default STS and TSE forms. As with other functions in TraMineR, a different input format can however be specified, in which case an automatic conversion into the default format is applied before the state-event transformation. Similarly, an option allows to further transform the output in any supported output format.

7 Conclusion

The aim of this chapter was to respond to an obvious lack in the literature of a general reference for all questions regarding the preparation of categorical sequence

data. The comprehensive overview of the different ways of organizing discrete sequence data and of the possibilities to pass from one presentation to the other one makes this chapter unique. The overview was built on our experiences in the analysis of life trajectory data. The chapter presents thus also original data transformation solutions such as those adopted for converting state data into event sequences. The material assembled here should undoubtedly help others in preparing data for sequence analysis. At least it corresponds to what we would have liked to find when we started to work with sequence data.

The data organization strongly depends indeed on the nature of the sequence and it is therefore important to identify the kind of sequence data at hand. We have seen that an important distinction that should be done is between chronological sequences and sequences without time content. Behind positions and sequence lengths, the former hold indeed time information that we should care to preserve when manipulating and converting sequences. Then, for time stamped sequences, a second important distinction is between state and event sequence data. The conversion from one of these types into the other one may be awkward and the solutions proposed here constitute perhaps the most original part of the chapter. Now, though the overview looks complete we can imagine some further developments, for instance regarding the automatic detection of the data organization or in the designing of additional solutions for automatizing the conversion between states and events.

Acknowledgements. This work was realized within a research project entitled "Mining event histories: Towards new insights on personal Swiss life courses". The authors are grateful to the Swiss National Foundation for scientific research who supported this project under grant SNF-100012-113998.

References

1. Aassve, A., Billari, F., Piccarreta, R.: Strings of adulthood: A sequence analysis of young British women's work-family trajectories. European Journal of Population 23(3), 369–388 (2007)
2. Blossfeld, H.P., Golsch, K., Rohwer, G.: Event History Analysis with Stata. Lawrence Erlbaum, Mahwah (2007)
3. Brock, G.N., Shaffer, J.R., Blakesley, R.E., Lotz, M.J., Tseng, G.C.: Which missing value imputation method to use in expression profiles: A comparative study and two selection schemes. BMC Bioinformatics 9, 12 (2008)
4. Gabadinho, A., Ritschard, G., Studer, M., Müller, N.S.: Mining sequence data in R with TraMineR: A user's guide for version 1.1. Technical report, Department of Econometrics and Laboratory of Demography, University of Geneva, Geneva (2009), http://mephisto.unige.ch/traminer
5. Gauthier, J.A., Widmer, E.D., Bucher, P., Notredame, C.: Multichannel sequence analysis applied to social science data, University of Lausanne (2007) (manuscript) (under review)
6. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Transactions on Asian Language Information Processing 3(1), 66–85 (2004)

7. Karweit, N., Kertzer, D.: Data organization and conceptualization. In: Giele, J.Z., Elder, G.H. (eds.) Methods of Life Course Research: Qualitative and Quantitative Approaches, pp. 81–97. Sage, Thousand Oaks (1998)
8. Little, R.J.A.: Modeling the drop-out mechanism in repeated-measures studies. Journal of the American Statistical Association 90(431), 1112–1121 (1995), http://www.jstor.org/stable/2291350
9. Ritschard, G., Oris, M.: Life course data in demography and social sciences: Statistical and data mining approaches. In: Levy, R., Ghisletta, P., Le Goff, J.M., Spini, D., Widmer, E. (eds.) Towards an Interdisciplinary Perspective on the Life Course, Advances in Life Course Research, vol. 10, pp. 289–320. Elsevier, Amsterdam (2005)
10. Yamaguchi, K.: Event history analysis. In: ASRM 28. Sage, Newbury Park (1991)
11. Zaki, M.J.: SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42(1/2), 31–60 (2001)

Considerations on Logical Calculi for Dealing with Knowledge in Data Mining*

Jan Rauch

Summary. An attempt to develop and apply logical calculi in exploratory data analysis was made 30 years ago. It resulted in a definition and study of observational logical calculi based on modifications of classical predicate calculi and on mathematical statistics. Additional results followed the definition and first implementations of the GUHA method of mechanizing hypothesis formation. The GUHA method can be seen as one of the first data mining methods. Applications of modern and enhanced implementation of the GUHA method confirmed the generally accepted need to use domain knowledge in the process of data mining. Moreover it inspired considerations on the application of logical calculi for dealing with domain knowledge in data mining. This paper presents these considerations.

1 Introduction

This paper is inspired by both results related to observational and theoretical calculi [5, 13] and experience with the application of the GUHA procedures implemented in the LISp-Miner system [16, 20]. Observational and theoretical calculi are defined in [5] as tools to answer the questions:

- Can computers formulate and verify scientific hypotheses?
- Can computers analyze empirical data in a rational way and produce a reasonable reflection of the observed empirical world? Can this be done using mathematical logic and statistics?

An additional result presented in [5] is a formal definition of the GUHA method of the mechanizing hypothesis formation, which aims to offer all interesting facts arising from the analyzed data to a given problem. The method is

Jan Rauch
Faculty of Informatics and Statistics, University of Economics, Prague
nám. W. Churchilla 4, 130 67 Prague 3, Czech Republic
e-mail: rauch@vse.cz

* The work described here has been supported by Grant No. 201/08/0802 of the Czech Science Foundation and by Grant No. ME913 of Ministry of Education, Youth and Sports, of the Czech Republic.

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 177–199.
springerlink.com

realized using GUHA-procedures. Procedure input consists of analyzed data and a simple definition of a usually large set of relevant (i.e. potentially interesting) patterns. The procedure automatically generates each particular pattern and tests if it is true in the analyzed data. Procedure output consists of all prime patterns. The pattern is prime if it is true in the analyzed data and does not immediately follow from other more simple output patterns [5]. Several GUHA procedures were implemented, see e.g. [6, 7, 8, 9, 11]. The LISp-Miner system contains six GUHA procedures [16, 17, 23] that were applied to solve various data mining tasks. The most frequently used GUHA procedure is the ASSOC procedure, which mines for patterns that are a generalization of association rules widely defined and studied in the field of data mining. In addition, the most important results on observational calculi can be understood as the slogic of association rules.

The GUHA procedures implemented in the LISp-Miner system have new features that make it possible to fine-tune the set of relevant patterns in a precise way so that some aspects of semantics can be included. This led to the implementation of additional tools for maintaining various items of domain knowledge related to the analyzed data. Stored items of domain knowledge are used in both the process of data mining and the presentation of results.

Analytical questions are understood here as the core of the data mining process. The data mining process starts with the formulation of an analytical question interesting from the user's point of view and the process is completed with a well written analytical report answering the given analytical question. The possibilities of GUHA procedures to deal with some aspects of semantics, the results concerning observational calculi and domain knowledge stored in the LISp-Miner system play an important role in dealing with analytical questions and reports. Current experience concerning such analytical questions and reports leads to considerations on logical calculi for dealing with knowledge in data mining. The goal of this paper is to present these considerations. Note that the presented approach differs from the approach based on Inductive Logic Programming, e.g. [1].

The main features of the GUHA method and LISp-Miner system are introduced in Section 2. Observational calculi and logic of association rules are summarized in Section 3. Domain knowledge stored in the LISp-Miner system is described in Section 4. The possibilities of using stored domain knowledge to formulate interesting analytical questions and solve them using GUHA procedures implemented in LISp-Miner system are discussed in Section 5. A very important step in answering a given analytical question is to filter out the consequences of stored items of domain knowledge. The logic of observational calculi plays an important role in this step, see Section 6. Resulting analytical reports are discussed in Section 7. The considerations on logical calculi for dealing with analytical questions and reports in data mining are summarized in Section 8. Concluding remarks and a description of further work are included in Section 9.

2 GUHA Procedures and the LISp-Miner System

There are six GUHA procedures implemented in the LISp-Miner system. All of them deal with data matrices that are the results of the transformation of input data tables stored in the analyzed database. Transformations are realized by the LMDataSource module, which is an inherent part of the LISP-Miner system. These transformations are important from the point of view of our considerations and they are related to both various properties of analyzed data and the given analytical question. The details of analyzed data are included in Section 2.1.

The most frequently used GUHA procedure is the 4ft-Miner procedure, which is the enhanced ASSOC procedure defined in [5]. It mines for patterns that can be understood as the generalized association rule $X \rightarrow Y$ where X and Y are sets of items. The intuitive meaning of the $X \rightarrow Y$ rule is that transactions (e.g. supermarket baskets) containing a set X of items tend to contain a set Y of items [2].

The 4ft-Miner procedure deals with association rules of the form $\varphi \approx \psi$ where φ and ψ are Boolean attributes derived from columns of the analyzed data matrix. The meaning of the rule $\varphi \approx \psi$ is that the Boolean attributes φ and ψ are associated in a way corresponding to the symbol $\approx$. The symbol $\approx$ is called the *4ft-quantifier*. It defines the relation of φ and ψ using the contingency table of φ and ψ. Both various simple relations and relations corresponding to the statistical hypothesis tests can be used.

The procedure 4ft-Miner also deals with conditional association rules of the form $\varphi \approx \psi/\chi$ where φ, ψ, and χ are Boolean attributes derived from columns of the analyzed data matrix. The meaning of the pattern $\varphi \approx \psi/\chi$ is that the Boolean attributes φ and ψ are associated in a way corresponding to the symbol $\approx$ when a condition given by the Boolean attribute χ is satisfied.

Basic information on the 4ft-Miner procedure and association rules is presented in section 2.2. Procedure input consists of analyzed data, a specification of the 4ft-quantifier $\approx$ and a definition of the set of association rules to be generated and verified. It is important that there are very fine tools to define this set. They make it possible to involve various aspects of semantics in the definition. Some details are shown in Section 2.3. Other GUHA procedures implemented in the LISp-Miner system are briefly outlined in section 2.4.

2.1 Analyzed Data

Input for all GUHA procedures implemented in the LISp-Miner system is the data matrix $\mathcal{M}$; see Fig. 1. The data matrix $\mathcal{M}$ has n rows corresponding to the observed objects $o_1, \ldots, o_n$. It also has K columns corresponding to the attributes $A_1, \ldots, A_K$ describing particular objects. The value of the attribute A_j for the object o_i is denoted as $v_{i,j}$. We assume the set of possible

$\mathcal{M}$	A_1	A_2	...	A_K
o_1	$v_{1,1}$	$v_{1,2}$	...	$v_{1,K}$
⋮	⋮	⋮	⋮ ⋮ ⋮	⋮
o_n	$v_{n,1}$	$v_{n,2}$	...	$v_{n,K}$

$\mathcal{T}$	C_1	C_2	...	C_K
o_1	$c_{1,1}$	$c_{1,2}$	...	$c_{1,K}$
⋮	⋮	⋮	⋮ ⋮ ⋮	⋮
o_n	$c_{n,1}$	$c_{n,2}$	...	$c_{n,K}$

Fig. 1 Data Matrix $\mathcal{M}$ and Original Database Table $\mathcal{T}$

values of the attribute A_i is $\{a_1^{(i)}, \ldots, a_{t_i}^{(i)}\}$. The values $a_1^{(i)}, \ldots, a_{t_i}^{(i)}$ are called *categories* of the attribute A_i.

The data matrix $\mathcal{M}$ is a result of the transformation of the database table $\mathcal{T}$. The database table $\mathcal{T}$ is also understood to be a data matrix with n rows corresponding to observed objects $o_1, \ldots, o_n$. It also has K columns $C_1, \ldots, C_K$ corresponding to particular attributes. The column C_i is transformed into the attribute A_i, $i = 1, \ldots, K$. The core of the transformation of column C_i into the attribute A_i is the definition of suitable subsets of the set of all possible values of C_i.

The particular subsets define the categories $a_1^{(i)}, \ldots, a_{t_i}^{(i)}$ of the attribute A_i. We must define t_i mutually disjoint subsets $\gamma_1, \ldots, \gamma_{t_i}$; the subset γ_j defines the category $a_j^{(i)}$ for $j = 1, \ldots t_i$. The term $A_i(o)$ below denotes the value of the attribute A_i for the object o. This means that $A_i(o_k) = v_{i,k}$ for $j = 1, \ldots, n$ and $k = 1, \ldots, K$. Similarly, $C_i(o)$ denotes the value of the column C_i for the object o and $C_i(o_k) = c_{i,k}$. The categories are defined so that

$$A_i(o) = a_j^{(i)} \text{ if and only if } C_i(o) \in \gamma_j \text{ for } j = 1, \ldots, t_i \ .$$

There are various ways of defining the subsets $\gamma_1, \ldots, \gamma_t$ in the LISp-Miner system [23]. Some aspects of semantics can also be included, e.g. interval boundaries of original body mass index values that define standard levels of obesity. Note that the union $\bigcup_{j=1}^{t_i} \gamma_j$ need not to cover the whole set of values of the attribute C.

2.2 Association Rules and 4ft-Miner Procedure

The 4ft-Miner procedure mines for association rules of the form $\varphi \approx \psi$ where φ and ψ are Boolean attributes derived from attributes – columns of the analyzed data matrix $\mathcal{M}$. *Basic Boolean attributes* are created first. The basic Boolean attribute is an expression of the form $A(\alpha)$ where $\alpha \subset \{a_1, \ldots a_t\}$ and $\{a_1, \ldots a_t\}$ is the set of all categories of the attribute A. The basic Boolean attribute $A(\alpha)$ is true in row o of $\mathcal{M}$ if it is $A(o) \in \alpha$ where $A(o)$ is the value of the attribute A in row o. Boolean attributes φ and ψ are derived from basic Boolean attributes using connectives $\vee$, $\wedge$ and $\neg$ in the usual way.

The meaning of the rule $\varphi \approx \psi$ is that the Boolean attributes φ and ψ are associated in a way corresponding to the 4ft-quantifier $\approx$. The 4ft-quantifier $\approx$ defines the relation of φ and ψ using a four-fold contingency table of φ and

Table 1 4ft Table $4ft(\varphi, \psi, \mathcal{M})$ of φ and ψ in $\mathcal{M}$

$\mathcal{M}$	ψ	$\neg\psi$
φ	a	b
$\neg\varphi$	c	d

ψ, see Tab. 1. The four-fold contingency table of φ and ψ in the data matrix $\mathcal{M}$ is the quadruple $\langle a, b, c, d\rangle$ of natural numbers, where a is the number of rows of $\mathcal{M}$ satisfying both φ and ψ, b is the number of rows of $\mathcal{M}$ satisfying φ and not satisfying ψ, etc. The four-fold contingency table (the *4ft table*) of φ and ψ in $\mathcal{M}$ is denoted by $4ft(\varphi, \psi, \mathcal{M})$.

There are 16 *basic 4ft quantifiers* implemented in the 4ft-Miner procedure. Four important examples of basic 4ft-quantifiers follow. All of them were defined in relation to the GUHA method, however most of them were defined sooner or later in additional sources; see e.g. [4] and used as measures of the interestingness of association rules. An overview of various measures of the interestingness of association rules and their relation to 4ft-quantifiers is included in [19].

The quantifier $\Rightarrow_p$ of *founded implication* is defined for $0 < p \leq 1$ in [5] by the condition $\frac{a}{a+b} \geq p$. The association rule $\varphi \Rightarrow_p \psi$ means that at least $100p$ per cent of the rows of $\mathcal{M}$ satisfying φ also satisfy ψ. The ratio $\frac{a}{a+b}$ defines a measure of interestingness called *confidence* [2].

The 4ft-quantifier $\Leftrightarrow_p$ of *founded double implication* is defined for $0 < p \leq 1$ in [8] by the condition $\frac{a}{a+b+c} \geq p$. The association rule $\varphi \Leftrightarrow_p \psi$ means that at least $100p$ per cent of the rows of $\mathcal{M}$ satisfying φ or ψ satisfy both φ and ψ. The ratio $\frac{a}{a+b+c}$ defines a measure of interestingness called *Jaccard* [4].

The 4ft-quantifier $\equiv_p$ of *founded equivalence* is defined for $0 < p \leq 1$ in [8] by the condition $\frac{a+d}{a+b+cd} \geq p$. The association rule $\varphi \equiv_p \psi$ means that φ and ψ have the same value (either *true* or *false*) for at least $100p$ per cent of all rows of $\mathcal{M}$. The ratio $\frac{a+d}{a+b+c+d}$ defines a measure of interestingness called *accuracy* [4] or *success rate* [3].

The 4ft-quantifier $\Rightarrow_q^+$ of *above average dependence* also called the *AA quantifier* is defined for $0 < q$ in [16] by the condition $\frac{a}{a+b} \geq (1+q)\frac{a+c}{a+b+c+d}$. The association rule $\varphi \Rightarrow_q^+ \psi$ means that among the objects satisfying φ there are at least $100p$ per cent more objects satisfying ψ than among all observed objects.

The 4ft-quantifier $\odot_{Base}$ called *Base* is defined for integer $0 < Base$ in [5] by the condition $a \geq Base$. The association rule $\varphi \odot_{Base} \psi$ means that there are at least $Base$ rows of $\mathcal{M}$ satisfying both φ and ψ.

The 4ft quantifier can be defined as any conjunction of basic 4ft quantifiers. Note that the conjunction of $\Rightarrow_p$ and $\odot_{Base}$ is traditionally denoted as $\Rightarrow_{p,Base}$ and that the quantifier $\Rightarrow_{p,Base}$ is defined in [5] as the 4ft quantifier of founded implication. This is similar for the quantifiers $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^+_{p,Base}$.

The 4ft-Miner procedure also mines for *conditional association rules* of the form $\varphi \approx \psi/\chi$ where φ, ψ and χ are Boolean attributes. The intuitive meaning of $\varphi \approx \psi/\chi$ is that φ and ψ are in the relation given by the 4ft-quantifier $\approx$ when the condition χ is satisfied, for details see [16].

2.3 Defining the Set of Relevant Boolean Attributes

Input for the 4ft-Miner procedure consists of analyzed data and of a definition of the set of relevant association rules $\varphi \approx \psi$ or of the set of relevant conditional association rules $\varphi \approx \psi/\chi$ to be generated and tested. The definition of the set of relevant association rules is given by:

- a definition of a *set of relevant antecedents*, which we denote Φ
- a definition of a *set of relevant succedents*, which we denote Ψ
- if we are interested in conditional association rules, then we also include a definition of a *set of relevant conditions*, which we denote X
- a definition of the 4ft-quantifier $\approx$.

The set of relevant antecedents, the set of relevant succedents and the set of relevant conditions are defined in a same way. We describe in more details the definition of the set of relevant antecedents. Each antecedent φ is a conjunction $\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_k$ where $\varphi_1, \varphi_2, \ldots, \varphi_k$ are *partial antecedents*. Each partial antecedent is either a conjunction of literals or a disjunction of literals. A literal is a basic Boolean attribute $A(\alpha)$ or a negation $\neg A(\alpha)$ of a basic Boolean attribute. The definition of the set Φ of relevant antecedents is given by a definition of the *sets* $\Phi_1, \ldots, \Phi_k$ *of relevant partial antecedents*, $k \geq 1$ and by minimal and maximal numbers of literals in antecedents. Each set of partial antecedents is defined by:

- *the type of partial antecedent*, either *Conjunction* or *Disjunction*
- *the minimum number of literals* and *the maximum number of literals*, i.e. $0 \leq$ *minimum number of literals* $\leq$ *maximum number of literals*
- a set of attributes from which literals will be generated
- a simple definition of the set of all literals for each given attribute
- additional details - *basic attributes* and *classes of equivalence* see [16].

Note that the minimum length of the partial antecedent can be 0 and this results in an empty partial antecedent. The value of the empty partial antecedent is identically *true*, the empty partial antecedent is not considered to be a literal and in this way the number of literals in an antecedent can become smaller than k, see above.

A literal is a basic Boolean attribute $A(\alpha)$ or its negation $\neg A(\alpha)$. The set α is called a *coefficient of the basic Boolean attribute* $A(\alpha)$ or a *coefficient of the literal*. The *length of the literal* is the number of categories in its coefficient. The set of all literals to be generated for a particular attribute is given by:

- the type of coefficient; there are seven types of coefficients: *subsets*, *intervals*, *cyclic intervals*, *left cuts*, *right cuts*, *cuts*, *one particular category*
- the minimum and maximum length of the literal
- positive/negative option: (i) – generate only positive literals, (ii) – generate only negative literals (iii) – generate both positive and negative literals.

The set of relevant antecedents, the set of relevant succedents and the set of relevant conditions can overlap. However, association rules with more than one literal created from the same attribute are not generated.

We use the attribute A with categories {1, 2, 3, 4, 5} to give examples of particular types of coefficients:

- *subsets*: the definition of subsets with a length of 2-3 gives literals A(1,2), A(1,3), A(1,4), A(1,5), A(2,3), ..., A(4,5), A(1,2,3), A(1,2,4), A(1,2,5), A(2,3,4), ..., A(3,4,5)
- *intervals*: the definition of intervals with a length of 2-3 gives literals A(1,2), A(2,3), A(3,4), A(4,5), A(1,2,3), A(2,3,4) and A(3,4,5).
- *cyclic intervals*: the definition of cyclic intervals with a length of 2 gives literals A(1,2), A(2,3), A(3,4), A(4,5), and A(5,1)
- *left cuts*: the definition of left cuts with a maximum length of 3 defines literals A(1), A(1,2) and A(1,2,3)
- *right cuts*: the definition of right cuts with a maximum length of 4 defines literals A(5), A(5,4), A(5,4,3) and A(5,4,3,2)
- *cuts* means both left cuts and right cuts
- *one particular value* means one literal with one chosen category, e.g. A(2).

We should emphasize that the appropriate use of types of coefficients makes it possible to consider various aspects of semantics. For example, left cuts of the attribute A can be used to define Boolean attributes saying that the values of A are small, similarly for right cuts and high values of A.

2.4 Additional GUHA Procedures

There are five additional GUHA procedures in the LISp-Miner system [17].

The SD4ft-Miner procedure mines for SD4ft-patterns $\alpha \bowtie \beta : \varphi \approx \psi / \chi$. Such *SD4ft-patterns* mean that the subsets given by Boolean attributes α and β differ in the relation of the Boolean attributes φ and ψ when the condition given by the Boolean attribute χ is satisfied.

The KL-Miner procedure mines for KL-patterns $R \sim C / \chi$. Such *KL-patterns* mean that the category attributes R and C are in a relation given by the symbol $\sim$ when the condition given by the Boolean attribute χ is satisfied.

The SDKL-Miner procedure mines for SDKL-patterns $\alpha \bowtie \beta : R \sim C / \chi$. Such *SDKL-patterns* mean that the subsets α and β differ in what concerns the relation of the category attributes R and C when the condition given by the Boolean attribute χ is satisfied.

The CF-Miner procedure mines for CF-patterns $\sim R/\chi$. Such *CF -patterns* mean that the frequencies of categories of the attribute R satisfy the condition given by the symbol $\sim$ when an other condition given by the Boolean attribute χ is satisfied.

The SDCF-Miner procedure mines for SDCF-patterns $\alpha \bowtie \beta : \sim R/\chi$. Such *SDCF-patterns* mean that the subsets α and β differ in the frequencies of the particular categories of the attribute R when the condition given by the Boolean attribute χ is satisfied.

3 Logical Calculi of Association Rules

It is important that association rules of the form $\varphi \approx \psi$ can be considered as formulas of special logical calculi. There are both practically important and theoretically interesting results concerning logical calculi of association rules [15, 19] that can be used when dealing with domain knowledge in data mining. Logical calculi of association rules belong to observational calculi introduced in [5].

The results we are going to present are closely related to classes of association rules. Classes of association rules are defined by classes of 4ft quantifiers. The association rule $\varphi \approx \psi$ belongs to the *class of implicational association rules* if the 4ft quantifier $\approx$ belongs to the *class of implicational quantifiers*. We say that the association rule $\varphi \approx \psi$ is an *implicational rule* and that the 4ft quantifier $\approx$ is an *implicational quantifier*. This is the same for other classes of association rules.

There are various important classes of 4ft quantifiers defined by *truth preservation conditions* [5, 15, 19]. We say that class $\mathcal{C}$ of 4ft-quantifiers is defined by the truth preservation condition $TPC_{\mathcal{C}}$ if there is a Boolean condition $TPC_{\mathcal{C}}(a,b,c,d,a',b',c',d')$ concerning two four-fold contingency tables $\langle a,b,c,d\rangle$ and $\langle a',b',c',d'\rangle$ so that the following is true: *4ft quantifier* $\approx$ *belongs to class* $\mathcal{C}$ *if, and only if,* $\approx(a,b,c,d) = 1 \wedge TPC_{\mathcal{C}}(a,b,c,d,a',b',c',d')$ *implies* $\approx(a',b',c',d') = 1$ *for all 4ft tables* $\langle a,b,c,d\rangle$ *and* $\langle a',b',c',d'\rangle$. Important examples of classes of 4ft-quantifiers are shown in Tab. 2. The quantifiers $\Rightarrow_{p,Base}$, $\Leftrightarrow_{p,Base}$, and $\equiv_{p,Base}$ used in table 2 are defined in Section 2.2.

The results used in dealing with knowledge in data mining concern deduction rules between association rules. The application of deduction rules of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ where φ, ψ, φ', and ψ' are general Boolean attributes is outlined in Section 6. If the deduction rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ is correct, and if the rule $\varphi \approx \psi$ is true in the data matrix $\mathcal{M}$, then the association rule $\varphi' \approx \psi'$ is also true in the data matrix $\mathcal{M}$. It is important that known results give a simple criterion of correctness for such rules. The criterion depends on the class of 4ft-quantifier $\approx$. Note that there are criteria for all the quantifiers $\Rightarrow_{p,Base}$, $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{p,Base}$ as introduced in Section 2.2.

Below we outline the criterion for the implication quantifiers, for details see [13]. The class of *interesting implicational quantifiers* is defined first; all

Table 2 Examples of Classes of Association Rules

class	truth preservation condition		example of 4ft-quantifier
implicational	$TPC_{\Rightarrow}$	$a' \geq a \ \wedge \ b' \leq b$	$\Rightarrow_{p,Base}$
Σ-double implicational	$TPC_{\Sigma,\Leftrightarrow}$	$a' \geq a \ \wedge \ b' + c' \leq b + c$	$\Leftrightarrow_{p,Base}$
Σ-equivalency	$TPC_{\Sigma,\equiv}$	$a' + d' \geq a + d \ \wedge \ b' + c' \leq b + c$	$\equiv_{p,Base}$

the important implicational quantifiers are interesting implicational quantifiers. It is then shown that for each deduction rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ with an interesting implicational quantifier $\approx$ there are formulas Δ, Γ, Λ of propositional calculus so that the rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ is correct if, and only if, both Δ and Γ are tautologies or Λ is a tautology of propositional calculus. It is important that the formulas Δ, Γ and Λ can be easily created from the attributes φ, ψ, φ', and ψ'.

4 Domain Knowledge

A part of the LISp-Miner system is called *LM KnowledgeSource*, whose goal is to maintain various types of domain knowledge. Particular items of stored knowledge are already used or meant to be used in various ways when applying particular GUHA procedures, see e.g. [18]. The goal of this section is to present this knowledge in a form suitable for our considerations. Knowledge stored in *LM KnowledgeSource* is related to the meta-attributes of a given domain. Meta-attributes correspond to columns of database tables that are transformed into data matrices - the inputs of particular GUHA procedures. There are two types of meta-attributes.

We used examples relating to the domain of cardiology enhanced by certain sociological aspects; e.g. level of education. There are two data matrices STULONG and ADAMEK [18] that belong to this domain. An example of the first type of meta-attribute is the meta-attribute *Weight*. It is measured in kg and the categories of resulting attributes are usually defined as intervals of original values. There is no problem defining the attributes *Weight* with the same categories for both data matrices STULONG and ADAMEK. An example of the second type of meta-attribute is the meta-attribute *Education*. The attribute *Education* in the STULONG data matrix has the categories *basic*, *apprentice*, *secondary*, and *university*. The attribute *Education* in the ADAMEK data matrix has the categories *basic*, *secondary*, *higher*. It is not possible to simply convert the categories of the attribute *Education* in the STULONG data matrix into the categories of the attribute *Education* in the ADAMEK data matrix.

Various items of knowledge related to the meta-attributes of both types are stored in *LM KnowledgeSource*. We must consider two types of such items here: *groups of meta-attributes* and *mutual influence among particular meta-attributes.*

Attribute	*Age*	*Education*	*Hypertension*	*Beer*	*Wine*	*BMI*	*Obesity*	*City*	...
Age		$\otimes$	$\uparrow^+$			$\uparrow\uparrow$			...
Education						$\uparrow\downarrow$			...
Hypertension									...
Beer							$\uparrow^+$		...
Wine			$\uparrow^-$						...
BMI							$\mathcal{F}$		...
Obesity			$\rightarrow^+$						...
City					$\approx$		?		...
...	...	...	...	...	...	...	...	...	...

Fig. 2 Mutual Influence of Meta-attributes

Groups of meta-attributes: An important part of domain knowledge is given by the structure of a set of meta-attributes. Examples of groups of meta-attributes are the group *Social Characteristics* consisting of attributes such as *Marital Status*, *Education* , etc. and the group *Physical Examination* consisting of attributes such as *Weight (kg)*, *Height (cm)*, *Systolic Blood Pressure (mm Hg)*, etc.; see [18]. We assume there is a system of mutually disjoint groups of meta-attributes $BG_1, \ldots, BG_G$ so that their union covers all meta-attributes related to data matrices in the given domain. We call these groups *basic groups of attributes*. There are also additional important groups of meta-attributes. An example is the group *Cardiovascular Risk Factors* that contains attributes such as *Hypertension*, *Mother Hypertension*, *Obesity*, *Smoking* etc. coming from several basic groups [18].

Both basic groups and additional groups of meta-attributes are perceived by domain experts as reasonable sets of attributes. The information on groups of meta-attributes is used e.g. in the formulation of local analytical questions see Section 5.

Mutual influence among particular meta-attributes: An important type of knowledge stored in LM KnowledgeSource is information on mutual influence among particular meta-attributes. It is assumed that this is generally accepted knowledge. However it can also be related to particular domain experts and stored as own opinion. This knowledge is stored in the way outlined in Fig. 2, where several meta-attributes relating to the cardiology domain and the data matrices STULONG and ADAMEK are used.

There are four types of meta-attributes:

- *Boolean*; e.g. *Obesity* (i.e. patient is obese) or *Hypertension* (i.e. patient has hypertension)
- *nominal*; e.g. *City* with categories - particular cities
- *ordinal*; e.g. *Age* with categories - particular years or *Education* with a set of categories {*basic*, *apprentice*, *secondary*, *university* } for the STULONG data matrix and with a set of categories {*basic*, *secondary*, *higher*} for the ADAMEK data matrix

- *rational*, which can be represented by rational numbers in computers; e.g. *BMI* (i.e. Body Mass Index) or *Beer / Wine* (i.e. Beer / Wine consumption in liters per week)

There are several types of influences among meta-attributes, most of them are relevant to specified types of attributes. The following types of influences are used in Fig. 2:

- $\uparrow\uparrow$ – if the row meta-attribute increases then the column meta-attribute increases too, both attributes are ordinal or rational
- $\uparrow\downarrow$ – if the row meta-attribute increases then the column meta-attribute decreases, both attributes are ordinal or rational
- $\uparrow^{+}$ – if the row meta-attribute increases then the relative frequency of patients satisfying the column attribute increases, the row attribute is ordinal or rational and the column attribute is Boolean
- $\uparrow^{-}$ – if the row meta-attribute increases then the relative frequency of patients satisfying the column attribute decreases, the row attribute is ordinal or rational and the column attribute is Boolean
- $\rightarrow^{+}$ – truthfulness of the row attribute increases then relative frequency of true values of the column attribute, both attributes are Boolean.
- ? – there could be an influence, no detail is known
- $\mathcal{F}$ – means that there is a strong dependency like function, e.g. *Obesity* is equivalent to BMI ≥ 32
- $\otimes$ – there is some influence but we are not interested
- $\approx$ – there is some influence but is not yet known.

Note that there are additional dependencies and that it is also possible to describe a conditional influence where some influence between two meta-attributes is observed only if a certain condition is satisfied.

5 Analytical Questions and GUHA Procedures

5.1 Principles – Patterns of Analytical Questions

Various analytical questions can be formulated on the basis of domain knowledge stored in *LM KnowledgeSource* and answered by applications of the GUHA procedures implemented in the LISp-Miner system [18, 20]. The principle is an application of suitable "analytical question patterns" to items of domain knowledge. Analytical question patterns are formulated in such a way that the patterns produced by particular GUHA procedures can be used to answer the particular analytical questions. Note that $\mathcal{AQ}$ means analytical question below.

The input for the GUHA procedure $\mathcal{G}$ can be seen as a couple $\langle \mathcal{M}, \mathcal{D}(\mathcal{S}_{RP}) \rangle$ where $\mathcal{M}$ is the analyzed data matrix and $\mathcal{D}(\mathcal{S}_{RP})$ is a definition of the set $\mathcal{S}_{RP}$ of relevant patterns. The output $\mathcal{G}(\mathcal{M}, \mathcal{D}(\mathcal{S}_{RP}))$ of the GUHA procedure $\mathcal{G}$ is the set $\mathcal{S}_{PP}$ of all prime patterns. It is crucial that the particular GUHA

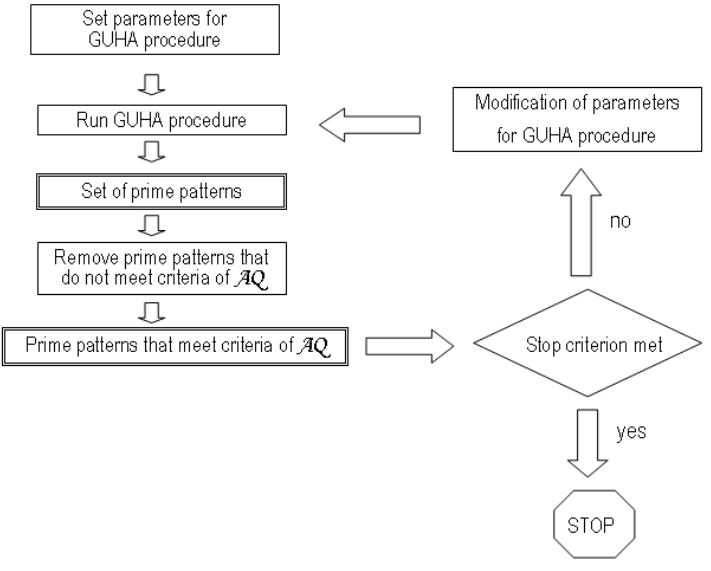


Fig. 3 The Process of Answering a Given $\mathcal{AQ}$

procedures implemented in the LISp-Miner system have broad possibilities for fine tuning the definition $\mathcal{D}(\mathcal{S}_{RP})$ of the set $\mathcal{S}_{RP}$ of relevant patterns to fit the solved $\mathcal{AQ}$ in a very precise way.

However the set $\mathcal{S}_{PP}$ of prime patterns still usually involves various uninteresting patterns. In solving the $\mathcal{AQ}$ like "Which patterns that do not follow from given items of domain knowledge can be found in the data matrix $\mathcal{M}$?" we must filter out all prime patterns that can be understood as consequences of given items of domain knowledge. To solve the $\mathcal{AQ}$ like "What known items of domain knowledge can be found in the data matrix $\mathcal{M}$?" we must filter out all prime patterns that cannot be understood as consequences of given items of domain knowledge. It can be said we must remove all prime patterns that do not satisfy criteria given by the solved $\mathcal{AQ}$. Note that our goal is to present the set of remaining patterns as the result of data mining in the form of a well structured analytical report. The process of answering a given $\mathcal{AQ}$ is usually done in several iterations, see Fig. 3.

We will discuss this process on the basis of a simple analytical question pattern "What strong unknown relations among attributes of two given groups of attributes are valid in the given data matrix $\mathcal{M}$?". We will use the data matrix ADAMEK [20] mentioned above. We apply this pattern to two groups of attributes. The first group called *Personal Characteristics* consists of the attributes *Sex*, *Age*, *Education*, *City*, *Beer*, and *Wine*. The second group called *Health Status* consists of the attributes *Hypertension*, *BMI*, and *Obesity*. Note that in Fig. 2 there are relevant items of background knowledge concerning

meta-attributes with the same names as attributes in the ADAMEK data matrix.

The first step in solving an analytical question is the selection of the GUHA procedure. Several GUHA procedures implemented in the LISp-Miner system can usually be used and their results can be combined to get the best solution. In our example we used the 4ft-Miner procedure. The main reason for this choice is that 4ft-Miner mines for association rules and that there are well known logical calculi of association rules; see section 3. The initial set up of parameters for the GUHA 4ft-Miner procedure for our task is described in Section 5.2. Removing prime patterns that do not meet $\mathcal{AQ}$ criteria is a complex step. In our case it requires the application of deduction rules in the logical calculus of association rules; see Section 6. The stop criterion is discussed in Section 5.3. Some remarks on modifications of parameters for the GUHA procedure 4ft-Miner are in Section 5.4. However it must be emphasized that a more detailed description of these problems is not within the scope of this paper. Some additional details are given in [18].

5.2 Initial Set Up of 4ft-Miner Parameters

The GUHA 4ft-Miner procedure mines for association rules $\varphi \approx \psi$ where both φ and ψ are Boolean attributes derived from the analyzed data matrix. Our task is to find strong unknown relations among attributes in the groups *Personal Characteristics* and *Health Status*. This means we will search strong unknown association rules $\varphi \approx \psi$, where φ is a Boolean attribute derived from the group *Personal Characteristics*, ψ is a Boolean attribute derived from the group *Health Status*, and $\approx$ is a 4ft-quantifier that can be easily interpreted.

The initial set up of parameters for the 4ft-Miner procedure depends on both the solved task, the general properties of attributes used and the properties of the particular analyzed data matrix. A detailed description is not within the scope of this paper. We only show one of the possibilities of how this task is done and give certain notes.

We need an easily interpretable 4ft-quantifier, thus we use the quantifiers $\Rightarrow_{p,Base}$, $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{p,Base}$. This means that four particular applications of the 4ft-Miner will be used, one for each of these quantifiers. The set of relevant antecedents will be initially defined as the set of all possible conjunctions of 1 - 4 basic Boolean attributes created from attributes in the group *Personal Characteristics* This can be done, e.g. in the following way: (Remember, for details on coefficients see Section 2.3.)

The basic Boolean attributes *Sex*(*male*) and *Sex*(*female*) will be used (coefficients *subsets* with a length of 1). Sets of basic Boolean attributes *Education*(α) and *City*(α) will also be defined as coefficients – *subsets* with a length of 1.

The attribute *Age* with categories - intervals of years $(0, 10\rangle$, $(10, 20\rangle$, ..., $(90, 100\rangle$ will be defined and coefficients *intervals* with a length of 1 - 3 will be used. This means that the basic Boolean attributes $Age(0, 10\rangle$, $Age(0, 20\rangle$, $Age(0, 30\rangle$, $Age(10, 20\rangle$, ..., $Age(70, 100\rangle$, $Age(80, 100\rangle$, and $Age(90, 100\rangle$ will be generated.

The basic Boolean attributes $Beer(\alpha)$ and $Vine(\alpha)$ will be used similarly. The categories *very low*, *low*, *medium*, *high*, and *very high* are defined and coefficients *intervals* with a length of 1 - 2 will be applied. This means we have the basic Boolean attributes *Beer*(*very low*), *Beer*(*very low*, *low*), *Beer*(*low*), *Beer*(*low*, *medium*), *Beer*(*medium*), *Beer*(*medium*, *high*), *Beer*(*high*), *Beer*(*high*, *very high*), and *Beer*(*very high*) and similarly for *Wine*(α).

The set of relevant succedents will be initially defined as the set of all possible conjunctions of 1 - 3 basic Boolean attributes created from attributes in the group *Personal Characteristics*. The basic Boolean attributes *Hypertension*(*yes*), *Hypertension*(*no*) and *Obesity*(*yes*) will be created from the attributes *Hypertension* and *Obesity*. The attribute *BMI* has rational values from interval $\langle 15, 40)$, thus categories – intervals $\langle 15, 16), \ldots, \langle 39, 40)$, will be created and basic Boolean attributes – intervals with a length of 5 (i.e. "sliding window"' with a length of 5) will be used.

5.3 Stop Criterion

A stop criterion must be defined for each application of 4ft-Miner with one particular quantifier from $\Rightarrow_{p,Base}$, $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{p,Base}$. The most natural criterion is an interval for the number of remaining association rules after filtering out consequences of given items of domain knowledge. However in some cases additional criteria must be used, e.g. impossibility of modifying parameters or too much time needed to run 4ft-Miner with modified parameters.

5.4 Modification of Parameters for the 4ft-Miner Procedure

There are various ways of modifying particular parameters of the 4ft-Miner procedure. If the number of remaining association rules after filtering rules that are consequences of given items of domain knowledge is too low, we can e.g. decrease the parameters p and $Base$ of the quantifiers used, increase the maximal length of antecedents or increase the maximal length of coefficients of particular attributes. The inverse operations can be done if the number of remaining association rules is too high . Note that the various properties of particular attributes should be considered when modifying parameters.

6 Consequences of Stored Items of Domain Knowledge

A very important step in the process of answering a given analytical question $\mathcal{AQ}$ is to remove prime patterns that do not satisfy $\mathcal{AQ}$ criteria. We will discuss the related problems for our simple $\mathcal{AQ}$ "What strong unknown relations among attributes in the groups *Personal Characteristics* (*Sex*, *Age*, *Education*, *City*, *Beer*, and *Wine*) and *Health Status* (*Hypertension*, *BMI*, and *Obesity*) are valid in the data matrix ADAMEK?", see Section 5.1. We use the GUHA 4ft-Miner procedure with initial input parameters according to Section 5.2, the prime patterns produced by the 4ft-Miner are association rules.

We accept the criteria that "unknown relations" means that the found patterns are not consequences of items of knowledge concerning the mutual influence of particular attributes presented in Tab. 2. This means we have to remove the consequences of the items *Age* $\uparrow^+$ *Hypertension*, *Age* $\uparrow\uparrow$ *BMI*, *Education* $\uparrow\downarrow$ *BMI*, *Beer* $\uparrow^+$ *Obesity*, *Wine* $\uparrow^-$ *Hypertension*, *Obesity* $\rightarrow^+$ *Hypertension* of domain knowledge. We use the following principle to find all association rules that are consequences of these items of domain knowledge:

- We find the simplest association rules that can be understood as *atomic consequences* of particular items of the domain knowledge, see Section 6.1.
- We will consider all association rules that are logical consequences of atomic consequences as consequences of items of domain knowledge in question, see Section 6.2.

6.1 Atomic Consequences

An example of items of background knowledge is *Age* $\uparrow^+$ *Hypertension* which means that "*If Age increases then the relative frequency of patients with Hypertension also increases*". Thus we have to find the simplest relevant association rules $\varphi \approx \psi$ that can be understood as consequences of the item *Age* $\uparrow^+$ *Hypertension*. The relevant association rules $\varphi \approx \psi$ that can be understood as consequences of the item *Age* $\uparrow^+$ *Hypertension* must contain at least one basic Boolean attribute created from the attribute *Hypertension* and at least one basic Boolean attribute created from the attribute *Age*. Thus the simplest association rules $\varphi \approx \psi$ will have the form $\mathcal{B}(Age) \approx \mathcal{B}(Hypertension)$ or $\mathcal{B}(Hypertension) \approx \mathcal{B}(Age)$ where $\mathcal{B}(Age)$ is a basic Boolean attribute created from the attribute *Age* and similarly for $\mathcal{B}(Hypertension)$.

The basic Boolean attributes *Hypertension*(*yes*) and *Hypertension*(*no*) will be created for the attribute *Hypertension* and the attribute *Age* with categories - intervals of years $(0, 10\rangle, (10, 20\rangle, \ldots, (90, 100\rangle$ will be defined and coefficients *intervals* with a length of 1 - 3 will be used (i.e. the basic Boolean attributes $Age(0, 10\rangle$, $Age(0, 20\rangle$, $Age(0, 30\rangle$, $Age(10, 20\rangle$, ..., $Age(70, 100\rangle$, $Age(80, 100\rangle$, $Age(90, 100\rangle$ will be generated, see section 5.2.

We will use basic Boolean attributes $\mathcal{B}(Age)$ in the form $Age(\omega_{Age})$ which can be understood as *Age is high* or *patient is old*. Remember that $\omega_{Age} = \{a_{i_1}, \ldots, a_{i_K}\} \subset \{a_1, \ldots, a_L\}$ and $a_1, \ldots, a_L$ are intervals of natural numbers. Let us denote $I_{\omega_{Age}} = \bigcup_{j=1}^{K} a_{i_j}$, then the sets ω_{Age} can be defined e.g. such that we require that $k \in I_{\omega_{Age}}$ implies $k \geq 70$.

We use the quantifiers $\Rightarrow_{p,Base}$, $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$, thus the following *atomic consequences* will be used for the item of background knowledge $Age \uparrow^{+} Hypertension$:

- $Age(\omega_{Age}) \Rightarrow_{p,Base} Hypertension(yes)$ where $p \geq 0.9$ and $Base \geq 20$
- $Age(\omega_{Age}) \Leftrightarrow_{p,Base} Hypertension(yes)$ where $p \geq 0.9$ and $Base \geq 20$
- $Age(\omega_{Age}) \equiv_{p,Base} Hypertension(yes)$ where $p \geq 0.9$ and $Base \geq 20$
- $Age(\omega_{Age}) \Rightarrow^{+}_{q,Base} Hypertension(yes)$ where $q \geq 0.3$ and $Base \geq 20$.

We emphasize that we do not consider atomic consequences of the form $Hypertension(yes) \Rightarrow_{p,Base} Age(\omega_{Age})$ because we do not consider them as consequences of $Age \uparrow^{+} Hypertension$. Note that the 4ft quantifier $\Leftrightarrow_{p,Base}$ is symmetric and thus the rule $Age(\omega_{Age}) \Leftrightarrow_{p,Base} Hypertension(yes)$ means the same as the rule $Hypertension(yes) \Leftrightarrow_{p,Base} Age(\omega_{Age})$. In addition, the quantifiers $\equiv_{p,Base}$ and $\Rightarrow_{p,Base}$ are also symmetric [15]. Also note that the boundaries $p \geq 0.9$, $q \geq 0.3$ and $Base \geq 20$ can be changed on the basis of experience.

We present the meaning of particular atomic consequences of the item $Age \uparrow^{+} Hypertension$ of domain knowledge (we assume $p \geq 0.9$, $q \geq 0.3$ and $Base \geq 20$) in more detail below.

The rule $Age(\omega_{Age}) \Rightarrow_{p,Base} Hypertension(yes)$ means that at least $100p$ per cent of patients described in the data matrix ADAMEK who are old have hypertension and that there are at least 20 old patients with hypertension.

The rule $Age(\omega_{Age}) \Leftrightarrow_{p,Base} Hypertension(yes)$ means that at least $100p$ per cent of patients described in the data matrix ADAMEK who are old or have hypertension are both old and have hypertension and moreover that there are at least 20 old patients with hypertension.

The rule $Age(\omega_{Age}) \equiv_{p,Base} Hypertension(yes)$ means that at least $100p$ per cent of patients described in the data matrix ADAMEK are either both old and have hypertension or are both not old and do not have hypertension and moreover that there are at least 20 old patients with hypertension.

$Age(\omega_{Age}) \Rightarrow^{+}_{q,Base} Hypertension(yes)$ means that among old patients there are least $30q$ per cent more patients with hypertension than among all patients described in the data matrix ADAMEK and moreover that there at least 20 old patients with hypertension.

Tab. 3 presents an overview of all of the atomic consequences of the considered items of domain knowledge (again assuming $p \geq 0.9$, $q \geq 0.3$ and $Base \geq 20$). All the atomic consequences are constructed similarly to the atomic consequences of $Age \uparrow^{+} Hypertension$.

The following basic Boolean attributes are used in Tab. 3:

- $Age(\omega_{Age})$, which is explained above
- $Hypertension(yes)$, $Hypertension(no)$, and $Obesity(yes)$, which are obvious

Table 3 Atomic Consequences of Items of Domain Knowledge

No.	Item of Domain Knowledge	Atomic Consequences
1	$Age \uparrow^{+} Hypertension$	$Age(\omega_{Age}) \Rightarrow_{p,Base} Hypertension(yes)$ $Age(\omega_{Age}) \Leftrightarrow_{p,Base} Hypertension(yes)$ $Age(\omega_{Age}) \equiv_{p,Base} Hypertension(yes)$ $Age(\omega_{Age}) \Rightarrow^{+}_{q,Base} Hypertension(yes)$
2	$Age \uparrow\uparrow BMI$	$Age(\omega_{Age}) \Rightarrow_{p,Base} BMI(\omega_{BMI})$ analogously for $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$
3	$Education \uparrow\downarrow BMI$	$Education(\omega_{Education}) \Rightarrow_{p,Base} BMI(\delta_{BMI})$ analogously for $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$
4	$Beer \uparrow^{+} Obesity$	$Beer(\omega_{Beer}) \Rightarrow_{p,Base} Obesity(yes)$ analogously for $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$
5	$Wine \uparrow^{-} Hypertension$	$Wine(\omega_{Wine}) \Rightarrow_{p,Base} Hypertension(no)$ analogously for $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$
6	$Obesity \rightarrow^{+} Hypertension$	$Obesity(yes) \Rightarrow_{p,Base} Hypertension(yes)$ analogously for $\Leftrightarrow_{p,Base}$, $\equiv_{p,Base}$, and $\Rightarrow^{+}_{q,Base}$

- $BMI(\omega_{BMI})$ and $BMI(\delta_{BMI})$, which are explained below
- $Education(\omega_{Education})$, which is explained below
- $Beer(\omega_{Beer})$ and $Wine(\omega_{Wine})$ which are similar, see below.

The basic Boolean attribute $BMI(\omega_{BMI})$ should be understood as *BMI is high*. Remember that $\omega_{BMI} = \{a_{i_1}, \ldots, a_{i_K}\} \subset \{a_1, \ldots, a_L\}$ and $a_1, \ldots, a_L$ are intervals of rational numbers. Let us denote $I_{\omega_{BMI}} = \bigcup_{j=1}^{K} a_{i_j}$, then the sets ω_{BMI} can be defined e.g. such that we require that $r \in I_{\omega_{BMI}}$ implies $r \geq 30$.

The basic Boolean attribute $BMI(\delta_{BMI})$ should be understood as *BMI is low*. Let us denote $I_{\delta_{BMI}} = \bigcup_{j=1}^{K} a_{i_j}$, then the sets δ_{BMI} can be defined e.g. such that we require that $r \in I_{\delta_{BMI}}$ implies $r \leq 20$.

The basic Boolean attribute $Education(\omega_{Education})$ should be understood as *Education is high*. Remember that the attribute *Education* in the ADAMEK data matrix has the categories *basic*, *secondary*, and *higher*. Thus we define $\omega_{Education}$ such that $\omega_{Education} = \{secondary, higher\}$ or $\omega_{Education} = \{higher\}$.

The basic Boolean attribute $Beer(\omega_{Beer})$ should be understood as Beer consumption in liters per week is high. The attribute *Beer* has categories *very low*, *low*, *medium*, *high*, and *very high*. Thus we define ω_{Beer} such that $\omega_{Beer} = \{high, very\ high\}$ or $\omega_{Beer} = \{high\}$ or $\omega_{Beer} = \{very\ high\}$.

The basic Boolean attribute $Wine(\omega_{Wine})$ should be understood as Wine consumption in liters per week is high. It is defined similarly to $Beer(\omega_{Beer})$.

6.2 Logical Consequences of Atomic Consequences

Our goal is to filter out all association rules that can be understood as logical consequences of the items $Age \uparrow^{+} Hypertension$, $Age \uparrow\uparrow BMI$, $Education \uparrow\downarrow BMI$, $Beer \uparrow^{+} Obesity$, $Wine \uparrow^{-} Hypertension$, $Obesity \rightarrow^{+} Hypertension$ of

domain knowledge. Our approach is to consider such logical consequences as logical consequences of the atomic consequences listed in Tab. 3.

For each association rule $\varphi \approx \psi$ that is an output of the 4ft-Miner procedure,we have to decide if it will be filtered out or not. This means that we have to decide if the association rule $\varphi \approx \psi$ logically follows from at least one of the atomic consequences $\varphi_A \approx \psi_A$ listed in Tab. 3. In other words we have to decide if the deduction rule $\frac{\varphi_A \approx \psi_A}{\varphi \approx \psi}$ is correct or not. It can be decided however using the criteria introduced in Section 3.

`1. Introduction`
Formulation of the analytical question, i.e.
"What strong unknown relations among attributes in the groups *Personal Characteristics* and *Health Status* are valid in the data matrix ADAMEK?"
Explanation of how to answer the question and description of the structure of the report.

`2. Analyzed Data`
Overview of basic statistics of the used attributes.

`3. Answering the Analytical Question`
Explanation of association rules, basic Boolean attributes and 4ft-quantifiers, the possibilities of the 4ft-Miner procedure and of dealing with known items of domain knowledge.

`4. Domain Knowledge`
Explanation of known items of domain knowledge used (i.e. *Age* $\uparrow^+$ *Hypertension*, ...) and their consequences in the form of association rules.

`5. Results Overview`
Statistics of all unknown relations found, suitable statistics on particular attributes occurrences, assertions of "second order" such as *there is no unknown rule concerning Age*, description of suitable groups of unknown relations (e.g. new strong relations, exceptions from known relations, etc.).

`6. Detailed Results`
Detailed results structured in the way described in Chapter 5.
`6.1 New Strong Relations`
`6.2 Exceptions from Known Relations`
`6.3 ...`

`7. Conclusions`
Conclusions and suggestions of additional analytical questions.

Fig. 4 Outline of the Analytical Report

7 Resulting Analytical Report

The data mining process starts with the formulation of an analytical question which is interesting from the user's point of view. The result of data mining is understood here as a well written analytical report answering given analytical question. A detailed discussion of such reports exceeds the scope of this paper. Some remarks on this topic can be found in [20, 21, 22] Here we only outline the analytical report answering our analytical question, see Fig. 4.

Note that the structure of the report can be modified in various ways depending on the detailed results of data mining procedures applied. It is assumed that it will be possible to define a suitable skeleton (or family of related skeletons) for each analytical report.

8 Logical Calculi for Analytical Questions and Reports

Both the GUHA 4ft-Miner procedure and the logic of association rules presented in this paper are closely related to the results presented in [5]. The book [5] tries to give answers to the questions (i) *Can computers formulate and verify scientific hypotheses?* and (ii) *Can computers in a rational way analyze empirical data and produce a reasonable reflection of the observed empirical world? Can this be done using mathematical logic and statistics?*, see also Introduction. The answers given in [5] are based on the following scheme of inductive inference:

$$\frac{\text{theoretical assumptions, observational statement}}{\text{theoretical statement}} .$$

This scheme means that if we accept theoretical assumptions and verify a particular statement concerning observed data then we accept the conclusion - a theoretical statement. It is important that suitable statements about data rather than observed data lead to the theoretical conclusions. The questions **L0 - L4** are formulated in [5]:

(L0) In what languages does one formulate observational and theoretical statements? (What is the syntax and semantics of these languages? What is their relation to the classical first order predicate calculus?)

(L1) What are rational inductive inference rules bridging the gap between observational and theoretical sentences? (What does it mean that a theoretical statement is justified?)

(L2) Are there rational methods for deciding whether a theoretical statement is justified (on the basis of given theoretical assumptions and observational statements)?

(L3) What are the conditions for a theoretical statement or a set of theoretical statements to be of interest (importance) with respect to the task of scientific cognition?

(L4) Are there methods for suggesting such a set of statements, which is as interesting, as possible?

Answers to questions (L0) - (L2) constitute a *logic of induction*, answers to questions (L3) - (L4) constitute a *logic of suggestion*, and answers to questions (L0) - (L4) constitute a *logic of discovery*. Observational calculi are defined in [5] as the language in which observational statements are formulated. A typical feature of observational calculi is the effective calculability of the (true) value of each sentence in each observational structure. Theoretical calculi developed in [5] are statistically motivated. Theoretical sentences refer to systems of "possible worlds" and probability is understood as a measure on such a system of possible worlds. Observational and theoretical calculi are interrelated by inductive inference on statistical hypothesis tests. It is important that in many inductive inference rules hypotheses correspond one-to-one with certain specific observational statements. The GUHA method is formally defined in [5] as a tool for suggesting such a set of observational statements which are as interesting as possible.

The LISp-Miner system contains six GUHA procedures [16, 17, 23] that were recently applied to various data sets to solve numerous data mining tasks. The 4ft-Miner procedure is a slightly enhanced ASSOC procedure defined in [5], the additional five GUHA procedures in LISp-Miner are new procedures. However the statistical features of the GUHA method developed in [5] were not used in the aforementioned applications. We must emphasize that this does not mean that the statistical features of the GUHA method are not applicable, some information on their applications is included in e.g. in [6, 7, 8, 9]. The statistical features of the GUHA procedures were not used because of the analyzed data sets were not suitable for the application of statistical tests of hypothesis and/or the users of the GUHA method were not able to apply the statistical features of the GUHA procedures. Despite these limitations, these applications led to reasonable results that were appreciated by the owners of data sets.

The applications also led to the following conclusions and actions aimed at enhancing the possible applications of the LISp-Miner system.

- There are various important analytical questions that can be answered using the six GUHA procedures implemented in the LISp-Miner system. However it is not easy to identify all analytical questions that can be answered in the given situation.
- Such analytical questions are often related to various items of domain knowledge that can be easily formalized. Examples of such items of domain knowledge are presented in Section 4. A special new part of the LISp-Miner system called *LM KnowledgeSource* was implemented to store such items of domain knowledge.
- Analytical questions related to items of domain knowledge stored in *LM KnowledgeSource* can be easily formulated using patterns of analytical questions, see Section 5.1.

- We can distinguish two types of analytical questions. Analytical questions of the first type concern one given data matrix. An example is the analytical questions "What strong unknown relations among attributes in the groups *Personal Characteristics* and *Health Status* can be found in the data matrix ADAMEK?", see section 5. We call such questions *local analytical questions*.
- The second type of analytical questions concerns several data sets. An example is the analytical question "What differences can be found between the data sets ADAMEK and STULONG that concern generally accepted knowledge?". It is crucial that the structure of the data matrices can differ even in relation to the attributes in question (see e.g. the attribute *Education*). We call such questions *global analytical questions*.
- The consumer of the results is not usually interested in particular patterns related to the given analytical question. He requires a comprehensive report dealing with all relevant aspects of the given analytical question. We call such analytical reports *local analytical reports* and *global analytical reports* depending on the corresponding analytical question.
- The preparation of the analytical report is not an easy task. The structure of the report and its additional properties depend on the solved analytical question. It is reasonable to implement a software tool that will assist in the preparation of the analytical report answering a given analytical question.
- The possibilities of dealing with both *local* and *global analytical reports* led to considerations on the SEWEBAR system for the dissemination of analytical reports through the Semantic web [20].

The considerations presented in Sections 4 - 7 show that logical calculi of association rules introduced in Section 3 can be useful tools for dealing with knowledge in the data mining process. However there are numerous open related problems. We can formulate questions in a similar way to the way the questions **(L0)** - **(L4)** are formulated:

(K0) Are there methods for the formulation of interesting analytical questions? How can domain knowledge be used in such methods?

(K1) In what languages does one formulate useful domain knowledge? What is the syntax and semantics of these languages? What is their relation to known observational and theoretical calculi and other languages used to deal with domain knowledge (e.g. to ontologies and ILP)?

(K2) In what languages does one formulate and present analytical reports as answers to analytical questions. What are the ways of bridging the gap between observational statements produced by GUHA methods and the statements used in analytical reports for the presentation of the results of data mining?

(K3) Are there methods for producing good "knowledge intensive" analytical reports?

9 Conclusions and Further Work

We have presented an approach to data mining based on analytical questions and analytical reports answering particular analytical questions. We have demonstrated that GUHA procedures implemented in the LISp-Miner system and logical calculi of association rules are useful tools for dealing with such analytical questions and related reports. Their applications are closely related to the formal representation of various items of domain knowledge. We have introduced a new part of the LISp-Miner system called *LM KnowledgeSource* whose goal is to maintain various types of domain knowledge. The stored items of domain knowledge can be used to both formulate suitable analytical questions and to answer formulated questions.

Logical calculi of association rules and GUHA procedures provide particular answers to questions formulated in [5] with the goal of developing an approach to the logic of discovery. We have formulated similar questions that could help to develop logical calculi for dealing with analytical questions, analytical reports and domain knowledge. The goal of further work is to develop and investigate such logical calculi.

References

1. Lavrac, N., Dzeroski, S.: Inductive Logic Programming: Techniques and Applications. Ellis Horwood, Chichester (1994)
2. Aggraval, R., et al.: Fast Discovery of Association Rules. In: Fayyad, U.M., et al. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI Press, Menlo Park (1996)
3. Hébert, C., Crémille, B.: A Unified View of Objective Interestingness Measures. In: Perner, P. (ed.) MLDM 2007. LNCS, vol. 4571, pp. 533–547. Springer, Heidelberg (2007)
4. Geng, L., Hamilton, H.J.: Interestingness Measures for Data Mining: A survey. ACM Computing Surveys 38, 33 (2006)
5. Hájek, P., Havránek, T.: Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory. Springer, Heidelberg (1978)
6. Hájek, P. (guest ed.): International Journal of Man-Machine Studies, special issue on GUHA 10 (January 1978)
7. Hájek, P. (guest ed.): International Journal of Man-Machine Studies, second special issue on GUHA 15 (1981)
8. Hájek, P., Havránek, T., Chytil, M.: GUHA Method (in Czech). Academia, Prague (1983)
9. Hájek, P., Sochorová, A., Zvárová, J.: GUHA for personal computers. Computational Statistics & Data Analysis 19, 149–153 (1995)
10. Yang, Q., Wu, X.: 10 Challenging Problems in Data Mining Research. International Journal of Information Technology & Decision Making 5(4), 597–604 (2006)
11. Ralbovský, M., Kuchař, T.: Using Disjunctions in Association Mining. In: Perner, P. (ed.) ICDM 2007. LNCS, vol. 4597, pp. 339–351. Springer, Heidelberg (2007)

12. Rauch, J.: Logical Calculi for Knowledge Discovery in Databases. In: Proc. Principles of Data Mining and Knowledge Discovery, Trondheim, Norway, pp. 47–57 (1997)
13. Rauch, J.: Logic of Association Rules. Applied Intelligence 22, 9–28 (2005)
14. Rauch, J.: Definability of Association Rules in Predicate Calculus. In: Lin, T.Y., Ohsuga, S., Liau, C.J., Hu, X. (eds.) Foundations and Novel Approaches in Data Mining, pp. 23–40. Springer, Heidelberg (2005)
15. Rauch, J.: Classes of Association Rules - an Overview. In: Lin, T., et al. (eds.) Datamining: Foundations and Practice. Studies in Computational Intelligence, vol. 118, pp. 283–297. Springer, Heidelberg (2008)
16. Rauch, J., Šimůnek, M.: An Alternative Approach to Mining Association Rules. In: Lin, T.Y., Ohsuga, S., Liau, C.J., Tsumoto, S. (eds.) Data Mining: Foundations, Methods, and Applications, pp. 219–238. Springer, Heidelberg (2005)
17. Rauch, J., Šimůnek, M.: GUHA Method and Granular Computing. In: Hu, X., et al. (eds.) Proceedings of IEEE conference Granular Computing, pp. 630–635 (2005)
18. Rauch, J., Šimúnek, M.: Dealing with Background Knowledge in the SEWEBAR Project. In: Berendt, et al. (eds.) Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery. Springer, Heidelberg (2009) (to appear)
19. Rauch, J.: Logical Aspects of the Measures of Interestingness of Association Rules. In: Koronacki, J., et al. (eds.) Recent Advances in Machine Learning. Springer, Heidelberg (2009) (to appear)
20. Rauch, J., Šimůnek, M.: Semantic Web Presentation of Analytical Reports from Data Mining - Preliminary Considerations. In: Lin, T.Y., et al. (eds.) Web Intelligence 2007 Proceedings, pp. 3–7 (2007)
21. Rauch, J., Šimůnek, M.: LAREDAM - Considerations on System of Local Analytical Reports from Data Mining. In: An, A., Matwin, S., Raś, Z.W., Ślezak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 143–149. Springer, Heidelberg (2008)
22. Rauch, J., Tomečková, M.: System of Analytical Questions and Reports on Mining in Health Data – a Case Study. In: Roth, J., et al. (eds.) Proceedings of IADIS European Conference Data Mining 2007, pp. 176–181. IADIS Press (2007)
23. Šimůnek, M.: Academic KDD Project LISp-Miner. In: Abraham, A., et al. (eds.) Advances in Soft Computing – Intelligent Systems Design and Applications. Springer, Heidelberg (2003)
24. Svátek, V., Rauch, J., Ralbovský, M.: Ontology-Enhanced Association Mining. In: Ackermann, M., Berendt, B., Grobelnik, M., Hotho, A., Mladenič, D., Semeraro, G., Spiliopoulou, M., Stumme, G., Svátek, V., van Someren, M., et al. (eds.) EWMF 2005 and KDO 2005. LNCS, vol. 4289, pp. 163–179. Springer, Heidelberg (2006)

Part III
Information Integration and Data Security

A Study on Recent Trends on Integration of Security Mechanisms

Paul El Khoury, Mohand-Saïd Hacid, Smriti Kumar Sinha,
and Emmanuel Coquery

Abstract. Business solutions and security solutions are designed by different authorities at different coordinates of space and time. This engineering approach not only makes the lives of security and the business solution developers easy but also provide a proof of concept that the concerned business solution will have all the security features as expected. But it doesn't provide a proof that the integration process will not lead to conflicts between the security features in the security solution and also between security features and the functional features of the business solution. For providing a conflict-free secured business solution, both the developers of security solution as well as of the secure business solution need a mechanism to identify all possible cases of conflicts, so that the developers can redesign the corresponding solutions and thus resolve the conflicts if any. Conflict arises due to different authorities and configuration and other resource sharing among the solutions under integration. In this chapter, we discuss conflicts during integration of security solutions with business solutions covering the wide spectrum of social, socio-technical and purely technical perspectives. The investigated recent approaches for automated detection of conflicts are also discussed in brief. The ultimate objective of the chapter is to discover the best suited approaches for detecting conflicts by software

Paul El Khoury
SAP Research, SAP Labs France
University Claude Bernard Lyon 1, LIRIS CNRS UMR 5205, France
e-mail: paul.el.khoury@sap.com

Mohand-Saïd Hacid
University Claude Bernard Lyon 1, LIRIS CNRS UMR 5205, France
e-mail: mshacid@liris.cnrs.fr

Smriti Kumar Sinha
SAP Reseach, SAP Labs France
Tezpur University, Tezpur, India
e-mail: smriti.kumar.sinha@sap.com

Emmanuel Coquery
University Claude Bernard Lyon 1, LIRIS CNRS UMR 5205, France
e-mail: ecoquery@liris.cnrs.fr

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 203–224.
springerlink.com

developers. It spans over approaches from cryptographic level to policy level weaving over the feature interaction problem typically suited for software systems. The assessment of these approaches is demonstrated by a remote healthcare application.

Keywords: Integration, Security Solutions, Conflicts.

1 Introduction

In general, companies talk about security products whereas all of them are actually aiming at secure products. Far from being an afterthought, security is an ongoing requirement whose alignment with business goals and technologies throughout the software application is critical to success. Security experts developed and standardization bodies adopted several security solutions sufficient to satisfy security requirements for specific contexts. Although these security solutions cover a wide spectrum of contexts for IT applications, best practices and researchers [1, 4, 10, 15, 26, 28, 30, 32, 36, 41, 47, 69] showed that they are deficient in case they are not combined properly. Combining (i.e. composing) and integrating security solutions is an unavoidable requirement for secure products. First, one security solution cannot secure a complete application. To be compliant with the authoritative's regulations, several security requirements over a shared system have to be covered through independent security solutions. Second, it is cheaper and easier to combine security solutions rather than designing new ones, a fact that even security experts tend to procreate. These two motivations clearly show the unavoidable situations where security solutions are to be integrated. Still integration becomes of a greater challenge in cases where IT applications produced by competing companies (including their security mechanisms) have to be integrated [75, 76]. Last but not least, security solutions are now available as Commercial, Off-The-Shelf (COTS) [50]. Providing security solutions in this friendly manner (through security patterns [51, 52, 54] and the like) easiness the means for non security experts to combine them inappropriately assuming that: combining security solutions is equivalent to satisfying combined security requirements.

To better understand the nature for these conflicts we turned onto social science. It is well known in social science that for resolving conflicts[1] one has to trace them back to their roots. These roots for conflicts are classified in three categories: (i) *Position* as for an authority's point of view on a certain topic, hence conflict is due to opposed *positions* (ii) *Interest* as of strategic importance, hence conflict is caused by conflict between two authority's *interest* in a certain topic and finally (iii) *Need* as the requirement/the necessity to persist, knowing that conflicts of this category are caused by the authority's *need* of a certain topic for survival [57]. Drawing the parallel of social science to security solutions, it becomes clear that solving conflicts between integrated security solutions fall into category (iii) of social conflicts. However the different strategies discussed in [57] are not relevant to this chapter.

[1] Conflict management between humans, organizations and countries whether these latter are self-authoritative or not.

This chapter presents a survey on the available approaches for conflict management in socio-technical and technical security solutions. It presents a variety of methods used for analyzing and reasoning about integration of security solutions [1, 4, 10, 15, 26, 28, 30, 32, 36, 41, 47, 69, 70, 87, 62]. Particularly for the technical part it reviews most of the different conflict management approaches studied in the literature for different application layers. Although security experts studied and developed different tools for conflict detection and management, the sound integration of security solutions remained among the incomplete challenges [44, 36, 37] as we will point out as well in our analysis for the remote healthcare case study. We will first highlight the security requirements of the case study, second identify the required combinations, third map the combination of security solutions to the state of the art approaches for conflict management and finally show the open challenges for software developers in detecting conflict for the combination of security solutions.

The rest of this chapter is organized as follows: In Section 2 we introduce the genesis of conflicts. Then in Section 3 we describe several approaches displaying commonalities in addressing conflict detection and varying in dealing with conflicts resolution. In Section 4 we present our case study and points out to the scalability challenges for the adoption of the conflicts management approaches discussed in Section 3. Finally in in Section 5 we conclude this chapter pointing out to future directions.

2 Genesis of Conflicts

Security solutions have been considered as the non-functional or extra-functional aspects of the system's behavior. An environment where a security solution is hosted in the system is called a context. As long as the security solutions under integration share no context then no harm is possible. Many designs rely on this hypothesis to lower the percentage of potential conflicts; those designs fall out the context of this chapter. For the larger set of solutions under integration we consider set out the shared context setting as a blueprint for all conflict detection approaches classifiying their integration approaches from motivation until solution.

The analysis of security incidents and frauds has revealed that security is often compromised by exploiting organizational vulnerabilities [17, 7]. The call to arms for integrating security aspects during the entire development process overlooked the importance of the socio-technical aspect. Attackers bypass such security measures by exploiting weaknesses of the socio-technical system as a whole. Obviously, more than one socio-technical security solutions needed to be deployed for satisfying a certain security requirement for the system organizational structure. Integrating socio-technical security solutions have been tackled in the literature by focusing on the human aspect. Particularly at this level, reasoning over the *Interest* discloses potential conflicts *of interest*. The crossing of different interests is a shared *Trust* common to the security solutions. Furthermore, the shared context among security solutions under integration could coat technical aspects of the organization. Specifically, this technical context includes (i) functional and (ii) non-functional segments

of the system. First, in (i)'s case potential conflicts are specific to typical dependency types of conflicts. Sure we need to point out to an old debate over the 'nature' of security solutions in the research community, where the question was to considered as non-functional, extra-functional or as functional. In our case we can say that integration of security solutions is not immune to dependency conflicts, that is typical for the types of conflicts in functional systems, while surely considering its non-functional nature. Second, the (ii)'s case requires further analysis especially at more technical layers of the system where intruder models are introduced for each security solution. The combination of security solutions, could enrich the intruder's strategies. This is typically the case of integration of multi party security solutions such as security protocols.

This review chapter covers the integration solutions ranging from cryptographic ones to organizational ones such as the access control. Our main guideline in studying these integration methods is to try to classify the roots and the reasoning methodologies as much as possible. We aim at identifying approaches (and their corresponding tools) useful for verifying the soundness of the integration of security solutions in conventional applications such as remote e-Health assistance described in the end of this chapter.

3 Composition of Security Solutions

We outline the review for this survey in Figure 1. The shared context as previously defined is the elemental root of the classification of causes for conflict. This shared context can be leaning towards technical system managed by humans known by Socio-Technical, or strictly Technical systems. The integration of security solutions in Socio-Technical is presented in Section 3.1, whereas in Technical systems it is presented in Section 3.2. Furthermore we identify two additional categories for technical systems, Functional and Non-Functional. As shown in Figure 1 these two categories are types of root causes for conflicts caused by security solutions under integration. Based on the literature we identify three major technologies where conflicts are studied, namely Policy-based Models, Security Protocols and Cryptographic Solutions. In Figure 1 we showed the context of cryptographic solutions taken from the security protocols, further details on the rational behind this classification is provided in Section 3.2 where we review security protocols and in Section 3.2 where we review cryptographic solutions. The policy-based models are reviewed in Section 3.2. As with regards to the functional category we identify Feature Interaction for conflict detection and resolution. Further details about this category are shown in Section 3.3.

3.1 Composition of Socio-technical Security Solutions

Contracts establish trust between organizations and bound their parties, i.e. employees, and organizations, to deliver authentically the intended job. In order to enforce

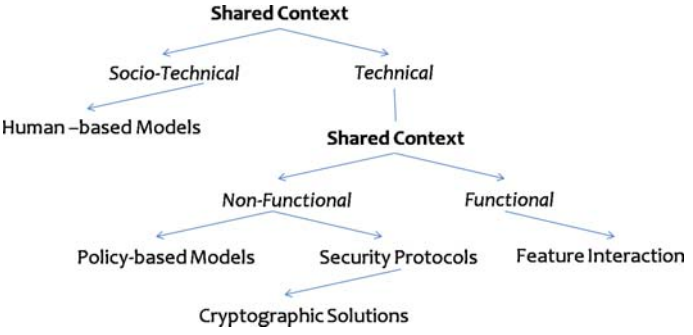


Fig. 1 Classification of the Common Context for Conflicts in Integration of Security Solutions

these contracts at the service level, several studies were pursued on discovering correct organizational configuration. Such configuration require no improper act resulting in undermining confidence in the organization. *Conflicts of interest* which is solved through *Separation of Duties* constraints occurs when an individual or organization has an interest that might compromise their reliability. Conflicts of interest are often discussed in the context of role-based access control (RBAC) models [4, 1, 2, 3] where the particular focus was for members (user/groups) belonging to two or more roles. This type of conflict is denoted as role-role conflict that is also studied as Separation of Duty (SoD). Advanced explanation based on user-roles assignments given in [4] distinguished between static and dynamic SoD. They conducted their study over a reference model that considers three distinct entities to be combined: users, privileges, and roles. Privilege denotes an access mode on an object, and a role denotes a set of privileges. They defined authorization relations among User-Role, Role-Role and Role-Privilege. Their analysis for detecting SoD violation was based on their previous work in [5] that analyzed role graph model representation for these SoD constraints violation.

Additional studies dealt with situation where users are assigned to mutually exclusive privileges [5, 1], i.e. privilege-privilege conflicts of interest. In addition to the previous conflict of interest or the separation of duties enumeration they referred to the object-based separation of duty. An object-based SoD controls that a user *may* perform two different operations on different objects, but may not perform these two operations on the same object. These works targeted more refined vision on organization than the previous ones. Furthermore, in [11, 12, 13] authors proposed to model, using UML-with-classes or UML-with-OCL, the RBAC models for detecting conflicts of interest. These approaches use specific domain constraints that will be checked statically or dynamically. However, they do not analyze organizational requirements to understand why such constraints should be introduced and the effects of their introduction, so major constraints could be omitted or minor constraints could affect system functionalities.

Indeed, security is often compromised by exploiting loopholes in the security policies adopted by the organization rather than by breaking protection mechanisms such as encryption or security protocols [77]. On the other hand, different

approaches mainly targeting requirement engineering argue against this kind of deeper analysis [6, 8, 10]. They consider a major source of system vulnerabilities the presence of conflicts among requirements of the system prior to the service level. The importance of all previous works is crucial but it is somehow true that [11, 12, 13, 5, 1, 4] enumerated conflicts rather than understanding why and when they occur. Specifically in [10] the authors described how those studies were not driven by legal requirement[2] which is the essence of 'contract' which makes [11, 12, 13, 5, 1, 4]'s definitions intuitive, but not justified.

The approach presented in [10] extends those works by considering both entitlements and objectives rather than only objectives. The Secure Tropos methodology adopts the SI*[3] modeling language [10] for the acquisition of the requirements model. SI* employs the concepts of actor, goal, and resource: an *actor* is an intentional entity that performs actions to achieve goals; a *goal* is a strategic interest of an actor; a *resource* represents a physical or an informational entity. A graphical representation for SI* presents actors as circles, goals as ovals and resources as rectangles. The Secure Tropos methodology allow for every actor a set of objectives, entitlements, capabilities. The objectives are goals intended to be achieved or resources required by the actor; The entitlements are goals and resources controlled by the actor; and finally, the capabilities are goals and resources that the actor is able to respectively achieve and furnish. Interesting and proper to this socio-technical nature is that SI* adopts the notions of *trust of permission* and *trust of execution* to model the expectation of one actor (the *trustor*) about the behavior of another actor (the *trustee*) on a goal or resource (the *trustum*). It also employs the notion of *permission delegation* to model the transfer of entitlements (the *delegatum*) from an actor (the *delegator*) to another actor (the *delegatee*), and the notion of *execution dependency* to model the transfer of objectives (the *dependum*) from an actor (the *depender*) to another actor (the *dependee*).

Using the Secure Tropos methodology and ST-Tool formal framework, the authors of [6] modeled the organization requirements with the corresponding relations among actors and detected conflicts of interest during requirements analysis. They classified conflicts of interest as Attorney-in-fact conflict, where some (possibly personal) interests of the delegatee interfere with the interests of the delegator; Role conflict, where an agent is assigned a role whose interests collide with those of the agent; and finally self-monitoring conflict, where an actor is responsible for monitoring his own behavior. The tool was able to detect the presence of such models inside the global organization model and highlight a conflict of interest.

These kind of conflicts are elemental to fulfill any methodology toward a secure product. Nevertheless, as we will show in Section 4 we do not consider them fundamental for software developers in front of direct application of composable security solutions provided as COTS. These approaches are to be used prior to when software developers starts their work, in contrast to approaches we show hereafter.

[2] The authors in [10] considered the study presented in [14] as their argument.

[3] SI* is read as "see star".

3.2 Composition of Security Solutions Sharing Technical Context

The composed security solutions might overlap over some of their own data (i.e. not the one belonging to the system, belonging to the solutions under integration). Examples are key distribution and roles. Hereafter we go through known security solutions starting by composition of security protocols in Section 3.2, then the composition of cryptographic solutions in Section 3.2 and finally with integration of policy-based solutions in Section 3.2.

3.2.1 Composition of Security Protocols

Security protocols can be seen as multi-party algorithms. Their role is to provide security requirements among several peers. The abstraction models used in this process guided researchers to abstract cryptographic primitives [31] and consider them as black boxes. Moreover, security experts identified intruders models (e.g. Dolev Yao [31]) that could fail the fulfillment of these requirements. Using formal models and theorem provers [33, 34, 35], security experts are able to prove whether a security protocol is correct. Although remarkable efforts have been devoted to these issues, still they do not scale to bigger protocols; For instance, protocols specified to run different instances concurrently over the web [33, 35]. Even when running a complex protocol (composed from smaller protocols) in the same environment, it is highly possible to unsatisfy the security requirements satisfied independently. New and unpredicted behaviors for intruder can possbily take advantage of several data of the protocol to act as adversary [37]. In [38] they established the multi-protocol attack that requires more than one protocol to occur. This has been illustrated in [39] using configurations that particularly increase the potential for this attack.

Divide and conquer is a typical strategy for complex protocols, particularly when smaller protocols are already proved to be correct independently [32]. The works presented in [42, 40, 41] provide sufficient conditions for composition. Furthermore, they claimed a complex protocol to be secure when the correctness of all its smaller protocols is guaranteed under shared context. The standard protocols found in literature usually do not meet these requirements, and thus the theoretical possibility of multi-protocol attacks remains.

Still such an approach may not lead to fruitful results since the resulting protocols could be large, and do not scale to current methods [36]. The conducted research in that direction has shown two ways for composing security protocols, (i) sequentially where they run one after the other and (ii) integrally where they run concurrently while sharing different non-functional data such as encryption keys, session identifiers [36]. In those directions we selected the works of Cremers [37] and Datta et al. [43] in order to illustrate the major efforts.

In [37], the author redefined the multi-protocol attack and experimented on the composition using the tool in [33] that uses a hybrid theorem-proving/model checking algorithm. The experiment was conducted on a set of security protocols for three security properties: secrecy and non-injective agreement as well as non-injective synchronization that are two forms of authentications. Their conclusion showed the

non-possibility to verify an environment with all these protocols in parallel. Instead, they tested all possible combinations of two or three protocols from their set. When such a test yielded an attack, it was verified automatically whether the attack actually required multiple protocols, or could be mounted against a single protocol. During this process the authors discovered 163 new multi-protocol attacks. Moreover they discovered 23 out of 30 protocols, that had security correct claims separated but for which multi-protocol attacks existed. The analysis of the behavior for these new multi-protocol attacks led to two recurring behaviors: (i) Protocol updates and (ii) Ambiguous authentication. (i) is present when a second protocol, shares the same key structure of the initially deployed security protocol, which is very similar to the first one. Such a situation makes multi-protocol attacks (e.g. to a man-in-the-middle) very likely. Moreover, in (ii) the authentication protocol sets up session keys for other follow-up protocols. The resulting composition consists of the authentication protocol and the protocol that uses the session key. In this work the author showed that there can be a multiprotocol attack involving different follow-up protocols. Finally the author showed possible prevention methods for the multi-protocol attacks using context-aware tagging scheme that is ensuring that two protocols are different using unique tagging schemes.

One of the recent significant developments in compositional protocol analysis is Protocol Composition Logic (PCL) [43]. PCL provides support for compositional reasoning, and has been applied in a number of case studies. While the PCL approach is quite general, it cannot, in contrast to the previous approach, be easily automated [44]. In PCL, a first notation is introduced to define terms, which in turn are used to define protocols. For such protocols, an execution model is defined, assigning to each protocol a set of possible execution histories, called runs. Then, a protocol logic is defined in order to reason about (sets of) runs of a protocol. This logic is proved sound with respect to the execution model. This means that if one proves a property in terms of the protocol logic, such as Receive(. . ., m), then a similar property should hold for the corresponding set of runs in the execution model, such as "*receive* . . ., m has occurred in the protocol run". While the authors proved several protocol compositions [45], lately Cremers [44] pointed out several weak points in the proof provided by this composition language opening the scope for enhancements.

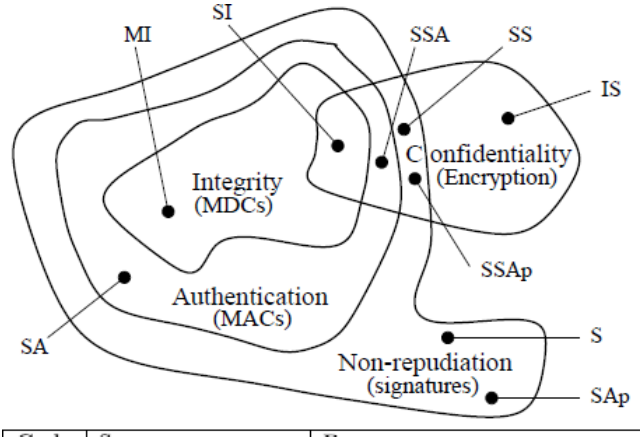
In general and as specified earlier the work on security protocols abstracted the cryptography complexity by considering it as a black box, a claim that is not intuitively obvious. Actually in the cryptography community there have been big efforts to discover whether it is possible to abstract sign, encrypt and other primitives as one abstract type of signature (like its actual representation in the conducted formal studies on security protocols through *sign*).

3.2.2 Composition of Cryptographic Solutions

Proofs of security protocols are independent of cryptographic details by abstracting the cryptographic operations. This is typically the case of Dolev-Yao model, which treats cryptographic operations as a specific term algebra. It may create issues when specific security primitives are combined naïvely, for instance we consider Sign

& Encrypt vulnerability to *surreptitious forwarding* in asymmetric cryptography. Let Alice sign & encrypt 'I love you' to Bob, but Bob re-encrypts Alice's signed message for Charlie. In the end, Charlie believes Alice wrote to him directly 'I love you', and can't detect Bob's subterfuge. Bridging the gap between formal methods and cryptography received fare amount of attention in the literature [46, 64, 55, 65].

Starting with [46], the authors provided a crypto-library with cryptographic composable operations where the abstract and the cryptographic versions are sound within the context of arbitrary surrounding interactive protocols. They proved a set of primitives for arbitrary, cryptographically secure public-key encryption and signature systems, enhanced by additional operations like tagging and randomization. Therefore the protocol designed via the abstraction of these primitives surely won't lead to efficiency problems. The presented primitives in the abstract library consist of Dolev-Yao-style primitives that are safely realized by a cryptographic implementation under different assumptions. The library provided works in a reactive setting. It means that the library can provide cryptographic primitives for more complex protocols with more powerful adversaries while preserving their security



Code	Scope	Purpose
IS	Confidentiality	provide secrecy of information
MI	Integrity	detect corruption of a message
SA	Authentication	authenticate the origin of a message
S	Non-repudiation	provide the authorship of a message
SAp	Non-repudiation	separate message from signature
SI	Confidentiality and Integrity	detect corruption of a secret
SSA	Confidentiality and Authentication	authenticate the origin of a secret
SS	Confidentiality and Non-repudiation	prove the authorship of a secret
SSAp	Confidentiality and Non-repudiation	separate secret from signature

Fig. 2 Cryptographic Design Patterns and their Purposes from [55]

properties. Interestingly the PCL discussed in 3.2 was complemented with Computational PCL, which is sound with respect to the complexity-theoretic model of modern cryptography [43].

In contrast to security solutions, already the composition of cryptographic function by non security experts is provided. Nevertheless, the composition of cryptographic mechanisms has to comply to very strong constraints, particularly the order in which the composition occurs.

Beyond best practice, the work in [55] presented more insight on the cryptographic operations. They considered four requirements, i.e. confidentiality, integrity, authentication and non-repudiation, and their corresponding four solutions, i.e. encryption/decryption, generation/verification of Modification Detection Code (MDC), generation/verification of Message Authentication Code and digital signing/verification. While all the requirements are typical for most scenarios, the composition of their solutions is limited. Therefore the authors presented all possible compositions as patterns using their *Tropic* pattern cryptographic language and corresponding Cryptographic APIs. To summarize, figure 2 shows the resulting composed cryptographic design patterns that are supported. Furthermore, in this study the authors have shown how their Cryptographic API (whether standalone or composed) can be used easier compared with other Cryptographic API provided by others, for example IBM (Common Cryptographic Architecture [66]), RSA (Cryptoki [67]) and Microsoft (CryptoAPI [68]).

The work presented in this category can be considered exploitable by non-security experts. Except that novice security users, such as software developers, are unable to compare and contrast all these low level cryptographic solutions.

3.2.3 Composition of Policy Based Models

The right for privacy and the need to meet the confidentiality requirement of sensible data are protected by laws and regulations spanning over the four continents [16, 18]. This motivation to research encouraged studies on access control solutions varying from firewalls, to operating systems, database management systems, network routers, and lately web services. Different access control models and frameworks were proposed [22, 21, 23, 24, 25]. In those models different kinds of conflicts might occur. We already highlighted conflicts of interest in Section 3.1 which could violate the Separation of Duty constraints. Obviously, the specified access control policies for these models could share some of the access control model primitives, such as subject, role, resource... For as many models as there exist, there is at least one corresponding approach for conflict management when different evaluations are retrieved from different applicable policies [30, 29, 26, 27, 28, 15]. In [30] they presented an off-line, static analysis of authorizations and obligations policies to determine modality conflicts (i.e. also known as clash) and application specific conflicts that is specified by external constraints expressed as Meta-policies.

An emerging access control model for organization is Or-BAC[4] [24]. Or-BAC is based on Rule based access control (Rule-BAC) where access control policies

[4] Organization Based Access Control.

are centered around the concept of organization. In these models access control policies are defined as set of rules, i.e. Condition → Authorization where Condition is a set of constraints over the subjects, actions and objects. In [27, 28] the authors analyzed conflict management for Rule-BAC Model and pointed out several failures , e.g. Rule-BAC is only capable of detecting actual conflicts not potential ones as the required computation is shown to be undecidable. Later, they showed how to manage conflicts in Or-BAC where the concepts of role, activity and view are used to specify the policy independently from concrete implementation of subjects, actions and objects in the system. Similarly to Rule-BAC, they assigned priorities to access control rules for managing conflicts in Or-BAC. Nevertheless, they overcame the difficulties of Rule-BAC using inheritance mechanisms and separated constraints specification [28] with tractable problems computable in polynomial time. They also described a tool called MotOrBAC, for managing conflicts.

eXtensible Access Control Meta Language (XACML) [19] is nowadays a standard for fine grained authorization for Web Services (WS). Underlying this language there is an access control model that supports several enforcement points connected to one or more centralized decision points. While this is the major benefit for specifying distributed access control policies in companies with distributed architectures [20], it could ends up as a major drawback. Subjects, Resources, Roles and other RBAC primitives might be shared in specific XACML policies applicable to requesters. Internally to XACML, the standard provides rules and policies combination algorithms (Deny-overrides, Permit-overrides, First-one-applicable and Only-one-applicable) with different strategies (based on priorities) for solving clashes between XACML policies sharing RBAC primitives[5]. Nevertheless, in some cases, administrators prefer to highlight the clashes between the access control policies prior to the application of any combination algorithm. In [29] the authors proposed a conflict analysis approach using Free Variable Tableaux. Using this approach they were able to detect modality, propagation, separation of duties and time constraint conflicts. Furthermore, they provided a friendly representation of the conflict's cause. With a different motivation, [15] studied the potential access control conflicts in virtual organizations (VO). They motivated their approach by using two possibly conflicting decision makers, i.e. the resource owner and the data owner. Even if regulations are firm toward the decision that have to be taken in those cases, still at the service level these conditions have to be verified. On contrary to previous approaches, this approach allows parties of the VO to specify their preferences to the administrator concerning the integration approaches for their policies with the policies of the other parties.

Policy based security at the network layer had its share of attention [69, 70]. Configuring security policies for firewalls that provide flexible traffic control and data protection schemes for IP networks, is a critical task. The thousands of policies that exist in different devices of the network have to be checked for intra- and inter-policy conflicts. In [70] they indicated that 30% of expert system administrators made configuration mistakes that lead to serious policy conflicts. In this work

[5] Technically in the target tag of XACML language.

the authors classified conflicts through filtering network security policies. They presented guidelines for identifying and rectifying conflicts in traffic flow control and protection. Specifically, they highlighted shadowing and spuriousness conflicts in traffic flow control, as well as for nested/overlapping security sessions in traffic protection. In another work [69] the author presented the Firewall Policy Advisor for filtering and protecting firewall policy from rule anomalies. They defined a number of firewall policy anomalies in both centralized and distributed firewalls and then proved that these are the only conflicts that could exist in firewall policies. They provided a tool that alarms administrators to change conflicting policies.

The composition of security solutions might not occur on security shared data in all cases. Interestingly an old, but still tickling, area of research enlightened us on such contexts. In Section 3.3 we show how conflicts are managed in feature-interaction approaches.

3.3 Composition of Security Solutions Sharing Technical and Functional Context

It is obvious that a functional resource could be the data to be secured by a composed security solution. This part of conflicts have been tackled mainly through the feature interaction community, initially dedicated for Telecommunication [87, 71]. The typical scenario for feature interaction is the following telephony one that we show briefly for clarity (taken from [87]).

A subscribes to originating call screening (OCS), with user C on the screening list, and user B subscribes to call forwarding (CF) to user C. If A calls B, and the call is forwarded to C, as prescribed by Bs feature CF, then As feature OCS is compromised. Clearly, if the call is not forwarded, then the CFB feature is compromised. These kinds of interactions can be very difficult to detect (and resolve), particularly since different features may be activated at different stages of the call cycle, and indeed at different locations both outside and within the network.

In a software system, functionality can be thought of as a feature. In order to make the complexity of modern software systems manageable their functionality is increasingly being decomposed into features. A survey dedicated to feature interaction is presented in [71]. In [71] we find an intuitive definition for *feature* as a set of logically-related requirements and their specifications, intended to deliver a particular behavioral effect. A feature often delivers tangible end user value [78, 79]. In contrast to security requirements, in this case a requirement is an expected functional behavior from the system. Adding up features or subtracting them in the system might conduct to unpredicted situations. This is logically sound to our earlier outline presented all along this chapter. The system in this case is the shared context and each feature is similar to a security solution that is added. Hence, if more than one feature share the same resource(s) then they can influence each others behavior. This interaction could be good in the sense when the modification of the system behavior is functionning as the planned desired one. Nevertheless if it results in undesired behavior then it is a bad interaction. This problem is known as feature

interaction problem [80, 71, 82, 81]. Feature interactions which lead to conflicts are obviously bad ones and are generally the subject matter of feature interaction problem. Undesirable means that there won't exist a system that can run properly with the integrated features mutually available and running. Feature interaction problem is one of the major challenges of feature-based software development. Research in feature interaction deals with the avoidance, detection and resolution of feature interactions [83, 80, 82, 81, 84, 85].

Manual detection of all conflicts and dependencies is inefficient and error-prone. Automatic detection of conflicts and dependencies is the requirement of feature-based software engineering today. There are different formal approaches, like logic-based, state-based, graph-based, etc. [87]. Graph transformation is an approach supported by available tools like AGG [86] to study feature interaction problem and detect conflicts and dependencies automatically. Graph is used as an abstract representation of many problems in Computer research. We select such a common approach to describe it based on a special type of directed graph called attributed graph.

An attributed graph $G = (V, E, s, t)$ is a directed graph consisting of a set V of vertices and a set E of edges.
Source and target functions $s, t : E \rightarrow V$ respectively return the initial node and the final node of an edge. Moreover, both the vertices and the edges of G are decorated by a number of attributes, i.e. names with a value and a type.

The graph defined above is basically a multigraph which allows multiple edges between two given vertices. Graph transformation is a rule-based modification of a Graph G into a graph H. The rules are the production of the graph grammar. Graph grammar approaches transform attributed graphs using graph grammars. There are different graph grammar approaches, mainly matrix graph grammars [88] and category theory-based graph grammars [88]. The second one is supported by practical tools such as AGG [86] based on graph transformation. For conflict and dependency detection for our problem it is sufficient.

In a graph transformation two kinds of non-determinism are observed: for each production several matches may exist and several rules might be applicable in different orders. Considering the case where two graph transformations can be applied to the same host graph, the result might be the same, regardless of the application order. Otherwise, if one of two alternative transformations is not independent of the second, the first will disable the second. In this case, the two rules are in conflict. Conversely, two transformations are said to be parallel independent if they modify different parts of the host graph and ultimately end up to the same graph. This situation can be described by the mathematical property called confluence in the parlance of general rewriting system. It is known that graph transformation can be thought of as a graph rewriting system [87].

In brief, confluence can be defined using *reduction sequence* that terminates to an element after several reduction steps[6]. This can be illustrated as follow: Let a, b,

[6] A reduction sequence that terminates for w_n, is written as: $W \rightarrow_I^* W_n$.

$c \in S$, with $a \rightarrow_* b$ and $a \rightarrow_* c$. If a is confluent, there exists a $d \in S$ with $b \rightarrow_* d$ and $c \rightarrow_* d$.

There are different variants of confluence, like local confluence, semi-confluence, strong confluence, etc. Interested readers may refer to [87] for more details.

If a system is not confluent, it gives rise to conflicting situations. Detection of the conflicting situations can be done by Critical Pair Analysis (CPA). CPA is known from term rewriting and used there to check if a term rewriting system is confluent. It has been generalized to graph rewriting. Critical pairs formalize the idea of a minimal example of a conflicting situation. From the set of all critical pairs we can extract the objects and links which cause conflicts or dependencies. We invite readers to consult the short manual provided by [53] for deeper understanding of AGG and CPA.

Conflicting and depending rules are called critical pairs. AGG (Attributed Graph Grammar) tool provides all the graph transformation steps discussed above. Using AGG we can find out conflicts and dependencies using CPA. The CPA GUI gives clear and sufficient information about critical pairs [86].

Nevertheless, as feature is an abstract concept that represents the functional behavior of part of a system, several other approaches conducted different kind of analysis. In software engineering, for instance, they are represented in requirement, static and dynamic views of UML [83]. UML is one of the common means for representing static and dynamic views through case, sequence and state diagrams. Moreover, in Aspect Oriented Programming support for Separation of Concerns in software development was modeled as features. Research studies were conducted to discover undesirable interactions between different concerns or aspects using feature interactions[48, 49]. These feature interaction based approaches are not appropriated to security solutions rather only used as means for checking a consistent combination. Scalability could be the burden for such research direction. Therefore they are incapable of handling our requirement for the integration of security solutions.

4 Smart Items Case Study

We have introduced the necessity for conflict management approaches earlier in this chapter, then we have detailed several approaches targeting different application layers. Specific assessments were provided correspondingly to these approaches. A realistic assessment should position all these approaches (in addition to their own objectives) with respect to the industrial needs in developing secure products.

This section discusses these approaches with respect to an industrial case study (simulating real life applications) developed to promote remote healthcare assistance to elderly people. This application extends traditional Tele-Cardiology applications using the facilities of a domestic house[7] and other intelligent devices. This

[7] The domestic house, or smart home, is provided by the Domus Laboratory at the University of Sherbrooke, online description is available at **http://domus.usherbrooke.ca/?locale=en**.

system should support the discovery, interaction and collaboration among doctors, pharmacists, patients, social workers and emergency medical teams in the health care realm and, in particular, during emergency situations. Obviously, in our quest toward secure applications we need to combine security requirements, possibly conflicting ones.

Briefly, patient's health condition can be monitored through various wearable medical sensors worn as washable smart T-shirts. All these sensors form the Body Sensor Network (BSN). The measured data are collected and pre-processed by a personal mobile hub such as Smart Phones. Similarly, the patient's house is equipped with a sensor network and a local server, which centrally processes the sensor data for monitoring the activity of the patient and the environmental setting. In the remainder, we refer to it as the smart home [73, 72]. The information collected by the BSN and smart home are sent to the Monitoring and Emergency Response Centre (MERC), the organization responsible for the maintenance and storage of patient medical data, such as the Electronic Health Record (EHR). MERC processes such data to have a constant snapshot of the patients' health status so as to promptly initiate proper healthcare procedures when a potential emergency alert is identified. Each actor (e.g., doctors, social workers, etc.) is provided with an eHealth terminal, i.e. a PDA, which runs eHealth software designed to support medical requests and reports in compliance with MERC. In this setting, MERC and the other actors within the system have to process collected data and protect them from unauthorized access along the lines set by the actual data protection regulations, like the Directive 95/46/EC [16] of the EU.

Among the possible application scenarios in the remote healthcare system, we focus on an emergency situation. In the case of alert, the rescue request with patient's location is sent by MERC to the emergency team asking for assisting the patient. The assigned rescuers are granted access to the patient's EHR and last medical data collected by the BSN. When the patient is found and rescued, the emergency team sends a notification to MERC with comments regarding medicines administrated to the patient. The details of this scene are depicted in Figure 3 through the Web Services and clients' Graphical User Interfaces orchestrated by MERC.

The prototype is implemented using the Service Oriented Architecture paradigm (SOA). SOA is a blueprint for an adaptable, flexible, and open IT architecture for developing service-based, enterprise-scale business solutions. An enterprise service is typically a set of Web Services combined with business logic that can be accessed and used repeatedly to support a particular business process. In our implementations, Business Process Execution Language (BPEL) is used for orchestrating the set of Web services involved in an enterprise service. The emergency scene[8] presents strong security, dependability and privacy requirements. We reported few of them in Table 1. We highlight the security requirements for one particular actor in one act of this scenario, namely Req 4 and Req 6. The MERC shall keep the EHR data confidentially stored, the communication with any of the other actors guaranteeing integrity and confidentiality of the data exchanged.

[8] The emergency scene is one of the two scenes demonstrated at Information and Communication Technologies (ICT) 2008 [74].

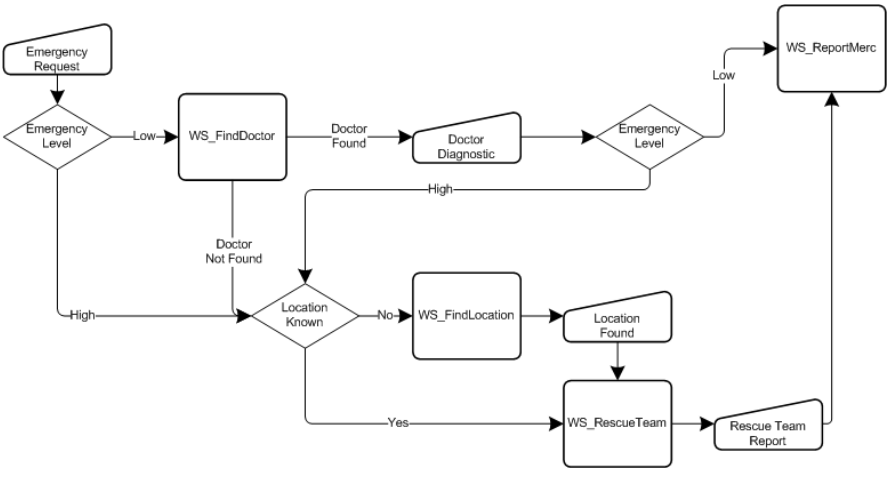


Fig. 3 Emergency Situation of the Smart Items Case Study

Table 1 Some sample security, dependability and privacy requirements for the Emergency scene

Req 1	The Smart T-Shirt data should be kept confidential for all requesters except the MERC.
Req 2	The doctor discovery process shall successfully terminate in 1 minute (i.e., one and only one doctor shall proceed in assisting the patient). Beyond one minute, the MERC shall manage the situation as an emergency and invoke the rescuers.
Req 3	The system shall guarantee that the commitment of actors (e.g., doctors, rescue teams) to actions cannot be later repudiated.
Req 4	Each communication between MERC and the e-health terminals of the selected doctor and of the medical team shall guarantee integrity and confidentiality of the data exchanged.
Req 5	Similarly, each communication between the e-health terminal of the selected doctor, the medical team and the patient e-health terminal shall guarantee integrity and confidentiality of the data exchanged.
Req 6	The selected doctor and the medical team using the e-health terminal shall remotely be identified, authenticated and granted access to the patient's Electronic Health Record to retrieve his health status.

Technically, the MERC has to orchestrate between all the actors of the scenario. The means used relies on different architectures such as $client-server$, e.g. when the doctor is consulting the patient's diagnostics or the MERC agents localizing the patient, $peer-to-peer$, e.g. in cases where the Web Service (WS) firing the patient's assistant request is invoking the emergency WS that initiates the activeBPEL workflow of the MERC, and others such as $publish-\&-subscribe$ that falls into our other scenes.

To satisfy these requirements we adopted the security patterns approach where it is possible to deploy security solution as COTS with easy to use implementation [9, 74]. Still our aim is to apply much more security solutions from different layers captured as security patterns[9]. We deployed (i) fine grained authorization with XACML implementation [51], (ii) Brokered authentication using WS Trust implementation [9], (iii) Secure Channel using SSL with client certificate [9], (iv) Trusted Platform Module [89] and still need much more. The approach we have demonstrated in our previous work showed how easily software developers can apply security solutions by means of easy-to-use implementations of security patterns [74]. Nevertheless these solutions still without any proof that their integraton over shared context is still correct as it was independently. Actually this isn't feasible because it challenges the scalability of all the approaches presented in Section 3. As a matter of fact the security solutions/patterns listed from (i) to (iv) can be verified, in some cases, separately by the conflict management approaches shown in this chapter. From Section 3.1 *experts* can detect if the policies administrated at the MERC level violate SoD constraints, from Section 3.2 *experts* can check whether the combined communication channels still provide their corresponding requirements, and as a final example *experts* can borrow approaches from 3.2 to provide tools for security administrators to correctly configure the access control policies. It is true that combining security protocols might not be required in our scenario but it is of use in online video store application and the like. Our work on security patterns bridge the gap between experts and novice security users and provide security to software developers. Still such an approach can not be adopted without corresponding approaches for managing conflicts of combined security solutions suitable for software developers. Eventually, feature interaction problems and their approach in handling conflicts from the integration of security solutions seems the most reasonable starting point toward conflict management approaches for security patterns.

5 Conclusion and Future Work

Today's pioneer organizations recognize that performance accelerates when information security is driven into the very framework of a business. Business applications are moving from standalone systems to Service-Oriented Architectures. This collaboration can use Information Technology to achieve a closer integration and better management of relationships between internal and external parties. However, the current practice of security engineering is hampered by the fact that it is not considered as an integral part of system engineering. As a result, implemented security solutions often need to be integrated. In this chapter we reviewed the main approaches regarding integration of security mechanisms. We classified and then described the types of conflicts that may arise when integrating security mechanisms. We reviewed several approaches and outlined their commonalities

[9] A security pattern describes a particular recurring security problem that arises in specific contexts, and presents a well-proven generic solution for it [52, 51, 54]. Our approach for security patterns adopts the SERENITY approach [9].

regarding their ways of addressing conflict detection and their differences regarding their ways of dealing with conflicts resolution. We've already started the work on our new approach based on features interaction which is also scalable to other security solutions including security protocols. We consider to improve it by considering the support of intruder models for analysing and comparing existing approaches. Furthermore we plan on considering also the integration of security solution over modelling languages such as UML, particularly UMLsec ans SecureUML to allow reasoning mechanisms for software developers adopting those kind of approaches.

Acknowledgements. This work is partially funded by the EU projects COMPAS.

References

1. Simon, R., Zurko, M.E.: Separation of duty in role based access control environments. In: Proceedings of the 10th IEEE Workshop on Computer Security Foundations, Rockport, MA, June 10-12, pp. 183–194. IEEE Computer Society Press, Los Alamitos (1997)
2. Kuhn, D.R.: Mutual exclusion as a means of implementing separation of duty requirements in role-based access control systems. In: Proceedings of the 2nd ACM Workshop on Role-Based Access Control, Fairfax, VA, pp. 23–30. ACM Press, New York (1997)
3. Nyanchama, M., Osborn, S.: Role-based security, object oriented databases and separation of duty. SIGMOD Rec. 22(4), 45–51 (1993)
4. Nyanchama, M., Osborn, S.: The role graph model and conflict of interest. ACM Transactions on Information and System Security (TISSEC) 2(1), 3–33 (1999)
5. Nyanchama, M., Osborn, S.: Access rights administration in role-based security systems. In: Proceedings of the IFIP Working Group 11.3 Working Conference on Database Security. Elsevier North-Holland, Inc., Amsterdam (1994)
6. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling Security Requirements Through Ownership, Permission and Delegation. In: Proceedings of the 13th IEEE International Requirements Engineering Conference (RE 2005), pp. 167–176. IEEE Computer Society Press, Los Alamitos (2005)
7. Johnston, D.: Russian accused of citibank computer fraud, August 18. The New York Times (2007)
8. van Lamsweerde, A., Darimont, R., Letier, E.: Managing Conflicts in Goal-Driven Requirements Engineering. TSE 24(11), 908–926 (1998)
9. Benameur, A., Khoury, P.E., Seguran, M., Sinha, K.S.: Serenity in e-Business and Smart Items Scenarios. In: Spanoudakis, G., Mana Gomez, A., Spyros, K. (eds.) The Security and Dependability for Ambient Intelligence Series: Advances in Information Security, vol. 55, pp. 375–392. illus (2009) ISBN: 978-0-387-88774-6
10. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Detecting Conflicts of Interest. In: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE 2006), pp. 315–318. IEEE Computer Society Press, Los Alamitos (2006)
11. Basin, D., Doser, J., Lodderstedt, T.: Model Driven Security: from UML Models to Access Control Infrastructures. TOSEM 15(1), 39–91 (2006)
12. Shin, M.E., Ahn, G.-J.: UML-Based Representation of Role-Based Access Control. In: Proc. of WETICE 2000, pp. 195–200. IEEE Press, Los Alamitos (2000)
13. Ray, I., Li, N., France, R., Kim, D.-K.: Using UML to visualize role-based access control constraints. In: Proc. of SACMAT 2004, pp. 115–124. ACM Press, New York (2004)

14. Trimarchi, P.: Istituzioni di diritto privato, XVI edn. Giuffr'e Editore (2005)
15. Mazzoleni, P., Bertino, E., Crispo, B., Sivasubramanian, S.: XACML policy integration algorithms: not to be confused with XACML policy combination algorithms! In: Proceedings of the eleventh ACM symposium on Access control models and technologies, Lake Tahoe, California, USA, June 07-09 (2006)
16. European Parliament. European directive on data privacy 95/46/CE (1995), http://www.cdt.org/privacy/eudirective/EUDirective.html (accessed December 1, 2008)
17. Fusaro, P.C., Miller, R.M.: What Went Wrong at Enron: Everyone's Guide to the Largest Bankruptcy in U.S. History. Wiley, Chichester (2002)
18. HIPAA. U.s. government department of health and human services health. Insurance Portability and Accountability Act (1996)
19. OASIS. Security services technical committee. eXtendible Access Control Markup Language Committee specification 2.0 (2005)
20. Lorch, M., Proctor, S., Lepro, R., Kafura, D., Shah, S.: First experiences using xacml for access control in distributed systems. In: XMLSEC 2003: Proceedings of the 2003 ACM workshop on XML security, pp. 25–37. ACM Press, New York (2003)
21. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: An access control model supporting periodicity constraints and temporal reasoning. ACM Transactions on Database Systems (TODS) 23(3), 231–285 (1998)
22. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-Based Access Control Models. Computer 29(2), 38–47 (1996)
23. Joshi, J., Bertino, E., Latif, U., Ghafoor, A.: Generalized Temporal Role Based Access Control Model. IEEE Transactions on Knowledge and Data Engineering 7(1) (2005)
24. Abou El Kalam, A., El Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miège, A., Saurel, C., Trouessin, G.: Organization Based Access Control. In: IEEE 4th International Workshop on Policies for Distributed Systems and Networks, Policy 2003 (2003)
25. Jajodia, S., Samarati, P., Sapino, M.L., Subrahmanian, V.S.: Flexible support for multiple access control policies. TODS 26(2), 214–260 (2001)
26. Samak, T., Al-Shaer, E., Li, H.: QoS Policy Modeling and Conflict Analysis. POLICY (2008)
27. Cuppens, F., Cuppens-Boulahia, N., Ben Ghorbel, M.: High-level conflict management strategies in advanced access control models. In: Workshop on Information and Computer Security, Timisoara, Romania (2006)
28. Cuppens, F., Miège, A.: Conflict management in the Or-BAC model, Technical report, ENST Bretagne, France (2003)
29. Kamoda, H., Yamaoka, M., Matsuda, S., Broda, K., Sloman, M.: Policy Conflict Analysis Using Free Variable Tableaux for Access Control in Web Services Environments. In: WWW2005 Workshop 14th International World Wide Web Conference (2005)
30. Lupu, E.C., Sloman, M.: Conflicts in policy-based distributed systems management. IEEE Transactions on Software Engineering 25(6), 852–869 (1999)
31. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory IT-29, 198–208 (1983)
32. Derek, A.: Formal Analysis of Security Protocols: Protocol Composition Logic, Ph.D thesis, Computer Science Department, Stanford University (2006)
33. Cremers, C.: Scyther - Semantics and Verification of Security Protocols. Ph.D thesis, Computer Science Department, Eindhoven University of Technology (2006)

34. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuellar, L., Drielsma, P.H., Heam, P., Kouchnarenko, O., Mantovani, J., Modershei, S., von Oheimb, D., Rusinowitch, M., Santiago, J., Turuani, M., Vigano, L., Vigneron, L.: The AVISPA tool for the automated validation of internet security protocols and applications. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 281–285. Springer, Heidelberg (2005)
35. Holzmann, G.: Design and Validation of Computer Protocols. Prentice Hall, Englewood Cliffs (1991)
36. Cremers, C.: Compositionality of security protocols: a research agenda. In: Vodca 2004, Bertinoro, Italy. ENTCS, vol. 142(3), pp. 99–110 (2006)
37. Cremers, C.: Feasibility of Multi-Protocol Attacks. In: Proceedings of The First International Conference on Availability, Reliability and Security, pp. 287–294. IEEE Computer Society Press, Los Alamitos (2006)
38. Kelsey, J., Schneier, B., Wagner, D.: Protocol interactions and the chosen protocol attack. In: Security Protocols Workshop, pp. 91–104 (1997)
39. Tzeng, W., Hu, C.: Inter-protocol interleaving attacks on some authentication and key distribution protocols. Inf. Process. Lett. 69(6), 297–302 (1999)
40. Gong, L., Syverson, P.: Fail-stop protocols: An approach to designing secure protocols. In: Proc. of the 5th International Working Conference on Dependable Computing for Critical Applications, pp. 44–55 (1995)
41. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report (2000)
42. Guttman, J., Thayer, F.: Protocol independence through disjoint encryption. In: PCSFW: Proc. of the 13th Computer Security Foundations Workshop IEEE (2000)
43. Datta, A., Derek, A., Mitchell, J.C., Roy, A.: Protocol Composition Logic (PCL). Electronic Notes in Theoretical Computer Science, vol. 172, pp. 311–358 (2007)
44. Cremers, C.: On the Protocol Composition Logic PCL. In: ASIACCS 2008: Proceedings of the ACM Symposium on Information, Computer and Communications Security, Tokyo, Japan, pp. 66–76 (2008)
45. Datta, A., Derek, A., Mitchell, J., Pavlovic, D.: A derivation system and compositional logic for security protocols. Journal of Computer Security 13(3), 423–482 (2005)
46. Backes, M., Pfitzmann, B., Waidner, M.: A universally composable cryptographic library. In: Proceedings of the 10th ACM Conference on Computer and Communications Security (2003)
47. Ngo, L., Tarkoma, S., Laud, P.: Extending a universally composable cryptographic library. Master thesis. Helsinki University of Technology (2008)
48. Beltagui, F.: Features and Aspects: Exploring feature-oriented and aspect-oriented programming interactions. Technical Report No: COMP-003-2003. Computing Department, Lancaster University (2003)
49. Kojarski, S., Lorenz, D.: FIdentifying Feature Interactions in Multi-Language Aspect-Oriented Frameworks. In: Proceedings of the 29th International Conference on Software Engineering (ICSE 2007), Minneapolis, MN, May 20-26, pp. 147–157. IEEE Computer Society, Los Alamitos (2007)
50. Liu, Z.: Manage Component-Specific Access Control with Differentiation and Composition, Technical Report Indiana University (2001)
51. Sanchez-Cid, F., Munoz, A., El Khoury, P., Compagna, L.: XACML as a Security and Dependability (S&D) pattern for Access Control in AmI environments. In: Proc. of AmI.d 2007, pp. 143–155. Springer, Heidelberg (2007)

52. Compagna, L., El Khoury, P., Massacci, F., Thomas, R., Zannone, N.: How to capture, communicate, model, and verify the knowledge of legal, security, and privacy experts: a pattern-based approach. In: Proc. of ICAIL 2007, pp. 149–154. ACM Press, New York (2007)
53. Taentzer, G.: AGG: A Graph Transformation Environment for Modeling and Validation of Software. In: Applications of Graph Transformations with Industrial Relevance, pp. 446–453 (2004) ISBN: 978-3-540-22120-3
54. Cuevas, A., El Khoury, P., Gomez, L., Laube, A.: Security Patterns for Capturing Encryption-Based Access Control to Sensor Data. In: Proc. of SECURWARE 2008, pp. 62–67. IEEE Press, Los Alamitos (2008)
55. Braga, A., Dahab, R., Rubira, C.: Composing Cryptographic Services: A Comparison of Six Cryptographic APIs. Technical Report IC-99-05, Institute of Computing, State University of Campinas, Sao Paulo, Brazil (1999)
56. Braga, A., Dahab, R., Rubira, C.: A Meta-Object Library for Cryptography. Technical Report IC-99-06, Institute of Computing, State University of Campinas. Campinas, Sao Paulo, Brazil (1999)
57. Borisoff, D., Victor, D.: Conflict Management: A Communication Skills Approach, 2nd edn. Allyn & Bacon (October 24, 1997) ISBN-13: 978-0205272945
58. Schneier, B.: Applied Cryptography, 2nd edn. John Wiley and Sons, Chichester (1996)
59. Menezes, A., van Orschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
60. Stroud, R., Wu, Z.: Using Metaobject Protocols to Satisfy Non-Functional Requirements. In: Object-Oriented Meta-Level Architectures and Reflection, ch. 3, pp. 31–52 (1996)
61. Fabre, J.-C., Perennou, T.: Friends: A Flexible Architecture for implementation of Fault Tolerant and Secure Distributed Applications. In: Hlawiczka, A., Simoncini, L., Silva, J.G.S. (eds.) EDCC 1996. LNCS, vol. 1150, pp. 3–20. Springer, Heidelberg (1996)
62. Davis, D.: Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML. In: USENIX Annual Technical Conference, General Track, pp. 65–78 (2001)
63. RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2
64. Pfitzmann, B., Waidner, M.: Composition and Integrity Preservation of Secure Reactive Systems. CCS, Greece (2000)
65. Backes, M., Pfitzmann, B., Waidner, M.: Symmetric authentication within a simulatable cryptographic library. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 271–290. Springer, Heidelberg (2003)
66. Johnson, D., Dolan, G., Kelly, M., Le, A., Matyas, S.: Common Cryptographic Architecture Cryptographic Application Programming Interface. IBM Systems Journal 30(2), 130–150 (1991)
67. Kaliski, B.: Cryptoki: A Cryptographic Token Interface, Versopn 1.0 (1995), http://www.rssa.com/rsalabs/pubs/PKCS/html/pkcs-11.html
68. Microsoft Corporaton. Application Programmer's Guide: Microsoft CryptoAPI. Version 2.0 (1996)
69. Al-Shaer, E., Hamed, H.: Taxonomy of Conflicts in Network Security Policies. IEEE Communications Magazine 44(3), 134–141 (2006)
70. Al-Shaer, E., Hamed, H., Boutaba, R., Hasan, M.: Conflict Classification and Analysis of Distributed Firewall Policies. IEEE Journal on Selected Areas in Communications 23(10), 2069–2084 (2005)
71. Nhalabatsi, A., Laney, R., Nseibeh, B.: Feature Interaction: The Security Threat from Within the Software Systems. Progress in Informatics, Special Issue: The future of software engineering for security and privacy 5, 75–89 (2008)

72. Busnel, P., Khoury, P.E., Giroux, S., Li, K.: Achieving Socio-Technical Confidentiality using Security Pattern in Smart Homes. In: Proceedings for the Third International Symposium on Smart Home (2008)
73. Pigot, H., Mayers, A., Giroux, S.: The intelligent habitat and everyday life activity support. In: Proceedings of the 5th international conference on Simulations in Biomedicine, Slovenia, pp. 507–516 (2003)
74. Khoury, P.E., Li, K., Busnel, P., Giroux, S.: Serenity demo: Secure remote healthcare environment using serenity. In: Information and Communication Technologies, Lyon, France (2008)
75. Bauer, L., Garriss, S., Reiter, M.K.: Detecting and resolving policy misconfigurations in access-control systems. In: SACMAT 2008: Proceedings of the 13th ACM symposium on Access control models and technologies, pp. 185–194. ACM, New York (2008)
76. Khoury, P.E., Coquery, E., Hacid, M.: Consistency Checking of Role Assignments in Inter-Organizational Collaboration. In: Proceedings for the 1st ACM GIS Workshop on Security and Privacy in GIS and LBS. ACM, New York (2008)
77. Anderson, R.: Why cryptosystems fail. COMM 37(11), 32–40 (1994)
78. Cheng, K.E., Ohta, T. (eds.): Feature Interactions in Telecommunications Systems III. IOS Press, Amsterdam (1995)
79. Dini, P., Boutaba, R., Logrippo, L. (eds.): Feature Interactions in Telecommunication Networks IV. IOS Press, Amsterdam (1997)
80. Felty, A., Namjoshi, K.: Feature Specification and Automated Conflict Detection. ACM Transactions on Software Engineering and Methodology 12(1), 3–27 (2003)
81. Kamoun, J., Logrippo, L.: Goal-oriented feature interaction detection in the intelligent network model. In: Feature Interactions in Telecommunications and Software Systems V (1998)
82. Keck, D.O., Kuehn, P.J.: The feature and service interaction problem in telecommunications systems: A survey. IEEE Trans. Softw. Eng. 24(10), 779–796 (1998)
83. Jayaraman, P., Whittle, J., Elkhodary, A., Gomaa, H.: Model Composition in Product Lines and Feature Interaction Detection Using Critical Pair Analysis. In: Engels, G., Opdyke, B., Schmidt, D.C., Weil, F. (eds.) MODELS 2007. LNCS, vol. 4735, pp. 151–165. Springer, Heidelberg (2007)
84. Douence, R., Fradet, P., Sudholt, M.: Composition, reuse, and interaction analisys of stateful aspects. In: Proceedings of the 3rd international Conference of Aspect-oriented Software Development, Lancaster, UK. ACM, New York (2004)
85. Kolberg, M., Magill, E., Marples, D., Tsang, S.: Feature interactions in services for networked appliances. In: IEEE International Conference on Communications, New York, USA (2002)
86. AGG Homepage, http://tfs.cs.tu-berlin.de/agg
87. Calder, M., Kolberg, M., Magill, E., Reiff-Marganiec, S.: Feature Interaction: A Critical Review and Considered Forecast. Computer Networks: The International Journal of Computer and Telecommunications Networking 41(1), 115–141 (2003)
88. Biggs, N., Lloyd, E., Wilson, R.: Graph Theory, pp. 1736–1936. Oxford University Press, Oxford (1986)
89. Gurgens, S., Rudolph, C., Mana, A., Munoz, A.: Facilitating the Use of TPM Technologies through S&D Patterns. In: SPatterns, DEXA Workshops, pp. 765–769 (2007)

Monitoring-Based Approach for Privacy Data Management*

H. Meziane, S. Benbernou, F. Leymann, and M.P. Papazoglou

Abstract. This chapter addresses the problem of managing private data in service based applications ensuring end-to-end quality of service(QoS) capabilities. The proposed approach is processed through monitoring the compliance of privacy agreement that spells out a consumer's privacy rights and how consumer private information must be handled by the service provider. A state machine based model is proposed to describe the Private Data Use Flow (PDUF) toward monitoring which can be used by privacy analyst to observe the flow and capture privacy vulnerabilities that may lead to non-compliance. The model is built on top of (i) properties and timed-related privacy requirements to be monitored that are specified using LTL (Linear Temporal Logic) (ii) a set of identified privacy misuses.

1 Introduction

The huge recent increase in web-based applications carried out on the Internet has accompanied by an exponential amount of data exchanged by the interacting entities through web-services and the growth of consumer awareness of their lack of

H. Meziane
LIRIS, University Claude Bernard Lyon1, France and University of Oran, Algeria
e-mail: meziane.hassina@univ-oran.dz

S. Benbernou
LIRIS, University Claude Bernard Lyon1, France
e-mail: sbenbern@liris.univ-lyon1.fr

F. Leymann
IAAS, University Stuttgart, Germany
e-mail: frank.leymann@iaas.uni-stuttgart.de

M.P. Papazoglou
INFOLAB, Tilburg University, Netherlands
e-mail: M.P.Papazoglou@uvt.nl

* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 225–247.
springerlink.com

privacy. Web services are available for banking, shopping, learning, healthcare, and government online. In the beginning, the interest of researchers and practitioners has converged on the functional aspects of those software services and their description. Because of the increasing agreement on the implementation and management of the functional aspects of those services, the interest of researchers is shifting toward the 'non-functional' or quality aspects of web-enabled services including security, privacy, availability, accessibility, etc. Most of these services require the consumer's personal information in one form or another which makes the service provider in the possession of a large amount of consumer private information along with the accompanying concerns over potential loss of consumer privacy. In fact, as the amount of exchanged information exponentially grows, the number of inappropriate usage and leakage of personal data is increasing, privacy has emerged and is becoming one of the most important and the most crucial concerns and challenging issues. It is today one of the major concerns of users exchanging information through the web, including service requesters, service providers and legislators. Everyone who has purchased anything from the Internet had led the experience of pausing and wondering if is "safe" to enter one's credit card information. Clearly, the more one is exposed to new services on the Internet and the varied personal information that is demanded, by theses services, the more one wonders whether the personal information that ones enters would be kept safe. The search problem faced by Internet users today is not the lack of information from searches, but the challenge is how the web-based applications are more trustworthy to control the private data usage to keep more confidentiality. Such a need, leads to built and manage service-based systems which provide desired end-to-end QoS awareness. Traditionally, access control to any kind of data (e.g.private) has dealt only with authorization decisions on a subject's access to target resources. Obligations are requirements that have to be fulfilled by the subject for allowing access. Conditions are subjects and object-independent environmental requirements that have to be satisfied for access. In today's highly dynamic, distributed environment, obligations and conditions are also crucial decision factors for richer and finer controls on usage of data resources. More precisely, the challenge of private data management is how to do *usage control*, knowing that the private data is already used. In fact, while *access control* aspect of security and privacy is well understood, it is unclear of how to do *usage control*.

The need is to assess the health of systems that implement Web services. We investigate the self-protecting service management. We are sensitive to build system which anticipates, detects hostile activities dealing with the private data, identifies, and protects against threats.

In response to the privacy concerns quoted above, in [5] we proposed a *privacy agreement* model that spells out a set of requirements related to consumer's privacy rights in terms of how service provider must handle privacy information. The properties and private requirements can be checked at a design time prior to execution, however, the monitoring of the requirements at run-time has strong motivations since those properties can be violated at run time. Thus, checking at run-time the

compliance of the requirements defined in the privacy agreement is a challenging issue. That issue must be properly addressed otherwise it could lead to agreement breaches and to lower service quality. Indeed, the private data use flow must be observed which means monitoring the behaviour of the privacy agreement. From the results of the observations, analysis can be done to come up to an understanding, why the non-compliance took place and what remedy will be provided enhancing the privacy agreement.

The common approach developed to support requirements monitoring at run-time assumes that the system must identify the set of the requirements to be monitored. In fact, as part of the privacy agreement model, the set of privacy requirements to be monitored are needed from which *monitoring private units* are extracted and their occurrences at run-time would imply the violation of the requirements. Besides the functional properties (e.g operations of the service), the time-related aspects are relevant in the setting of the privacy agreement. In addition, the non-compliance or failing to uphold the privacy requirements are manifested in terms of vulnerabilities must be identified.

In this chapter, we propose an approach for the management of privacy data terms defined in the privacy agreement at run-time. The approach features a model based on state machine which is supported by *abstractions* and *artifacts* allowing the run-time management. Our contribution articulates as follows:

1. From the privacy requirements defined in the privacy agreement, we extract a set of *monitoring private units* specified by the means of Linear Temporal Logic (LTL) formulas,
2. The set of privacy misuses is most likely met throughout the private data use is provided. That set is not limited and can be enriched by those promptly revealed when they occur in run-time and captured by the analysis,
3. A state machine based model is provided in order to describe the activation of each privacy agreement clauses, that is, it spells out the Private Data Use Flow (PDUF). The state machine supports abstractions and by the means of previous artifacts, the behaviour observations are expressed. It will *observe* which and when a clause is activated, or which and when a clause is violated and what types of vulnerabilities happened, or which clause is compliant and etc. Such observations lead to do reasoning to enhance the privacy agreement and enrich the knowledge on misuses.

The remainder of the chapter is structured as follows. We start by presenting an overview of the privacy agreement developed in our previous work in Section 2. In Section 3, we describe the architectural support for privacy data use flow monitoring. Section 4 proposes an LTL-based approach to specify the monitoring private units and presents a set of privacy misuses. In Section 5, we present the private data use flow model. In Section 6,7 we present the architecture of the framework and we discuss the prototype that we have developed to implement this framework. We discuss related work in Section 8 and conclude with a summary and issues for future work in Section 9.

2 Privacy Agreement Model

To make the chapter self containing, in this section we recall the privacy agreement model specified in our previous work [5, 7]. We proposed a framework for privacy management in Web services. A privacy policy model has been defined as an agreement supporting a lifecycle management which is an important deal of a dynamic environment that characterizes Web services based on the state machine, taking into account the flow of the data use in the agreement. Hence, WS-Agreement has been extended including privacy aspects. In this setting, the features of the framework are:

- The privacy policy and data subject preferences are defined together as one element called *Privacy-agreement*, which represents a contract between two parties, the service customer and the service provider within a validity. We provided abstractions defining the expressiveness required for the privacy model, such as rights and obligations.
- The framework supports lifecycle management of privacy agreement. We defined a set of events that may occur in the dynamic environment, and a set of change actions used to modify the privacy agreement. An *agreement-evolution* model is provided in the privacy-agreement.
- An *agreement-negotiation protocol* is provided to build flexible interactions and conversations between parties when a conflict happens due to the events occurring in the dynamic environment of the Web service.

Informally speaking the abstraction of privacy model is defined in terms of the following requirements:

- *data-right*, is a predefined action on data the data-user is authorized to do if he wishes to.
  We distinguish two types of actions (i) actions used to complete the service activity for the current purpose for which it was provided (ii) actions used by a service to achieve other activities than those for which they are provided.
- *data-obligation*, is the expected action to be performed by service provider or third parties (data- users) after handling personal information in data-right. This type of obligation is related to the management of personal data in terms of their selection, deletion or transformation.

Let us illustrate the motivations through the following example dealing with a purchase service where the transactions between the customer and the service is not considered in the chapter. Let us assume that the privacy policy of the service provider accepted by the customer is defined as follows: the service has the authorization to collect email address (email) and credit card number (ccn) to complete its activity for the current purpose i.e. the email is used to send invoices and credit card number for the payment of the invoices. Furthermore, the service provider can also use email address to achieve an extra activity for instance marketing purpose i.e. the email is used to send the available products and their prices.

Formally speaking, we define data-right and data-obligation as follows :

Definition 1. ***(data-right.)*** *A data-right r_d is a tuple (u,d,p,μ_{rd}), with $u \subseteq \mathscr{U}$ and $d \subseteq \mathscr{D}$ and $p \subseteq \mathscr{PO}$ and $\mathscr{R}^d = \{\{r_d^i\}_j \,/\, i>0\ j>0\}$, where $\mathscr{U}$ is the ontology of data users and $\mathscr{D}$ is the ontology of personal data and $\mathscr{PO}$ is the set of authorized operations identifying purposes of the service and μ_{rd} is the period of data retention (the data-right validity), and $\mathscr{R}^d$ is the set of data-rights.*

Example 1

1. $r^1_{email}(sp, email, send\ Invoice, \mu_{r1email})$,
   specifies that the service provider *sp* has the right to use *email* for sending invoices during the period $\mu_{r1email}$.
2. $r^2_{email}(sp, email, send\ Offer, [d_s, d_s + 1\ month])$,
   specifies that the service provider *sp* has also the right to use *email* for sending the available products and their prices during the period $\mu_{r2email}$ which is $1\ month$ after both sides have signed the agreement at d_s date.
3. $r_{ccn}(sp, ccn, payment\ Invoice, \mu_{rccn})$,
   specifies that the service provider *sp* has the right to use *ccn* for the payment of the invoices during the period μ_{rccn}.

Definition 2. ***(data-obligation.)*** *A data-obligation o_d is a tuple (u,d,a_o,μ_{od}) with $u \subseteq \mathscr{U}$ and $d \subseteq \mathscr{D}$ and $a_o \in \mathscr{A}_o$ and $\mathscr{O}^d = \{\{o_d^i\}_j \,/\, i>0\ j>0\}$, where $\mathscr{U}$ is the ontology of data users and $\mathscr{D}$ is the ontology of personal data and $\mathscr{A}_o$ a set of actions that must be taken by the data user and μ_{od} is an activated date of the obligation, and $\mathscr{O}^d$ is the set of data-obligations.*

Example 2

1. $o_{ccn}(sp, ccn, crypt, [d_{pay} + 1\ day])$,
   specifies that the service provider *sp* must crypt the *ccn* for a given data subject at the end of each payment process, for instance, at d_{pay}+1 day (μ_{occn}).
2. $o_{email}(sp, email, hide, \mu_{oemail})$
   specifies that the service provider *sp* must hide the email for a given data subject at μ_{oemail} i.e. when the authorization of email retention time is elapsed.

Based on those requirements, we formalized a privacy data model as follows :

Definition 3. ***(A privacy data model.)*** *A privacy data model $\mathscr{P}^d$ is a couple $< \mathscr{R}^d, \mathscr{O}^d >$, where $\mathscr{R}^d$ is the set of data-rights and $\mathscr{O}^d$ is the set of data-obligations.*

By means the proposed privacy model, we extended current WS-Agreement specifications which do not support the privacy structure and do not include the possibility to update the agreement at runtime. In fact, a guarantee is not fulfilled because of an event occurring in the service behavior and may change the personal data use. The proposed extension is reflected in a new component in a WS-Agreement called ***privacy-agreement***.

A privacy-agreement structure is represented in two levels:

1. ***policy level***, it specifies the *Privacy-Data term* defined as a set of *clauses* of the contract denoted by $\mathscr{C}$ between the provider and the customer. The description of the elements defined in the privacy-data model is embedded in this level, including guarantees dealing with privacy-data model.
2. ***negotiation level***, it specifies all possible events that may happen in the service behavior, thus evolving the privacy guarantee terms defined in the policy level. Negotiation terms are all possible actions to be taken if the guarantee of privacy terms is not respected, then a conflict arises. They are used through a negotiation protocol between the service provider and the customer.

We also defined in this level the validity period of the privacy agreement and a set of penalties when the requirements are not fulfilled.

In the rest of the chapter, we are interested in the first level. We will present a way to observe the use of the private data throughout the run time, and how to capture the compliance of the agreement related in the privacy data terms.

3 Overview of the Monitoring Framework

We devise a privacy-compliance architecture for monitoring. It incorporates three main components discussed in this chapter, they are depicted in Fig. 1 and are namely a *private requirements specification*, a *PDUF Observer*, a *monitor*. The figure assumes the web service executes a set of operations using private data. While

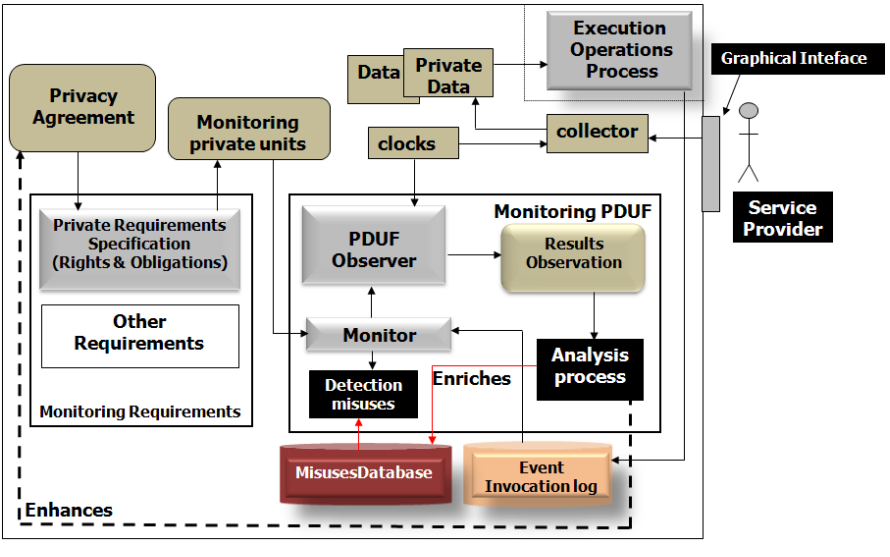


Fig. 1 Monitoring framework

executing the operations of the service, the process generates events stored in a database as logs.

In order to check the privacy compliance, the monitoring private units are extracted from the *private requirements specification* defined in the privacy agreement. Monitoring private units are specified by the means of LTL formulas taking into account the privacy time-related requirements using a set of clocks.

The *monitor* collects the raw information to be monitored regarding the monitoring private units from the event logs database. The collected data and private data misuses stored in a database are fitted together in the *PDUF Observer* component in order to check the non-compliance.

The PDUF observer observes the behavior of the private data use flow. The privacy agreement clauses are observed, which means, when a clause is activated, or which and when a clause is violated and what types of vulnerabilities happened, or which clause is compliant etc. A model to represent such behavior is provided. At the end of the observations the *observation results* report is generated to the Analysis process depicted in the figure.

From the previous observed results and reasoning facilities, the analysis process will provide diagnosis of violations, for instance understanding why the non-compliance took place and what remedy will be provided enhancing the privacy agreement. It can also enrich the database of misuses by those promptly revealed when they occur at run-time. Finally, the detection misuses component consumes the misuses recorded in the database and identify the violation types from compliant usage behavior. We will not give more details about the analysis and detection components, they are out of the scope of this chapter.

4 Requirements for Monitoring Privacy

One of the key aspects for the reliability of the service is the trustworthiness of the compliance of its collected private data use to the agreement. To ensure the privacy agreement compliance, the observation of the service behaviour and its private data use becomes a necessity. For making the compliance happen, keep track of all uses is a fact, that is, from the result of the observations, if needed when violations are detected, the revision of the agreement can be held and relaxed. Indeed, to make the observation effective, two essential ingredients are required, we need to define what kind of knowledge must be monitored and the knowledge which makes the agreement not compliant. In this section we discuss the two aspects.

4.1 Monitoring Units for Privacy

We distinguish four types of unit to be monitored: *private data unit*, *operation unit*, *temporal unit* and *role unit*.

• *Private data unit.* The private data unit d is the core of our monitoring framework. In fact, from the log, we need to observe only the private data and its behavior.
• *Operation unit.* We distinguish two types of actions (i) actions used to complete the service activity for the current purpose for which it was provided and are denoted by $Op_{current}$ (ii) actions used by a service to achieve other activities than those for which they are provided, called $Op_{extra-activity}$. Those two kinds of operations are proposed in order to know when a compliance is compromised, while the service is running for which it was provided or for some operations else. The set of the operations is denoted Op.
• *Role unit.*We need to observe who will use the private data.
• *Temporal unit.* The analysis of time-related aspects of the privacy monitoring requires the specification of operation durations and timed requirements. The instance monitor i.e. temporal unit is defined as a temporal formula using Linear Temporal Logic (P,S,H, operators) [14]. We identify four types of temporal units, and we denote the set of temporal units by $\mathscr{T}$:

Definition 4. ***(Right triggering time).*** *For each collected private data d, the right triggering time denoted ε_{rd} is the activation time of the operation associated to the right:*
$\forall R^i_d \in \mathscr{C} \rightarrow \exists\ \varepsilon^i_{rd} \in T \mid (op^i_d.R^i_d)^{\varepsilon^i_{rd}}$ *is activated, where i is the i th right associated to the private data d, $\mathscr{C}$ is a set of clauses in the agreement, and T is a domain of time values. We need to satisfy the LTL formula $\models_{\varepsilon^i_{rd}} P\ op^i_d.R^i_d$, by means the past temporal operator P i.e., in the past at ε^i_{rd} time the operation is true.*

Definition 5. ***(Right end time).*** *For each collected private data d, the right end time denoted β_{rd} is the end time of the data use (operation) associated to the right :*
$\forall R^i_d \in \mathscr{C} \rightarrow \exists\ \beta^i_{rd} \in T \mid (op^i_d.R^i_d)^{\beta^i_{rd}}$ *is finished, and the LTL formula is $\models_{\beta^i_{rd}} P \neg op^i_d.R^i_d$ at β_{rd} time the operation is not valid.*

Definition 6. ***(Obligation triggering time).*** *For each collected private data d, the obligation triggering time denoted μ_{od} is the activation time of the action associated to the obligation: $\forall O_d \in \mathscr{C} \rightarrow \exists\ \mu_{od} \in T \mid (a_d.O_d)^{\mu_{od}}$ is activated. We need to satisfy the LTL formula $\models_{\mu_{od}} (a_d.O_d) S (\neg op_d.R_d)$, by means the since operator S i.e., $a_d.O_d$ is true since $\neg op_d.R_d$ (The formula is valid when each right associated to the obligation is achieved).*

Definition 7. ***(Obligation end time).*** *For each collected private data d, the obligation end time denoted α_d is the end time of the action associated to the obligation: $\forall O_d \in \mathscr{C} \rightarrow \exists\ \alpha_d \in T \mid (a_d.O_d)^{\alpha_d}$ is ended, the LTL formula is $\models_{\alpha_d} P\ \neg a_d.O_d$ at α_d time the action is not valid.*

4.2 Privacy Misuses

In this section, we identify the non-compliance or failing to uphold the agreement manifested in terms of vulnerabilities or misuses. We provide a privacy misuses

Table 1 Misuses identification through privacy data use flow

Requirement	Compliance Category	Misuses	Type of misuses
Data-right	Use	no-authorized operation op_d *[wrong-use]*; the misuse happens when the following formula is not valid: $\not\models Hop_d.R_d$, in all the past op_d is not admitted.	Explicit
	Retention time	violation of data retention period: the misuse happens when the formula $\models P((\beta_{rd} - \varepsilon_{rd}) > \mu_{rd})$ is valid.	Explicit
	Disclose-To	a *[wrong collector]* as third party; the following formula is not valid: $\not\models Hu.R_d$, in all the past u is a wrong user.	Explicit
Data-obligation	Obligation Activation date	violation of the obligation activation, the misuse happens when the formula $\models P(\beta_{rd} > \mu_{od})$ is valid.	Explicit
	Security on data (delete, update, hide, unhide,...)	Lack or failure of mechanism or procedure.	Implicit
Security	/	1)Loss of confidentiality and integrity of data for flows from the Internet, 2) external attacks on the processes and platform operating systems since they are linked to the Internet, 3) external attacks on the database,...	Implicit

which is most likely met throughout the private data use. We have classified them into two classes *explicit* and *implicit misuses*. The former one can be visualized in our private data use flow model whereas the latter can not be identified. For instance, *security on data*, *accountability* can not be identified in our model, so it is not in the scope of this chapter. We classified three types of explicit misuses, *temporal misuses*, *operation misuses* and *role misuses*. Table 1 summarizes such misuses. However, the listed misuses are not unique, while run-time, some new misuses can be detected and come to enrich the misuse database. How to detect such misuses is not discussed in this chapter.

5 Monitoring Private Data Use Flow

In order to describe the lifecycle management privacy data terms defined in the agreement, we need to *observe* the data use flow. Such observations will allow us to make analysis, diagnosis and to provide reasoning on violations, for instance why the violations happen, what we can improve in the agreement for making the compliance of the agreement happens etc. The analysis aspect is not handled in this chapter.

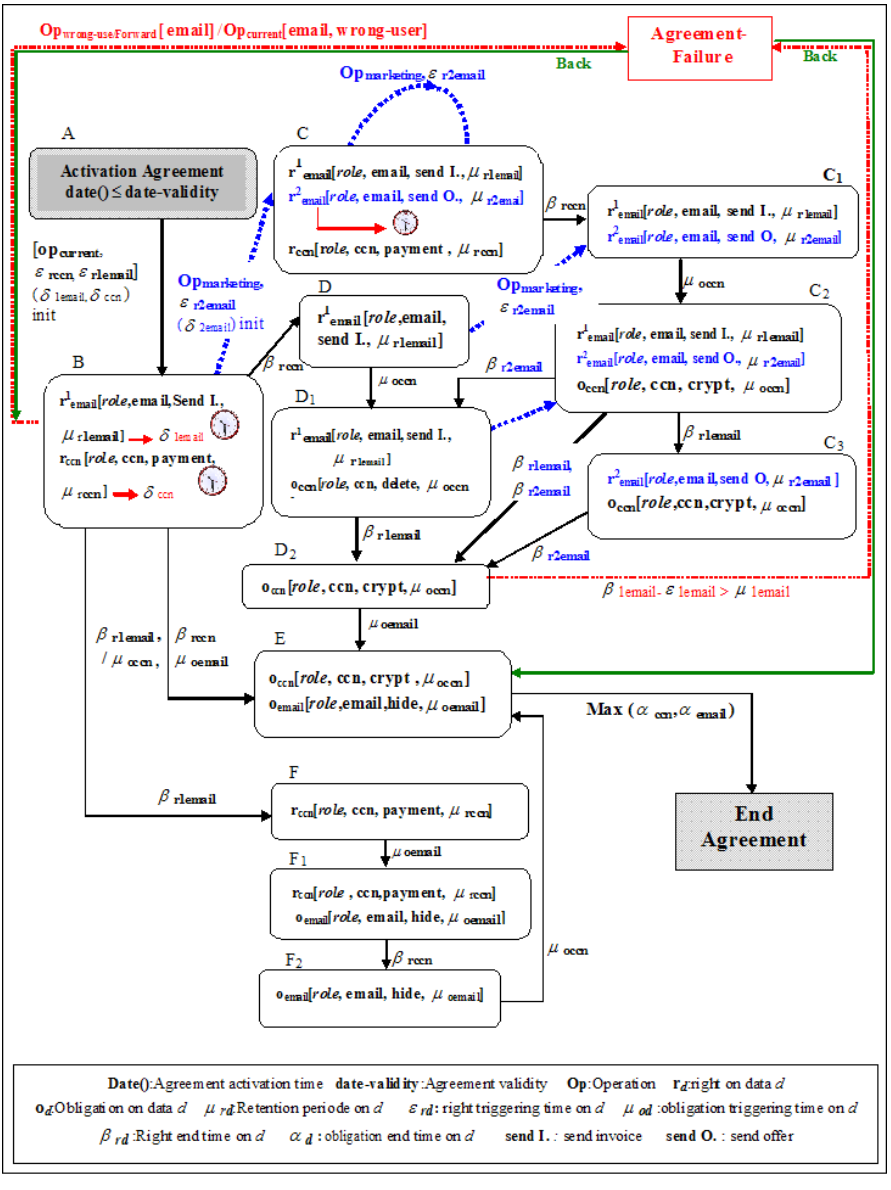


Fig. 2 Private Data Use Flow (PDUF)

We propose to express the Private Data Use Flow (PDUF) as a state machine because of its formal semantic, well suited to describe the activation of different clauses of the privacy agreement. It is an effective way to identify privacy vulnerabilities, where a service 's compliance to privacy regulations may be compromised. It will show *which* and *when* a clause is activated toward the monitoring or which

and when a clause is violated. The time-related requirement properties set in the agreement are depicted explicitly in the state machine. It will specify the states of each activated clause in the policy level. The semantic of the state machine is to define all the triggered operations involving private data from the activation of the agreement (initial state) to the end of the agreement (final state) . We need to keep track of all private data use with or without violations. Fig. 2 shows an example of the privacy data term activation for the purchase service provider.

We have identified several abstractions in relation to private data flow, *private data use* abstractions and *authorization* abstractions. The first abstractions describe the different states in which the agreement is -which private data is collected and when it is used and for what and who use it- . The authorization abstractions provide the conditions that must be met for transitions to be fired.

In this formalism, the fact that the private data has a time retention for a right (respectively the activation time of an obligation) called *fixed guard time*, the private data use time is represented by time increment in the state, followed by the end of the right (respectively obligations) with success or a violation of that time. Intuitively, PDUF is a finite state machine for which a set of clock variables is assigned denoted by Δ. A variable is assigned for each activation of the clauses (rights and obligations). The values of these variables increase with passing the time. The transition will take place when an operation is activated or monitoring time units are triggered. If the temporal units are compliant to the guard times, it will happen the transition will take place with success and no violation is recorded in that state. However, if non-compliance is detected, the transition will take place with violation, then the state is marked as violated.

Definition 8. ***(PDUF.)*** *A PDUF is a tuple* $(\mathscr{S}, s_i, s_f, \mathscr{M}, \mathscr{R}, \mathscr{Q})$

- $\mathscr{S}$ *is a set of states;*
- $s_i \in \mathscr{S}$ *is the initial state, and* $s_f \in \mathscr{S}$ *is the final state ;*
- $\mathscr{M}$ *is a set of monitoring private units: set of triggered operations and/or set of temporal units,* $\mathscr{M} = \{OP, \mathscr{T}\}$*;*
- $\mathscr{R} \subseteq \mathscr{S}^2 \times \mathscr{M} \times 2^{\Delta}$ *is a set of transitions with a set of operations or a set of triggering time and a set of clocks to be initialized* $\delta_{d-init} \in \Delta$*;*
- $\mathscr{Q} : \mathscr{S} \rightarrow \{\delta_i \mid \delta_i \in \Delta, i \geq 1\}$ *assigns a set of clocks to the states.*

The effect of each transition $\mathscr{R}(s, s', m, c)$ from the source state s to the target state s' is to set a status of the clauses in the agreement which means to perform an operation $op \in \mathscr{OP}$ using a private data or a monitoring time unit $t \in \mathscr{T}$ is activated.

Let's define the semantic of PDUF through the following example for the agreement with a set of clauses (rights and obligations).

Example 3. Let us consider the example of a purchase service without giving details about transactions between the customer and the service. An agreement has been signed between them setting up a set of clauses with a validity period denoted by *validity-date*. Those clauses are specified as follows: at the date *date()* the agreement is activated and the service collects email address (email) and credit card number (ccn). Those private data are used for two types of operations **(1)**

to complete the service activity for the current purpose i.e. the email is used to send invoices and credit card number for the payment of invoices. The operations are expressed by the following rights $r^1_{email}(role, email, send\ invoice, \mu_{r1email})$ and $r_{ccn}(role, ccn, payment\ invoice, \mu_{rccn})$ **(2)** to achieve other activities than those for which they are provided, for instance marketing purpose i.e. the email is used to send the available products and their prices, that clause is expressed by the right $r^2_{email}(role, email, send\ offer, \mu_{r2email})$.

When the retention times of the private data email and ccn $(\beta_{r1email}, \beta_{r2email}, \beta_{rccn})$ are elapsed, the corresponding obligations are triggered, $O_{email}(role, email, hide, \mu_{oemail})$ and $O_{ccn}(role, ccn, delete, \mu_{occn})$. Those obligations specifying the role must hide (respectively delete) as soon as the activation date μ_{oemail} (respectively μ_{occn}) is reached.

In what follow, due to the space limitation we will not comment on all the state machine, and for the sake of clarity, we omit some details about it, such as the clocks on the states and all the misuses etc.

States: we define four types of states:

- The initial state s_i represents the activation of the agreement where the first private data of the customer is collected. In Fig. 2, s_i is defined by A.
- The intermediary states represent the flow of the collected private data use. By entering a new state, a private data is used.
  - to complete the activity of the service for which it was provided, identified in Fig. 2 by $Op_{current}$. In the state B, the current operations are *SendInvoice* and *payment*. In this state, the clocks δ_{1email} and δ_{ccn} are activated respectively to r^1_{email} and r_{ccn} and incremented passing the time.
  - and/or to achieve an extra activity as depicted in Fig. 2 by $Op_{marketing}$. The right r^2_{email} is activated in the state C as soon as the marketing operation is triggered. The same operation can be activated as many times as the data time retention $\mu_{r2email}$ is valid. It is represented by a *loop* in the state C. The privacy agreement remains in the same state.
  - and the data use is finished (the right). For instance, the agreement will be in the state C_1 since the data retention guard time is reached, which means the finishing time of the right is over and is denoted by β_{rccn}.
  - and/or to activate an operation dealing with the security (e.g. obligations) when the retention time of the private data defined as a fixed time in the right is elapsed and the time for triggering the obligations starts. For instance, such case is depicted in Fig. 2 in the state C_2, where o_{ccn} is activated when the usage time of the date β_{rccn} is reached and the obligation time starts defined in the transition by μ_{occn}.
- The *virtual* state labeled *Failure agreement* will be reached when a private data is used to achieve the operation misuse, and/or role misuse and/or time misuse happens regarding the clock variable values and fixed times. For instance, the first type of misuse is identified by $Op_{wrong-use/Forward}[email]$ between state B

and Failure agreement state. We call this state as a virtual state because it is considered only like a flag of misuses.
- The final state s_f represents the end of the agreement which means the validity of the agreement is over, and either the data use in all its shapes is compliant to the agreement or the agreement is not respected due to the misuses. The best case is to reach the end of the agreement without any misuses as depicted in the figure from the state E to the end-agreement state.

Transitions: Transitions are labeled with conditions which must be met for the transition to be triggered. We have identified three kinds of authorization abstractions:

- Activation conditions. We define two types of activation (i) an operation has the authorization to collect private data to achieve the current aim of the service, for instance, $op_{current}$ condition on the transition from the state A to the state B, an operation dealing with an extra activity of the service has the authorization to be triggered. For instance, the operation $op_{marketing}$ from the state B to the state C.
- Temporal conditions. The transition is called *timed transition*. Regarding the temporal monitoring unit, we define four types of timed transitions (1) *right triggering time* ε_{rd}, for instance from the state B to the state C the timed transition is labeled by $\varepsilon_{r2email}$ along with the activation of the clock δ_{2email} assigned to the right r^2_{email} (2) *Right end time* β_{rd}, from the state C to state C_1 the transition is labeled β_{rccn}, which means the ccn use is over (3) *Obligation triggering time* μ_{od}, the authorization to keep the private data is finished and the obligation is triggered, for instance from the state C_1 to C_2, the transition is labeled μ_{occn}, the operation of security must be fired (4) *Obligation end time* α_d, the obligation is over, for instance from the state E to the end-agreement state, we calculate the maximum of the two end times α_{email} and α_{ccn}, in our case it is the best way to finish the compliance of the agreement.
- Misuse Conditions. The transition can be labeled by all the misuses identified in Sect. 4.2. For the misuse dealing with the operations , the target state of the transition is *failure-agreement* and *Back* to the previous state, for instance, the operation $op_{wrong-use/forward}$ on the transition between the state B and the failure-agreement state, and back to the state B. For the temporal misuse the target state of the transition is *failure-agreement* and no back to the previous state rather to the next state, for instance, a time violation happens in D_2 and the system passes to the next state E.

6 Architecture and Implementation

Architecture

The architecture described in this section incorporates a set of components depicted in Fig. 3, namely a *Web service simulator*, a *requirement extractor*, an *event filter*, a *monitor* and a *PDUF Observer*.

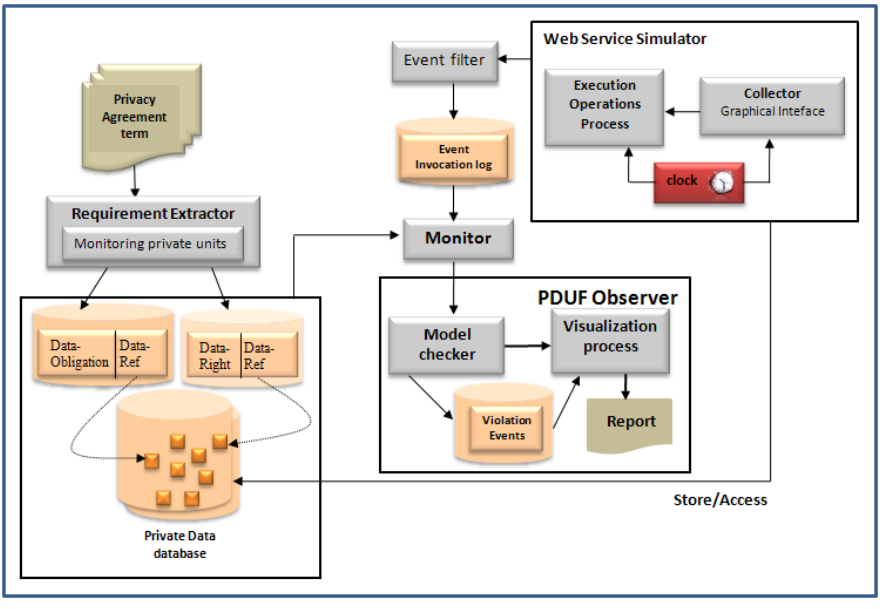


Fig. 3 Architecture of the Framework

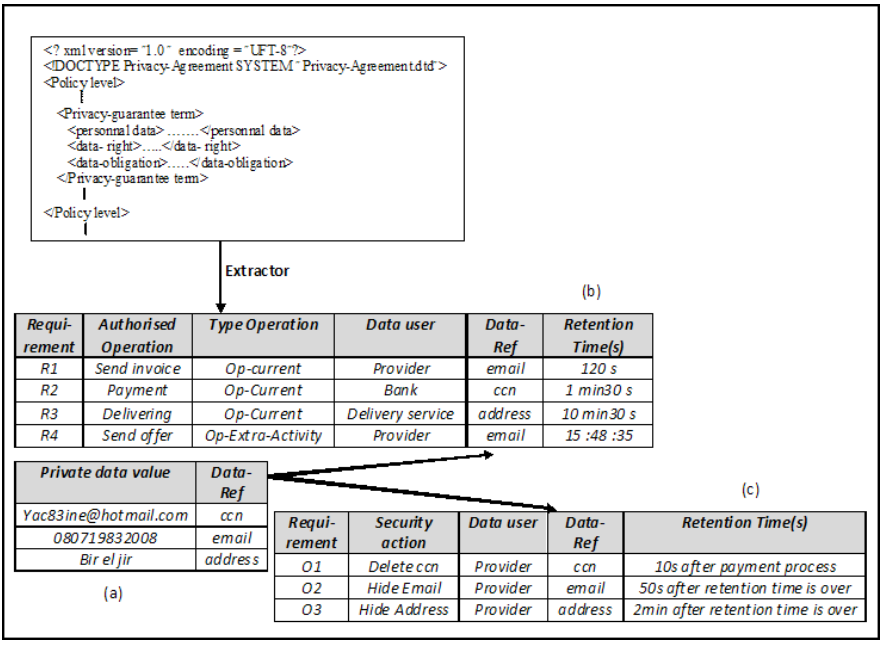


Fig. 4 (a) private data table (b) rights private units table (c) Obligation private units table

- ***Web service simulator*** is an environment simulating web services interactions. It includes two sub-components, the ***operation execution process*** and a ***collector***. The former executes a set of operations defined in the system to fulfill the service requirements. The ***collector*** is an editing interface, allows the provider to collect the private data from the attached operation while it is executed. The private data collection and the execution of the operations are considered as the events provided by the simulator. It is important to stress that there is an assignation of a clock variable for each activated operation and for each collected data . The values of these variables increase with passing the time.
- ***Requirement extractor*** takes as input privacy agreement represented in an XML-based language and extracts the monitoring private units to be monitored from the requirements specification (rights , obligations). Theses private units are recorded in right and obligation tables (see Fig. 4).
- ***Event filter***, while executing the operations of the service or an invocation of the client for providing private data, the process generates events which are sent to the event filter. After reception, the event filter identifies its type and its relevance to the privacy data term of the agreement being monitored, and records it in the event log database of the framework. All the non relevant events are not tackled.
- ***Monitor*** collects the raw information to be monitored regarding the monitoring private units from the event logs in the order of their occurrence. The collected data and the monitoring private units stored in requirement database are fitted together in the *PDUF Observer* component in order to check the compliance.
- ***PDUF Observer*** observes the behavior of the private data use flow by using two components *Compliance checker* and *visualization process*. The checker checks the compliance of the recorded events with the monitoring private units stored in the requirements tables. In case of non compliance with privacy data term of the agreement, the compliance checker stored the deviation in the violation event database. The visualization process visualizes the result of the checking process. Thus, we can see the behavior of private data use flow and identify the violations and the details of the events that have caused it. The observations and violations are reported in a report which can be viewed as an XML document.

To provide and generate the events that will be used during the monitoring framework, we simulate an execution of different operations by web services, which are, service client, service provider and partners services (Bank service, delivery service and maintenance service). In this simulator we specify two kinds of operations, (1) *internal operations* (2) *external operations*. The former are executed by the service provider while the latter are executed by the partner services. It is important to note that both kinds of operations can use or not the private data. Table 2 summarizes such operations .

Implementation

A prototype of the framework is written in Java. We have developed an execution operation engine to simulate the collaboration between the different services of the system by the activation of the aforementioned operations. Figure 5 (A) shows

Table 2 Specification of operations and activities used in the simulator

Status operation	Type operation	Description	Example
Collect activity ***Col-A***	Internal operation	Collection of the private data	Collection of data email, ccn, address.
Invocation current operation $I-OP_{current}$	Internal operation	Activation of the current operation	Execution of send invoices by using email
Invocation extra operation $I-OP_{extra-A}$	Internal operation	Activation of the extra operation which can be executed concurrently with current operation	Execution of send offer by using email
Receive operation *Rec-OP*	External operation	Partner waits for the invocation of current or extra-activity by the provider	Request invocation of the payment with the transfer of the private data ccn
Reply operation ***Rep-OP***	External operation	The partner service responds to a request for the execution of an operation previously accepted through a receive operation	Execution of the payment operation using the transfered ccn
End operation ***End-OP***	External operation	The partner service informs the provider the end of the operation execution	Send invoices operation is over
Invoke security action ***Sec-A***	Internal/External operation	Activation of the security action (obligation) by the provider or/and the partner service when the retention time of data is over	Hide email
Other operation ***Other***	External/Internal operation	The provider and/or the partner can activate the operations which are not specified in the agreement. Such operations may use or not the private data	Statistic, Maintenance process
Clock assignation activity		Assignment of clock variable for each activated operation (activation time, end time) or for each collected data	

different operations that can be activated by the system, while Figure 5 (B) shows the execution of the receive operation between service provider and bank service.

The engine generates logs of the events during the execution process. This event log is fed into our framework in order to provide the runtime information that is necessary for monitoring. The events can be the activity of data collection, the execution of the internal operation by the provider, or the activation of external operation through the exchanged messages between the services partner and the provider. Each operation has two clock variables, the activation time and the end time(during the simulation we use the minute and second unit). The events are described by the attributes that have the following form :

- *Event* is a unique identifier of the event.
- *oper* is the signature of the operation or the collect activity.
- *status* represents the type of the operation (see Table 2) (e.g. $Col-A$, $I-OP_{current}$, $I-OP_{extra-A}$, *Rec-OP*, *Rep-OP*, *End-OP*, *Other*,*Sec-A*)
- *data-user* is the identifier of the service executing an operation.
- *data-ref* is the identifier of the used private data.

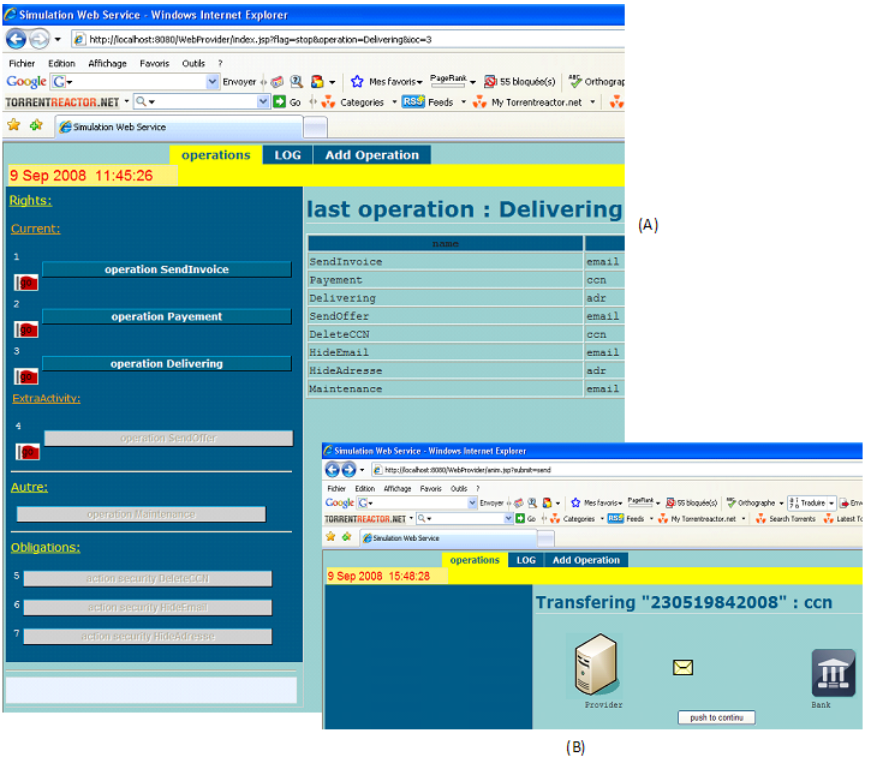


Fig. 5 (A) List of operations of the Simulator (B) Execution of the receive operation (Payment operation with the transfer of ccn)

- *data-value* is the value of the private data.
- *clock* is the triggering time or end time assigned to the event.

An example of event structure and instances is depicted in Fig. 6.

According to the model presented in section 5, a set of paths representing the possible usage flow of all private data from the initial state (activation agreement) to the end state (end agreement) is provided. We identify a sequence of execution events stored as the event log (green row related to email of Fig. 6) with one of expected behavior path. However, if an event of the sequence is deviated from the expected behavior, then the violation is detected. For illustration, in Figure 7 is depicted three usage flow paths of the email private data with specific colors, (1) the black identifies the flow *A*-*B*-*C*-C_1-C_2, path,(2) the pink path *A*-*B*-$C^{'}$-*D*-$C^{'}_2$ (3) green path *A*-*B*-$C^{''}_2$.

To visualize the paths of private data use flow, we used OpenJGraph which is a Java library to create and manipulate graphs. The events are classified into two categories: (1) *State event* : $SEvent = \{I-OP_{current}, I-OP_{extra-A}, Rep-A, Sec-A\}$ (2) *Transition events:* $TEvent = \{Col-A, End-A, clock-Sec-A\}$.

Event	Oper	status	Data-user	Data-ref	Data-value	Clock
E1	Collect	Col-A	Provider	Email	Yac83ine@hotmail.com	15:48:19
E2	Send invoice	$I\text{-}OP_{current}$	Provider	Email	Yac83ine@hotmail.com	15:48:24
E3	Send offer	$I\text{-}OP_{extra\text{-}A}$	Provider	Email	Yac83ine@hotmail.com	15:48:35
E4	Maintenance	$I\text{-}OP_{current}$	Company C	Email	Yac83ine@hotmail.com	15:48:44
E5	Collect	Col-A	provider	Ccn	080719832008	15:49:33
E6	Payment	REP-Op	Bank	Ccn	080719832008	15:49:36
E7	Send Offer	End-Op	Provider	Email	Yac83ine@hotmail.com	15:49:42
E8	Payment	End-Op	Provider	Ccn	080719832008	15:49:45
E9	Collect	Col-A	provider	Address	Bir el jir	15:50:01
E10	Delivering	$I\text{-}OP_{current}$	Delivery	Address	Bir el jir	15:50:05
E11	Delivering	End-Op	Provider	Address	Bir el jir	15:50:20
E12	HideAddress	Sec-A	Provider	Address	Bir el jir	15:52:06
E13	Send Invoice	End-Op	Provider	Email	Yac83ine@hotmail.com	15:52:10
E14	Delete CCN	Sec-A	Provider	Ccn	080719832008	15:52:15
E15	Hide Email	Sec-A	Provider	Email	Yac83ine@hotmail.com	15:56:17

Fig. 6 Event log of a purchase service

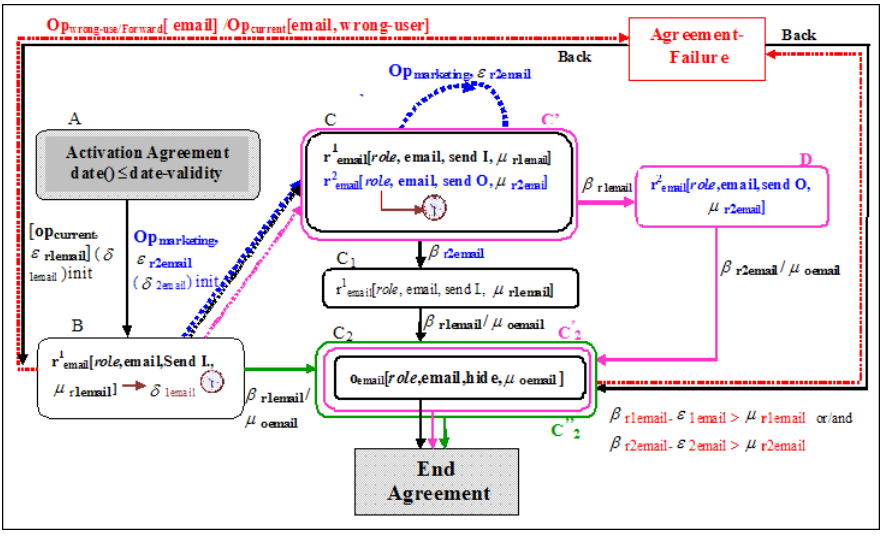


Fig. 7 PDUF related to the private data email

Figure 8 keeps track of the behavior related to email data from the sequence of event stored in the log. It identifies the black path *A*-*B*-*C*-C_1-C_2. It also shows tree types of violations discussed in section 4.2. These violations are characterized by [*wrong-collector and wrong-use*], *violation of data retention period* and *the violation of the obligation activation*. The corresponding notations in the graph are respectively (OPwrong-use[email]/Maintenance [email,wrong-user]),(error in retention right time [right current : send invoice, retention 226>120]),(error in triggering obligation time[obligation email activation= 247>240]). Figure 9 shows the usage flow of all the private data manipulated in the system.

Fig. 8 Data use flow related to the private data email

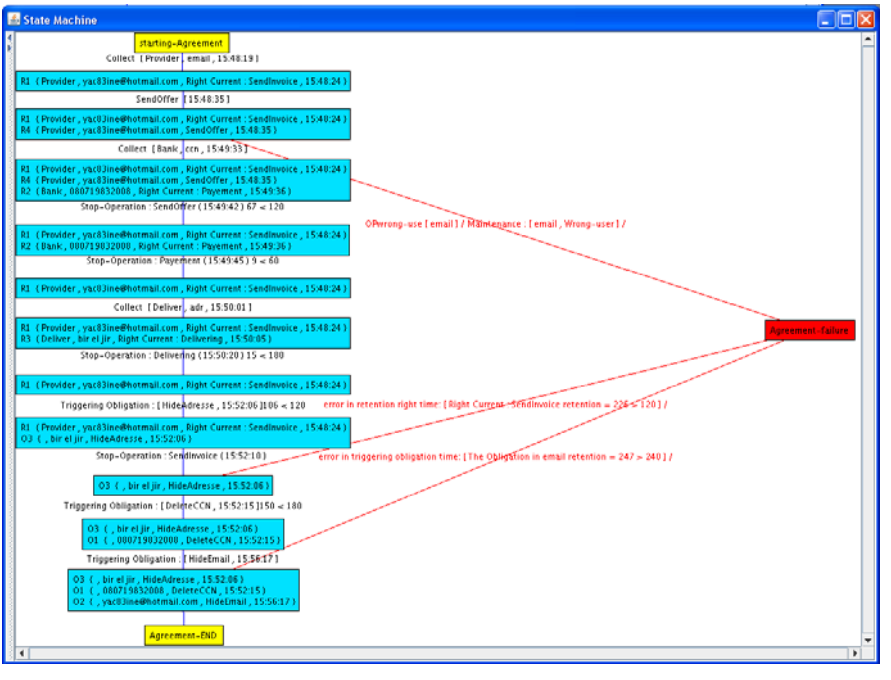


Fig. 9 Generation of the global PDUF including all private data collected in thesystem

7 Related Work

The literature is very scarce on works dealing with monitoring the privacy compliance in web service. However, the problem of web services and distributed business processes monitoring is investigated in the works [12, 19, 13, 3, 4, 2, 9, 17, 6]. The research in [12, 19] is focusing on monitoring of service-based software (SBS) systems specified in BPEL. They use event calculus for specifying the requirements that must be monitored. The run-time checking is done by an algorithm based on integrity constraint checking in temporal deductive databases. In [13], the authors present a framework to support the monitoring of service level agreements. The agreements that can be monitored are expressed in an extension of WS-Agreement.The main characteristic of the proposed extension is that it uses an event calculus based language, called EC-Assertion, for the specification of the service guarantee terms in a service level agreement that need to be monitored at runtime.

Barezi et al in [3, 4] developed a tool that instruments the composition process of an SBS system in order to make it call external monitoring services that check assertions at runtime. The work in [2] is close to the previous works, the authors present a novel approach to web services described as BPEL processes. The approach offers a clear separation of the service business logic from the monitoring functionality. Moreover, it provides the ability to monitor both the behaviours of single instances of BPEL processes, as well as behaviours of a class of instances.

In [17], the authors propose an approach to the automated synthesis and the runtime monitoring of web service compositions. Automated synthesis, given a set of existing component services that are modeled in the BPEL language, and given a composition requirement. The latter expresses assumptions under which component services are supposed to participate in the composition, as well as conditions that the composition is expected to guarantee. Run-time monitoring matches the actual behaviors of the service compositions against the assumptions expressed in the composition requirement, and reports violations.

Beeri et al in [6] present BP-Mon, a novel query language and system for monitoring BPs. BP-Mon offers a high level intuitive design of monitoring tasks. A novel optimization technique exploits available knowledge on the BP structure to speed up computation.

Lazovik et al. [11] propose an approach based on operational assertions and actor assertions. They are used to express properties that must be true in one state before passing to the next, to express an invariant property that must be held throughout all the execution states, and to express properties on the evolution of process variables. While providing facilities for the verification of processes these approaches do not take privacy requirements into account.

In terms of privacy compliance, there exist few works including [8, 16, 21, 15, 20, 10]. In [8], the authors examine privacy legislation to derive requirements for privacy policy compliance systems. They propose an architecture for a privacy policy compliance system that satisfies the requirements and discuss the strengths and weaknesses of their proposed architecture.

In [16], the authors introduce the concept of an 'information transfer registry' as a mechanism to track compliance in a business to business network. The registry stores signed contracts specifying the consent an individual has given for information transfers for specific business purposes. Organizations register all information transfers from individual to business or business to business against the appropriate contract to document their compliance.

In [21] the author proposes a graphical visualization notation to facilitate the identification of private information vulnerabilities that can lead to privacy legislation non-compliance. In [15], the authors automate the management and enforcement of privacy policies (including privacy obligations) and the process of checking that such policies and legislation are indeed complied with. This work is related to enterprise. While providing tools for privacy compliance in the previous works, however, these approaches do not take private data use flow into account and no formal method along with reasoning and also no time-related properties are discussed.

In order to provide better protection for personal data, the authors in [18] propose PRMF, a privacy rights management framework which enforces personal data processing compliance with privacy policies related to organizational, legislative, and regulatory needs. PRMF can satisfy many aspects of privacy legislation, including security, transparent processing, lawful basis, and finality - purpose limitation.

In [20], the authors propose an approach for compliance checking of agreed privacy policies and preferences in a federated identity management context. They introduce mechanisms and algorithms for policy compliance checking between federated service providers, based on an innovative policy subsumption approach.

In [10], the author focus their attention on the discovery of private data. Their objective for private data discovery is to develop ways to extract private data efficiently and effectively from unstructured and semistructured content soas not to interfere with work activities. The private data may emerge from any type of computer-based activity, whether it is collaborative or not.

In terms of privacy analysis, there exist few works including [1]. This work has proposed a straightforward method for visual analysis of privacy risks in web services, focusing the analysts attention at locations that hold personal information at one time or another. The method only identifies possible privacy risks and does not evaluate the likelihood of a risk being realized.

8 Conclusion

In this chapter we pointed out the challenge of privacy in web service based applications ensuring the end-to-end non functional QoS awarness. We proposed an effective and formal approach to observe and verify the privacy compliance of web services at run-time. We have emphasized private data use flow monitoring of privacy-agreement requirements, which is an important issue to date has not been addressed. It is a state machine based approach, that allows to take into account the timed-related properties of privacy requirements and to facilitate the identification of private information misuses. The privacy properties to be monitored are

specified in LTL. The monitored units are extracted from the privacy agreement requirements. The approach supports the monitoring of a set of identified misuses that lead to non-compliance, and which can be enriched from the observation diagnosis. The approach is still under development. Our ongoing work and a promising area for the future include: (1) The development of reasoning facilities to provide a diagnosis of misuses, (2) The development of tools for detecting the misuses (3) The development of tools along with metrics for enhancing the privacy-agreement from the observations (4) Expanding the approach to handle the composition of the services.

References

1. Yee, G.: Visual analysis of privacy risks in web services. In: 2007 IEEE International Conference on Web Services (ICWS 2007), Salt Lake City, Utah, USA, July 9-13, pp. 671–678. IEEE Computer Society, Los Alamitos (2007)
2. Barbon, F., Traverso, P., Pistore, M., Trainotti, M.: Run-time monitoring of instances and classes of web service compositions. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2006), pp. 63–71 (2006)
3. Baresi, L., Ghezzi, C., Guinea, S.: Smart monitors for composed services. In ICSOC 2004. In: Proceedings of the 2nd international conference on Service oriented computing, pp. 193–202. ACM Press, New York (2004)
4. Baresi, L., Guinea, S.: Towards dynamic monitoring of ws-bpel processes. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 269–282. Springer, Heidelberg (2005)
5. Benbernou, S., Meziane, H., Li, Y.H., Hacid, M.: A privacy agreement model for web services. In: IEEE International Conference on Service Computing SCC 2007 (2007)
6. Milo, T., Pilberg, A., Beeri, C., Eyal, A.: Monitoring business processes with queries. In: Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, pp. 603–614 (2007)
7. Guermouche, N., Benbernou, S., Coquery, C.E., Hacid, M.: Privacy-aware web service protocol replaceability. In: IEEE International Conference on Web Services ICWS 2007 (July 2007)
8. Yee, G., Korba, L.: Privacy policy compliance for web services. In: Proc. of the IEEE International Conference on Web Services (ICWS 2004), Washington, USA, pp. 158–166 (2004)
9. Kazhamiakin, R., Pandya, P., Pistore, M.: Representation, verification, and computation of timed properties in web. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2006), Washington, USA, pp. 497–504. IEEE Computer Society, Los Alamitos (2006)
10. Korba, L., Wang, Y., Geng, L., Song, R., Yee, G., Patrick, A.S., Buffett, S., Liu, H., You, Y.: Private data discovery for privacy compliance in collaborative environments. In: 5th International Conference on Cooperative Design, Visualization, and Engineering, CSVE 2008, Calvi'a, Mallorca, Spain, September 21-25, pp. 142–150. Springer, Heidelberg (2008)
11. Lazovik, A., Aiello, M., Papazoglou, M.: Associating assertions with business processes and monitoring their execution. In: ICSOC 2004 Proceedings of the 2nd international conference on Service oriented computing, New York, USA, pp. 94–104 (2004)

12. Mahbub, K., Spanoudakis, G.: Run-time monitoring of requirements for systems composed of web-services: Initial implementation and evaluation experience. In: 2005 IEEE International Conference on Web Services (ICWS), December 2005, pp. 257–265 (2005)
13. Mahbub, K., Spanoudakis, G.: Monitoring ws-agreement: An event calculus-based approach. In: Test and Analysis of Web Services, pp. 265–306. Springer, Heidelberg (2007)
14. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer, Heidelberg (1992)
15. Casassa Mont, M., Pearson, S., Thyne, R.: A systematic approach to privacy enforcement and policy compliance checking in enterprises. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 91–102. Springer, Heidelberg (2006)
16. Peyton, L., Nozin, M.: Tracking privacy compliance in b2b networks. In: Proceedings of the 6th International Conference on Electronic Commerce, ICEC 2004, Delft, The Netherlands, October 25-27, pp. 376–381 (2004)
17. Pistore, M., Traverso, P.: Assumption-based composition and monitoring of web services. In: Test and Analysis of Web Services, pp. 307–335 (2007)
18. Song, R., Korba, L., Yee, G.: Privacy rights management for privacy compliance systems. In: 21st International Conference on Advanced Information Networking and Applications (AINA 2007), Workshops Proceedings, Niagara Falls, Canada, May 21-23, vol. 1, pp. 620–625. IEEE Computer Society, Los Alamitos (2007)
19. Spanoudakis, G., Mahbub, K.: Non intrusive monitoring of service based systems. International Journal of Cooperative Information Systems (2006)
20. Squicciarini, A.C., Casassa Mont, M., Spantzel, A.B., Bertino, E.: Automatic compliance of privacy policies in federated digital identity management. In: 9th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2008), Palisades, New York, USA, June 2-4, pp. 89–92. IEEE Computer Society, Los Alamitos (2008)
21. Yee, G.: Visualization for privacy compliance. In: VizSEC 2006: Proceedings of the 3rd international workshop on Visualization for computer security, pp. 117–122. ACM Press, New York (2006)

Achieving Scalability with Schema-Less Databases

David A. Maluf and Christopher D. Knight

Abstract. Large enterprises continue to struggle with information and critical decision-making data being widely distributed, stored in a number of proprietary and heterogeneous formats, and remaining inaccessible for mining of critical information that spans the collected knowledge of the organization. NETMARK is an easy to use, scalable system for storing, decomposing, and indexing enterprise-wide information developed for NASA enterprise applications. Information is managed in a contextualized form, but one that is schema-less for immediate storage and retrieval without the need for a schema manager or database administrator. NETMARK is accessed via the WebDAV (HTTP) standard protocol for remote document management and a simple HTTP query algebra for immediate retrieval of information in an XML structured format for processing by applications such as Web 2.0 (AJAX) systems.

1 Introduction

This paper describes the conceptualization, design, implementation and application of an approach to scalable and cost-effective information integration for large-scale enterprise information management applications. Our work was motivated by requirements in the United States National Aeronautics and Space Administration (NASA) enterprise where many information and process management applications demand access to and integration of information from large numbers of information sources (in some cases up to as many as 50 different sources) across multiple divisions and with information of different kinds in different formats. An example is the application of assembling an agency level annual report that requires information such as project status, division updates, budget information, personnel progress etc. from different data sources in different

David A. Maluf
NASA Ames Research Center
Moffett Airfield, CA 94035
e-mail: David.A.Maluf@nasa.gov

Christopher D. Knight
NASA Ames Research Center
Moffett Airfield, CA 94035
e-mail: Christopher.D.Knight@nasa.gov

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 249–274.
springerlink.com

departments, divisions, and centers within NASA. In the early 2000s which is when we started considering technology solutions to address such information access and integration challenges, data integration technology was already quite well-developed with commercial off-the-shelf solutions as well. Major intelligent information integration research projects such as SIMS, TSIMMIS, HERMES, InfoMaster, Information Manifold [1,4] to name a few, that were concerned with building data integration systems based on a *mediator* architecture had reached considerable maturity. We had solutions to challenging problems such as providing efficient query processing over multiple distributed data sources, schema mapping and integration tools, wrapper technology for legacy data sources and also Internet data sources, and technologies for entity resolution and matching across multiple sources. There were also a slew of vendors including spin-offs such as Nimble [5], Junglee, Mergent, Enosys [6] and Fetch, and bigger companies such as IBM touting off-the-shelf data integration technology that could address the required information integration needs. While functionally meeting the requirements, none of these technologies could provide scalable and cost-effective information integration solutions for large scale applications. The basic problem was that such middleware based technology being offered became rather "heavy-weight" in the face of large scale applications. A significant amount of investment was required in assembling new integration applications. Particularly the effort in managing models and meta-data i.e., in describing the many sources being integrated and also in providing an integrated view over the various sources became formidable, to the extent that this became one of the key impediments to the widespread adoption of EII technology in general. A testament to this is articulated in a review of EII technology [3] where a CTO of (a then prominent) EII start-up observes *"A connected thread to this (key impediments for EII) is to address modeling and metadata management, which is the* ***highest cost item*** *in the first place"*.

The above problems carried over to the area of the "Semantic-Web" [7], where most applications demand a heavy investment in creating various *ontologies* and further providing semantic linkages across such ontologies. The substantial effort and complexity in ontology creation and maintenance continues to be a major impediment in realizing practical semantic-web applications.

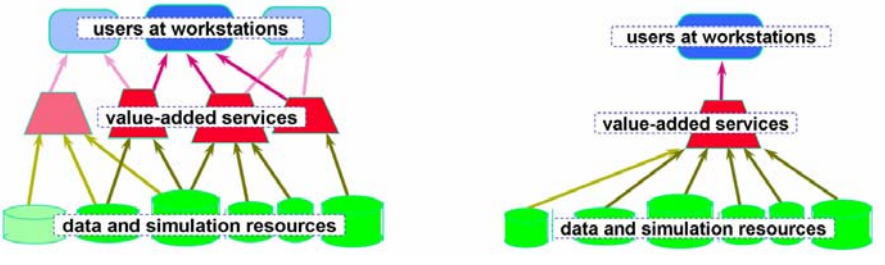


Fig. 1 Intelligent Information Integration (III)

The lack of scalable and cost-efficient data integration technologies was however not because this was something that could not be achieved, but rather because the original vision of *intelligent information integration* had gone awry. The original vision of intelligent information integration (or I^3) research sponsors such as DARPA [1] was a nimble and flexible approach where clients could at will select and integrate information from different sources in a manner suited to their particular applications and the complexity of each new application was confined to the application itself (Fig 1(a)). In practice however this degenerated to a situation where the complexity of *all* applications was added on to the mediation layer (Fig 1(b)). The reason this happened was due to some flawed assumptions about how enterprise data should be managed and integrated. These assumptions, along with our alternative solutions are presented below, namely:

"Data must always be stored and managed in DBMS systems" Actually, requirements of applications vary greatly ranging from data that can well be stored in spreadsheets, to data that does indeed require DBMS storage.

"The database must always provide for and manage the structure and semantics of the data through formal schemas" Alternatively, the "database" can be nothing more than intelligent storage. Data could be stored generically and imposition of structure and semantics (schema) may be done by clients as needed.

"Managing multiple schemas from several independent sources and interrelationships between them, i.e., "schema-chaos" is inevitable and unavoidable" Alternatively, any imposition of schema can be done by the clients, as and when needed by applications.

The above philosophy in our opinion captures the *original* vision of intelligent information integration which is what we present here. The centerpiece of this entire effort, arguably, is the realization of the NETMARK information integration and management system. NETMARK offers some key advantages as a system that significantly differentiates it from other alternative technologies in its general category. These are:

- The system is **easy-to-use**. NETMARK can be accessed in simple Web-accessible fashion where for both "providing data" (i.e., we wish to make a source or data in source accessible to some application) or querying data is done using simple desktop drag-and-drop or simple Web URL arguments as we shall see shortly. For most other COTS data integration systems a relatively higher level of expertise is required to be able to use the integration technologies.
- The system supports **large-scale applications** of different kinds. Contextual access to a wide variety of enterprise data ranging from text reports in formats such as Word, PDF, or files to presentations (PowerPoint) to spreadsheets and tables (Excel) is provided. Also as we shall demonstrate, query processing performance in NETMARK is an order of magnitude faster than other (XML) data management systems for large datasets.
- The system is **cost-effective**. There is little procurement cost beyond basic COTS hardware for the deployment of the NETMARK system. The configuration and system management requirements are minimal.

[1] The United States Defense Advanced Research Projects Agency.

Fig. 2 Theory to System Realization to Practice

Our work in realizing such a system has involved all aspects of the spectrum from theoretical concepts to an efficient implementation to real-world deployment (Fig 2), and which is what we present in this paper. We start in the next section (Section 2) with describing a theory of *flexible* knowledge sharing which is the basis for making integration applications scalable. We also present a *context sensitive* query paradigm which we offer and validate as a simple yet powerful paradigm for querying enterprise data. In Section 3 we describe the architecture and system details on the NETMARK data management and integration system which is based on the above flexible knowledge sharing and context sensitive querying paradigm. In Section 4 we present performance evaluation results showing the significant advantage we have with NETMARK over other semi-structured and XML data management systems in the domain of enterprise data. Section 5 presents case studies of the use of NETMARK in actual NASA applications and also the realization and usage of other more expansive systems for tasks such as process management that employ NETMARK as an integration engine. Section 6 describes API access and also the availability of NETMARK as open-source software. Finally in Section 7 we describe ongoing work and a conclusion.

2 Articulation Management and Ontology Algebra

Any information source is basically a knowledge source in more general terms. Thus information sharing and integration is, more generally, a problem of *knowledge* sharing and integration [8]. The complexity of the knowledge can vary from something as simple as a list i.e., data in a single column, to a more structured associated representation such as a relational database to a richer representation such as an object-oriented database, or a more complex knowledge representation such as LOOM [9], Classic [10], or an ontology [11]. The theory of information integration is built upon general theories of how knowledge should be shared and integrated. This is achieved through 2 fundamental constructs 1) Representation of knowledge – in each information source, as well as the "global" view of the integrated knowledge, and 2) *Articulations* – defining linkages across information sources and between any information source and the global view of knowledge [12]. For instance the articulation associated with application A1 illustrated in Fig 3 states that the concept permanent-employees in the JPL information source *is the same* (ist) as the concept full-time-employees in the Ames information

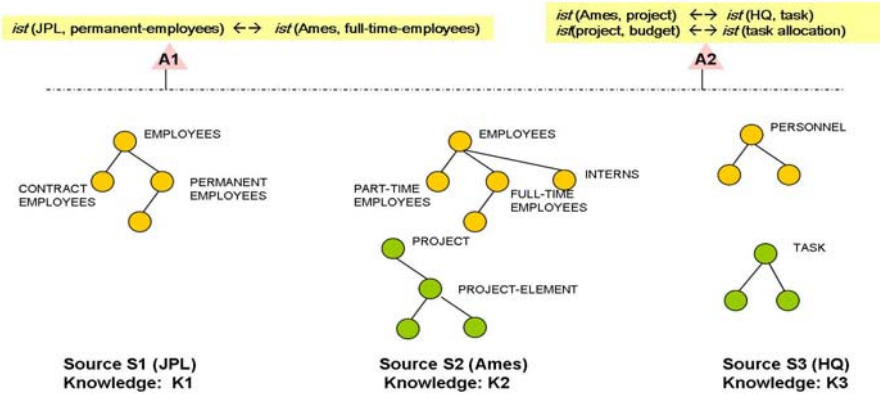


Fig. 3 Knowledge Representation and Articulation

source. Literally all of the major information integration systems proposed [1] are based on the above constructs of describing and linking knowledge across sources, albeit the particular knowledge representation schemes may vary.

This approach is functional but not scalable to large applications as the knowledge and articulations in the mediator simply add on as new applications are added. Our approach generalizes the notion of articulations and provides a more flexible framework for the integration of knowledge, specifically knowledge and articulations are incorporated on only an application specific and as-required basis. Consider a configuration where an information integration system i.e., mediator or other system provides integrated access to a certain (fixed) set of information sources. We refer to this as an integration configuration, for instance as illustrated in Fig 3 we have an integration configuration across 3 information sources (in reality of course the number of sources is typically larger). An integration configuration serves a number of applications, for instance the configuration in Fig 3 serves 2 applications A1 (say an employee payroll application) and A2 (say an agency wide project management application). The capabilities we incorporate in our approach are:

(i) The capability of selecting **relevant** articulations. For any integration configuration, the existing approach is to maintain all articulations for all its associated applications at the mediator. We advocate a more scalable approach which is to maintain articulations associated with *clients* i.e., with applications [13]. For example as shown in Fig 3, application A1 may require *only* the articulations between Ames and JPL budgets or application A2 may require *only* the articulations between JPL personnel and HQ personnel. Our framework provides the capability to create and select articulations that are relevant to a new application and on as as-required basis, also articulations are maintained at each client per application and not centralized for all applications at the mediator. Quantitatively, if $a_1,\ldots, a_n$ are applications and if $N(a_i)$ is the number of articulation rules (an assessment of complexity) for application a_i, then with existing approaches we have a total of $\Sigma_{\text{all i}} N(a_i)$ articulation rules at the mediator whereas with the application

specific approach we have a maximum of $\mathbf{MAX}(N(a_i))$ rules associated with any application. For large applications typically $\mathbf{MAX}(N(a_i)) << \Sigma_{\text{all i}} N(a_i)$; thus a large complexity at the mediator is now shifted to each client with the complexity at each client being much smaller than what would have been at the mediator.

(ii) **Algebra** for Knowledge Selection and Manipulation. In existing approaches the knowledge required for *all* applications is maintained at the mediator for a particular configuration. Applications however require only the knowledge that is relevant to that application. For example application A1 in Fig 3 may really require only the knowledge of BUDGETs from the Ames and JPL sources and other available knowledge such as that related to PERSONNEL may be irrelevant to this application.

We incorporate an ontology algebra [18] that enables us to systematically select and combine and define the knowledge for each application. The primary concepts in the algebra are:

Intersection: The intersection is the first concept of the domain algebra since it allows the algebra to bring together two domains. It is equivalent to an AND operator. The intersection of two knowledge sources (ontologies) results in an ontology that contains (only) the concepts that *have been articulated* as being the same concepts. For instance the intersection of the EMPLOYEES ontologies from the JPL and Ames sources i.e., K1 and K2 would be an ontology with the concept PERMANENT-EMPLOYEE (or FULL-TIME EMPLOYEE) as these are (all) the concepts that have been determined to be semantically the same by the articulation rules.

Union: The union concept allows the algebra to bring together two domains to form a new one. It is equivalent to an OR operator. However the algebra lacks a formal approach to eliminate redundant knowledge that is common to both. This leads to several ways of establishing the unions of multiple domains. It is convenient to think of knowledge as not being redundant if not explicitly specified by the articulation rules. Similarly to the natural join in relational databases, the domain algebra union joins knowledge sources when they link through shared articulation rules. The union is restricted only to the knowledge that the rules relate to. For instance the union of the EMPLOYEES ontologies in K1 and K2 would be the shared concept PERMANENT-EMPLOYEE (or FULL-TIME EMPLOYEE) *plus* all the other concepts such as INTERNS, CONTRACT-EMPLOYEES etc.

Difference: The difference concept completes the algebra and its presence compensates for the absence of negation. The difference operation retrieves the elements in domains that are NOT covered by another. Hence, the difference operation results in asymmetrical results and is not commutative.

The above algebraic constructs arms us with a systematic and comprehensive mechanism to select and manipulate knowledge specific to an application need. As with articulations the complexity is thus confined to the application.

(iii) **Context.** The 3rd fundamental construct we use to bring scalability is the notion of context. The notion of context provides a way to define the validity of a sentence relative to a situation [14,16]. Context logic provides the capability of translating encoding knowledge relative to its context and hence relates the knowledge to its domain. For instance one may specify the term “vision” as query

with the intent of the use of the term vision on the context of program management and future planning or in an entirely different sense of vision related equipment for astronauts. We provide for the ability of situating knowledge in particular contexts. When searching or querying information over large numbers of sources it is context, as we shall demonstrate, that is a simple but powerful enabler in achieving the relevance and scalability that is required.

We refer the reader to [13,17,18] where the above summarize theories of articulation management and ontology algebra are discussed in more detail. The ontology algebra and articulation management capabilities are essentially tools for the *integration configuration assembler* in forming the knowledge sharing and integration required for a new application. The notion of context results in a context sensitive querying capability for the *end user* that we shall elaborate on now.

2.1 Document Querying Based on Articulation and Algebra

The notion of context is practically realized as a simple yet powerful primitive for querying and searching heterogeneous, distributed document collections in a context sensitive fashion in NETMARK. Any document is essentially comprised of various sections and sub-sections; for instance the project summary document in Fig 4 above is comprised of a **PROJECT SUMMARY** section, and **Background** and **Purpose** sub-sections etc. These fragments such as the project summary section, background sub-section etc., are referred to as *context*. The information within the context, in this case the text within the fragment is referred to as *content*.

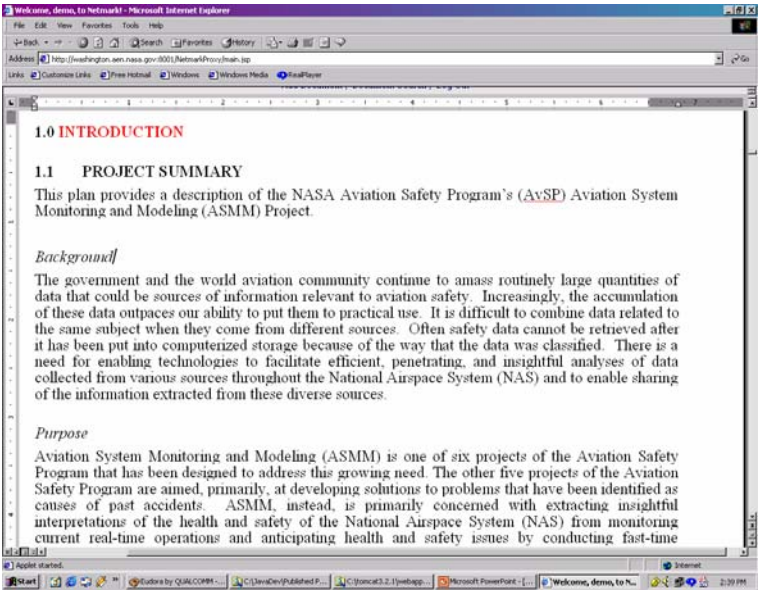


Fig. 4 Document Sections

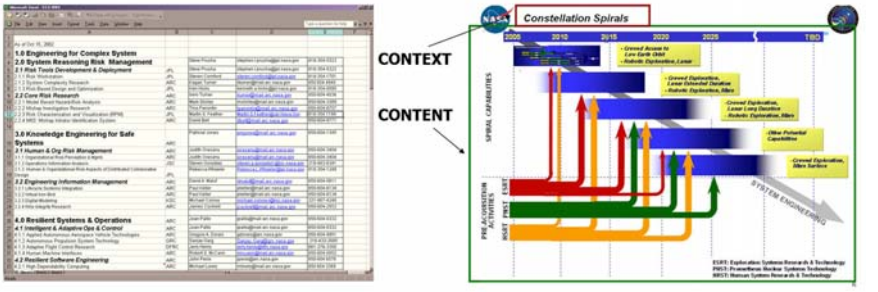


Fig. 5 Context and Content

These notions extend to documents beyond text documents as well, for instance a spreadsheet is also comprised of different fragments i.e., cells, rows and columns, and associated groups thereof, a (PowerPoint) slide comprises of a slide title (context) and the associated slides content (content), or an email message can be considered as comprising of the context of its subject and content as the actual email message text.

Querying

Such fragmentation, into context and associated content permits context sensitive search and querying. A key capability is that of *context search*. A context search query, such as "Context=Procurement" [2] will return the content portion in the 'Procurement' sections (the text in the Procurement section) in *all* the documents in a document collection, as illustrated in Fig 6. A context query thus extracts the specified context (section) from all documents and returns it to the user. Users can also specify *content searches,* which are essentially keyword searches that return all documents containing the specified search terms. For instance, a content query such as "Contract" will return all documents that contain the term 'Contract' anywhere in the document. One can combine context and content searches, for instance a query such as "Context=Procurement Comment&Content=Contract" returns the "Procurement" contexts (sections) of all documents where the term 'Contract' occurs *within the Procurement context (section)* as shown in Fig 6.

Essentially NETMARK provides keyword-based search over large (originally) unstructured document collections but with an added powerful capability of context sensitiveness, i.e., the user is able to ground the search terms in a particular context of interest. XDB Query is the query language for NETMARK. We will not go into the query syntax details here but the key features are that context and content search specifications are appended to a URL that is sent to NETMARK. An example of a formal XDB query, and also the XDB query syntax is illustrated in Table 1 below.

[2] We are using an informal syntax for illustration and will describe the actual query syntax shortly.

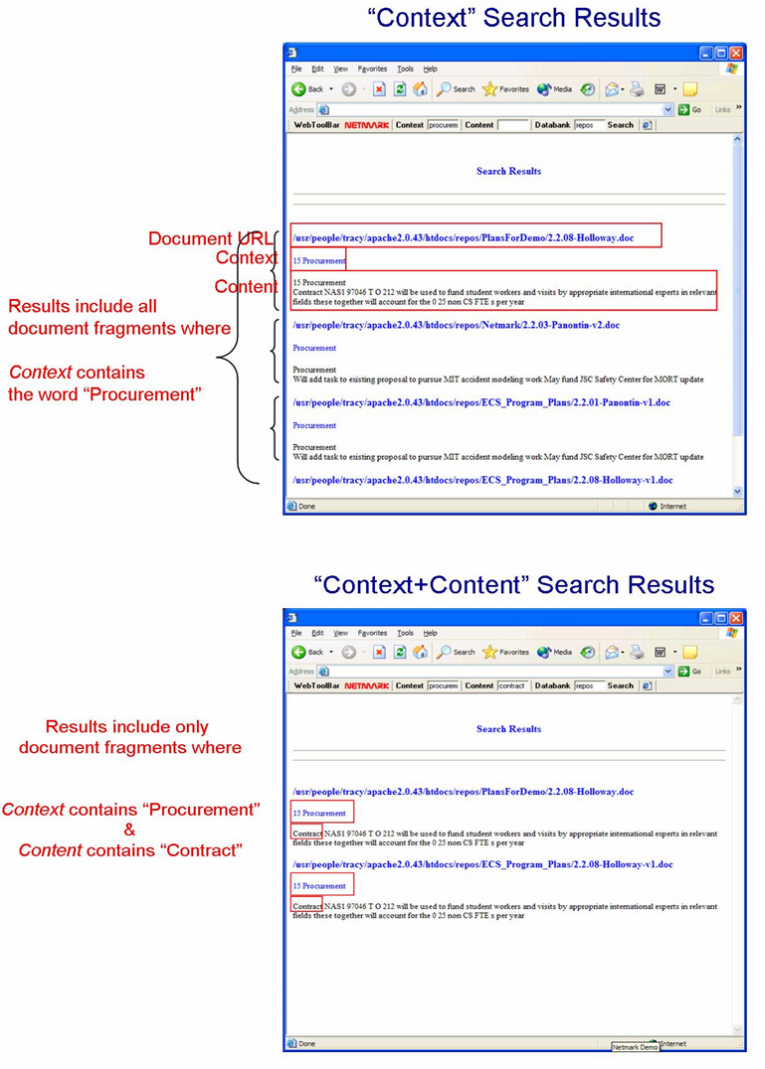
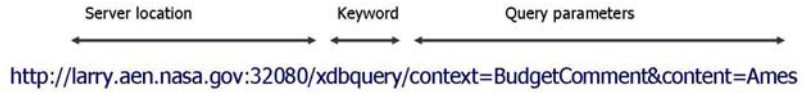


Fig. 6 Context Sensitive Querying and Retrieval

Table 1 XDB Query Syntax



https://<server_address>/xslt/**xdbquery**/{[**context=**<context_keys>] | [**&content=**<content_keys>]} | [**&scope=**<relative_url_to_folder>] | [**&syntax=**{html, xml, ascii}] | [**&sxslt=**<relative_url_to_xslt_file>]

Context and content parameters can be specified in the query parameters aspect of the XDB query. One can also specify additional parameters which can control the maximum number of documents returned, the (tree) depth of the result items, we refer to [19,21] for details.

3 NETMARK: Technical Details

We first briefly illustrate how integration applications are built using NETMARK and then describe the design and implementation of the NETMARK system itself. As regards how information integration applications are assembled NETMARK makes a significant departure from other such systems. Consider having to create an integration configuration across the three information sources illustrated in Fig 3. We first create a unique resource, a URI, corresponding to this configuration. This URI corresponds to the virtual integrated source across all three sources. Next, we load information i.e., enterprise documents corresponding to employee, project etc information into this URI. This is done by a simple drag-and-drop desktop operation at source (JPL, Ames, and HQ) where the desktop folder is actually a remote desktop folder corresponding to the URI. Now the information across all three sources can be simply queried by issuing XDB queries to the integration URI. For example a context query requesting EMPLOYEE fragments will return EMPLOYEE fragments in data (originally) from both JPL and Ames sources. Should any articulations be required they are created and attached to the specific application as needed. We refer to the system documentation [28,29] for more details.

3.1 NETMARK System Design

As a data management engine, NETMARK is based on a "schema-less" paradigm that provides high efficiency and throughput in retrieval. Before describing the architecture and technical details we wish to highlight some additional features that we have incorporated that address key tasks in the information pipeline. These are:

(i) Capability of ingesting information "as-is". No data preparation or mark-up whatsoever is required from any user that wishes to provide data for incorporation and integration into NETMARK. Enterprise data in a multitude of formats ranging from Word or PDF documents to Excel spreadsheets to PowerPoint presentations is provided to the system as is which then structures the data as we shall describe shortly.

(ii) Information composition and presentation capabilities. XDB Query also provides for associating XSLT style-sheets with a query, the query result thus gets presented in the desired format. Integrated data collected from multiple sources is often composed (back) into common business, documents; for instance project information integration from multiple divisions and departments would be composed and presented in business document format such as report or a slide presentation. Commonly used business documents can thus be used as the *interface* to integrated data.

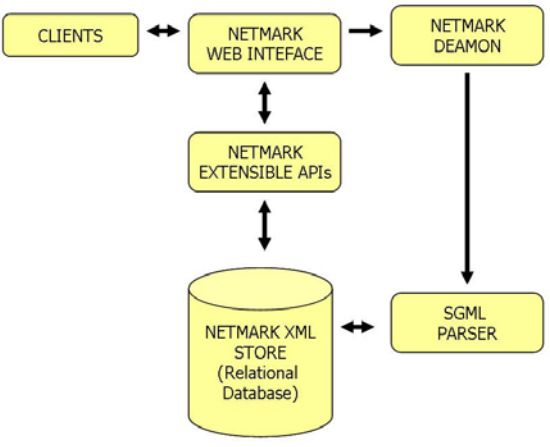


Fig. 7 NETMARK System Architecture

The NETMARK system architecture is outlined in Fig 7 below. All data is (ultimately) stored in a single data store which is an XML data store, implemented on top of an underlying relational database.

Clients i.e., data producers and providers and data consumers (or both) access NETMARK through a Web interface, which we illustrated in the examples in Section 2. Any data, such as say a folder of several PDF or Excel documents can be provided to NETMARK by a simple drag and drop operation (into a "NETMARK Folder" on the users desktop). The NETMARK Daemon and the SGML Parser provide functionality for loading data (documents) into NETMARK i.e., a continual process (the daemon) reads in any new documents inserted into a NETMARK folder and then invokes an SGML parser for structuring it and loading it into the NETMARK XML data store.

3.2 Data Storage

Data with varying degrees of structure ranging from data that originates from a well structured database to unstructured data that is in documents and spreadsheets, is integrated into and supported by NETMARK. For data that is unstructured, some structure is automatically imposed based on fragments, sections and sub-sections in the documents. The approach to data storage is to keep the underlying representation simple, yet expressive enough to store fragment and section oriented properties and relationships in documents.

Data Fragmentation, Structuring, and Storage

Any data to be stored into NETMARK, whether originally structured or unstructured, is first fragmented into sections and sub-sections which are then marked in XML, the XML data is then stored as a tree of "nodes", finally the nodes are stored in relational tables. This pipeline is illustrated in Fig 8, where we begin

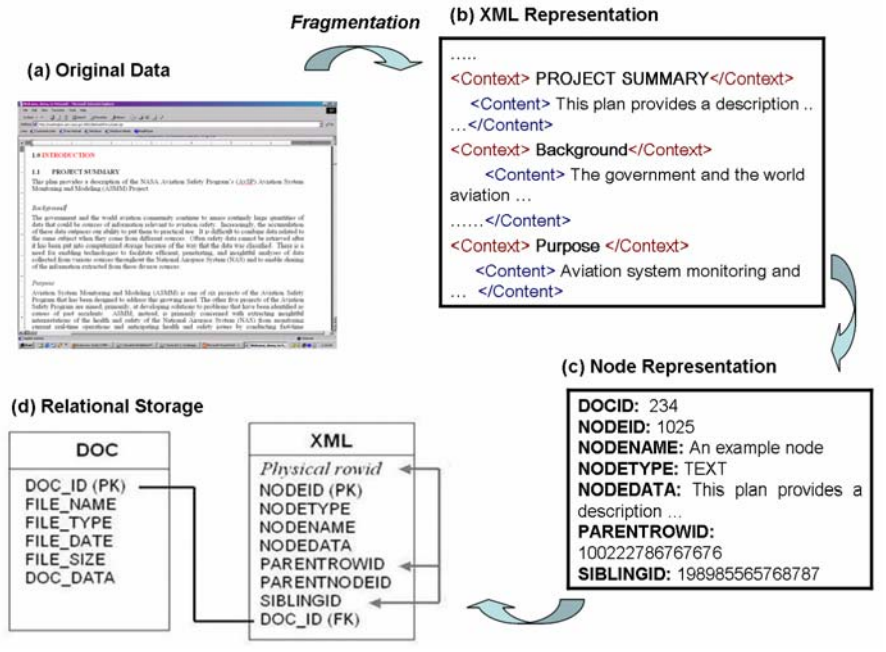


Fig. 8 Data Storage Pipeline

with the (originally unstructured) document in Fig 4. As a first step this is fragmented into different sections and sub-sections and marked up in XML as shown in Fig 8.

Such fragmentation is done by a suite of converters that are part of NETMARK. These converters have been built on top of (text extraction) frameworks such as Apache Jakarta POI [3] and JPedal for PDF [4] and employ heuristics to automatically fragment an unstructured document into various sections which are then marked up in XML. We next introduce the concept of a *node* which is the fundamental unit of data storage in the system. A node essentially captures the information in each context and content fragment in the document. Thus there is a node corresponding to *each* context or content fragment in the document. Every node carries in it certain information as described in Table 2 (a). As we see this is information such as a unique identifier for that node, or a type corresponding to the particular fragment it is capturing, for instance nodes of type 'TEXT' typically capture information in content fragments and nodes of type 'CONTEXT' capture information in context fragments.

Table 2 (b) illustrates a node of type 'TEXT' corresponding to a particular content fragment (encircled in Fig 8) where we see that the NODEDATA element of the node contains the text in that fragment.

[3] http://jakarta.apache.org/poi/
[4] http://www.jpedal.org/

Table 2 Nodes

DOCID: *A unique number assigned to the document.* **NODEID:** *A unique identifier for each node.* **NODENAME:** *A descriptive name for the node* **NODETYPE:** *Identifies the node type, which is one of a small list of mutually exclusive node types.* **NODEDATA:** *The actual content of the node.* **PARENTROWID:** *Contains the ROWID of a parent of the node (if any).* **SIBLINGID:** *Contains the ROWID of a sibling of the node (if any).*

DOCID: 234 **NODEID:** 1025 **NODENAME:** An example node **NODETYPE:** TEXT **NODEDATA:** This plan provides the … **PARENTROWID:** 100222786767676 **SIBLINGID:** 198985565768787

Hierarchical parent-child relationships are also maintained across certain nodes, specifically the following relationships are maintained:

(i) Any node of type 'TEXT' i.e., capturing a content fragment is placed as a *left* child of the node capturing its *corresponding* context.

(ii) Any node of type 'CONTEXT' is placed as a *right* child of the node corresponding to a context *immediately preceding* it in the document.

For the running example, some of the nodes and their parent-child relationships according to (i) and (ii) above are illustrated in Fig 9, the sharp-edged boxes represent context nodes and the rounded-edge boxes represent content nodes.

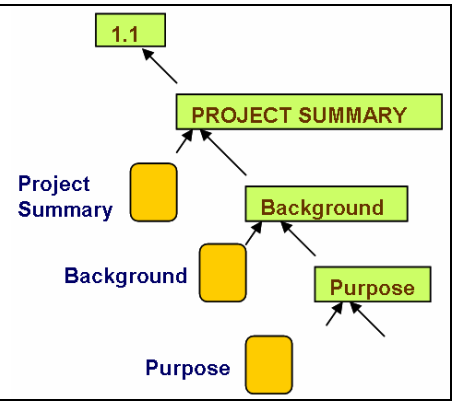


Fig. 9 Tree Structure

The rationale for this organization is that we wish to maintain adequate structuring information such as the association of content with particular context and at least immediate precedence relationships amongst different contexts, however we also wish to keep the hierarchy of relationships simple. In the nodes these relationships are captured by the 'PARENTROWID' and 'SIBLINGID' elements which maintain pointer relationships across nodes.

As regards the storage of XML data, in general it is ultimately stored as either tree-structures in XML databases that provide "native" XML implementations [30], or another popular approach is to store it in an underlying relational database and a variety of "shredding" algorithms exist to meaningfully convert the XML data to underlying relational tables. The number of such underlying relational tables and complexity of organization is dependent on the actual XML data in existing approaches. However in NETMARK we use just *two* relational tables to represent and store the data in *any* semi-structured document, these tables remain the same for any document or application. This is possible given the restrictions we have made on the hierarchical relationships across nodes above. As shown in Fig 8, these two tables are called "XML" and "DOC", the XML table contains all the nodes and the DOC table contains information about all the documents.

To summarize the above information processing pipeline from unstructured input data to storage in relational tables, we (i) Fragment an unstructured document into various sections and sub-sections and convert to XML, this results in context and content blocks defined in XML, (ii) Create nodes corresponding to each context or content block, (iii) Capture hierarchical structure, i.e., parent-child relationships between nodes (including that of associated context and content) through pointers across nodes, and (iv) Store node and document information in two relational tables, namely XML and DOC

3.3 Efficient Query Processing

Given an XDB query, query processing in NETMARK basically involves locating the relevant nodes and composing the requested result for these nodes. We have exploited the availability of the ROWID which is a data type available in Oracle 9i and later versions which store either *physical* or *logical* row addresses or each record in a table. A physical ROWID is the actual (absolute) address of a record through which we have the fastest access to any record in a table, with a guaranteed single-block read access. We refer to [19] for more details on the ROWID format details but would like to emphasize here that the use of ROWID s is a key to efficient query processing in NETMARK. The physical ROWID based technology is now patented [20].

Context and content search is performed by first querying the text index for the search key. Several matching nodes may be returned. For each such node we traverse the tree structure (through the node's parent or sibling nodes) until the first *context* node is found. Once a particular CONTEXT is found, traversing back down the tree structure via the sibling node retrieves the corresponding *content* text. The search result is then rendered and displayed appropriately. Accessing a

record based on its physical address ROWID provides an efficient, constant access time C (machine dependent; normally in the millisecond range) that is independent of the number of records or nodes in the database and regardless of maximum node depth within a node structure. The time to respond to a context or content query is thus approximately proportional to log(N) (first search time) plus a sum of the Cs for each successive search where N is the number of records or nodes.

3.4 The Schema-Less Aspect

A traditional i.e., an object-relational mapping from XML to a relational database schema models the data within the XML documents as a tree of objects that are specific to the data in the document. In this model, element type with attributes, content, or complex element types are generally modeled as classes. Element types with parsed character data (PCDATA) and attributes are modeled as scalar types. This model is then mapped to the relational database using traditional object-relational mapping techniques or via SQL3 object views. Therefore, classes are mapped to tables, scalar types are mapped to columns, and object-valued properties are mapped to key pairs (both primary and foreign). This mapping model is limited since the object tree structure is different for each set of XML documents. On the other hand, the NETMARK SGML parser models the document itself (similar to the DOM), and its object tree structure is the *same* for all XML documents. Thus, NETMARK is designed to be *independent* of any particular XML document schemas and is termed to be "*schema-less"*.

4 Performance

For any data management system, we want an assessment of its performance in query evaluation in absolute terms as well vis-à-vis other systems in its category. As emphasized, NETMARK is a really a semi-structured data management system targeted towards context and content kinds of queries and with support for XML as a representation and exchange mechanism. Despite this distinction, carefully designed (and now considerably widely used) *benchmarks* for XML query processing evaluation deserve consideration for the evaluation of NETMARK. We have employed XMARK [22] in particular, albeit with considerations about some aspects. First, the XMARK framework generates test data in the form of an XML document in a domain of transactions, people, and auctions using a data generator called xmlgen. Such generated data is indeed reasonable for evaluating NETMARK as it is representative of the kind of semi-structured data that NETMARK is designed to manage. Next however is the issue of test queries; XMARK includes a suite of 19 test queries, Q1-Q19, that are designed to evaluate a whole range of XML querying aspects ranging from aggregation to structural joins to handling of complex path expressions. NETMARK however is not designed or even intended to support such capabilities. We thus pick a relevant subset of these test queries, specifically ones that directly correspond to contextual search that

NETMARK supports. We have also added some additional contextual search test queries, at various levels of depth in the XMARK test document, for more exhaustive test coverage.

The evaluation below presents query response times for queries that correspond to context and content kinds of queries. In addition to absolute numbers, we also provide a comparison with (Oracle) Berkeley DB (which we refer to as BXML), an XML over relational system, under the same configuration. The results are provided for a single XML document with sizes ranging from 50MB all the way to 1GB. These evaluations were conducted on an i686 machine with 4 Intel Pentium (R) 2.8 GHz processors running GNU/Linux. We refer to [22] for details about the XMARK benchmark and associated data generator and test query suite.

4.1 Performance Results

We selected a subset of queries from the original test queries suite of the XMARK benchmark and also added some queries of our own for more exhaustive testing of relevant aspects. These queries are listed in Table 3 below, we provide the syntax for expressing these queries in both XDB Query (used for NETMARK) and XQuery (that we use for BXML).

Table 3 Test Queries

Query	XDB Query	Equivalent XQuery	Result Size ₸
Q1	context=id & content=person0	/site/people/person/[@id='person0']	1
Q6	context=item	/continents//items/String()	
Q14	content=gold	//site[dbxml:contains(./*, "gold")]	
NQ1	context=country	//country/string()	12716
NQ2	context=payment	//payment/string()	21750
NQ3	context=country & content=Tonga	//country[dbxml:contains(., "Tonga")]	12
NQ4	context=payment & content=cash	//payment[dbxml:contains(., "cash")]	10933
NQ3'	context=country & content=Tonga	//country[. = "Tonga"]	12
NQ4'	context= payment & content= Cash	//payment[. = "Cash"]	10933

₸ Number of XML elements in 100M XML document.

There are a few points we wish to highlight regards the selection of queries in Table 3.

(i) As mentioned above, queries in the original XMARK benchmark that relate to functionality not in XDB Query (such as complex path expressions, joins, aggregation, etc.) are not selected. We have thus selected only queries Q1, Q6, and Q14 from the original XMARK test suite.

(ii) Some additional queries relating to context and content have been added (NQ1-NQ4) that perform context and content searches on XML elements at various depths.

(iii) In XDB query for a content match we only provide the semantics of containment of which an exact match is a special case. This is different from XQuery where we make a distinction between requiring an element to exactly match a given string vs the element containing that string. Thus for the contextual search queries NQ3 and NQ4 we have considered both interpretations (i.e., exact match and containment) when expressing them in XQuery (NQ3' and NQ4')

Table 4 Performance Results

	50M		*100M*		*250M*		*500M*		*1G*	
	BXML	**NM**	**BXML**	**NM**	**BXML**	**NM**	**BXML**	**NM**	**BXML**	**NM**
Q1		0.08*		0.91		2.4		6.9		
Q6		24.9		65.1		108.1		276.5		
Q14	21.3	0.01	40.2	0.02	240.0	0.32		0.74		
NQ1	25.5	21.0	60.2	61	102.1	105.4		300.1		
NQ2	23.4	18.1	30.9	29	110.3	109.7		236.4		
NQ3	12.2	0.1	23.0	0.83	105.6	3.8		8.4		
NQ4	11.1	2.9	21.7	3.4	97.0	6.7		13.3		
NQ3'					†					
NQ4'										

* All response times are in seconds.

† No response for over 1 hour.

Table 4 provides the query response times for the test queries (Table 3) for both BXML and NETMARK for varying benchmark document sizes under the same configuration. There are two important observations to be made. (i) For context only queries, the performance of NETMARK appears to be comparable to that of BXML. For this class of queries NETMARK appears to perform "as good as" a representative XML database system. (ii) For context+content queries (i.e., XQuery queries involving a text search within an XML element) NETMARK is significantly faster compared to BXML, in fact as much as 25 times faster in some cases as demonstrated.

What we can claim to have achieved with NETMARK is a system that for the kinds of (context and content) queries it is designed to support is, depending on the type of query, comparable to or significantly faster than state-of-the-art XML database systems for the same functionality. Note also that such performance has been achieved with relatively much simpler query processing algorithms given the simple schema-less nature of the underlying relational database.

In Fig 10 we demonstrate how the query response time for NETMARK scales with document size for the various test queries (divided into 2 sets based on the actual response times). There is one other aspect to performance in NETMARK besides query response times, which is the time taken for loading documents into the system. Note that NETMARK automatically fragments and structures input data before storing it in the system and for large applications it is important that this document loading be efficient. Our earlier work [19] benchmarks this aspect as well demonstrating high-throughput rates for loading new input documents into the system.

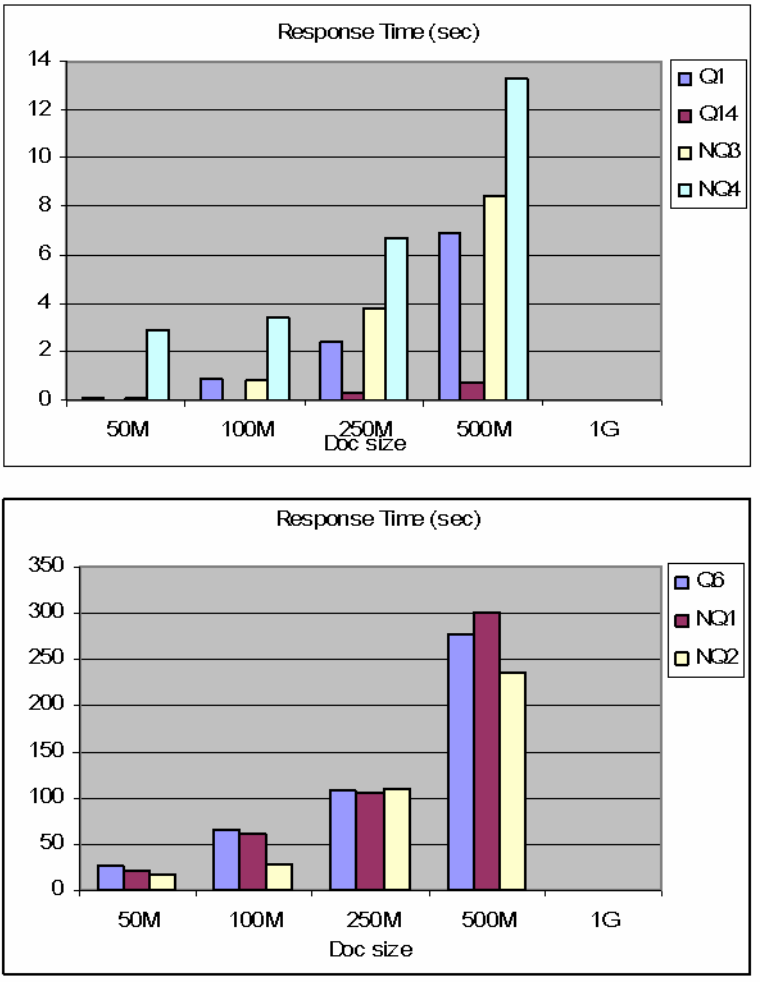


Fig. 10 Response Time and Document Size

5 Application, Case-Studies

Recall the key hypotheses on which this work is based. In essence we claimed that (i) a more flexible and application specific knowledge sharing approach and (ii) context-sensitive querying would result in an integration system that is both effective and scalable. Further the system is expected to be easy-to-use with minimal training and expertise through the use of desktop and Web-based system interaction. Such claims are best, and perhaps only, validated by actual system use and deployment in real-world applications, which is what we have done extensively in the NASA enterprise and beyond in the last few years. NETMARK has been deployed in several information integration applications in the NASA enterprise,

further it has been incorporated as the information integration engine for other information and process management systems that have a broader functionality. Specifically NETMARK has been integrated with the XEROX Docushare system which has resulted in a content and document management system called "NX". It has also been enhanced with several capabilities to support information flow in a project management lifecycle, resulting in the NASA Program Management Tool or what is referred to as "PMT". We provide a description of some of these systems and their enterprise application use. Literally all of the applications served are over thousands of enterprise documents of various kinds from multiple different NASA departments, centers and organizations.

NETMARK NETMARK has been deployed for many information integration applications within NASA and other organizations. As an example one of the NASA applications is the analysis of *mishap* reports as part of aviation safety analysis. Such analysis reports are (typically) text reports describing the analysis of a range of accidents involving NASA and non-NASA aircraft. The use of NETMARK permitted the abstraction and selection of particular sections of interest from reports and also the integration of information across multiple reports. The structured data was then fed to data analysis and visualization tools for tasks such as multi-dimensional analysis.

We must mention that a minimal effort and time was expended in the assembly of this particular application with required zero investment in additional software development and required just 2 man days for system setup and application assembly. Several hundreds of thousands of such reports from different sources have been integrated. Apart from several installations at NASA, NETMARK has been licensed to various public and private organizations including Black Tulip, XEROX, the State of Pennsylvania, NXAR Inc., Jumpstart Inc., and the University of California, Irvine.

NX The NX system is the result of a strategic collaboration between NASA and XEROX Corp, where NETMARK has been integrated with many XEROX Docushare capabilities for text and document management. NX offers a suite of capabilities in 1) Content management, including capabilities for content and document management and sharing, distribution and collaborative sharing, and 2) Content process management, i.e., business process activities such as tracking and compliance. The key benefit is the existing documents and applications get seamlessly incorporating into newly automated systems with NX. NASA has implemented the NX technology at six centers and in various programs, including the following 1) The International Space Station (ISS) which uses NX to mine information for historical decisions and safety assurance information, 2) NASA Program Analysis and Evaluation (PA&E) which adopted NX in 2005 and which led to adoption by the NASA's strategic management council, and 3) Most NASA centers use the NX platform, including Ames, LaRC, GSFC, Dryden, JPL, JSC and NASA Headquarters. There are over ten different kinds of applications that have been realized using NX, ranging from automated report generation to contextualized enterprise search to knowledge sharing and groupware applications. The number of seat licenses for NX at the NASA JPL, Ames, Langley and Dryden centers are currently 6000, 3500, 1400 and 1200 respectively.

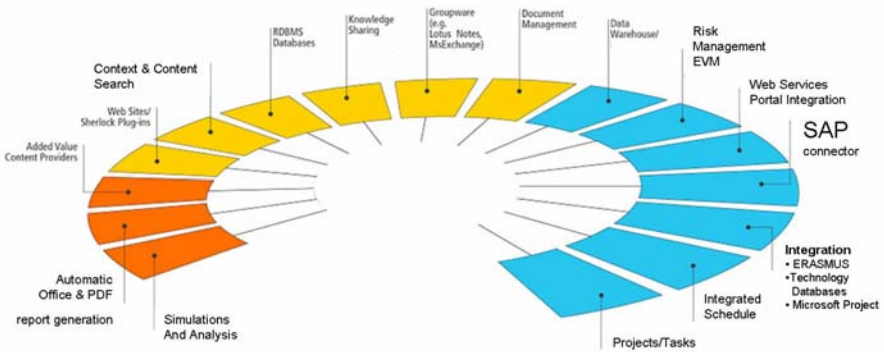


Fig. 11 PMT Applications

PMT Another key system that has NETMARK as the information integration engine is the Program Management Tool (PMT), which is a custom-built business intelligence solution developed at NASA to successfully manage large programs [24]. It enables program and task managers to communicate success critical information on the status and progress of all program levels in an efficient and always current manner by keeping track of project goals, risks, milestones and deliverables, and assisting with the proper allocation of financial, material, and human resources. PMT also provides integrated access to multiple distributed resources across the NASA agency, namely the "ERASMUS" reporting system (ERASMUS is an executive reporting system and project performance dashboard that includes performance metrics of all NASA centers, programs, projects, and safety and health activities), the NASA Technology Inventory Database (an inventory of technologies developed by or under development at NASA), and the Integrated Financial Management System IFMP (an agency-wide information system supporting NASA financial management activities). PMT has been used for processing of over a 1000 WBS at the NASA Chief Engineer's office and at various NASA mission directorates.

A comprehensive overview of PMT or a description of the development and realization of this system merits a separate discussion and we refer the reader to [24]. What we wish to emphasize here is that the system has been used extensively for key NASA technical management and financial management tasks such as program and project WBS[5] definition, resource planning and tracking, risk management, schedule management, periodic project status reports, performance reporting, budget formulation, phasing plans, financial roll-up reports, and guideline and funds received tracking. All the above applications are data intensive and it is NETMARK that is the information integration engine achieving the required information integration for all these applications. Given the wide acceptance and deployment of NETMARK within NASA and also beyond, positive feedback on

[5] A WBS i.e., Work Breakdown Structure is a plan associated with each NASA which breaks down each project into manageable pieces of work to facilitate planning and control of cost, schedule and technical content.

easy of use and configuration, the reduction of application assembly time from several months to a matter of days, and the near elimination of the requirement for expert consultants for assembling each application, it is fair to claim that the driving hypotheses in our work have indeed been validated. We refer to [25,26,27] for more detailed descriptions of the NASA applications of NETMARK.

6 System Interface and Availability

NETMARK includes an AJAX-SQL library [29] that provides an interface to the NETMARK XDB query server. The primary purpose of this library is to provide enterprise users i.e., at present NASA personnel, the capability of easily querying unstructured information in multiple NASA repositories based on both context and context and further recompose documents based on the results of the queries. A standard browser is all that is required for accessing the information in the many different proprietary information sources. The following Table 5 illustrates the fundamental features of an AJAX-SQL query.

The types of searches that AJAX-SQL enables over unstructured data include Context only search, Content only search, Combined context and content search, Combination search, XML data search, Unique value search, and Post Processing the query options.

Table 5 AJAX-SQL

Term	Description
Select	*Select data from a table*
Content	*Explained above*
Context	*Explained above*
From	*Address location of NETMARK XDB server*
Where	*Conditions select for type of data*
Distinct	*Conditions unique data results*

```
<script>
.....
      oTest.ajaxsqlQuery("select distinct [scr_number pvcs_id]
                              from [tinyft2.xml] where [title = shuttle]
                        cache,meta,offsets,snippet,ctx number of result");
</script>
...
```

Fig. 12 Embedding AJAX-SQL in Javascript

Fig 12 above provides a simple example of how AJAX-SQL can be embedded in Javascript for an application that accesses the NETMARK XDB server and displays the results on a Web page. Instead of XDB query we see the use of SQL like primitives, the above example for instance requests the `src_number` (a source number) and `pvcs_id` (a parts id of some sort) *elements* from fragments of an XML document (tinfyft2.xml) where 'shuttle' appears in the 'title' context. The parameters such as cache, offset etc. are configuration parameters.

System Availability Building upon and continuing the prior history of NASA in contributing to open-source software for the community, the NETMARK system has been made available for non-commercial research or academic uses under an open-source license from the NASA Ames Research Center. The available implementation is in Java with versions for both Linux and Windows. It requires either Oracle (9i or later) or MySQL as the underlying database. Interested groups may contact any of the chapter author on obtaining a copy of the software.

7 Related Work

As a comprehensive system with many aspects, the NETMARK work relates to work in several areas such as knowledge sharing, XML data management and query processing, XML and text search, and information integration technology in general. All of the above have been actively investigated by academia and industry. In comparison, the distinguishing features of our work can be summarized in the following contributions.

1) The flexible and scalable approach provided to knowledge sharing and integration. Our approach has made knowledge sharing and management for information integration more scalable, by keeping this application specific and making it a client (application) responsibility as opposed to a mediator responsibility. Articulation management and an ontology algebra are the formal tools provided to do this.

2) A significantly optimized data management system based on a schema-less approach. Implementing XML over relational systems has been an active area of research. In all such work the underlying relational representations are document dependent and complex as they must capture the full XML structure of the document. Efficient query processing of XML queries over these underlying relational tables is then a challenge, for which there is now an impressive array of efficient algorithms and solutions [32,35,36,37] for efficiently processing even rather complex queries. In NETMARK by keeping the document structuring of enterprise data adequate yet simple, we are able to translate the XML representations to a document independent simple representation in just 2 tables. Coupled with relational database features (i.e., the availability of physical ROWID s) we are able to provide very efficient query processing for XDB queries with a relatively much simpler query processing algorithm. The performance evaluation results presented validate the significantly better performance NETMARK has as compared to other XML database systems. The area of full-text search in XML [33,39] has also been investigated actively in recent years. Many solutions have also

advocated and developed capabilities for XML text search in context. Such work [33,39] is focused on determining meaningful fragments in XML in response to a text search, for instance based on the least-common-ancestor (LCA) and other criteria. Again, given our simpler representation we are able to provide high-performance text searches in context (although our notion of context is in some sense more straightforward than as investigated for more nested XML). At this point we should perhaps emphasize that the spirit of our work is not that we should not use formal and nested structuring of documents when required, but rather that structuring in a simple fragment oriented manner is adequate for a large class on enterprise applications and that then we should use simpler representations and query mechanisms given the scalability benefits as a result.

3) An end-to-end information integration system with easy desktop drag-and-drop and Web-based information ingest and retrieval capabilities. While this is mostly an engineering issue, our experience is that such capabilities have a) Helped alleviate the 'resistance' on behalf of an owner of a particular information source to provide data to and join an integration configuration. Such resistance is often largely due to additional investment required in brining his or her particular data to a right or agreed upon format before integration which in our case is addressed by the system. b) The information composition capabilities have further providing cost and time savings in that integrated data can quickly be composed into reports and presentations that it is ultimately intended for.

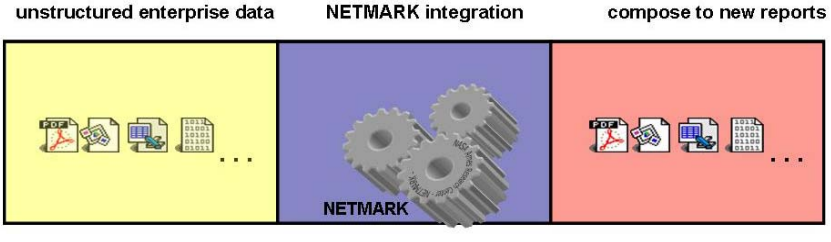


Fig. 13 Business Documents to Business Documents

8 Conclusions and Current Work

In this paper we described the conceptualization, design, and development of the NETMARK "schema-less" information integration system. We provided experimental results demonstrating the efficiency of schema-less data management systems and then through application case-studies demonstrating the effectiveness of the system in real-world information integration applications. There are active areas of ongoing work in both research and application of the system. In research and system development we are currently investigating incorporating secure access features in the system. An equal emphasis in this effort is on the application of NETMARK in real-world applications. Beyond domains such as project and personnel management, aviation safety data management etc., that we have applied the system to in NASA, we have begun investigating applicability in other

significantly different domains such as medical informatics. In fact the integration requirements in the medical and clinical informatics domain appear to pose many of the very problems that NETMARK is designed to efficiently address. As exemplified in [41], clinical data too is present in many different kinds of sources ranging from databases to text files (notes) to spreadsheets and applications often require integrated access to all the data. The UC-Irvine Center for Medical Informatics has recently obtained a NETMARK installation and is conducting a pilot information integration study with the system in this domain. Finally, we wish to re-emphasize our commitment to making this available as open-source software for research and/or non-profit use and welcome queries regards getting a license for the software.

References

[1] Halevy, A.Y., Rajaraman, A., Ordille, J.: Data Integration: The Teenage Years. In: Proc of VLDB (2006)
[2] Litwin, W., Mark, L., Roussopoulos, N.: Interoperability of Multiple Autonomous Databases. ACM Computing Surveys 22(3), 267–293 (1990)
[3] Halevy, A.Y., Ashish, N., Bitton, D., Carey, M.J., Draper, D., Pollock, J., Rosenthal, A., Sikka, V.: Enterprise information integration: successes, challenges and controversies. In: SIGMOD Conference 2005, pp. 778–787 (2005)
[4] Halevy, A.Y.: Data Integration: A Status Report. In: BTW 2003, pp. 24–29 (2003)
[5] Draper, D., Halevy, A.Y., Weld, D.S.: The Nimble XML Data Integration System. In: ICDE 2001, pp. 155–160 (2001)
[6] Papakonstantinou, Y., Borkar, V.R., Orgiyan, M., Stathatos, K., Suta, L., Vassalos, V., Velikhov, P.: XML queries and algebra in the Enosys integration platform. Data Knowl. Eng. 44(3), 299–322 (2003)
[7] Berners-Lee, T., Hendler, J., Lasilla, O.: The Semantic-Web. Scientific American (May 2001)
[8] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W.R.: Enabling Technology for Knowledge Sharing. AI Magazine 12(3), 36–55 (1991)
[9] MacGregor, R.M.: Inside the LOOM Description Classifier. SIGART Bulletin 2(3), 88–92 (1991)
[10] Brachman, R.J., McGuinness, D.L., Patel-Schneider, P.F., Borgida, A.: "Reducing" CLASSIC to Practice: Knowledge Representation Theory Meets Reality. Artif. Intell. 114(1-2), 203–237 (1999)
[11] Gruber, T.R.: The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. In: KR 1991 (1991)
[12] Collet, C., Huhns, M., Shen, W.: Resource Integration Using a Large Knowledge Base in Carnot. IEEE Computer 12(24) (December 1991)
[13] Maluf, D.A., Tran, P.: Articulation Management for Intelligent Integration of Information. IEEE Systems Man and Cybernetics (2001)
[14] Guha, R.V.: Context: A Formalization and Some Applications, Doctoral Dissertation, Stanford University (1991)
[15] Lenat, D., Guha, R.: The Evolution of CycL, The Cyc Representation language; Special Issue on Implemented Knowledge Representation System. ACM SIGART 2(3), 84–87 (1991)
[16] McCarthy, J.: Notes on Formalizing Context. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (1993)

[17] Maluf, D.A., Wiederhold, G.: Abstraction of Representation for Interoperation. In: Tenth International Symposium on Methodologies for Intelligent Systems. LNCS, pp. 441–455. Springer, Heidelberg (1997)
[18] Mitra, P., Wiederhold, G.: An Ontology-Composition Algebra. In: Handbook on Ontologies 2004, pp. 93–116 (2004)
[19] Maluf, D.A., Bell, D.G., Ashish, N., Knight, C., Tran, P.B.: Semi-Structured Data Management in the Enterprise: A Nimble, High-Throughput, and Scalable Approach. In: IDEAS 2005, pp. 115–124 (2005)
[20] Gawdiak, Y., La, T., Lin, Y., Maluf, D., Tran, P.: US Patent 6,968,338, Extensible database framework for management of unstructured and semi-structured documents, Awarded November 22 (2005)
[21] Maluf, D., Tran, P.: NETMARK: A Schema-Less Extension for Relational Databases for Managing Semi-structured Data Dynamically. In: Zhong, N., Raś, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2003. LNCS, vol. 2871, pp. 231–241. Springer, Heidelberg (2003)
[22] Schmidt, A.R., Waas, F., Ketersen, M.L., Florescu, D., Manolescu, I., Carey, M.J., Busse, R.: The XML Benchmark Project. In: CWI (2001)
[23] Paparizos, S., Al-Khalifa, S., Chapman, A., Jagadish, H.V., Lakshmanan, L.V.S., Nierman, A., Patel, J.M., Srivastava, D., Wiwatwattana, N., Wu, Y., Yu, C.: TIMBER: A Native System for Querying XML. In: SIGMOD Conference 2003, p. 672 (2003)
[24] Maluf, D.A., Bell, D.G., Ashish, N., Putz, P., Gawdiak, Y.: Business Intelligence in Large Organizations: Integrating Which Data? In: Esposito, F., Raś, Z.W., Malerba, D., Semeraro, G. (eds.) ISMIS 2006. LNCS, vol. 4203, pp. 248–257. Springer, Heidelberg (2006)
[25] Maluf, D., Tran, P.: Managing Unstructured Data With Structured Legacy Systems. In: IEEE Aerospace Conference, Montana (2008)
[26] Maluf, D.: Searching Across the International Space Station. In: IEEE Aerospace Conference, Montana (2007)
[27] Maluf, D.: Knowledge Mining Application in a IVHM Testbed. In: IEEE Aerospace Conference, Montana (2006)
[28] NETMARK XDB guide
[29] NETMARK API
[30] Jagadish, H.V., Khalifa, S., Chapman, A., Lakshmanan, L., Nierman, A., Paparizos, S., Patel, J., Srivastava, D., Wiwatwattanna, N., Wu, Y., Yu, C.: TIMBER: A Native XML Database. VLDB Journal 11, 274–291 (2002)
[31] Ives, Z., Halevy, A., Weld, D.: An XML query engine for network-bound data. VLDB Journal 11, 380–402 (2002)
[32] Funderbunk, J.E., Kiernan, G., Shanmugasundaram, J., Shekita, E., Wei, C.: XTABLES: Bridging relational technology and XML. IBM Systems Journal 41, 616–641 (2002)
[33] Li, Y., Yu, C., Jagadish, H.V.: Enabling Schema-Free XQuery with meaningful query focus. VLDB Journal (2008)
[34] Botev, C., Shanmugasundaram, J.: Context-Sensitive Keyword Search and Ranking for XML. In: WebDB 2005, pp. 115–120 (2005)
[35] Grust, T., Rittinger, J., Teubner, J.: Why off-the-shelf RDBMSs are better at XPath than you might expect. In: ACM SIGMOD Conference, pp. 949–958 (2007)
[36] Georgiadis, H., Vassalos, V.: Xpath on Steroids: Exploiting Relational Engines for Xpath Performance. In: ACM SIGMOD Conference, pp. 317–328 (2007)
[37] Boncz, P.A., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: a fast XQuery processor powered by a relational engine. In: ACM SIGMOD, pp. 479–490 (2006)

[38] Vagena, Z., Moro, M., Tsotras, V.: Twig Query Processing over Graph Structured XML Data. In: Workshop on Web and Databases WebDB 2004, Paris, France (2004)
[39] Xu, Y., Papakonstantinou, Y.: Efficient LCA based keyword search in XML data. In: EDBT 2008, pp. 535–546 (2008)
[40] Madhavan, J., Cohen, S., Dong, X.L., Halevy, A.Y., Jeffery, S.R., Ko, D., Yu, C.: Web-Scale Data Integration: You can afford to Pay as You Go. In: CIDR 2007, pp. 342–350 (2007)
[41] Anderson, N., Lee, E., Brockenbrough, J.S., Minie, M., Fuller, S., Brinkley, J., Tarczy-Hornoch, P.: Issues in Biomedical Research Data Management and Analysis: Needs and Barriers. Journal of the American Medical Informatics Association 14(4) (August 2007)
[42] Xalan, http://xml.apache.org/xalan-j/
[43] XML, http://www.w3.org/XML/
[44] Docushare, http://docushare.xerox.com/ds/
[45] Oracle
[46] MySQL
[47] Apache
[48] WebDAV

Part IV
Intelligent Agents

Managing Pervasive Environments through Database Principles: A Survey

Yann Gripay, Frédérique Laforest, and Jean-Marc Petit

Abstract. As initially envisioned by Mark Weiser, pervasive environments are the trend for the future of information systems. Heterogeneous devices, from small sensors to framework computers, are all linked though ubiquitous networks ranging from local peer-to-peer wireless connections to the world-wide Internet. Managing such environments, so as to benefit from its full potential of available resources providing information and services, is a challenging issue that covers several research fields like data representation, network management, service discovery... However, some issues have already been tackled independently by the database community, e.g. for distributed databases or data integration. In this survey, we analyze current trends in pervasive environment management through database principles and sketch the main components of our ongoing project SoCQ, devoted to bridging the gap between pervasive environments and databases.

Keywords: Pervasive environments, Databases, Continuous queries, Data streams, Services.

1 Introduction

As initially envisioned by Mark Weiser [61], pervasive environments are the trend for the future of information systems [43]. Heterogeneous devices, from small sensors to framework computers, are all linked though ubiquitous networks ranging

Yann Gripay
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
e-mail: yann.gripay@liris.cnrs.fr

Frédérique Laforest
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
e-mail: frederique.laforest@liris.cnrs.fr

Jean-Marc Petit
Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
e-mail: jean-marc.petit@liris.cnrs.fr

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 277–298.
springerlink.com

from local peer-to-peer wireless connections to the world-wide Internet. Managing such environments, so as to benefit from its full potential of available resources providing information and services, is a challenging issue that covers several research fields like data representation, network management, service discovery.

In order to cope with the development of autonomous devices and location-dependent functionalities, an abstraction of device functionalities as distributed services allows the pervasive system to automate some of the possible interactions between heterogeneous devices. As devices may be sensors or effectors, services may represent some interactions with the physical environment, like taking a photo from a camera or displaying a picture on a screen. These interactions bridge the gap between the computing environment and the user environment, and can be managed by the pervasive system through such services. Many projects of pervasive systems have been devised, e.g. [8, 13, 45, 46, 57, 58].

In this setting, even data tend to change their form to handle information dynamicity. The relational paradigm widely adopted in DataBase Management Systems (DBMS) for many years is too restrictive to manage pervasive environments with emerging data sources such as data streams and services. Queries in traditional DBMS are "snapshot queries" expressed in SQL: a query is evaluated with the current state of the database, and the result is a static relational table. The "snapshot" term expresses that the result represents only the state of the database at the moment of the query, and is never updated. With dynamic data sources, "snapshot queries" may be not sufficient as it would be computation-expensive to periodically execute them and obtain up-to-date results.

Data streams open new opportunities to view and manage dynamic systems, such as sensor networks. The concept of queries that last in time, called *continuous queries* [17], allows to define queries whose results are continuously updated as data "flow" in the data streams. This kind of data sources has drawn the attention of the database community for many years. Data Stream Management Systems (DSMS) have been studied in many works, e.g. [3, 7, 14, 18, 26, 55, 63].

From a data-centric point of view, traditional databases [28, 44] have to be used alongside with non-conventional data sources like data streams and services to deal with new properties such as dynamicity, autonomy and decentralization. Query languages and processing techniques need to be adapted to those data sources. Data management systems tend to evolve from DBMS (DataBase Management System) or DSMS (Data Stream Management System) to a more general concept of DataSpace Support Platform (DSSP) [25]. A DSSP is intended to deal with "large amount of interrelated but disparately managed data". However, many issues have already been tackled in the field of databases to extend databases in this new setting, like distributed databases or data integration.

In this survey, we study pervasive computing from a data-centric point of view. Current trends in pervasive environment management can be related to, and enhanced by, current research in the database community. In this setting, we set up an ongoing project, called SoCQ, as our attempt to bridge the gap between pervasive computing and database principles.

In Section 2, we first give an overview of pervasive systems and of the many issues in this field. We then show how database principles have been leveraged to answer to new constraints in those environments in Section 3. In Section 4, we tackle enabling technologies for pervasive systems. We then discuss our approach to manage pervasive environment through database principles in Section 5. Finally, we conclude this survey and discuss some open issues in Section 6.

2 Overview of Pervasive Systems

Pervasive computing, or ubiquitous computing [61], is "a paradigm for the 21st century" [50] that tackles "connecting the physical world" [23] to a ubiquitous network and discovering available resources [64] in the environment. With such settings, applications like "Data Space" [38, 25] or "Programmable Pervasive Space" [34] can be realized.

Pervasive information systems can be analyzed as the interaction of three layers, each one interacting with the "individual layer" representing the user [41]:

1. the infrastructure layer, that represents the technical part for supporting pervasive systems inducing capabilities (and limitations) for the second layer;
2. the service layer, that represents applications that can be built in pervasive systems to answer to user expectations;
3. the social layer, that imposes some restrictions upon the behavior of applications to enforce social rules, like legal aspects and user privacy.

In this overview, the focus is put on the infrastructure layer and the service layer through the presentation of the principles of pervasive systems and the description of some projects of pervasive systems. The social layer is tackled in the conclusion of this section.

2.1 Principles of Pervasive Systems

> Most important, ubiquitous computers will help overcome the problem of information overload. There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods. *Mark Weiser [61]*

The idea of ubiquitous computing, or pervasive computing, was initiated by Mark Weiser in his famous article "The Computer for the 21st Century" [61] in 1991. His vision of computers fully integrated in the human environment and gracefully providing information and services to users is still an open issue in computer science and computer engineering.

Pervasive computing results from the evolution of the computing paradigm from centralized mainframe computers with "dummy" terminals at an organizational

level to more decentralized networks of personal computers at a user level, and toward the multiplication of "smart" small-scale appliances, e.g. hand-held devices like smart phones or PDAs, or embedded devices integrated in the surrounding environment, like autonomous sensors and actuators. In so-called pervasive information systems [42], those smart objects can benefit from wireless and wired networks to remotely access to powerful computing and large distributed databases, and to be remotely accessed by other smart objects, thus creating what could be called the "Internet of Things" [56].

This integration of "computerized artifacts" blurs the distinction between computers and other electronic devices [43], leading to new application models. From a user point of view, applications can be mobile, localized and personalized: new interaction possibilities can make applications go "off the desktop", i.e. applications can run in the background, using the user environment itself as an ubiquitous interface. From a system point of view, sensors and actuators can be distributed in the environment and autonomously gather data and execute actions with no or few human interactions.

As presented in [10], developing applications in such complex computing environments leads to the need for middlewares. Middlewares offer a unified representation and access to those distributed resources. The following requirements are detailed:

1. abstraction of devices (sensors, actuators, *etc.*);
2. loosely coupled communications, including discovery mechanisms;
3. context management;
4. application developer support.

Abstraction of Devices

Pervasive systems are distributed systems of devices able to communicate with others through network links. Devices may range from isolated sensors to mainframe computers, including smart phones, PDAs, desktop computers, and may be embedded in the environment, mobile, handheld, or stationary. At an abstract level, devices can be viewed as entities providing some of the following functionalities:

- sensor: it can report one or more environment parameters or events;
- actuator: it can modify the environment through its actions;
- computation: it can compute some information given some input data;
- storage: it can store data and answer to queries about it.

Those devices are mainly represented by services distributed in the pervasive network. This abstraction enables interoperability between heterogeneous devices. The service representation tends however to be divided in two categories: reactive services and autonomous components. Reactive services can be invoked and composed by a supervision system in order to create applications [9, 29, 12]. On the other hand, autonomous components [32] can decide themselves to collaborate with some others in order to create coherent processes.

As devices may be sensors or actuators [23], services may represent some interactions with the physical environment, like taking a photo from a camera or displaying a picture on a screen. These interactions bridge the gap between the computing environment and the physical environment, that can both be managed by the pervasive system.

In summary, the set of devices in pervasive systems can be abstracted as an environment of distributed services providing sensor, actuator, computation and storage functionalities, where some services may be autonomous. We call such an environment a *pervasive environment*.

Loosely Coupled Communications

A common representation for data and services is required for services to understand each others. Tuple representations like for databases or standardized languages such as XML are commonly used for data exchange between services. Services are represented by their interface: it provides a list of methods that can be invoked and potentially the types of events that the service may publish.

At a lower level, system functionalities are often accessed through proxies and wrappers that translate commands and data between platform-independent and platform-specific representations.

Service discovery is a common issue [64] in distributed systems (pervasive systems, grids, or even Internet). As services may enter or exit the pervasive environment at any time (e.g. services provided by mobile devices), the discovery should be dynamic in order to reflect the currently available services.

Remote invocation, or more generally communication between services, can not always rely on a stable network infrastructure in pervasive systems. Asynchronous messaging is then preferred to synchronous communications: asynchronous messaging can handle more gracefully network latency and failures in this dynamic setting.

Asynchronous messaging also enables event mechanisms through publish/subscribe systems: a service can subscribe to some events provided by another service, and the expected events are sent asynchronously when they occur.

Context Management

Context management is a key element for dynamic adaptation of applications to their environment. As devices and services are spatially distributed in the environment and may be mobile, a strong need for localization appears in pervasive systems. A spatial indexation of the entities in the environment is necessary to allow location-aware processes.

In a more general way, the notion of context can be defined as “any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups, and computational and physical

objects." [22]. From [22], three layers of components are required to capture the context: *widgets* that acquire low-level information from sensors, *interpreters* abstracting this information and *aggregators* gathering information by entity. Applications can then use these components in order to provide context-aware behaviors.

A common representation for the context is also a requirement to enable interoperability. Whereas simple forms of context can be expressed using key-value pairs (e.g. `[name="carla", location="elysee"]`), more elaborate context models need graph-model representation like RDF (Resource Description Framework) or the more general concept of ontology (e.g. the Context Ontology Language (CoOL) [54]). Ontologies allow independent services to reason about the same concepts with a shared ontology or to agree about concepts with ontology alignments.

Application Developer Support

Distributed functionalities in a pervasive environment may appear or disappear dynamically. In order to make the development of applications easier, applications can be defined using abstract functionalities and dynamically linked to actual implementations at runtime, depending on the available resources. For instance, the OSGi "whiteboard pattern" [49] works as follows: service consumers use a given service interface and, at runtime, can search registered services that implement this interface, and then invoke their methods.

Middlewares like OSGi [48], combined with network protocols like UPnP [59] or DPWS [60], implement an abstraction of pervasive systems by providing a catalog of available services that are dynamically discovered, and by hiding communications details through unified interfaces to access to those services.

Interfaces with users or other software components try to hide the complexity of the pervasive system organization. Explicit interactions (reactivity) are often made with a declarative language using an abstract view of the environment, while implicit interactions (proactivity) provide useful automatically configured services to users depending on their context.

2.2 Projects of Pervasive System

Giant research projects on pervasive systems have been conducted in the greatest universities throughout the world. Among them, we quote the Oxygen project [46] of the MIT, the EasyLiving project [45] at Microsoft Research or the Aura project [13] of the Carnegie Mellon University. The examples they provide concern mainly intelligent workspaces and enhanced spaces (e.g. elderly homes). These projects encompass many research teams of different specialties (from hardware to software, including artificial intelligence, speech recognition and synthesis, multimodal and plastic user interfaces, local area networks, middleware, *etc.*), and have shown ambitious objectives in particular on the user interface.

In the Oxygen project [46, 27, 40], they have defined new devices for the end-user (called H21s) that include computing and communication facilities as well as

multimodal user interfaces. They have designed a dedicated network technology. They have also studied the software level by defining a technique for the adaptation of software to the ever changing pervasive environment using a goal-oriented programming technique [51]. It decomposes applications in two levels: the goal level abstracts the end-user task, and the software components (called pebbles) level contains effective code realization. Pebbles are platform-independent software components, capable of being assembled dynamically by the goals planning mechanism in response to evolving system requirements. A subsystem of the Oxygen environment concerns the management of user knowledge. It is based on a RDF representation and a learning system gathers information on the user habits and preferences. Collaborative tools have also been proposed, like the annotation of web documents.

The EasyLiving project [45, 12] also works on intelligent environments (in-home or in-office). The context sensing and modeling has been highly studied (combination of multiple sensor modalities, automatic sensor calibration) as well as the interaction with the end-user (computer vision and visual user interaction, adaptation of user interfaces...). They have also defined device-independent protocols for communication and data. Like Oxygen, adaptation is based on an abstraction of users' tasks and on the discovery and composition of effective services in the environment. The originality of this project comes from their geometric model of the world. This model represents objects (of the real world or of the software world) as entities, and geometric relationships between entities as measurements. A measurement is a polygon that represents the object physical expanse, and can be associated with uncertainty values. The precise and complete geometric model allows to specify precise situations involving different objects (e.g. an object in a certain area, close to a certain software component).

The Aura project [13, 29] aims at providing each user with an invisible halo of computing and information services that persists regardless of location. They have deployed efforts at every level: from the hardware and network layers, through the operating system and middleware, to the user interface and applications. Their project ambition goes one step further compared to the others, as they want the user not to be restricted to classical devices, but should be able to interact continuously with his "aura" that follows him everywhere and at every time, using any available appliance, even the coffee maker while the user stands in front of it (as the video available on their web site [13] shows it).

Other big projects could be described. One can cite the following ones:

- The Portolano project [58] at the Washington University focuses on sensors management, networking, transparency to the end-user and trust. They have studied an infrastructure based on mobile agents that interact with applications and users. Data-centric routing automatically migrates data among applications on the user's behalf. Data thus becomes "smart" and serves as an interaction mechanism within the environment.
- The Endeavour project [57] at Berkeley University has studied a planet-scale, self-organizing, and adaptive Information Utility. Their main objective is to arbitrarily and automatically distribute data among Information Devices". Data are

seen as software components that can advertise themselves, provide their own adaptable user interface and their own negotiation process for their integration in applications.
- The Sentient project [8] at AT&T Laboratories Cambridge is based on a device called a bat with a unique id, an ultrasound transmitter and a radio transceiver, 2 buttons and a beeper. It is located by a central controller, and the world model stores the correspondence between bats and their owners, applying algorithms to the bat location data to determine the location of the person or object which owns it.

All these projects focus on services and consider the environment as a halo of available services. The notion of data is not present in the front: data are embedded in software components. All information interesting the user or describing his way of working are represented in objects or services; the paradigm for the manipulation of artifacts are services or components. With the advent of the DataSpace notion [25], another vision has appeared, placing data at the centre of the pervasive system. Some projects have tried to focus on a data-like representation of the environment, including databases and data streams but also services. They have resulted in hybrid SQL-like systems that include remote services calls in queries (e.g. [31, 62]). They will be detailed in section 3.

2.3 Summary

In the previous section, we emphasized that pervasive systems need a certain degree of abstraction of devices about hardware, software and network capabilities. Middlewares and layered architectures are mainly used to achieve this level of abstraction, through a common representation of resources as services. Communications between services are often implemented as asynchronous messages that are independent of the underlying platform and network protocol. Context management is also a key element for dynamic adaptation of applications to their environment.

Among the restrictions imposed on pervasive applications by the social layer [41], a strong issue is to enforce security policy. As sharing of data and services among devices is one of the main point of pervasive systems, security is needed to protect access to resources and to ensure some level of user privacy. Despite many works on security for distributed systems, it remains an open issue in such complex environments. Other restrictions may come from usability issues, aesthetics issues and environmental issues, in particular in term of energy consumption.

3 Related Database Research

Current trends in pervasive environment management can be analyzed as the leveraging of database principles, applied to a more dynamic and distributed setting. We first tackle the representation and management of streaming data sources. We then

tackle data integration problems that occur in pervasive environment settings and describe the interplay between data and services.

3.1 Data Streams

Pervasive systems often include services that periodically or occasionally generate data, be it events or sensor readings. Managing such data sources in programs (e.g. in a supervision system) can be complex as it implies asynchronous data handling. In order to cope with this complexity, database principles can be applied: data sources are represented in a way similar to relations in databases, and queries can be formulated in a declarative way using a SQL-like query language from which query optimization techniques can be applied.

Furthermore, data streams and relations may be handled in a homogeneous way so as to enable queries combining both types of data sources. Data streams represent the integration of dynamic data sources in databases, leading to the definition of continuous queries providing dynamic results that are continuously updated. Queries may also still be one-shot as standard SQL queries, i.e. their results are evaluated once and not updated.

Many projects have been launched on data streams, among which we quote NiagaraCQ, TelegraphCQ, Cougar, TinyDB, STREAM, the Global Sensor Network and Cayuga.

NiagaraCQ [17] introduces some definitions of continuous queries over XML data streams. Queries are defined as triggers and produced event notification in real-time. The TelegraphCQ system [14] proposes adaptive continuous query processing over streaming data and historical data.

Cougar [63, 11] and TinyDB [30] handle continuous queries over sensor networks with a focus on the optimization of energy consumption for sensors using in-network query processing. STREAM [7] defines a homogeneous framework for continuous queries over relations and data streams.

In those systems, continuous queries are defines using a SQL-like language. Another approach is tackled with Borealis [37, 3, 18]: a Distributed Stream Processing System (DSPS) enables to define dataflow graphs of operators in a "box & arrows" fashion, making distributed query processing easier.

Continuous queries can be used to define some parts of pervasive applications in a declarative way: in [26, 39], the progressive cleaning process for data retrieved from numerous physical sensors is defined by a pipeline of continuous queries declaratively defined in SQL. A complex event-processing using state-machine operator producing data streams is also proposed. In [4, 5], the Global Sensor Network, a middleware for sensor networks, enables to specify continuous queries as virtual sensors whose processing is specified declaratively in SQL, with a subquery for preprocessing each input stream.

Cayuga [20, 19] is a stateful publish/subscribe system for complex event monitoring where events are also defined by SQL-like continuous queries over data streams.

3.2 Data and Services Integration

In this section, we discuss the interplay between data and services, and possible optimizations for queries involving both types of data sources.

Data integration has been a long standing theme of research over the past 30 years. Now, the broader notion of dataspaces [25, 38] has appeared to provide base functionality over all data sources and applications, regardless of how integrated they are and without having a full control over the underlying data [25]. For example, to answer a query when some data sources are unavailable, the data accessible at the time of the query have to be used to propose the best possible results.

In the setting of data integration, the notion of *binding patterns* appears to be quite interesting since they allow to model a restricted access pattern to a relational data source as a specification of "which attributes of a relation must be given values when accessing a set of tuples" [24]. A relation with binding patterns can represent an external data source with limited access patterns in the context of data integration [24]. It can also represent an interface to an infinite data source like a web site search engine [31], providing a list of URLs corresponding to some given keywords. In a more general way, it can represent a data service, e.g. web services providing data sets, as a virtual relational table like in [53].

The SQL standard itself supports some forms of access to external functionalities through User-Defined Functions (UDF). UDFs can be scalar functions (returning a single value) or table functions (returning a relation). UDFs are defined in SQL or in another programming language (e.g. C, Java), enabling to access to any external resources. Table functions are a way to implement the notion of virtual tables, however limited to having only one binding pattern determined by the function input parameters. UDFs are also tagged as deterministic or non-deterministic: query rewriting may not change the number of invocations for non-deterministic UDFs. Abstract Data Types can also be used to get an object-oriented view of sensors, like in the Cougar project [11, 63].

Optimization of queries involving expensive functions or methods leads to the redefinition of cost models to integrate the estimated cost of computation. This issue has been studied for standard databases [15, 16, 35, 36], and also for continuous query processing [21].

In a similar way to binding patterns, the ActiveXML language [1] allows to define XML documents containing extensional data, i.e. data that are present in the document, and intensional data, representing service calls that provide data when needed. Intensional data is close to the notion of virtual tables and binding patterns. ActiveXML is also a "framework for distributed XML data management" [6] and defines an algebra to model operations over ActiveXML documents distributed among peers, that enables query optimization.

In Aorta [62], continuous queries can implicitly interact with devices through an external function call. However, the relationship between functions and devices, as well as the optimization criteria, are not explicit and cannot be declaratively defined.

In [5], the Global Sensor Network allows to define virtual sensors abstracting implementation details of data sources, and provides continuous query processing facilities over distributed data streams.

3.3 Summary

Database research that can be related to pervasive environments span across several issues. From a data-centric point of view, the management of pervasive environments is the management of distributed dynamic data sources and services that should be accessed through declarative queries: therefore, there is a need for integration of data streams, external methods and services, into relational or XML databases. Continuous or one-shot queries over such extended databases need to be declaratively defined, for example using a SQL-like language, optimized for this new setting, and processed (in real-time for continuous queries).

4 Enabling Technologies

A pervasive environment is full of functionalities, but a user may be lost and not able to comprehend and optimally use all available data sources and services the environment can provide. Furthermore, applications are not easy to develop and maintain because of the heterogeneity and the dynamicity of the environment. Typically, low-level technical code using programming languages (Java, C#...) and network protocols has to be devised to come up with some pervasive applications.

In this section, we detail some technologies that enable to build pervasive environment systems. Those technologies tackle system problems like service discovery and remote invocation in a heterogeneous setting, but also some higher-level issues like a common data and service representation.

CORBA [47] (Common Object Request Broker Architecture) is an open architecture and infrastructure that enables applications to interoperate over network links. It can be defined as an object bus: applications can access to local or remote objects without worrying about underlying network issues (including serialization issues). A lookup allows to search objects by name and get object references. Objects are defined using the platform-independent IDL (Interface Description Language) that can be used to generate stub and/or skeleton in many programming languages.

Some systems tackle the same issues, but are more platform- or language-dependent, like Microsoft DCOM (Distributed Component Object Model) or Java RMI (Remote Method Invocation).

Whereas those systems are a sort of object bus, other systems focus on a messaging protocol between services to achieve interoperability. Those protocol are more data-oriented. The open standard XML (eXtended Markup Language) is often used as the message format for such protocols, like for the simple yet efficient XML-RPC (XML - Remote Procedure Call) or its more complex but powerful evolution SOAP (Simple Object Access Protocol) for Web Services. REST (REpresentational State

Transfer) relies on the HTTP API to transfer messages, but do not define a message format: it is rather an architecture style using the well-established HTTP protocol to simplify communications. For those protocols, service discovery needs to be done by external registries, like UDDI (Universal Description, Discovery and Integration) for Web Services.

UPnP [59] (Universal Plug and Play) and the more recent DPWS [60] (Device Profile for Web Services) are based on some messaging protocols and include automatic discovery mechanisms, using network broadcast facilities. UPnP/DPWS entities are devices that host several services providing methods and events.

JMX (Java Management eXtension) and OSGi [48] are two Java framework that can host some (potentially active) java objects as services and enable a local and remote access to them through various network protocols like RMI, Web Services, or even UPnP and DPWS if dynamic discovery is needed.

Nowadays, these relatively recent technologies have become mature, some of them being used in the industry (in particular for application servers). Focused on interoperability issues in a heterogeneous setting, they can be re-used in the context of pervasive environment management systems.

5 SoCQ: A Comprehensive PEMS

Managing pervasive environments, in particular heterogeneous devices, remains a complex task: a certain level of abstraction and loosely coupled communications can be achieved with current middlewares, but application developer support still can not hide resource heterogeneity. However, with the adoption of a data-centered point of view, this heterogeneity can be further abstracted: devices can be represented as distributed data sources providing data, data streams and services that are manageable in a homogeneous way.

The **S**ervice-**o**riented **C**ontinuous **Q**uery project, or SoCQ project [33, 52], is devoted to making the development of pervasive applications easier through database principles. It aims at contributing in the area of Dataspaces [25, 38] through a unified view of data and service spaces mandatory in pervasive environments.

We are currently working on the definition of an approach to homogeneously represent such pervasive environments through database principles. The basic idea is to present to application developers a database-like view of the environment resources, so that they can visualize this environment as a set of tables and launch declaratively-defined continuous queries involving available data sources and services. This approach is built on an extension of the relational model and uses a SQL-like query language.

DBMSs (DataBase Management Systems) provide a homogeneous view as well as storage and query facilities for relational data. DSMSs (Data Stream Management Systems) also provide a homogeneous view and query facilities for both relational data and data streams. We then call PEMS, for Pervasive Environment Management System, a system that manages in a similar way an environment containing data relations, data streams and services.

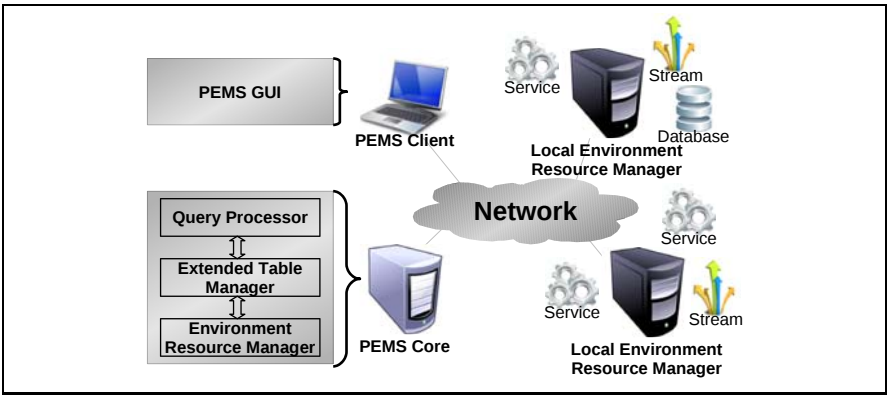


Fig. 1 Overview of a PEMS environment

In this project, we tackle the following challenges:

- definition of a homogeneous representation for databases, data streams and services from the pervasive environment,
- definition of a query language over pervasive environments allowing to easily develop pervasive applications,
- design of a Pervasive Environment Management System (PEMS) supporting both the homogeneous representation and the query processing facilities.

In Figure 1, the different elements of a PEMS are shown. A distributed resource manager handles service discovery and remote invocations, with local resource managers as proxies for local devices that provide data, streams and services. An extended table manager builds a homogeneous representation of non-conventional data sources, and the query processor allows to define, optimize and execute queries.

5.1 Example Scenario

In the example scenario, we monitor temperatures in an office building: when a temperature exceeds some threshold in a room, an alert message is sent to the manager of this room. A photo of the room can be joined to the message. We simulate an environment, illustrated in Figure 2, containing the following data sources and services:

- two data relations: one containing some information about the rooms (manager, temperature threshold...), the other one being a list of contacts (including contacts of the managers),
- some temperature sensors distributed in several rooms, providing data streams,
- some cameras installed in the rooms, providing photo services,
- some messenger services (by mail, instant message, SMS).

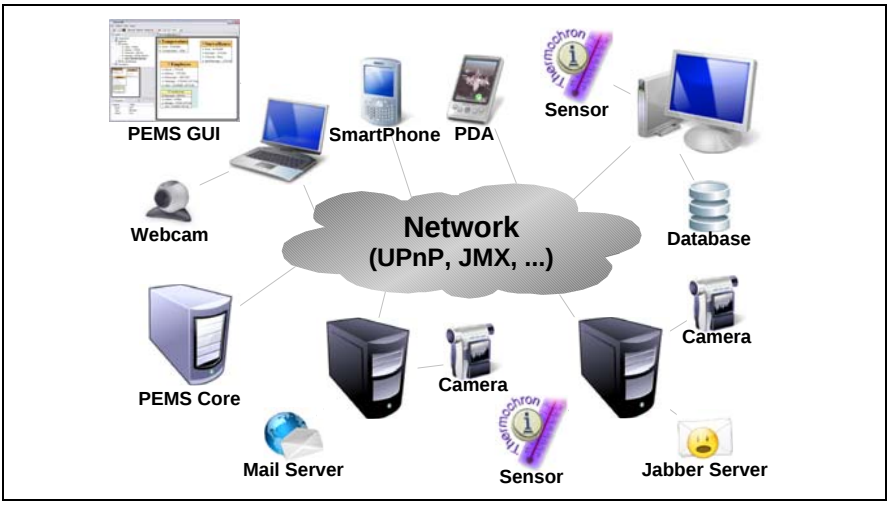


Fig. 2 Illustration of the scenario environment

This environment can be represented homogeneously with relations and streams extended with virtual attributes and binding patterns. Virtual attributes are attributes that do not have a value and may be provided a value through a query, due to binding patterns that indicate their relationship with method prototypes from services. We call such relations *XD-Relations*, standing for eXtended Dynamic Relations, and such environments *relational pervasive environments*. For the example scenario, we can view a DDL representation of the schema of this environment in Table 1.

With such environments, the use of distributed functionalities provided by services is declaratively specified in SQL-like queries by the virtual attributes that need to be realized, i.e. that need to be provided a value. In order to realize those attributes, the corresponding binding patterns are invoked for every involved tuples, leading to several service invocations. We call these queries SoCQ, for Service-oriented Continuous Queries.

Over this environment, many different SoCQ queries could be launched. For example, the temperature monitoring can be declaratively defined as a continuous query. The three XD-Relations are joined (the stream "temperature" must be windowed) on the manager name and the area, the threshold is checked and the message body is set. The binding pattern "sendMessage" will be invoked in order to fetch a value for the virtual attribute "sent". The SQL-like query in Table 2 is a typical example of a pervasive application that is defined at the declarative level, without worrying about low-level technical considerations (programming languages, network protocols).

The role of a PEMS is to manage a relational pervasive environment, with its dynamic data sources and set of services, along with the execution of the continuous queries over this environment. In the following sections, we first sketch the data model that supports XD-Relations, and the algebra that enables SoCQ queries. We then give an overview of our PEMS implementation.

Table 1 DDL description of prototypes and XD-Relations for the environment of the example scenario

```
PROTOTYPE sendMessage( address STRING, text STRING ) :
  (sent BOOLEAN) ACTIVE;

PROTOTYPE takePhoto( ) :
  ( photo BLOB );

RELATION surveillance (
  area         STRING,
  manager      STRING,
  threshold    REAL,
  alertMessage STRING
);

RELATION employees (
  name      STRING,
  address   STRING,
  messenger SERVICE,
  text      STRING   VIRTUAL,
  sent      BOOLEAN  VIRTUAL
)
USING BINDING PATTERNS (
  sendMessage[messenger] ( address, text ) : ( sent )
);

RELATION cameras (
  camera  SERVICE,
  area    STRING,
  photo   BINARY  VIRTUAL
)
USING BINDING PATTERNS (
  takePhoto[camera] ( ) : ( photo )
);

STREAM temperatures (
   area        STRING,
   temperature REAL
);
```

Table 2 SoCQ query for the example scenario

```
SELECT surveillance.area, surveillance.manager, employees.sent
FROM   temperatures [now], employees, surveillance
WHERE  surveillance.manager = employees.name
  AND  surveillance.area = temperatures.area
  AND  surveillance.threshold < temperatures.temperature
  AND  employees.text IS surveillance.alertMessage
```

5.2 Modeling of Pervasive Environments

In order to homogeneously represent data sources and other resources from pervasive environments, we propose a model that integrate distributed functionalities of

resources within data sources. Our model, based on the relational model, is built on the following notions: prototypes, services and extended relations with virtual attributes and binding patterns.

Distributed functionalities can be represented as services implementing prototypes. For example, a webcam and an IP camera are two services from the environment that implement a prototype `takePhoto():(photo)` that takes zero input attribute and provides one output attribute `photo`; a mail server, an instant messaging server and a SMS gateway are three services that implement a prototype `sendMessage(text,address):(sent)` that takes two input attributes `text` and `address` and provides one output attribute `sent`. Invoking a prototype on a service realizes the implied actions, like taking a photo for a camera and sending a message to the given address for the mail server.

As service invocations can have an impact on the physical environment, e.g. invoking a service that sends a message, we need to consider two categories of prototypes: *active prototypes* and *passive prototypes*. Active prototypes are prototypes having a side effect on the physical environment that can not be neglected (e.g. in Table 1, `sendMessage` is tagged as active). On the opposite, the impact of passive prototypes is non-existent or can be neglected, like reading sensor data (e.g. `takePhoto`).

Prototypes can be integrated into data relations schemas through virtual attributes and binding patterns. Virtual attributes are attributes from the relation schema that do not have a value at the tuple level. They represent input and output attributes of prototypes. A binding pattern is associated with a relation schema and specifies one non-virtual attribute as the service reference attribute, the prototype and which attributes are linked with the prototype input and output attributes. For example, the `employees` relation (see Table 1) is associated with one binding pattern that uses the prototype `sendMessage`, the service reference attribute `messenger` and that links the attributes `address` and `text` with the prototype input attributes, and the attribute `sent` with the prototype output attribute. Output attribute should be virtual attributes, whereas input attributes can also be real (i.e. , non-virtual) attributes, like the attribute `address` in this example.

We call such relations, X-Relations, standing for eXtended Relations. Virtual attributes represent possible interactions with services: when a query needs the virtual attribute `sent`, a value is required for the virtual attribute `text` due to the binding pattern (the attribute `address` being real), and it implies an invocation of the prototype `sendMessage`. The required value should be provided by the query itself. The services on which the prototype is invoked are defined by the value of the service reference attribute (here, attribute `messenger`), at the tuple level.

In the following table, an example of content for the X-Relation `employees` is presented. The constants “mailer” and “jabber” are two service references, the former for the mail server, the latter for the instant messaging server. The star (*) symbol reminds that virtual attributes do not have a value.

name	address	messenger	text	sent
nicolas	nicolas@elysee.fr	mailer	*	*
carla	carla@elysee.fr	mailer	*	*
françois	francois@im.gouv.fr	jabber	*	*

Pervasive environments being dynamic, data sources may include streaming data. We extend our model to integrate data sources like data streams. We call XD-Relations, for eXtended Dynamic Relations, X-Relations that are time-dependent: XD-Relations can be either finite (relations where tuples can be inserted and deleted) or infinite (append-only relations, i.e. data streams). An environment represented by a set of XD-Relations is defined as a relational pervasive environment.

5.3 Service-Oriented Continuous Queries

Queries over relational pervasive environments allow to define interactions between dynamic data sources and services, i.e. pervasive applications. Such queries are defined to be continuous queries, i.e. queries that are executed continuously to maintain their results up-to-date, like in the example scenario. They are called Service-oriented Continuous Queries, or SoCQ queries. However, some queries may be snapshot queries, i.e. queries executed once that produce their results and do not maintain them, like standard SQL queries in DBMS.

SoCQ queries are based on the so-called Serena algebra (**Ser**vice-**ena**bled algebra) that defines query operators over XD-Relations. Standard relational operators are redefined over finite XD-Relations, and new operators are defined. Realization operators handle the transformation of virtual attributes either by providing them a value (a constant or the value of another attribute) or by invoking a binding pattern. Window operators and streaming operators handle infinite XD-Relations: window operators transform an infinite XD-Relations into a finite XD-Relations (e.g. a relation that contains the tuples inserted during the last 5 minutes into the stream operand), and streaming operators transform finite XD-Relations into infinite XD-Relations (e.g. a stream of the tuple inserted into the relation operand).

A SQL-like query language has been defined to declaratively express SoCQ queries. For example, for the example scenario, the query in Table 2 involves several operators: windows (the `[now]` is a window of size 1 applied on the stream `temperatures`), selections, joins, realizations, streaming. This query produces a stream of alerts (when a threshold is exceeded) while invoking the `sendMessage` prototype when needed (to actually send messages to area managers).

5.4 Implementation of PEMS

The PEMS core is composed of three logical layers (see Figure 1). A global resource manager handles service discovery and remote invocations, with local resource managers as distributed proxies for local devices that provide services. An

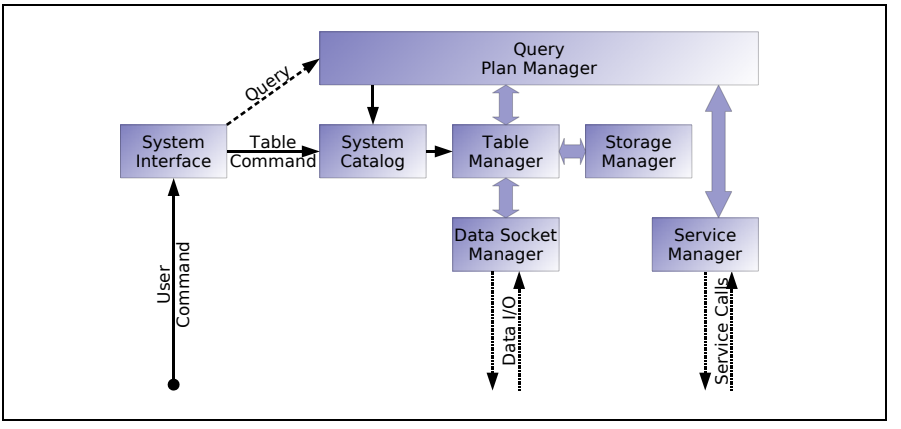


Fig. 3 Internal modules of the PEMS core

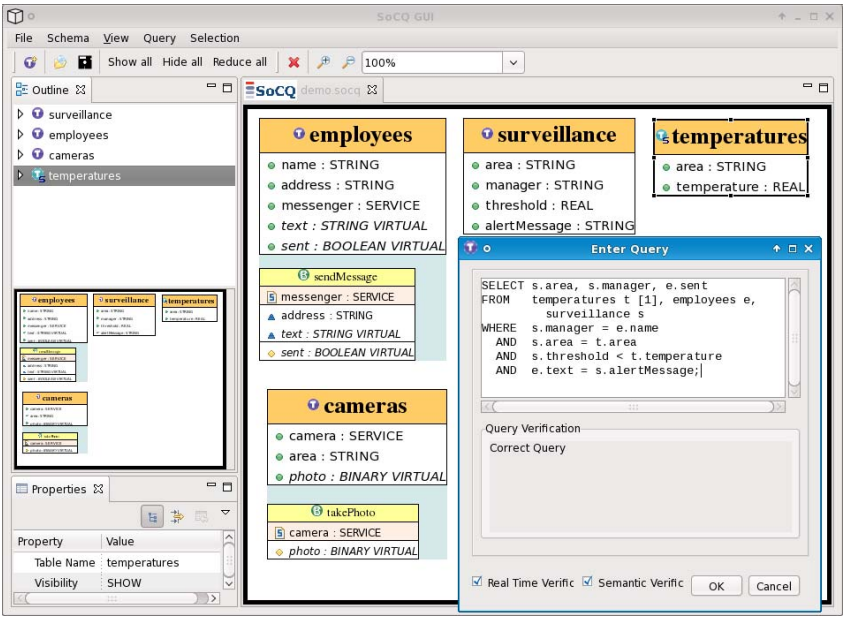


Fig. 4 The PEMS GUI

extended table manager builds a homogeneous representation of non-conventional data sources, and the query processor allows to define, optimize and execute Service-oriented Continuous Queries. These layers are composed of several internal modules sketched in Figure 3.

The PEMS prototype is developed in the Java/OSGi framework [48]. Each module of the PEMS is an OSGi bundle and communicates with each other through the OSGi service life cycle management. The chosen network protocol for service discovery and remote invocations is UPnP [59]: the prototype uses the dedicated standard OSGi bundles for this protocol.

The PEMS GUI, shown in Figure 4, is also developed in the Java/OSGi framework, as an Eclipse RCP Plugin, i.e. the GUI is integrated in the Eclipse platform. It communicates remotely with the PEMS core through a JMX interface. It enables to visualize existing XD-Relations and their content, to add/alter/delete XD-Relations, and to launch/stop SoCQ queries.

6 Conclusion

Pervasive systems intend to take advantage of the evolving user environment so as to provide applications adapted to the environment resources. As far as we know, bridging the gap between data management and pervasive applications has not been fully addressed yet. A clear understanding of the interplays between databases, data streams and services is still lacking and is a major bottleneck toward the declarative definition of pervasive applications.

Pervasive environments are complex environments that raise issues in several research domains. Studying pervasive computing from a data-centric point of view raises some similarity with current database research like data streams, data and services integration, or distributed databases.

In this setting, the SoCQ project is our attempt to bridge the gap between pervasive computing and the database domain. It demonstrates the following points: 1) a homogeneous database-like view on pervasive environments containing dynamic data sources and services is possible as a set of XD-Relations, through the notions of virtual attributes and binding patterns; 2) Service-oriented Continuous Queries (SoCQ queries) over relational pervasive environment allow to define pervasive applications combining data sources and services. A formal algebra has been devised allowing to apply query optimization techniques in pervasive environments. Such declarative definitions of SoCQ queries make the definition and the evolution of pervasive applications easier.

References

1. ActiveXML, http://www.activexml.net/
2. Ioannidis, Y., Scholl, M.H., Schmidt, J.W., Matthes, F., Hatzopoulos, M., Böhm, K., Kemper, A., Grust, T., Böhm, C. (eds.): EDBT 2006. LNCS, vol. 3896. Springer, Heidelberg (2006)
3. Abadi, D.J., et al.: The Design of the Borealis Stream Processing Engine. In: CIDR 2005, Proceedings of Second Biennial Conference on Innovative Data Systems Research (2005)

4. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment. In: VLDB 2006, Proceedings of the 32nd International Conference on Very Large Data Bases (2006)
5. Aberer, K., Hauswirth, M., Salehi, A.: Infrastructure for data processing in large-scale interconnected sensor networks. In: MDM 2007, Proceedings of the 8th International Conference on Mobile Data Management (2007)
6. Abiteboul, S., Manolescu, I., Taropa, E.: A framework for distributed xml data management. In: EDBT [2], pp. 1049–1058
7. Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Motwani, R., Nishizawa, I., Srivastava, U., Thomas, D., Varma, R., Widom, J.: STREAM: The Stanford Stream Data Manager. IEEE Data Engineering Bulletin 26(1), 19–26 (2003)
8. ATT Laboratories, Cambridge: Sentient Computing Project, http://www.cl.cam.ac.uk/research/dtg/attarchive/spirit/
9. Becker, C., Handte, M., Schiele, G., Rothermel, K.: PCOM – A Component System for Pervasive Computing. In: PerCom 2004, Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications, p. 67 (2004)
10. Biegel, G., Cahill, V.: Requirements for middleware for pervasive information systems. Kourouthanassis and Giaglis [42], vol. 10, pp. 86–102 (2007)
11. Bonnet, P., Gehrke, J., Seshadri, P.: Towards sensor database systems. In: Tan, K.-L., Franklin, M.J., Lui, J.C.-S. (eds.) MDM 2001. LNCS, vol. 1987, pp. 3–14. Springer, Heidelberg (2000)
12. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S.: EasyLiving: Technologies for intelligent environments. In: Thomas, P., Gellersen, H.-W. (eds.) HUC 2000. LNCS, vol. 1927, pp. 12–29. Springer, Heidelberg (2000)
13. Carnegie Mellon University: Project Aura, Distraction-free Ubiquitous Computing, http://www.cs.cmu.edu/~aura/
14. Chandrasekaran, S., et al.: TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In: CIDR 2003, Proceedings of the First Biennial Conference on Innovative Data Systems Research (2003)
15. Chaudhuri, S., Shim, K.: Query optimization in the presence of foreign functions. In: VLDB 1993: Proceedings of the 19th International Conference on Very Large Data Bases, pp. 529–542. Morgan Kaufmann Publishers Inc., San Francisco (1993)
16. Chaudhuri, S., Shim, K.: Optimization of queries with user-defined predicates. ACM Trans. Database Syst. 24(2), 177–228 (1999), http://doi.acm.org/10.1145/320248.320249
17. Chen, J., DeWitt, D.J., Tian, F., Wang, Y.: NiagaraCQ: A Scalable Continuous Query System for Internet Databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 379–390 (2000)
18. Cherniack, M., et al.: Scalable Distributed Stream Processing. In: CIDR 2003, Proceedings of the First Biennial Conference on Innovative Data Systems Research (2003)
19. Demers, A.J., Gehrke, J., Hong, M., Riedewald, M., White, W.M.: Towards expressive publish/subscribe systems. In: EDBT [2], pp. 627–644
20. Demers, A.J., Gehrke, J., Panda, B., Riedewald, M., Sharma, V., White, W.M.: Cayuga: A general purpose event monitoring system. In: CIDR, pp. 412–422 (2007), www.crdrdb.org
21. Denny, M., Franklin, M.J.: Operators for expensive functions in continuous queries. In: ICDE 2006: Proceedings of the 22nd International Conference on Data Engineering, p. 147. IEEE Computer Society, Washington (2006)
22. Dey, A.K., Abowd, G.D., Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction 16(2), 97–166 (2001)

23. Estrin, D., Culler, D., Pister, K., Sukhatme, G.: Connecting the Physical World with Pervasive Networks. IEEE Pervasive Computing 1(1), 59–69 (2002)
24. Florescu, D., Levy, A., Manolescu, I., Suciu, D.: Query Optimization in the Presence of Limited Access Patterns. In: SIGMOD 1999: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pp. 311–322 (1999), http://doi.acm.org/10.1145/304182.304210
25. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: a new Abstraction for Information Management. SIGMOD Rec. 34(4), 27–33 (2005)
26. Franklin, M.J., et al.: Design Considerations for High Fan-In Systems: The HiFi Approach. In: CIDR 2005, Proceedings of Second Biennial Conference on Innovative Data Systems Research (2005)
27. Gajos, K., Fox, H., Shrobe, H.: End user empowerment in human centered pervasive computing. In: Pervasive 2002, Zurich, Switzerland (2002)
28. Garcia-Molina, H., Widom, J., Ullman, J.D.: Database System Implementation. Prentice-Hall, Inc., Upper Saddle River (1999)
29. Garlan, D., et al.: Project Aura: Toward Distraction-Free Pervasive Computing. IEEE Pervasive Computing 1(2), 22–31 (2002)
30. Gehrke, J., Madden, S.: Query processing in sensor networks. IEEE Pervasive Computing 3(1), 46–55 (2004)
31. Goldman, R., Widom, J.: WSQ/DSQ: A Practical Approach for Combined Querying of Databases and the Web. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 285–296 (2000)
32. Grimm, R., et al.: System Support for Pervasive Applications. ACM Transactions on Computer Systems 22(4), 421–486 (2004)
33. Gripay, Y.: Service-oriented Continuous Queries for Pervasive Systems. In: EDBT 2008 PhD Workshop (2008), http://liris.cnrs.fr/publis/?id=3428
34. Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., Jansen, E.: The gator tech smart house: A programmable pervasive space. Computer 38(3), 50–60 (2005)
35. Hellerstein, J.M.: Optimization techniques for queries with expensive methods. ACM Transactions on Database Systems 23(2), 113–157 (1998), http://doi.acm.org/10.1145/292481.277627
36. Hellerstein, J.M., Stonebraker, M.: Predicate migration: Optimizing queries with expensive predicates. In: SIGMOD 1993, Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 267–276 (1993)
37. Hwang, J.H., Xing, Y., Cetintemel, U., Zdonik, S.: A cooperative, self-configuring high-availability solution for stream processing. In: ICDE 2007, Proceedings of the 23rd International Conference on Data Engineering (2007)
38. Imielinski, T., Nath, B.: Wireless graffiti: data, data everywhere. In: VLDB 2002, pp. 9–19 (2002)
39. Jeffery, S.R., Alonso, G., Franklin, M.J., Hong, W., Widom, J.: Declarative support for sensor data cleaning. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) PERVASIVE 2006. LNCS, vol. 3968, pp. 83–100. Springer, Heidelberg (2006)
40. Koile, K., Tollmar, K., Demirdjian, D., Shrobe, H., Darrell, T.: Activity zones for context-aware computing. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 90–106. Springer, Heidelberg (2003)
41. Kourouthanassis, P.E., Giaglis, G.M.: The design challenge for pervasive information systems. Advances in Management Information Systems [42], vol. 10, pp. 29–85 (2007)
42. Kourouthanassis, P.E., Giaglis, G.M. (eds.): Pervasive Information Systems. Advances in Management Information Systems, vol. 10. M.E. Sharpe, Armonk (2007)

43. Kourouthanassis, P.E., Giaglis, G.M.: Toward pervasiveness. In: Advances in Management Information Systems [42], vol. 10, pp. 3–25 (2007)
44. Levene, M., Loizou, G.: A Guided Tour of Relational Databases and Beyond. Springer, Heidelberg (1999)
45. Microsoft Research: EasyLiving, http://research.microsoft.com/easyliving/
46. MIT: Oxygen Project, Pervasive, Human-centered Computing, http://oxygen.csail.mit.edu/
47. OMG: CORBA, http://www.corba.org/
48. OSGi Alliance: http://www.osgi.org/
49. OSGi Alliance: Listeners Considered Harmful: The "Whiteboard" Pattern. Technical Whitepaper (2004), http://www.osgi.org/wiki/uploads/Links/whiteboard.pdf
50. Saha, D., Mukherjee, A.: Pervasive computing: a paradigm for the 21st century. Computer 36(3), 25–31 (2003)
51. Saif, U., Pham, H., Paluska, J.M., Waterman, J., Terman, C., Ward, S.: A case for goal-oriented programming semantics. In: UbiSys 2003: Workshop on System Support for Ubiquitous Computing, 5th International Conference on Ubiquitous Computing, UbiComp 2003 (2003)
52. SoCQ Project: http://socq.liris.cnrs.fr/
53. Srivastava, U., Munagala, K., Widom, J., Motwani, R.: Query Optimization over Web Services. In: VLDB 2006, Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 355–366 (2006)
54. Strang, T., Linnhoff-popien, C.: Service interoperability on context level in ubiquitous computing environments. In: SSGRR 2003w, Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (2003)
55. Tian, F., DeWitt, D.J.: Tuple Routing Strategies for Distributed Eddies. In: VLDB 2003, Proceedings of the 29th International Conference on Very Large Data Bases, pp. 333–344 (2003)
56. Union, I.T.: The Internet of Things. ITU Internet Reports. International Telecommunication Union (2005)
57. University of California, Berkeley: The Endeavour Expedition: Charting the Fluid Information Utility, http://endeavour.cs.berkeley.edu/
58. University of Washington: Portolano: An Expedition into Invisible Computing, http://portolano.cs.washington.edu/
59. UPnP Forum: Universal Plug and Play, http://www.upnp.org/
60. Web Services for Devices (WS4D): Devices Profile for Web Services (DPWS), http://ws4d.org/
61. Weiser, M.: The Computer for the 21st Century. Scientific American 265(3), 94–104 (1991)
62. Xue, W., Luo, Q.: Action-Oriented Query Processing for Pervasive Computing. In: CIDR 2005, Proceedings of the Second Biennial Conference on Innovative Data Systems Research (2005)
63. Yao, Y., Gehrke, J.: Query Processing in Sensor Networks. In: CIDR 2003, Proceedings of the First Biennial Conference on Innovative Data Systems Research (2003)
64. Zhu, F., Mutka, M., Ni, L.: Service Discovery in Pervasive Computing Environments. IEEE Pervasive Computing 4(4), 81–90 (2005)

Toward a Novel Design of Swarm Robots Based on the Dynamic Bayesian Network

Einoshin Suzuki, Hiroshi Hirai, and Shigeru Takano

Abstract. In this chapter, we describe a novel design method of swarm robots based on the dynamic Bayesian network. Recently, an increasing attention has been paid to swarm robots due to their scalability, flexibility, cost-performance, and robustness. Designing swarm robots so that they exhibit intended collective behaviors is considered as the most challenging issue and so far ad-hoc methods which heavily rely on extensive experiments are common. Such a method typically faces a huge amount of data and handles them possibly using machine learning methods such as clustering. We argue, however, that a more principled use of data with a probabilistic model is expected to lead to a reduced number of experiments in the design and propose the fundamental part of the approach. A simple but a real example using two swarm robots is described as an application.

1 Introduction

Swarm robots are autonomous agents each of whom interacts with their environment locally based on a relatively simple program but as a system they exhibit complex collective behaviors. Recently, swarm intelligence [2] has emerged as a scientific research field for studying such systems from various viewpoints including computer science, engineering, and biology. The main advantages of a system composed of swarm robots are scalability, flexibility, cost-performance, and robustness.

Designing swarm robots, however, is prone to several difficulties which stem from the limited capabilities of individual robots and the nature of the system. A swarm

Einoshin Suzuki
Kyushu University, Fukuoka 819-0395, Japan
e-mail: suzuki@i.kyushu-u.ac.jp

Hirai Hiroshi
Kyushu University, Fukuoka 819-0395, Japan
e-mail: sc105048@s.kyushu-u.ac.jp

Shigeru Takano
Kyushu University, Fukuoka 819-0395, Japan
e-mail: takano@i.kyushu-u.ac.jp

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 299–310.
springerlink.com

robot must cope with uncertainties in its perception and its action with its relatively simple program, its limited hardware (e.g. motors, wheels, sensors, actuators, onboard computer), and its limited information (e.g. signals from sensors, images, communication). Designing individual robots so that they exhibit intended behaviors as a swarm is recognized as one of the most important research issues. Currently, most solutions may be classified as ad-hoc methods which heavily rely on simulation and/or real experiments. Such a method typically faces a huge amount of data and handles them possibly using machine learning methods such as clustering. Clearly a systematic approach based on a solid theoretical foundation with essential data and less experiments is desirable.

Recently, statistical machine learning [1] has been successfully used in designing autonomous vehicles [5]. Especially, autonomous vehicles which succeeded in running a 131-mile course which is most along narrow, unpaved desert trails in a competition called the DARPA Grand Challenge have validated the usefulness of statistical machine learning in designing a complex autonomous agent [5, 11]. Another notable example is an application of the dynamic Bayesian network to planning problems of a manipulator robot [12]. The approach returns the joint probability distribution of all possible trajectories based on a probabilistic model and thus provides a novel method for designing an individual robot.

In this chapter, we explain our ongoing work toward a novel design of swarm robots based on the dynamic Bayesian network. It makes use of data in a more principled way with a probabilistic model and the number of experiments in the design is expected to be highly reduced. In section 2, we explain issues in designing swarm robots. Section 3 briefly introduces the dynamic Bayesian network and we propose our novel approach for designing swarm robots in section 4. Section 5 shows an example of two real swarm robots which are designed to patrol around with a small probability of collision with the principled method. Section 6 gives concluding remarks.

2 Designing Swarm Robots

2.1 Swarm Robots

Swarm robotics is a new approach to the coordination of multi-robot systems [9]. It adopts a decentralized approach in which the intended collective behaviors emerge from the local interaction between robots and their environment. The main advantage lies in its robustness and applications are found in any task in which distributed robots need to explore, survey, collect, harvest, rescue, or assemble into structures.

Interesting applications including a path formation of swarm robots [7] have been presented. In the work, the task of the swarm robots is to form a path from a home location to objects. The robot is equipped with two external wheels, a camera, a semispherical mirror for the camera, and several LEDs in one of three colors used for communication. The basic data is collected in experiments using

the real robots but the experiments for the path formation is done by computer simulation. The system is shown to be scalable, robust, and fault-tolerant. Though we admire the work, we are concerned about its rather artistic nature for designing swarm robots to produce the desired collective behaviors. Especially the number of simulations is considered to be huge otherwise a sufficient level of confidence is not guaranteed.

Studying complex collective behaviors of social insects gives useful hints in designing swarm robots [3]. One of the most important keywords here may be stigmergy: simple interactions of social insects with its environment produce complex collective behaviors such as a nest construction by wasps. Other issues such as a categorization of collective behaviors of social insects, modulation of self-organized behaviors, and their management of uncertainty and complexity are considered as precious sources of ideas for designing swarm robots. We, however, consider that a systematic method for the design is necessary to use the ideas in practice.

2.2 Difficulties in the Design

A hardware of a swarm robot is relatively simple due to its size and its design principle. A swarm robot obtains limited information since the quality of the signals from sensors and communication is relatively low. It has a limited capability to process information due to its simple program and its on-board computer. Its actions are highly uncertain as its actuators such as motors and wheels are of low quality. Therefore, a typical swarm robot has to cope with uncertainties in its perception and its action under an unfavorable condition in terms of computing.

The design of a complex system which consists of many elements is more difficult than that of a simple system with one element. The case of swarm robots belongs to the former and it is generally agreed that a large number of experiments are necessary. Such experiments produce a huge amount of data and so far they are processed in an ad-hoc manner. Clearly a systematic approach based on a solid theoretical foundation with essential data and less experiments is desirable.

3 Dynamic Bayesian Network

3.1 Expressing Probabilistic Knowledge

We assume that the state of a system at time t is described with a set $\mathbf{S}_t$ of probabilistic variables which cannot be observed i.e., state variables that are latent. We denote the set of probabilistic variables which can be observed, i.e., evidence variable, with $\mathbf{E}_t$. Following a common assumption [10], we assume that the state variables begin at $t = 0$ i.e., $\mathbf{S}_0, \mathbf{S}_1, \ldots$ while the evidence variables begin at $t = 1$ i.e., $\mathbf{E}_1, \mathbf{E}_2, \ldots$. A specific assignment $\mathbf{S}_t = \mathbf{s}_t$ of values $\mathbf{s}_t$ to $\mathbf{S}_t$ may be denoted with $\mathbf{s}_t$. Likewise a specific assignment $\mathbf{E}_t = \mathbf{e}_t$ of values $\mathbf{e}_t$ to $\mathbf{E}_t$ may be denoted with $\mathbf{e}_t$.

A Bayesian network $(\mathbf{V},\mathbf{L})$ is a directed graphical representation of conditional dependencies, where $\mathbf{V}$ and $\mathbf{L}$ are sets of nodes and links (i.e., edges), respectively. A node of a Bayesian network represents a probabilistic variable Z with a conditional table which describes the conditional probability $P(Z = z|\Pi(Z,\mathbf{V},\mathbf{L}))$ of Z taking an arbitrary value z given its parent nodes $\Pi(Z,\mathbf{V},\mathbf{L})$ in $(\mathbf{V},\mathbf{L})$. The joint distribution of the set $\mathbf{Z}$ of all variables in $(\mathbf{V},\mathbf{L})$ is given as follows.

$$\mathbf{P}(\mathbf{Z}) = \prod_{Z\in\mathbf{Z}} P(Z = z|\Pi(Z,\mathbf{V},\mathbf{L})) \tag{1}$$

In the rest of the chapter, we denote the probability distribution $\mathbf{P}(\mathbf{Z} = \mathbf{z})$ of a set $\mathbf{Z}$ of variables taking an arbitrary value $\mathbf{z}$ with $\mathbf{P}(\mathbf{Z})$.

A dynamic Bayesian network is a kind of Bayesian network in which time t is explicit. Each of the hidden Markov model and the Kalman filter is known to be a kind of the dynamic Bayesian network. Details of the Bayesian network can be found in [1, 6, 10].

3.2 *Inferencing the Joint Probability Distribution*

The most popular kinds of inference with the dynamic Bayesian network are filtering, prediction, and smoothing [10]. The filtering computes the a posteriori probability distribution of the states of the present $(t = T)$ given all evidences until the present i.e. it computes $\mathbf{P}(\mathbf{S}_T|\mathbf{e}_{1:T})$ given $\mathbf{e}_{1:T}$, where $\mathbf{e}_{1:T}$ is the set of values of $\mathbf{E}_t$ for $t = 1,2,\ldots,T$. The prediction, on the other hand, computes the a posteriori probability distribution of the states in the future $(t = T+k)$ given all evidences until the present i.e., it computes $\mathbf{P}(\mathbf{S}_{T+k}|\mathbf{e}_{1:T})$ given $\mathbf{e}_{1:T}$, where k is a positive integer. The smoothing computes the a posteriori probability distribution of the past $(t = h)$ given all evidences until the present i.e., it computes $\mathbf{P}(\mathbf{S}_h|\mathbf{e}_{1:T})$ given $\mathbf{e}_{1:T}$, where h is an integer which satisfies $0 \le h < T$.

It is common to assume that the system is a first-order Markov process: the current state $\mathbf{S}_t$ depends only on the previous state $\mathbf{S}_{t-1}$ i.e., $\mathbf{P}(\mathbf{S}_t|\mathbf{S}_{0:t-1}) = \mathbf{P}(\mathbf{S}_t|\mathbf{S}_{t-1})$ for any t. With this assumption, the kinds of inference are significantly simplified. For instance the filtering becomes the following recursive estimation known as the forward message passing.

$$\mathbf{P}(\mathbf{S}_T|\mathbf{e}_{1:T}) = \beta\mathbf{P}(\mathbf{e}_T|\mathbf{S}_T)\sum_{\mathbf{s}_{T-1}} \mathbf{P}(\mathbf{S}_T|\mathbf{s}_{T-1})\mathbf{P}(\mathbf{s}_{T-1}|\mathbf{e}_{1:T-1}), \tag{2}$$

where β represents the regularizer [10].

The recursive estimation such as that of (2) is efficient when the shape of the dynamic Bayesian network is simple such as a chain. In other cases, one has to resort to an approximate inference method such as various sampling methods, Markov Chain Monte Carlo methods, variational methods, and loopy belief propagation methods [4, 10].

In this chapter, we, as designers, are interested in the observable evidences of swarm robots in the future. Therefore the objective of our prediction is to compute the joint probability distribution $\mathbf{P}(\mathbf{E}_{T_{\text{fin}}}|\mathbf{e}_1)$ of the evidences in the future ($t = T_{\text{fin}}$) given the initial evidence $\mathbf{e}_1$. The computation of the prediction can be treated similarly to the smoothing: it is an iterative application of the following equation from $t = 1, 2, \ldots, T_{\text{fin}} - 1$.

$$\mathbf{P}(\mathbf{E}_{t+1}|\mathbf{e}_t) = \sum_{s_t} \mathbf{P}(\mathbf{E}_{t+1}|\mathbf{e}_t, \mathbf{s}_t)\mathbf{P}(\mathbf{s}_t|\mathbf{e}_t) \tag{3}$$

3.3 Application to Robot Control

In a robot control planning problem, the desired target $\mathbf{S}_{T_{\text{fin}}}$ at $t = T_{\text{fin}}$ may be given in addition to the initial state $\mathbf{s}_1$. Toussaint et al. have shown that the planning problem can be solved by expressing knowledge on the structure of the robot as a dynamic Bayesian network and calculating the optimal intermediate states $\mathbf{s}_{1:T_{\text{fin}}}$ under given constraints [12, 13]. In [12], they demonstrated an example of a single humanoid robot with the angles of its multiple joints as the state variables and the positions of its end-effector as the observations. Unfortunately, we think that the model is too simple to be applied to the design of the swarm robots.

Pfeffer and Tai have proposed a model based on the dynamic Bayesian networks for autonomous agents that interact with each other in a distributed, asynchronous manner [8]. Their model mainly assumes as its application a sensor network for monitoring a dynamic system, where message transmissions among agents play a central role. A swarm system, however, is not necessarily a monitoring system thus we think that the model is limited for our purpose.

4 Toward a Novel Design of Swarm Robots

Suppose we are designing autonomous swarm robots each of which has a deterministic controller $C(\theta)$ with a set θ of parameters. We assume that the environment is dynamic and all robots and the environment are synchronized i.e., they act simultaneously at time slices $t = 1, 2, \ldots$. The sets of the latent and observable variables of a robot i at time t are denoted with $\sigma_{t,i}$ and $\varepsilon_{t,i}$, respectively. Since we are designers, $\sigma_{t,i}$ and $\varepsilon_{t,i}$ represent the internal state such as its sensor values and hence its actuator values, and the physical evidence such as its position, of the robot i at time t, respectively. We assume that the corresponding sets of the dynamic environment are denoted with $\sigma_{t,0}$ and $\varepsilon_{t,0}$, respectively. The corresponding variables of $\sigma_{t,i}$ and $\varepsilon_{t,i}$ are denoted with $\mathbf{S}'_{t,i}$ and $\mathbf{E}'_{t,i}$, respectively.

The situation may be described with a dynamic Bayesian network where $\mathbf{S}_t = \{\mathbf{S}'_{t,0}, \mathbf{S}'_{t,1}, \ldots\}$ and $\mathbf{E}_t = \{\mathbf{E}'_{t,0}, \mathbf{E}'_{t,1}, \ldots\}$. We assume that the state $\mathbf{S}'_{t,i}$ of robot i at

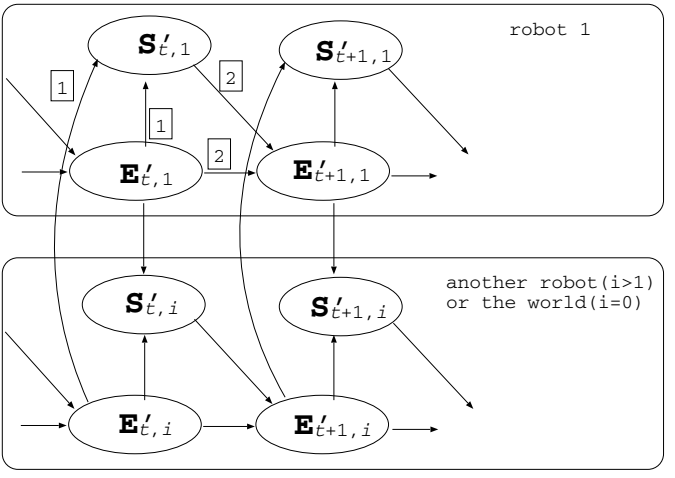


Fig. 1 Dynamic Bayesian network for the swarm robot design problem

time t depends on the current evidence $\mathbf{e}_t$ and thus try to obtain its conditional probability distribution $\mathbf{P}(\mathbf{S}'_{t,i}|\mathbf{e}_t)$. Note that another possibility would be to assume that $\mathbf{S}'_{t,i}$ also depends on $\sigma_{t-1,i}$, which corresponds to equipping the swarm robot with a memory of the previous state. We do not take this possibility as we are interested in analyzing reactive swarm robots.

For the evidence $\mathbf{E}'_{t+1,i}$ of robot i at time $t+1$, we assume that it depends on the previous evidence $\varepsilon_{t,i}$ and the previous state $\sigma_{t,i}$. We also try to obtain its conditional probability distribution $\mathbf{P}(\mathbf{E}'_{t+1,i}|\varepsilon_{t,i},\sigma_{t,i})$. We show the dynamic Bayesian network in Figure 1. Note that $\mathbf{P}(\mathbf{S}'_{t,i}|\mathbf{e}_t)$ and $\mathbf{P}(\mathbf{E}'_{t+1,i}|\varepsilon_{t,i},\sigma_{t,i})$ may be called as a status model and an evidence model of robot i, respectively. Each of them is a conditional probability table used in the inference on the dynamic Bayesian network.

In this case, (3) may be written for each robot as follows.

$$\mathbf{P}(\mathbf{E}'_{t+1,i}|\mathbf{e}_t) = \sum_{\mathbf{s}_t} \mathbf{P}(\mathbf{E}'_{t+1,i}|\mathbf{e}_t,\mathbf{s}_t)P(\mathbf{s}_t|\mathbf{e}_t) \tag{4}$$

A failure $F(\mathbf{E}_{T_{\text{fin}}})$ at time T_{fin} is a proposition which is defined in terms of $\mathbf{E}_{T_{\text{fin}}}$. We may try to design $C(\theta)$ so that the probability $P(F(\mathbf{E}_{T_{\text{fin}}}))$ of the failure $F(\mathbf{E}_{T_{\text{fin}}})$ at time T_{fin} is minimized. This problem is difficult as the number of the possible initial conditions is usually infinite. Thus we assume a specific initial condition $\mathbf{e}_1$, which we call a critical case, and modify θ so that $P(F(\mathbf{E}_{T_{\text{fin}}})|\mathbf{e}_1)$ is minimized.

To calculate $P(F(\mathbf{E}_{T_{\text{fin}}})|\mathbf{e}_1)$, we obtain the probabilities of the states $\mathbf{P}(\mathbf{S}_{2:T_{\text{fin}}}|\mathbf{e}_1)$ and the probabilities of the evidences $\mathbf{P}(\mathbf{E}_{2:T_{\text{fin}}}|\mathbf{e}_1)$ given the initial evidence $\mathbf{e}_1$. The modification method of the swarm robot, especially its controller, for decreasing $P(F(\mathbf{E}_{T_{\text{fin}}})|\mathbf{e}_1)$ depends on the application problem and thus is not given in a general form in this chapter.

5 Example of Two Swarm Robots with a Static Environment

5.1 Collision Avoidance Problem and Our Swarm Robot

We consider the design of $C(\theta)$ of two swarm robots in a room with fixed obstacles i.e., the number of robots is two and the environment is static so $\mathbf{S}_t = \{\mathbf{S}'_{t,1}, \mathbf{S}'_{t,2}\}$ and $\mathbf{E}_t = \{\mathbf{E}'_{t,1}, \mathbf{E}'_{t,2}\}$. The failure $F(\mathbf{E}_{T_{\text{fin}}})$ is defined as a collision between a robot with the other robot, an obstacle, or the wall. Two examples of an initial evidence (i.e., critical case) $\mathbf{e}_1$ are shown in Figure 2. Each robot has a sensor to detect obstacles in front of it and thus a robot i has one variable $s_{t,i}$ which represents whether it has detected an obstacle at time t. The corresponding values of $s_{t,i}$ are true and false, respectively. We assume that the controller of a robot is deterministic and thus its action depends on the value of $s_{t,i}$ so $\sigma_{t,i} = \{s_{t,i}\}$.

We show a picture of our robot swarm in Figure 3. The robot kit is called Robo Designer and is commercially available from Japan Robotech ltd. We have placed three distance measurement sensors in front of each robot. Here the distance measurement sensor emits an infra-red signal and if it detects a reflection of the emitted signal it returns an electric signal to the on-board computer. The sensor is capable of detecting objects in the range of approximately 20 cm - 80 cm in a range of 5 degrees.

We assume that the velocity of a robot is constant. Therefore the evidence $\varepsilon_{t,i}$ of a robot i at time t consists of its 2-dimensional coordinate $(x_{t,i}, y_{t,i})$ on the surface and its orientation $\alpha_{t,i}$ i.e., $\varepsilon_{t,i} = \{x_{t,i}, y_{t,i}, \alpha_{t,i}\}$. $X_{t,i}, Y_{t,i}, A_{t,i}$ are defined as the respective variables. We assume that a robot can either move forward or make a 90-degree right-turn so the simplest form of a controller $C(\theta)$ may be a decision list such as "if $s_{t,i}$ = true make a 90-degree right-turn, otherwise go straight". We adopt this controller in the rest of the chapter hence the parameter θ consists of the values related with $s_{t,i}$ = true and 90. For instance, the robot can sense 3 times in 0.1 second then judge the existence of an obstacle with a threshold value for its electric signal with the majority vote.

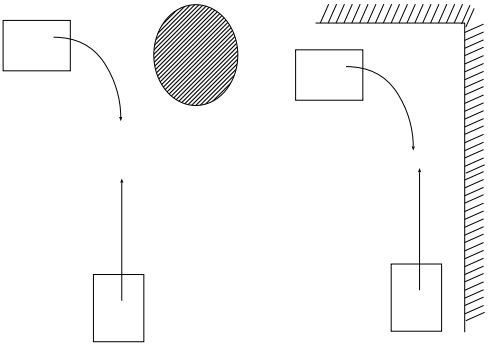


Fig. 2 Two examples of an initial evidence (i.e., critical case)

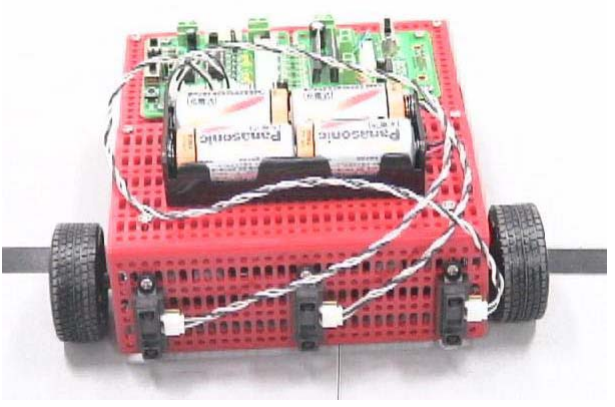


Fig. 3 Our swarm robot

5.2 *Prediction Based on a Dynamic Bayesian Network*

In our problem, (4) may be rewritten as follows.

$$\begin{aligned}
&\mathbf{P}(X_{t+1,1}, Y_{t+1,1}, A_{t+1,1}, X_{t+1,2}, Y_{t+1,2}, A_{t+1,2} | x_{t,1}, y_{t,1}, \alpha_{t,1}, x_{t,2}, y_{t,2}, \alpha_{t,2}) \\
&= \sum_{s_{t,1}} \sum_{s_{t,2}} \mathbf{P}(X_{t+1,1}, Y_{t+1,1}, A_{t+1,1}, X_{t+1,2}, Y_{t+1,2}, A_{t+1,2} | x_{t,1}, y_{t,1}, \alpha_{t,1}, x_{t,2}, y_{t,2}, \alpha_{t,2}, \\
&\quad s_{t,1}, s_{t,2}) \mathbf{P}(s_{t,1}, s_{t,2} | x_{t,1}, y_{t,1}, \alpha_{t,1}, x_{t,2}, y_{t,2}, \alpha_{t,2}) \qquad (5) \\
&= \sum_{s_{t,1}} \sum_{s_{t,2}} \mathbf{P}(X_{t+1,1}, Y_{t+1,1}, A_{t+1,1} | x_{t,1}, y_{t,1}, \alpha_{t,1}, s_{t,1}) \mathbf{P}(X_{t+1,2}, Y_{t+1,2}, A_{t+1,2} | x_{t,2}, y_{t,2}, \\
&\quad \alpha_{t,2}, s_{t,2}) P(s_{t,1} | D(x_{t,1}, y_{t,1}, \alpha_{t,1}, x_{t,2}, y_{t,2}, \alpha_{t,2})) P(s_{t,2} | D(x_{t,2}, y_{t,2}, \alpha_{t,2}, x_{t,1}, y_{t,1}, \\
&\quad \alpha_{t,1})) \qquad (6)
\end{aligned}$$

where $D(x_{t,i}, y_{t,i}, \alpha_{t,i}, x_{t,j}, y_{t,j}, \alpha_{t,j})$ is a proposition which becomes true when at least one of the other robot j, the wall, and the obstacle is located in the visible range of the sensors of the robot i.

In (6), we assume that the new coordinates and the new orientations are independent given the previous coordinates, orientations, and the states.

$$\begin{aligned}
\mathbf{P}(X_{t+1,i}, Y_{t+1,i}, A_{t+1,i} | x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i}) = \mathbf{P}(X_{t+1,i}, Y_{t+1,i} | x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i}) \\
\mathbf{P}(A_{t+1,i} | x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i}) \qquad (7)
\end{aligned}$$

We also assume that each of $\mathbf{P}(X_{t+1,i}, Y_{t+1,i} | x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i})$ and $\mathbf{P}(A_{t+1,i} | x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i})$ follows a Gaussian distribution when $s_{t,i}$ = false. The former Gaussian has a mean $(L\cos(\alpha_{t,i}) + x_{t,i}, L\sin(\alpha_{t,i}) + y_{t,i})$, where L represents the average length that the robot moves in one time slice, and a two-dimensional covariance matrix.

Fig. 4 Our environment of the experiments

The latter Gaussian is one-dimensional and has a mean $\alpha_{t,i}$. The number of the parameters is 1+4+1=6 and if the covariance matrix has non-zero elements only in its diagonal it is 4. These parameters may be estimated by a sufficient number of experiments with a single swarm robot, which makes forward moves.

We take similar assumptions when $s_{t,i}$ = true. The difference to those in the case of $s_{t,i}$ = false is that $\mathbf{P}(X_{t+1,i}, Y_{t+1,i}|x_{t,i}, y_{t,i}, \alpha_{t,i}, s_{t,i})$ is a Gaussian with a mean $(M + x_{t,i}, M + y_{t,i})$, where M represents the average length that the robot moves in one time slice when it makes a 90-degree right-turn and the Gaussian for the orientation has a mean of $\alpha_{t,i} + 90$ degree. The number of the parameters is again 6 in the general case or 4 with the specific type of the covariance matrix. These parameters may be estimated by a sufficient number of experiments with a single swarm robot, which makes a 90-degree right-turn.

Note that we assume that L and M are independent of the position on the surface. The time slice should be adjusted so that the velocity of a robot becomes stable in reality.

We assume that $P(s_{t,i}|D(x_{t,i}, y_{t,i}, \alpha_{t,i}, x_{t,j}, y_{t,j}, \alpha_{t,j}))$ is identical for $i = 1, 2$. Therefore, the probabilities may be represented by a 2×2 confusion matrix i.e., the degree of freedom is 2 and it suffices to estimate the probabilities of the false positive and the false negative. Again these parameters may be estimated by a sufficient number of experiments with a single swarm robot, which tries to detect objects in front of it and not to mis-detect non-existent objects.

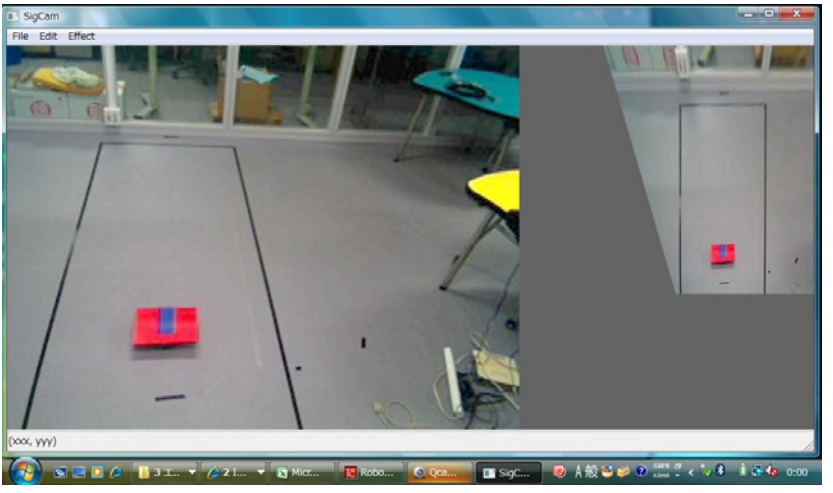


Fig. 5 Projection transformation of the field

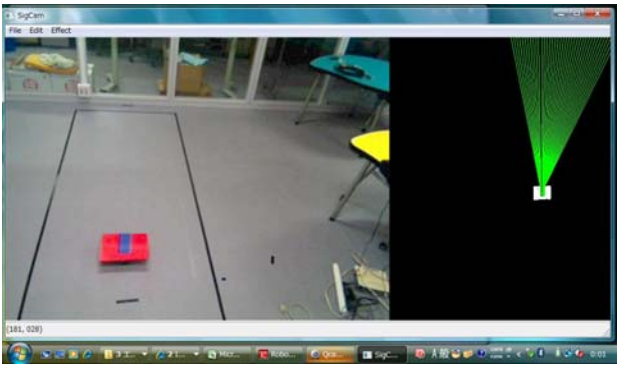


Fig. 6 Calculation of the position and the orientation of the swarm robot

5.3 Measurement of Basic Statistics

We are currently measuring the basic statistics of our swarm robot. Figure 4 shows our environment of the experiments. We use a USB camera (Logicool Qcam Orbit) of 300,000 pixels and we set it on a position of 160 cm height with a 45-degree angle to the ground. The frame rate of the movie is 10 frames per second and the size of each image is 640 pixels (W) $\times$ 480 pixels (H).

The size of the field in which the swarm robot moves is 70 cm (W) $\times$ 200 cm (H). We have performed a projection transformation to the camera image of resolution 140 pixels (W) $\times$ 400 pixels (H). Figure 5 shows our projection transformation of the field. A move of a robot of 0.5 cm in reality corresponds to its move of 1 pixel on the camera image.

To measure the current position and the orientation of a swarm robot, a color-based image processing has been used. We have colored the robot as shown in Figure 6 and assume that the center of gravity of red pixels corresponds to the position of the robot. The orientation is estimated as the direction to which the length of the blue line segment is longest.

6 Concluding Remarks

We have proposed a method for interpreting the behavior of swarm robots for their efficient design. The method provides the joint probability distribution of the future evidences, which is based on the basic statistics of a single swarm robot. Since the joint probability distribution of the future evidences may be used for computing the probability of the failure, the number of real experiments is significantly reduced. Our approach is based on statistical theories and is thus considered to be superior to ad-hoc approaches relying on easily-available methods.

We have made a very general assumption on the swarm robots and the environment. The price that we pay is the combinatorial explosion of the future states but most of the combinations may be eliminated if we assume some confidence level for an approximate reason. Another direction of research is to assume constraints on the form of the collective behaviors of the swarm robots such as aligning them in one line. This kind of constraints typically reduces the numbers of possible states and evidences, allowing us to simulate behaviors of longer periods compared to the general case.

Acknowledgements. A part of this research was supported by Strategic International Cooperative Program funded by Japan Science and Technology Agency (JST).

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
2. Dorigo, M.: Editorial. Swarm Intelligence 1(1), 1–2 (2007)
3. Garnier, S., Gautrais, J., Theraulaz, G.: The Biological Principles of Swarm Intelligence. Swarm Intelligence 1(1), 3–31 (2007)
4. Jordan, M.I., Ghahramani, Z., Jaakkola, T., Saul, L.K.: An Introduction to Variational Methods for Graphical Models. Machine Learning 37(2), 183–233 (1999)
5. Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D., Strohband, S.: Winning the DARPA Grand Challenge with an AI Robot. In: Proc. AAAI (2006)
6. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. dissertion, University of California, Berkeley (2002)
7. Nouyan, S., Campo, A., Dorigo, M.: Path Formation in a Robot Swarm. Swarm Intelligence 2(1), 1–23 (2008)
8. Pfeffer, A., Tai, T.: Asynchronous Dynamic Bayesian Networks. In: Proc. UAI, pp. 467–476 (2005)

9. Şahin, E., Winfield, A.: Special Issue on Swarm Robotics. Swarm Intelligence 2(2-4), 69–72 (2008)
10. Russel, S., Norvig, P.: Artificial Intelligence, a Modern Approach, 2nd edn. Prentice Hall, Upper Saddle River (2003)
11. Thrun, S.: Why We Compete in DARPA's Urban Challenge Autonomous Robot Race. CACM 50(10), 29–31 (2007)
12. Toussaint, M., Goerick, C.: Probabilistic Inference for Structured Planning in Robotics. In: Int. Conf. on Intelligent Robots and Systems (IROS), pp. 3068–3073 (2007)
13. Toussaint, M., Storkey, A.J.: Probabilistic Inference for Solving Discrete and Continuous State Markov Decision Processes. In: Proc. ICML, pp. 945–952 (2006)

Current Research Trends in Possibilistic Logic: Multiple Agent Reasoning, Preference Representation, and Uncertain Databases

Henri Prade

Abstract. Possibilistic logic is a weighted logic that handles uncertainty, or preferences, in a qualitative way by associating certainty, or priority levels, to classical logic formulas. Moreover, possibilistic logic copes with inconsistency by taking advantage of the stratification of the set of formulas induced by the associated levels. Since its introduction in the mid-eighties, multiple facets of possibilistic logic have been laid bare and various applications addressed: handling exceptions in default reasoning, modeling belief revision, providing a graphical Bayesian-like network representation counterpart to a possibilistic logic base, representing positive and negative information in a bipolar setting with applications to preferences fusion and to version space learning, extending possibilistic logic for dealing with time, or multiple agents mutual beliefs, developing a symbolic treatment of priorities for handling partial orders between levels and also improving computational efficiency, learning stratified hypotheses for coping with exceptions. The chapter aims primarily at offering an introductory survey of possibilistic logic developments. Still, it also outlines new research trends that are relevant in preference representation, or in reasoning about epistemic states.

1 Introduction

Possibilistic logic has been developed for about twenty-five years [49]. Possibilistic logic has been first introduced in artificial intelligence as a tool for handling uncertainty in a qualitative way in a logical setting. Each classical logic formula is associated with a certainty level. Later on, it has appeared that possibilistic logic can also be used for representing preferences [62]. Then, each logic formula represents a goal to be reached with its priority level (rather than a statement that is believed to be true with some certainty level). An interesting feature of possibilistic logic is

Henri Prade
IRIT, 118 route de Narbonne, CNRS and University of Toulouse,
31062 Toulouse Cedex 9, France
e-mail: prade@irit.fr

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 311–330.
springerlink.com

its ability to deal with inconsistency. Indeed a possibilistic logic base B, i.e. a set of possibilistic logic formulas, is associated with an inconsistency level $inc(B)$, which is such that the formulas associated with a level strictly greater than $inc(B)$ form a consistent subset of formulas.

These are the basic features of standard possibilistic logic, which are especially at work for encoding non monotonic reasoning and belief revision. However, there exist further extensions that this chapter more particularly addresses. The intended purpose of this introductory survey is to provide a broad overview of possibilistic logic developments, lying bare the basic ideas and the main mechanisms. For the technical details and examples, the reader is referred to the rich bibliography that is provided.

The chapter is organized in four main parts. The representational framework of possibilistic logic is briefly restated in Section 2, emphasizing the existence of different representation formats, including the use of constrained symbolic levels for the handling of partially ordered levels. Then in Section 3, two further representation issues, which have been more recently considered, are presented. First, the bipolar representations that handles both classical logic-like information that delimits a (fuzzy) set of models where the actual world may be (which correspond to non impossible interpretations), and so-called positive information expressing that subsets of interpretations are *actually possible* to some extent. Second, extensions of possibilistic logic for handling nested formulas and dealing with multiple agent situations are discussed, including some hints for reasoning about epistemic states. Section 4 suggests further developments in preference representation. Lastly, Section 5 outlines a possibilistic logic-like treatment of uncertainty in databases.

2 Background

A possibilistic logic base is semantically equivalent to a possibility distribution that restricts the set of interpretations (w. r. t. the considered language) that are more or less compatible with the base. Instead of an ordinary subset of models as in classical logic, we have a fuzzy set of models, since the violation by an interpretation of a formula that is not fully certain (or imperative) does not completely rule out the interpretation. Moreover, there exists two other noticeable representation frameworks that are equivalent to a possibilistic logic base (and are also semantically associated to a possibility distribution), which explicitly refer to contexts: i) a conditionals-based representation that expresses default statements, and ii) a Bayesian-like network representation that provides a graphical counterpart to a possibilistic logic base.

2.1 Standard Possibilistic Logic

Standard possibilistic logic [41] has been essentially developed as a formalism for handling qualitative uncertainty and preferences with an inference mechanism that is a simple extension of classical logic. Then a possibilistic logic formula is a pair

made of i) any well-formed classical logic formula, propositional or first-ordered, and ii) a weight expressing its certainty or priority. Such classical logic formulas can be only true or false, and fuzzy statements with intermediary degrees of truth are not allowed in standard possibilistic logic (although extensions exist for handling fuzzy predicates [52, 2, 3]).

A standard possibilistic logic expression is a pair (ϕ, α), where ϕ is a classical logic formula and $\alpha \in (0,1]$ is interpreted as a lower bound of a necessity measure N, i.e., (ϕ, α) is semantically interpreted as $N(\phi) \geq \alpha$, where N is a necessity measure. Formulas of the form $(\phi, 0)$, which do not contain any information ($N(\phi) \geq 0$ always holds), are not part of the possibilistic language. The interval [0,1] can be replaced by any linearly ordered scale, and even by distributive lattice structures (as in [39], where a logical formula is associated with the fuzzy set of time instants where the formula is more or less certainly true). A possibilistic logic base is a set of possibilistic logic formulas.

Necessity measures N are monotonic functions w. r. t. entailment, i.e. if $\phi \models \psi$ then $N(\phi) \leq N(\psi)$, and they characterized by the decomposability property

$$N(\phi \wedge \psi) = \min(N(\phi), N(\psi)),$$

and are dual of possibility measures Π (namely $N(\phi) = 1 - \Pi(\neg\phi)$). Mind that we only have $N(\phi \vee \psi) \geq \max(N(\phi), N(\psi))$. This goes well with the idea that one may be certain about the general statement $\phi \vee \psi$, without being really certain about more specific statements such as ϕ and ψ. In the same spirit, the *min* decomposability property of necessity measures N w. r. t. conjunction expresses that to be certain about $\phi \wedge \psi$, one has to be certain about ϕ and to be certain about ψ. Thanks to this decomposability property, a possibilistic logic base can be always put in a clausal equivalent form.

Dual possibility and necessity measures Π and N are based on the same possibility distribution that essentially encodes a preorder that rank-orders interpretations. A possibility measure Π is defined from a possibility distribution π as $\Pi(p) = max_{\omega \models p} \pi(\omega)$ [77].

A propositional possibilistic logic base $B = \{(p_i, \alpha_i) | i = 1, n\}$ is semantically associated with the possibility distribution

$$\pi_B(\omega) = \min_{i=1,n} \pi_{(p_i,\alpha_i)}(\omega)$$

with $\pi_{(p_i,\alpha_i)}(\omega) = 1$ if $\omega \models p_i$, and $\pi_{(p_i,\alpha_i)}(\omega) = 1 - \alpha_i$ if $\omega \models \neg p_i$.

Thus, π_B is obtained as the min-based conjunctive combination of the representations of each formula in B. Moreover, an interpretation ω is all the more possible as it does not violate any formula p_i with a high certainty level α_i (since if ω violates p_i, $\pi_{(p_i,\alpha_i)}(\omega) = 1 - \alpha_i$ and then the possibility $\pi_B(\omega)$ of ω would be small, as $1 - \alpha_i$ is).

The basic inference rule in possibilistic logic put in clausal form is the following resolution rule, here written in the propositional case:

$$(\neg p \vee q, \alpha); (p \vee r, \beta) \models (q \vee r, \min(\alpha, \beta)).$$

Using this rule repeatedly, a refutation-based proof procedure that is sound and complete w. r. t. the semantics exists for propositional possibilistic logic [41]. Algorithms and complexity evaluation (similar to the one of classical logic) can be found in [63]. It is worth pointing out that a similar approach with probability lower bounds would not ensure completeness [42]. Indeed the repeated use of the probabilistic counterpart of the above resolution rule, namely $(\neg p \vee q, \alpha); (p \vee r, \beta) \models (q \vee r, \max(0, \alpha + \beta - 1))$ (where (ϕ, α) now means $Probability(\phi) \geq \alpha$), is not always enough for computing the best probability lower bounds on a formula, given a set of probabilistic constraints of the above form.

Moreover, a formula such as $(\neg p \vee q, \alpha)$ can be rewritten under the semantically equivalent form $(q, \min(^t(p), \alpha))$, where $^t(p) = 1$ if p is true and $^t(p) = 0$ if p is false. This latter formula now reads "q is α-certain, provided that p is true" and can be used in hypothetical reasoning in case (p, γ) is not deducible from the available information (for some $\gamma > 0$) [12, 46].

Lastly, an important feature of possibilistic logic is its ability to deal with inconsistency. The level of inconsistency $inc(B)$ of a possibilistic logic base B (i.e. a set of possibilistic logic formulas) is defined as $inc(B) = \max\{\alpha \mid B \models (\bot, \alpha)\}$. All formulas whose level is strictly greater than $inc(B)$ are safe from inconsistency. It can be shown that $1 - inc(B)$ is nothing but the height $h(\pi_B)$ of π_B, defined by $h(\pi_B) = max_\omega \pi_B(\omega)$. All formulas in B whose level is less or equal to $inc(B)$ are ignored in the standard possibilistic inference process; they are said to be drawn. However, other inferences that salvage formulas that are below the inconsistency level but are not involved in some inconsistent subsets of formulas, have been defined and studied [17].

Remark: Representing ignorance. In standard possibilistic logic, formulas of the form $(p, 0)$ are ignored, since they correspond to the trivial piece of information $N(p) \geq 0$. The piece of information (p, α) stating that $N(p) \geq \alpha$ is compatible with any possibility distribution π such that $\pi \leq \pi_{(p,\alpha)}$. Then, the information is represented by the largest possibility distribution compatible with the constraint, i.e., $\pi_{(p,\alpha)}$ by virtue of the minimal specificity principle, which privileges the less restrictive distributions, i.e. those that keep the levels of possibility as large as possible taking into account the constraints. Thus, $N(p) \geq 0$ would be represented by $\pi_{(p,0)} = 1$. Thus, the semantics underlying a possibilistic logic base $B = \{(p_i, \alpha_i) | i = 1, n\}$ is that an empty base leaves all the interpretations equally possible and that any new non trivial piece of information (p_i, α_i) cannot but reduce the levels of possibility of the interpretations.

Besides, formulas associated with lower bounds of possibility measures (rather than necessity measures) have been also introduced [41, 44] and can be used for acknowledging partial ignorance, where fully ignoring p amounts to write that $\Pi(p) = 1 = \Pi(\neg p)$. This is another form of ignorance, which might be termed "alleged ignorance" and corresponds more generally to the situation where $\Pi(p) \geq \alpha > 0$ and $\Pi(\neg p) \geq \beta > 0$. This expresses that both p and $\neg p$ are somewhat *possible*, and contrasts with the type of uncertainty encoded by (p, α), which in turn is equivalent

to $\Pi(\neg p) \leq 1-\alpha$, which expresses that $\neg p$ is rather *impossible*. Alleged ignorance can be transmitted through equivalences. Namely from $\Pi(p) \geq \alpha > 0$ and $\Pi(\neg p) \geq \beta > 0$, one can deduce $\Pi(q) \geq \alpha > 0$ and $\Pi(\neg q) \geq \beta > 0$ provided that we have $(\neg p \vee q, 1)$ and $(p \vee \neg q, 1)$ [44, 69]. This form of alleged ignorance is different from the idea of "unawareness" about p, which corresponds to the absence of any formula involving p in the possibilistic base B.

2.2 Default Reasoning, Causality and Belief Revision

Such an ability to deal with inconsistency is exploited in default reasoning. A default rule "generally, if p then q" is represented by the conditional $\Pi(p \wedge q) > \Pi(p \wedge \neg q) \iff N(q|p) > 0$ (where Π denotes a possibility measure, and $N(q|p) = 1 - \Pi(\neg q|p)$ and $\Pi(q|p) = 1$ if $\Pi(p \wedge q) \geq \Pi(p \wedge \neg q)$ and $\Pi(q|p) = \Pi(p \wedge q)$ if $\Pi(p \wedge q) < \Pi(p \wedge \neg q)$). Thus, $N(q|p) > 0$ expresses that in the context where p is true, having q true is strictly more possible than q false. Then by laying bare the largest possibility distribution underlying a consistent set of defaults $\Pi(p_i \wedge q_i) > \Pi(p_i \wedge \neg q_i)$ for $i = 1, n$, it is possible to stratify the set of defaults according to their specificity (roughly speaking the most specific defaults receive the higher levels), and then to encode them by possibilistic logic formulas [13]. This encoding takes advantage of the fact that when new certain information is received, the level of inconsistency of the base cannot decrease, and if it strictly increases, some inferences that were safe before are now drawn in the new inconsistency level of the base and are thus no longer allowed, hence a non monotonic consequence mechanism takes place. Such an approach has been proved to be in full agreement with the Kraus-Lehmann-Magidor [61] postulates-based approach to nonmonotonic reasoning [15]. Moreover, a default rule maybe itself associated with a certainty level (in such a case each formula will be associated with two levels, namely a priority level reflecting its relative specificity in the base, and its certainty level) [53].

Let us also point out that since $N(q|p) > 0$ expresses that in context p, q is true in the normal course of things, qualitative necessity measures may be used for describing how (potential) causality is perceived in relation with the advent of an abnormal event that precedes a change. Namely, if an agent has the two following beliefs represented by $N(q|p) > 0$ and $N(\neg q|p \wedge r) > 0$ about the normal course of things, and that it has been reported that we are in context p, and that q, which was true, has become false after r takes place, then the agent will be inclined to think that "p caused $\neg q$". See [32] for a detailed presentation and discussion of this model (both from a formal and from a cognitive psychology point of view).

Besides, qualitative necessity relations (which can be encoded by necessity measures) are nothing but the epistemic entrenchment relations [45] that underly well-behaved belief revision processes [57]. This enables the possibilistic logic setting to provide syntactic revision operators that apply to possibilistic knowledge bases, including the case of uncertain inputs [47, 18]. Note that in possibilistic logic, the epistemic entrenchment of the formulas is explicit.

2.3 Possibilistic Bayesian Networks

A possibilistic logic base can be also changed into a possibilistic directed acyclic graph and vice-versa [7]. Such a graph exhibits a conditional independence structure just like for Bayesian nets. There exist two types of possibilistic Bayesian nets, depending on the conditioning that is used. Indeed conditioning may be defined qualitatively through the equation $\Pi(p \wedge q) = min(\Pi(q|p), \Pi(p))$ (which gives birth to the definition of $\Pi(q|p)$ already mentioned), or quantitatively using the product in place of 'min' in the previous equation. They are counterparts of probabilistic Bayesian nets with specific computational procedures that take advantage of the idempotency of the combination operation in possibility theory [5, 20]. Tools for computing "interventions" for causality assessments have been recently introduced in this setting [25]. Such graphical structures may be also of particular interest for representing preferences [14].

An important feature of the possibilistic logic setting is the existence of equivalent representation formats: set of prioritized logical formulas, preorders on interpretations (possibility distributions) at the semantical level, set of conditionals (of the form $\Pi(p \wedge q) > \Pi(p \wedge \neg q)$), graphical nets, for which there are algorithms for translating one format in another [8]. Recently, hybrid representations formats have been introduced where local possibilistic logic bases are associated to the nodes of a graphical structure [26].

2.4 Fusion

The fusion of information can take place in the different representation formats of the possibilistic setting. In particular, the combination of possibility distributions can be equivalently performed in terms of possibilistic logic bases. Namely, the syntactic counterpart of the pointwise combination of two possibility distributions π_1 and π_2 into a distribution $\pi_1 \oplus \pi_2$ by any monotonic[1] combination operator $\oplus$ such that $1 \oplus 1 = 1$, can be computed, following an idea first proposed in [31]. Namely, if the possibilistic logic base B_1 is associated with π_1 and the base B_2 with π_2, a possibilistic base that is semantically equivalent to $\pi_1 \oplus \pi_2$ can be obtained in the following way [16]:

$$B_{1\oplus 2} = \{(\varphi_i, 1-(1-\alpha_i)\oplus 1) \text{ s.t. } (\varphi_i, \alpha_i) \in B_1\},$$
$$\cup \{(\psi_j, 1-1\oplus(1-\beta_j)) \text{ s.t. } (\psi_j, \beta_j) \in B_2\},$$
$$\cup \{(\varphi_i \vee \psi_j, 1-(1-\alpha_i)\oplus(1-\beta_j)) \text{ s.t. } (\varphi_i, \alpha_i) \in B_1, (\psi_j, \beta_j) \in B_2\}.$$

$$\text{For} \quad \oplus = \min, \text{we get} \quad B_{1\oplus 2} = B_1 \cup B_2 \quad \text{with} \quad \pi_{B_1 \cup B_2} = \min(\pi_1, \pi_2)$$

[1] $\oplus$ is supposed to be monotonic in the wide sense for each of its arguments: $\alpha \oplus \beta \geq \gamma \oplus \delta$ as soon as $\alpha \geq \gamma$ and $\beta \geq \delta$.

as expected (conjunctive combination). For $\oplus = \max$ (disjunctive combination), we get

$$B_{1\oplus 2} = \{(\varphi_i \vee \psi_j, \min(\alpha_i, \beta_j)) \text{ s.t. } (\varphi_i, \alpha_i) \in B_1, \text{ and } (\psi_j, \beta_j) \in B_2\}.$$

With non idempotent $\oplus$ operators, some reinforcement effects may be obtained. Moreover, this approach has been also applied to the syntactic encoding of the Hamming distance-based merging of classical logic bases (where distances are computed between each interpretation and the different classical logic bases, thus giving birth to the counterparts of possibility distributions) [9]. Besides, fusion can be applied directly to qualitative or quantitative possibilistic networks [27, 28].

2.5 Partially Ordered Priorities

When possibilistic logic formulas are coming from different sources, it is not always possible to stratify them according to a complete preorder. Only partial information may be available about the ordering between the levels associated to the formulas [21]. This partial information can be represented by classical logic formulas pertaining to symbolic levels. Thus, the formula (p, α) can be reinterpreted as a two-sorted classical formula $p \vee A$ (expressing that if the situation is not abnormal ($\neg A$), p should be true). The possibilistic formulas with their symbolic levels together with the partial ordering information between levels can then be encoded in a classical two-sorted logic [22]. For instance, a constraint such as $\alpha < \beta$ translates into formula $\neg A \vee B$. The possibilistic logic inference machinery can be recast in this symbolic setting, and efficient computation procedures can be developed taking advantage of the compilation of the base in a dNNF format [22], including the special case where the levels are totally ordered [23].

3 Further Representation Issues: Bipolarity and Agentivity

Possibilistic logic is a form of labelled deductive systems [55]. Indeed, classical logic formulas are associated with levels (or labels). Different types of labels may be associated to formulas, provided they belong to a lattice structure. In the following, we focus on two recently developed extensions. In the first one, a new type of lower bound is associated to formulas, which no longer assesses the certainty that the interpretations violating the formulas are excluded as possible worlds, but rather expresses to what extent the models of the formulas are actually possible in the real world. In the second extension, formulas are not only associated with certainty levels, but also with sets of agents who entertain the corresponding beliefs. These labels can be also seen as different kinds of modalities, which are however handled in a way that remains close to classical logic. See [54] for an early study of the links between modal logics and possibility theory; see also [37].

3.1 Bipolar Possibilistic Representations

The representation capabilities of possibilistic logic can be enlarged in the bipolar possibilistic setting [50, 11, 48, 37]. This setting allows the representation of both negative and positive information. Negative information reflects what is not (fully) impossible and thus remain potentially possible. It induces (prioritized) constraints restricting where the real world may be (when expressing knowledge), or delimiting the potentially satisfactory choices (when dealing with preferences). Negative information can be encoded by necessity-based possibilistic logic formulas. Indeed, (p,α) encodes $N(p) \geq \alpha$, which is equivalent to $\Pi(\neg p) \leq 1-\alpha$, and thus reflects the impossibility of $\neg p$, which is all the stronger as α is high. Positive information expressing what is actually possible, or what is really desirable, is encoded by a new type of formula based on a set function called guaranteed (or actual) possibility measure (which is to be distinguished from "standard" possibility measures that rather express potential possibility (as a matter of consistency with the available information). This bipolar setting is of interest for representing observations and knowledge, or for representing positive and negative preferences.

Positive information is represented by formulas denoted $[q,\delta]$, which expresses the constraint $\Delta(q) \geq \gamma$, where Δ denotes a measure of actual possibility defined from a possibility distribution δ by $\Delta(q) = min_{\omega \models q}\delta(\omega)$. As a consequence, measures of actual possibility satisfy the following characteristic decomposability property: $\Delta(p \vee q) = min(\Delta(p),\Delta(q))$. Indeed, all the models of $p \vee q$ are actually possible, if both all the models of p and all the models of q are actually possible. Consequently, Δ is decreasing with respect to logical entailment, which contrasts with necessity (or potential possibility) measures.

In other words, the piece of positive information $[q,\delta]$ expresses that any model of q is at least possible with degree δ. Let $P = \{[q_j,\gamma_j] | j = 1,k\}$ be a positive possibilistic logic base. Its semantics is given by the possibility distribution

$$\delta_P(\omega) = \max_{j=1,k} \delta_{[q_j,\gamma_j]}(\omega)$$

$$\text{with} \quad \delta_{[q_j,\gamma_j]}(\omega) = 0 \quad \text{if} \quad \omega \models \neg q_j, \quad \text{and} \quad \delta_{[q_j,\gamma_j]}(\omega) = \gamma_j \quad \text{if} \quad \omega \models q_j.$$

Note that δ_P is obtained as the max-based *disjunctive* combination of the representation of each formula in P. Indeed more positive information increases δ by making more interpretations more actually possible, while more negative information decreases π (in the sense of Section 2) by restricting more the possible worlds.

Fusion operations can be defined at the semantic and at the syntactic level in the bipolar setting [19, 10]. The fusion of the negative part of the information is performed by using the formulas of section 2.4. Their counterpart for *positive* information is

$$P_{1\oplus 2} = \left| \begin{array}{ll} \{[\varphi_i, \alpha_i \oplus 0] & \text{s.t. } [\varphi_i,\alpha_i] \in P_1\}, \\ \cup\ \{[\psi_j, 0 \oplus \beta_j] & \text{s.t. } [\psi_j,\beta_j] \in P_2\}, \\ \cup\ \{[\varphi_i \wedge \psi_j, \alpha_i \oplus \beta_j] & \text{s.t. } [\varphi_i,\alpha_i] \in P_1, [\psi_j,\beta_j] \in P_2\}, \end{array} \right.$$

while $\delta_{P_{1\oplus 2}} = \delta_{P_1} \oplus \delta_{P_2}$. This may be used for aggregating positive (together with negative) preferences given by different agents who state what would be really satisfactory for them (and what they reject more or less strongly). This may also be used for combining positive (together with negative) knowledge. Then positive knowledge is usually made of reported cases that testify what is actually possible, while negative knowledge excludes what is (more or less certainly) impossible.

A consistency condition is natural between positive and negative information, namely what is actually possible should be included in to what is not impossible. Since positive information is combined disjunctively, and negative information conjunctively in a fusion process, this consistency condition should be enforced in the result. This can be done by a revision step that gives priority either to the negative side (in general when handling preferences where rejections are more important), or to the positive side (it may apply for knowledge when reliable observations are conflicting with general beliefs). Besides, it has been shown that this double revision mechanism in the bipolar setting enables a stratified extension of the version space approach for learning concepts from examples (positive information) and counter-examples (negative information) [70].

3.2 Multiple Agent Possibilistic Logic

Possibilistic logic has been recently extended by allowing formulas to be associated with subsets of agents and to be nested in order to encode the beliefs of different agents and their mutual beliefs [51]. Let $\mathscr{U}$ denote the set of all agents.

Beliefs of a subset of agents. In particular in this extension, it is possible to express that (at least) all the agents in a subset $\mathscr{A} \subseteq \mathscr{U}$ have some belief p, denoted $(p, \mathscr{A})$, or that there is at least one agent in $\mathscr{A}$ believes p denoted $(p, [\mathscr{A}])$. It can be checked that the following rules are valid

$$(\neg p \vee q, \mathscr{A}); (p \vee r, \mathscr{B}) \models (q \vee r, \mathscr{A} \cap \mathscr{B})$$

$$(\neg p \vee q, \mathscr{A}); (p \vee r, [\mathscr{A}]) \models (q \vee r, [\mathscr{A}])$$

Note that the first rule takes advantage of the lattice structure of power sets, and parallels the standard possibilistic logic resolution rule. The above rules apply to all-or-nothing beliefs and can be easily extended to the handling of multiple agent graded beliefs of the form "all the agents in $\mathscr{A}$ believes p at least at level α" denoted $(p, \alpha/\mathscr{A})$. Namely, the following inference rule is valid

$$(\neg p \vee q, \alpha/\mathscr{A}); (p \vee r, \beta/\mathscr{B}) \models (q \vee r, \min(\alpha, \beta)/\mathscr{A} \cap \mathscr{B})$$

When $\alpha = 1 = \beta$, we retrieve the above resolution rule, identifying $(p, \mathscr{A})$ with $(p, 1/\mathscr{A})$. When $\mathscr{A} = \mathscr{U} = \mathscr{B}$, we retrieve the standard possibilistic resolution rule. The idea of associating formulas with agents may be found in many works nowadays. For instance, in [75], the agents are ranked according to their credibility, which is used for consistency maintenance when incoming information conflicts with

current agent's beliefs in a collaborative multi-agent system. A somewhat similar idea was initially developed in multiple source possibilistic logic [40].

Expressing mutual beliefs. Besides, the Booleanization of possibilistic formulas gives us the capability of embedding them inside other possibilistic formulas, and then to express mutual agent beliefs. Indeed, since (p, α) is semantically interpreted as $N(p) \geq \alpha$, a possibilistic formula can be manipulated as a formula that is true (if $N(p) \geq \alpha$) or false (if $N(p) < \alpha$). Then possibilistic formulas can be put inside possibilistic formulas, as in, e.g. $((p, \alpha/\mathscr{A}), \beta/\mathscr{B})$ (for expressing that agents in $\mathscr{B}$ believe at least at level β that the agents in $\mathscr{A}$ believe p at least at level α), or be combined with propositional connectives as, e.g. in the following expression $(\neg(p, \mathscr{A}) \vee (q, \gamma/\mathscr{B}), \alpha)$ (stating that it is α-certain that if the agents in $\mathscr{A}$ believe p (at level 1), those in $\mathscr{B}$ believe q at least at level γ). Inference from such nested formulas should be handled as a two-layer process (assuming for simplicity that nestedness is not iterated, i.e. the insertion of standard possibilistic logic formulas inside possibilistic logic formulas is not iterated). More precisely, possibilistic resolution may be applied "externally" to the possibilistic logic formulas of the highest level (regarding the possibilistic logic formulas inside (if any) as classical formulas), or "internally" to the possibilistic logic formulas inside, once the "context" has been properly uniformized (by weakening) as illustrated on the two following examples.

Suppose we have the two following formulas:

$$((p, \alpha/\mathscr{A}'), \rho/\mathscr{C})$$

$$(\neg(p, \beta/\mathscr{A}) \vee (q, \gamma/\mathscr{B}), \delta/\mathscr{D})$$

Assume $\alpha > \beta$ and $\mathscr{A}' \supset \mathscr{A}$. Then from the first premise, we get by weakening

$$((p, \beta/\mathscr{A}), \rho/\mathscr{C});$$

then by "external" resolution with the second expression, we obtain

$$((q, \gamma/\mathscr{B}), min(\rho, \delta)/\mathscr{C} \cap \mathscr{D}).$$

Suppose now we have

$$((p, \alpha/\mathscr{A}'), \rho/\mathscr{C})$$

$$((\neg p \vee q, \beta/\mathscr{B}), \delta/\mathscr{D})$$

Assume $\rho > \delta$ and $\mathscr{C} \supset \mathscr{D}$. By weakening, we get

$$((p, \alpha/\mathscr{A}'), \delta/\mathscr{D});$$

by inside resolution, we obtain

$$((q, min(\alpha, \beta)/\mathscr{A}' \cap \mathscr{B}), \delta/\mathscr{D}).$$

It is clear that the above weakening steps can be always applied by taking the minimum of the certainty levels, and the intersection of the sets of agents, even if the first statement does not involve higher certainty levels and larger sets of agents. Mind that getting an empty set of agents after intersection makes the result trivial.

Semantical issues and ignorance. The semantics of a possibilistic formula of the form (p,α), where p can be true or false only, is given by the possibility distribution $\pi_{(p,\alpha)}$ (see Section 2.1). The semantics of $(p,\alpha/\mathscr{A})$, where $\mathscr{A} \subseteq \mathscr{U}$ is given by a collection of possibility distributions $\pi^a_{(p,\alpha)}$, where $a \in \mathscr{U}$ (namely, $\pi^a_{(p,\alpha)} = \pi_{(p,\alpha)}$ if $a \in \mathscr{A}$, and $\pi^a_{(p,\alpha)} = 1$ if $a \notin \mathscr{A}$). The semantics of a formula of the form $((p,\alpha),\beta)$, viewing (p,α) as a true or false statement, will be given in terms of a possibility distribution over the possibility distributions π such that $\pi \leq \pi_{(p,\alpha)}$ (that makes $N(p) \geq \alpha$ true) and the other possibility distributions, with respective weights 1 and $1-\beta$. This may be then reduced to one possibility distribution corresponding to the semantics of $(p,min(\alpha,\beta))$, via the disjunctive weighted aggregation $max(min(\pi_{(p,\alpha)},1),min(1,1-\beta))$, which expresses that either it is the case that $N(p) \geq \alpha$ with a possibility level equal to 1, or one knows nothing with possibility $1-\beta$ [51]. Similarly, the semantics of $((p,\alpha/\mathscr{A}),\beta/\mathscr{B})$, is obtained by associating, to each agent $b \in \mathscr{B}$, a possibility distribution over a set of possibility distributions π^a (such that $\pi^a \leq \pi^a_{(p,\alpha)}$ or not) for each $a \in \mathscr{U}$. In case $a = b$, the above reduction may be applied.

Mind that while the formula $((p,1/\mathscr{A}),\alpha/\mathscr{A})$ may be regarded as equivalent to $(p,\alpha/\mathscr{A})$, the formula $(\neg(p,1/\mathscr{B}),\alpha/\mathscr{A})$ should not be confused with formula $((\neg p,1/\mathscr{B}),\alpha/\mathscr{A})$. The former focuses on the eventuality that $N(p) < 1$ for agents in $\mathscr{B}$ according to agents in $\mathscr{A}$ (with certainty α), while the latter is about the belief of agents in $\mathscr{A}$ (with certainty α) that $N(\neg p) = 1$ for agents in $\mathscr{B}$. Clearly, $N(\neg p) = 1$ which expresses the complete certainty that p is false, is much stronger than $N(p) < 1$ that expresses the absence of complete certainty that p is true, thus opening the possibility that p is false (since equivalently $\Pi(\neg p) > 0$). Such a difference is exemplified by the two following valid inferences :

$$(\neg(p,1),\alpha);((p,1) \vee q,\beta) \models (q,min(\alpha,\beta)).$$

$$(\neg p,\alpha);(p \vee q,\beta) \models (q,min(\alpha,\beta)).$$

Although they look quite similar, their respective meaning is different. Indeed the second premise of the first one expresses that if one is not fully certain of p then q is true is at least β-certain, while the corresponding premise in the second rule requires that it is at least β-certain that if p is false then q is true. Clearly, p is false entails that it is not fully certain that p is true, but the converse is wrong.

The two above inferences are instances of the possibilistic resolution rule. However, other inference rules may be introduced, such as

$$(\neg(p,1),\alpha);(p \vee q,\beta) \models (\neg(\neg q,1),min(\alpha,\beta)),$$

which means that if it is α-certain that it is possible that p be false (since $\neg(p,1)$ means $\Pi(\neg p) > 0$), and one is β-certain that p or q holds, then it is $min(\alpha,\beta)$-certain that it is possible that q be true. Thus, through the negation of standard (necessity-based) possibilistic logic formulas of the form $\neg(p,1-\alpha)$, which are semantically equivalent to $\Pi(\neg p) > \alpha$), one captures something close to the alleged ignorance discussed in the final Remark of Section 2.1, which is expressed by constraints of the form $\Pi(\neg p) \geq \alpha$.

4 Representation of Preferences

The possibilistic logic setting applies to the representation of both knowledge and preferences. In case of preferences, the level α associated to formula p in (p,α) should be understood as a priority (rather than a certainty level). Thus, a piece of preference such as "I prefer a to b and b to c" (where a, b, c are not mutually exclusive) can be represented by the possibilistic base $B = \{(a \vee b \vee c, 1), (a \vee b, 1-\gamma), (a, 1-\beta)\}$ with $\gamma < \beta < 1$, by translating the preference into a set of more or less imperative goals. Namely, B states that a is somewhat imperative, that $a \vee b$ is more imperative, and that $a \vee b \vee c$ is still more imperative (in fact here compulsory, assuming that my choice is between a, b, and c). Note that the preferences are here expressed negatively: "nothing is possible outside a, b, or c ", "nothing is really possible outside a, or b ", and "nothing is strongly possible outside a ".

The possibilistic base B is associated with the possibility distribution π_B (following the definition of Section 2.1), which rank-orders the alternatives: $\pi_B(abc) = 1$, $\pi_B(a\neg bc) = 1$, $\pi_B(ab\neg c) = 1$, $\pi_B(a\neg b\neg c) = 1$, $\pi_B(\neg abc) = \beta$, $\pi_B(\neg ab\neg c) = \beta$, $\pi_B(\neg a\neg bc) = \gamma$, $\pi_B(\neg a\neg b\neg c) = 0$.

From this possibility distribution, one may compute the associated measure of actual possibility for some events of interest:

$$\Delta(a) = min(\pi_B(abc), \pi_B(a\neg bc), \pi_B(ab\neg c), \pi_B(a\neg b\neg c)) = 1$$

$$\Delta(b) = min(\pi_B(abc), \pi_B(\neg abc), \pi_B(ab\neg c), \pi_B(\neg ab\neg c)) = \beta$$

$$\Delta(c) = min(\pi_B(abc), \pi_B(\neg abc), \pi_B(a\neg bc), \pi_B(\neg a\neg bc)) = \gamma.$$

It gives birth to the positive base

$$P = \{[a,1],[b,\beta],[c,\gamma]\},$$

itself associated with a possibility distribution as defined in 3.1:

$\delta_P(abc) = 1$, $\delta_P(a\neg bc) = 1$, $\delta_P(ab\neg c) = 1$, $\delta_P(a\neg b\neg c) = 1$,
$\delta_P(\neg abc) = \beta$, $\delta_P(\neg ab\neg c) = \beta$, $\delta_P(\neg a\neg bc) = \gamma$, $\delta_P(\neg a\neg b\neg c) = 0$.

It can be observed that $\pi_B = \delta_P$. This shows that the preferences here can be equivalently encoded under the form of the positive base P, or of the negative base

B. Thus, the preferences are here conveniently expressed as a "weighted" disjunction of the three choices a, b and c, stating that a is fully satisfactory, b is less satisfactory, and that c is still less satisfactory. Let us mention the representational equivalence [6] between qualitative choice logic [35, 24] and actual (guaranteed) possibility logic, which can be viewed itself as a DNF-like counterpart of standard (CNF-like) possibilistic logic at the representation level. More generally, it has been shown [60] that contextual preferences can be conveniently expressed by possibilistic logic formulas with symbolic weights, and can be favorably compared with the CP-net [34] approach. Such preferences are also of interest for flexible querying [58].

In the above example, the three choices a, b and c might be mutually exclusive or not. If they are not, satisfying both a and b is not better than just satisfying a. Let us briefly examine how preferences with additive structure could be represented in a possibilistic setting. Let us consider a choice situation where it may be possible to satisfy a collection of non-mutually exclusive requirements r_1, ..., r_m, where satisfying r_i alone provides a satisfaction degree equal to ρ_i, and the respective satisfaction degrees are added in case of satisfying several r_i. For instance, if one satisfies r_j and r_k, the satisfaction degree is $\rho_j + \rho_k$. For normalization purpose, we assume, that for all i, $\rho_i > 0$ and that $\rho_1 + ... + \rho_m = 1$. Such a kind of preference will be denoted $r_1(\rho_1) \vee ... \vee r_M(\rho_m)$. How can it be represented in the setting of possibilistic logic?

Let us consider an example: $a(0.5) \vee b(0.3) \vee c(0.2)$. Let $\delta(\omega)$ denotes the satisfaction level of interpretation ω. We have here

$$\delta(\omega) = \Sigma_{i:\omega \models r_i} \rho_i = 1 - \Sigma_{i:\omega \models \neg r_i} \rho_i,$$

i.e., $\delta(abc) = 1$, $\delta(a\neg bc) = 0.7$, $\delta(ab\neg c) = 0.8$, $\delta(a\neg b\neg c) = 0.5$, $\delta(\neg abc) = 0.5$, $\delta(\neg a\neg bc) = 0.2$, $\delta(\neg ab\neg c) = 0.3$, $\delta(\neg a\neg b\neg c) = 0$.

It is worth noticing that the distribution δ can be obtained as the pointwise combination of the three elementary pieces of information $[a, 0.5]$, $[b, 0.3]$, $[c, 0.2]$, expressing that if a, b, c are satisfied respectively, the satisfaction level is at least equal to 0.5, 0.3, 0.2 respectively. Let $\delta_{[a,0.5]}$, $\delta_{[b,0.3]}$, $\delta_{[c,0.2]}$ denote the distributions representing $[a, 0.5]$, $[b, 0.3]$, and $[c, 0.2]$ respectively in the sense of Section 3.1. Then, we have $\delta_{[a,0.5]}(abc) = \delta_{[a,0.5]}(a\neg bc) = \delta_{[a,0.5]}(ab\neg c) = \delta_{[a,0.5]}(a\neg b\neg c) = 0.5$ and $\delta_{[a,0.5]}(\omega) = 0$ otherwise. Similarly, we have $\delta_{[b,0.3]}(abc) = \delta_{[b,0.3]}(ab\neg c) = \delta_{[b,0.3]}(\neg abc) = \delta_{[b,0.3]}(\neg ab\neg c)$
$= 0.3$, and $\delta_{[b,0.3]}(\omega) = 0$ otherwise; $\delta_{[c,0.2]}(abc) = \delta_{[c,0.2]}(a\neg bc) =$
$\delta_{[c,0.2]}(\neg abc) = \delta_{[c,0.2]}(\neg a\neg bc) = 0.2$, and $\delta_{[c,0.2]}(\omega) = 0$ otherwise.

Then one can easily check that

$$\delta(\omega) = \delta_{[a,0.5]}(\omega) \oplus \delta_{[b,0.3]}(\omega) \oplus \delta_{[c,0.2]}(\omega),$$

where $\oplus$ is the associative operation $x \oplus y = \min(1, x + y)$.

The syntactic counterpart of $\delta(\omega)$ can be directly obtained from $[a,0.5]$, $[b,0.3]$, $[c,0.2]$, as

$$K_\Delta = \{[a \wedge b \wedge c, 1], [a \wedge b, 0.8], [a \wedge c, 0.7], [b \wedge c, 0.5], [a, 0.5], [b, 0.3], [c, 0.2]\}.$$

by applying the syntactic fusion operation of section 3.1. It can be shown that the construction illustrated by the above example is general.

5 Handling Uncertainty in Possibilistic Databases

For already a long time, there has been an interest for dealing with incomplete, uncertain or fuzzy data in database management systems [64]. A variety of representation frameworks have been proposed including modal logic-based approaches [59], probabilistic models [76], and possibilistic representations [68]. In the recent years, there has been a renewal of interest motivated by the fact that indeed data may be quite often in practice pervaded with uncertainty. This has led to proposals using either a probabilistic setting [71], [29], [73], or a possibilistic setting [33].

In these works, the available information on the value of some attribute a for an item x is usually represented by a distribution $dis_{\mathscr{A}(x)}$ defined on an attribute domain $D_{\mathscr{A}}$. This may be a probability distribution $p_{\mathscr{A}(x)}$, or a possibility distribution $\pi_{\mathscr{A}(x)}$. The use of a possibility distribution is slightly easier due to a normalization condition that is easier to handle. Indeed, it is only supposed that $\max_{x \in D_{\mathscr{A}}} \pi_{\mathscr{A}(x)} = 1$, while we should have $\sum_{x \in D_{\mathscr{A}}} p_{\mathscr{A}(x)} = 1$, which makes the assessment of probabilities more constrained. However, in both cases, it seems advisable to use a possible worlds semantics that leads to take into account all the possible extensions of the database. This makes the things tricky. For instance, for some basic relational operations such as the join of two relations, it becomes necessary to keep track that some uncertain values should remain equal in any extension. Methods based on lineage have been proposed to handle such problems in the probabilistic case [29], or in the possibilistic case [33]. Their computational cost remain heavy in practice.

It seems however that if we drastically restrict the type of distributions that is allowed, and we use the possibilistic setting, important types of uncertain data could be processed at a more affordable computational cost. Moreover, an additional benefit of the possibilistic setting is an easier elicitation of the possibility degrees. In the following, we explain why and outline the main ideas underlying the approach.

In possibility theory, given a possibility distribution π, we can associate to any event A, its possibility $\Pi(A) = \max_{x \in A} \pi(x)$ and its necessity $N(A) = 1 - \Pi(\bar{A}) = \min_{x \notin A} 1 - \pi(x)$, where $\bar{A}$ denotes the opposite event. $N(A)$ represents the certainty of A. Conversely, a piece of information (A, α) encoding the constraint $N(A) \geq \alpha$, and expressing that A is at least certain at the degree α, can be represented by a possibility distribution $\pi_{(A,\alpha)}(x) = \max(A(x), 1 - \alpha)$. This is the basic building block of the semantics of possibilistic logic, as recalled in Subsection 2.1. The idea that we discuss in the following is to only handle pieces of information of the form (A, α), in a relational database framework.

Let us consider an ordinary n-tuple $x = (a_1(x), \ldots, a_n(x))$, where $a_i(x)$ denotes the value of attribute $\mathscr{A}_i$ for the item x. In the following, we first discuss the handling of pieces of uncertain data of the form $((a_1(x), \alpha_1), \ldots, (a_n(x), \alpha_n))$, before considering the more general case of uncertain disjunctive information. First, it is worth noticing that the degree of certainty α_i does not need to be assessed individually for each cell of a relational table. Indeed, a level of uncertainty may be uniformly associated to an attribute if its value is provided by a not fully reliable source, or is subject to change. Then this level of uncertainty will apply to any of the value of this attribute for any item. Besides, if a whole tuple $x = (a_1(x), \ldots, a_n(x))$ is naturally associated with a level of uncertainty α, since for example the tuple is coming from a source having a level of reliability α, this level can be distributed to each component of the tuple, leading to the equivalent uncertain piece of data $((a_1(x), \alpha), \ldots, (a_n(x), \alpha))$. This is justified by the characteristic property of necessity measures, namely, $N(p_1 \wedge \ldots \wedge p_n) = \min_i N(p_i)$, which leads to the equivalence $N(p_1 \wedge \ldots \wedge p_n) \geq \alpha \iff \forall i, N(p_i) \geq \alpha$. However, it is not allowed that the value of the key attribute(s) of the relation to which the tuple belongs be uncertain.

Let us consider a database example with two relations R and S containing uncertain pieces of data. If we look here for the persons who are married and leave in a

R	*Name*	*Married*	*City*
1	John	(yes, α)	$(\text{Toulouse}, \mu)$
2	Mary	$(\text{yes}, 1)$	(Albi, ρ)
3	Peter	(no, β)	$(\text{Toulouse}, \phi)$

S	*City*	*Flea Market*
1	Albi	(yes, γ)
2	Toulouse	(yes, δ)

city with a flea market, we shall retrieve *John* with certainty $\min(\alpha, \mu, \delta)$ and *Mary* with certainty $\min(\rho, \gamma)$. Generally speaking, such databases can be seen as a layered set of classical databases which gather all attribute values whose certainty is at least equal to some threshold, replacing the values that are not sufficiently certain by null values. Then, the answers to a query, whose certainty is at least equal to some threshold are the answers that can be obtained from the classical database gathering the pieces of information whose certainty is greater than this threshold. This is the counterpart of the well-known fact that the consequences from a possibilistic logic base that are least α-certain can be obtained from the classical logic base made of the formulas whose certainty is greater or equal to α.

It seems also possible to accommodate some cases of disjunctive information in this setting. Assume for instance that the third tuple of relation R is now $(\text{Peter}, (\text{no}, \beta), (\text{Albi} \vee \text{Toulouse}, \phi))$. Then, if we look for persons who are not married and leave in a city with a flea market, one should retrieve *Peter* with certainty $\min(\beta, \phi, \gamma, \delta)$. Indeed we have in possibilistic logic that $(\neg\text{Married}, \beta)$ and $(\text{Albi} \vee \text{Toulouse}, \phi)$, $(\neg\text{Albi} \vee \text{Flea Market}, \gamma)$, $(\neg\text{Toulouse} \vee \text{Flea Market}, \delta)$ entail $(\neg\text{Married}, \beta)$ and $(\text{Flea Market}, \min(\phi, \gamma, \delta))$.

The above observations and remarks suggest to further investigate the potentials of a necessity measure-based approach to the handling of uncertain pieces of information, for precisely identifying the different types of queries that are computationally tractable (in particular in case of disjunctive information). Clearly, the limited setting of certainty-qualified information is less expressive than the use of general possibility distributions, but seems to be expressive enough in practice for deserving further studies.

6 Concluding Remarks

The chapter attempts at offering a broad overview of the basic ideas underlying the possibilistic logic setting, through the richness of its representation formats, and suggesting applications to many AI problems, in relation with the representation of epistemic states and their handling when reasoning from and about them. This framework can be compared to other approaches including nonmonotonic logics, Bayesian nets, modal and hybrid logics [30], and Markov logic [72].

Possibilistic logic has been developed in various other directions in the last past years, including possibilistic inductive logic programming. Indeed learning a stratified set of first-order logic rules as an hypothesis in inductive logic programming has been recently shown of interest for learning both rules covering normal cases and more specialized rules that handle more exceptional cases [74]. Let us also point out other applications to decision [43], logic programming [3, 65, 67], to possibilistic influence diagrams [56], to argumentation [36, 1, 66, 4], and to paraconsistent reasoning [38].

Acknowledgements. The author thanks Salem Benferhat, Patrick Bosc and Olivier Pivert for useful remarks and discussions.

References

1. Alsinet, T., Chesñevar, C., Godo, L.: A level-based approach to computing warranted arguments in possibilistic defeasible logic programming. In: Proc. 2nd Inter. Conf. on Computational Models of Argument (COMMA 2008), Toulouse, May 28-30, pp. 1–12. IOS Press, Amsterdam (2008)
2. Alsinet, T., Godo, L.: A complete calculus for possibilistic logic programming with fuzzy propositional variables. In: Proc. 16th Conf. on Uncertainty in Artificial Intelligence (UAI 2000), Stanford, CA, pp. 1–10. Morgan Kaufmann, San Francisco (2000)
3. Alsinet, T., Godo, L., Sandri, S.: Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description. Elec. Notes in Theor. Computer Sci. 66(5) (2002)
4. Amgoud, L., Prade, H.: Explaining qualitative decision under uncertainty by argumentation. In: Yolanda, G., Moo, R. (eds.) Proc. Conf. on Artificial Intelligence (AAAI 2006), Boston, USA, July 16-20, pp. 219–224. AAAI Press, Menlo Park (2006)

5. Ben Amor, N., Benferhat, S., Mellouli, K.: Anytime propagation algorithm for min-based possibilistic graphs. Soft Computing 8, 150–161 (2003)
6. Benferhat, S., Brewka, G., Le Berre, D.: On the relation between qualitative choice logic and possibilistic logic. In: Proc. 10th Inter. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004), Perugia, July 4-9, pp. 951–957 (2004)
7. Benferhat, S., Dubois, D., Garcia, L., Prade, H.: On the transformation between possibilistic logic bases and possibilistic causal networks. Inter. J. of Approximate Reasoning 29, 135–173 (2002)
8. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bridging logical, comparative and graphical possibilistic representation frameworks. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 422–431. Springer, Heidelberg (2001)
9. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Possibilistic merging and distance-based fusion of propositional information. Annals of Mathematics and Artificial Intelligence 34, 217–252 (2002)
10. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Bipolar possibility theory in preference modeling: Representation, fusion and optimal solutions. Information Fusion 7, 135–150 (2006)
11. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Modeling positive and negative information in possibility theory. Int. J. of Intelligent Systems 23, 1094–1118 (2008)
12. Benferhat, S., Dubois, D., Lang, J., Prade, H.: Hypothetical reasoning in possibilistic logic: basic notions and implementation issues. In: Wang, P.Z., Loe, K.F. (eds.) Between Mind and Computer, Fuzzy Science and Engineering, pp. 1–29. World Scientific Publ., Singapore (1994)
13. Benferhat, S., Dubois, D., Prade, H.: Practical handling of exception-tainted rules and independence information in possibilistic logic. Applied Intelligence 9, 101–127 (1998)
14. Benferhat, S., Dubois, D., Prade, H.: Towards a possibilistic logic handling of preferences. Applied Intelligence 14, 303–317 (2001)
15. Benferhat, S., Dubois, D., Prade, H.: Nonmonotonic reasoning, conditional objects and possibility theory. Artificial Intelligence 92, 259–276 (1997)
16. Benferhat, S., Dubois, D., Prade, H.: From semantic to syntactic approaches to information combination in possibilistic logic. In: Bouchon-Meunier, B. (ed.) Aggregation and Fusion of Imperfect Information, pp. 141–161. Physica-Verlag, Heidelberg (1998)
17. Benferhat, S., Dubois, D., Prade, H.: An overview of inconsistency-tolerant inferences in prioritized knowledge bases. In: Dubois, D., Klement, E.P., Prade, H. (eds.) Fuzzy Sets, Logic and Reasoning about Knowledge. Applied Logic Series, vol. 15, pp. 395–417. Kluwer, Dordrecht (1999)
18. Benferhat, S., Dubois, D., Prade, H., Williams, M.A.: A practical approach to revising prioritized knowledge bases. Studia Logica 70, 105–130 (2002)
19. Benferhat, S., Kaci, S.: Logical representation and fusion of prioritized information based on guaranteed possibility measures: Application to the distance-based merging of classical bases. Artificial Intelligence 148, 291–333 (2003)
20. Benferhat, S., Khellaf, F., Mokhtari, A.: Product-based causal networks and quantitative possibilistic bases. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 13, 469–493 (2005)
21. Benferhat, S., Lagrue, S., Papini, O.: Reasoning with partially ordered information in a possibilistic framework. Fuzzy Sets and Systems 144, 25–41 (2004)
22. Benferhat, S., Prade, H.: Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic. In: Proc. of the 9th Inter. Joint Conf. on Artificiel Intelligence (IJCAI 2005), Edinburgh, Scotland, pp. 1281–1286 (2005)

23. Benferhat, S., Prade, H.: Compiling possibilistic knowledge bases. In: Proc. of the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 337–341. IOS Press, Riva del Gardia (2006)
24. Benferhat, S., Sedki, K.: Two alternatives for handling preferences in qualitative choice logic. Fuzzy Sets and Systems 159, 1889–1912 (2008)
25. Benferhat, S., Smaoui, S.: Possibilistic causal networks for handling interventions: A new propagation algorithm. In: Proc. 22nd AAAI Conf. on Artificial Intelligence (AAAI 2007), Vancouver, Canada, July 22-26, pp. 373–378. AAAI Press, Menlo Park (2007)
26. Benferhat, S., Smaoui, S.: Hybrid possibilistic networks. Inter. J. of Approximate Reasoning 44, 224–243 (2007)
27. Benferhat, S., Titouna, F.: Min-based fusion of possibilistic networks. In: Proc. 4th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2005), Aachen, pp. 553–558 (2005)
28. Benferhat, S., Titouna, F.: Fusion and normalization of quantitative possibilistic networks. Applied Intelligence (2008) (to appear)
29. Benjelloun, O., Das Sarma, A., Halevy, A., Theobald, M., Widom, J.: Databases with uncertainty and lineage. The VLDB Journal 17, 243–264 (2008)
30. Blackburn, P.: Representation, reasoning, and relational structures: a hybrid logic manifesto. Logic Journal of the IGPL 8, 339–625 (2000)
31. Boldrin, L.: A substructural connective for possibilistic logic. In: Froidevaux, C., Kohlas, J. (eds.) ECSQARU 1995. LNCS, vol. 946, pp. 60–68. Springer, Heidelberg (1995)
32. Bonnefon, J.F., Da Silva Neves, R., Dubois, D., Prade, H.: Predicting causality ascriptions from background knowledge: Model and experimental validation. International Journal of Approximate Reasoning 48, 752–765 (2008)
33. Bosc, P., Pivert, O.: About projection-selection-join queries addressed to possibilistic relational databases. IEEE Trans. on Fuzzy Systems 13, 124–139 (2005)
34. Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., Poole, D.: CPnets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research 21, 135–191 (2004)
35. Brewka, G., Benferhat, S., Le Berre, D.: Qualitative choice logic. In: Proc. 8th Inter. Conf. on Knowledge Representation and Reasoning, KR 2002, Toulouse, pp. 158–169 (2002); Final version in Artificial Intelligence 157, 203–237 (2004)
36. Chesñevar, C.I., Simari, G.R., Godo, L., Alsinet, T.: Argument-based expansion operators in possibilistic defeasible logic programming: Characterization and logical properties. In: Godo, L. (ed.) ECSQARU 2005. LNCS, vol. 3571, pp. 353–365. Springer, Heidelberg (2005)
37. Dubois, D., Hajek, P., Prade, H.: Knowledge-driven versus data-driven logics. Journal of Logic, Language, and Information 9, 65–89 (2000)
38. Dubois, D., Konieczny, S., Prade, H.: Quasi-possibilistic logic and its measures of information and conflict. Fundamenta Informaticae 57, 101–125 (2003)
39. Dubois, D., Lang, J., Prade, H.: Timed possibilistic logic. Fundamenta Informaticae 15, 211–234 (1991)
40. Dubois, D., Lang, J., Prade, H.: Dealing with multi-source information in possibilistic logic. In: Neumann, B. (ed.) Proc. of the 10th Europ. Conf. on Artificial Intelligence (ECAI 1992), Vienna, Austria, August 3-7, pp. 38–42 (1992)
41. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D.M., et al. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)

42. Dubois, D., Lang, J., Prade, H.: Automated reasoning using possibilistic logic: semantics, belief revision and variable certainty weights. IEEE Trans. on Data and Knowledge Engineering 6(1), 64–71 (1994)
43. Dubois, D., Le Berre, D., Prade, H., Sabbadin, R.: Using possibilistic logic for modeling qualitative decision: ATMS-based algorithms. Fundamenta Informaticae 37, 1–30 (1999)
44. Dubois, D., Prade, H.: Resolution principles in possibilistic logic. Int. J. of Approximate Reasoning 4(1), 1–21 (1990)
45. Dubois, D., Prade, H.: Epistemic entrenchment and possibilistic logic. Artificial Intelligence 50, 223–239 (1991)
46. Dubois, D., Prade, H.: Combining hypothetical reasoning and plausible inference in possibilistic logic. J. of Multiple Valued Logic 1, 219–239 (1996)
47. Dubois, D., Prade, H.: A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. Int. J. Approx. Reasoning 17, 295–324 (1997)
48. Dubois, D., Prade, H.: Informations bipolaires: Une introduction. Information, Interaction, Intelligence (Revue I3) 3(1), 89–106 (2003)
49. Dubois, D., Prade, H.: Possibilistic logic: a retrospective and prospective view. Fuzzy Sets and Systems 144, 3–23 (2004)
50. Dubois, D., Prade, H.: A bipolar possibilistic representation of knowledge and preferences and Its applications. In: Bloch, I., Petrosino, A., Tettamanzi, A.G.B. (eds.) WILF 2005. LNCS (LNAI), vol. 3849, pp. 1–10. Springer, Heidelberg (2006)
51. Dubois, D., Prade, H.: Toward multiple-agent extensions of possibilistic logic. In: Proc. IEEE Inter. Conf. on Fuzzy Systems (FUZZ-IEEE 2007), London (UK), July 23-26, pp. 187–192 (2007)
52. Dubois, D., Prade, H., Sandri, S.: A possibilistic logic with fuzzy constants and fuzzily restricted quantifiers. In: Martin, T.P., Arcelli-Fontana, F. (eds.) Logic Programming and Soft Computing, pp. 69–90. Research Studies Press, Baldock (1998)
53. Dupin de Saint-Cyr, F., Prade, H.: Possibilistic handling of uncertain default rules with applications to persistence modeling and fuzzy default reasoning. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proc. 10th Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006), Lake District, UK, June 2-5, pp. 440–451. AAAI Press, Menlo Park (2006)
54. Fariñas del Cerro, L., Herzig, A.: A modal analysis of possibility theory. In: Jorrand, P., Kelemen, J. (eds.) FAIR 1991. LNCS, vol. 535, pp. 11–18. Springer, Heidelberg (1991)
55. Gabbay, D.: Labelled Deductive Systems, vol. 1. Oxford University Press, Oxford (1996)
56. Garcia, L., Sabbadin, R.: Possibilistic influence diagrams. In: Brewka, G., Coradeschi, S., Perini, A., Traverso, P. (eds.) Proc. 17th Europ. Conf. on Artificial Intelligence (ECAI 2006), Riva del Garda, Italy, August 29 - September 1, pp. 372–376. IOS Press, Amsterdam (2006)
57. Gardenförs, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. The MIT Press, Cambridge (1988)
58. HadjAli, A., Kaci, S., Prade, H.: Database preferences queries - A possibilistic logic approach with symbolic priorities. In: Hartmann, S., Kern-Isberner, G. (eds.) FoIKS 2008. LNCS, vol. 4932, pp. 291–310. Springer, Heidelberg (2008)
59. Imielinski, T., Lipski Jr., W.: Incomplete information in relational databases. J. of ACM 31, 761–791 (1984)
60. Kaci, S., Prade, H.: Mastering the processing of preferences by using symbolic priorities in possibilistic logic. In: Proc. 18th Europ. Conf. in Artificial Intelligence, Patras, Greece, July 21-25, p. 376, 380 (2008)

61. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44, 167–207 (1990)
62. Lang, J.: Possibilistic logic as a logical framework for min-max discrete optimisation problems and prioritized constraints. In: Jorrand, P., Kelemen, J. (eds.) FAIR 1991. LNCS, vol. 535, pp. 112–126. Springer, Heidelberg (1991)
63. Lang, J.: Possibilistic logic: complexity and algorithms. In: Gabbay, D., Smets, P. (eds.) Algorithms for Uncertainty and Defeasible Reasoning. Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, pp. 179–220. Kluwer Academic Publishers, Dordrecht (2001)
64. Motro, A., Smets, P. (eds.): Uncertainty Management in Information Systems. Kluwer Acad. Publ., Dordrecht (1997)
65. Nicolas, P., Garcia, L., Stephan, I., Lefevre, C.: Possibilistic uncertainty handling for answer set programming. Ann. Math. Artif. Intellig. 47, 139–181 (2006)
66. Nieves, J.C., Cortes, U.: Modality Argumentation Programming. In: Torra, V., Narukawa, Y., Valls, A., Domingo-Ferrer, J. (eds.) MDAI 2006. LNCS, vol. 3885, pp. 295–306. Springer, Heidelberg (2006)
67. Nieves, J.C., Osorio, M., Cortes, U.: Semantics for Possibilistic Disjunctive Programs. In: Baral, C., Brewka, G., Schlipf, J. (eds.) LPNMR 2007. LNCS, vol. 4483, pp. 315–320. Springer, Heidelberg (2007)
68. Prade, H.: Lipski's approach to incomplete information databases restated and generalized in the setting of Zadeh's possibility theory. Inf. Syst. 9, 27–42 (1984)
69. Prade, H.: Handling (un)awareness and related issues in possibilistic logic: A preliminary discussion. In: Dix, J., Hunter, A. (eds.) Proc. 11th International Workshop on Non-Monotonic Reasoning (NMR 2006), Lake District, UK, May 30-June 1, pp. 219–225. Clausthal University of Technology (2006)
70. Prade, H., Serrurier, M.: Bipolar version space learning. Int. J. of Intelligent Systems 23, 1135–1152 (2008)
71. Re, C., Suciu, D.: Managing probabilistic data with MystiQ: The can-do, the could-do, and the can't-do. In: Greco, S., Lukasiewicz, T. (eds.) SUM 2008. LNCS, vol. 5291, pp. 5–18. Springer, Heidelberg (2008)
72. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62, 107–136 (2006)
73. Sen, P., Deshpande, A.: Representing and querying correlated tuples in probabilistic databases. In: Proc. 23rd IEEE Inter. Conf. on Data Engineering (ICDE 2007), Istanbul, Turkey, pp. 596–605 (2007)
74. Serrurier, M., Prade, H.: Introducing possibilistic logic in ILP for dealing with exceptions. Artificial Intelligence 171, 939–950 (2007)
75. Tamargo, L.H., Garcia, A.J., Falappa, M.A., Simari, G.R.: Consistency maintenance of plausible belief bases based on agents credibility. In: Pagnucco, M., Thielscher (eds.) Proc. 12th International Workshop on Non-Monotonic Reasoning (NMR 2008). Technical Report UNSW-CSE-TR-0819, pp. 50–58 (2008)
76. Wong, E.: A statistical approach to incomplete information in database systems. ACM Trans. Database Syst. 7, 470–488 (1982)
77. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)

Part V
Data Management in Medical Domain

Atherosclerosis Risk Assessment Using Rule-Based Approach

Petr Berka and Marie Tomečková

Abstract. A number of calculators that compute the risk of atherosclerosis has been developed and made available on the Internet. They all are based on computing weighted sum of risk factors. We propose instead to use more flexible rule-based approach to estimate this risk. The used rules were created using machine learning methods and further refined by domain expert. Using our rule-based expert system NEST, we built a consultation module AtherEx, that helps (via Internet) a non-expert user to evaluate his atherosclerosis risk.

1 Introduction

Atherosclerosis is a slow, complex disease that typically starts in childhood and often progresses when people grow older. In some people it progresses rapidly, even in their third decade of age. Many scientists think it begins with damage to the innermost layer of the artery. Atherosclerosis involves the slow buildup of deposits of fatty substances, cholesterol, body cellular waste products, calcium, and fibrin (a clotting material in the blood) in the inside lining of an artery. The buildup (referred as a plaque) with the formation of the blood clot (thrombus) on the surface of the plaque can partially or totally block the flow of blood through the artery. If either of these events occurs and blocks the entire artery, a heart attack or stroke or other life-threatening events may result. People with a family history of premature cardiovascular disease (CVD) and with other risk factors of atherosclerosis have

Petr Berka
Dept. of Information and Knowledge Engineering, University of Economics, Prague and Centre of Biomedical Informatics, Institute of Computer Science of the Academy of Sciences, Prague, Czech Republic
e-mail: berka@vse.cz

Marie Tomečková
Centre of Biomedical Informatics, Institute of Computer Science of the Academy of Sciences, Prague, Czech Republic
e-mail: tomeckova@euromise.cz

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 333–350.
springerlink.com

an increased risk of the developing of atherosclerosis. Research shows the benefits of reducing the controllable risk factors for atherosclerosis: high blood cholesterol (level of LDL cholesterol over 100 mg/dL), cigarette smoking and exposure to tobacco smoke, high blood pressure (blood pressure over 140/90 mm Hg), diabetes mellitus, obesity (BMI over 25), physical inactivity. Atherosclerosis-related diseases are a leading cause of death and impairment in the United States, affecting over 60 million people. Additionally, 50% of Americans have levels of cholesterol that place them at high risk for developing coronary artery disease. Similar situation can be observed in other countries. So the education of patients about prevention of atherosclerosis is very important.

The chapter describes step-by-step the process of building a rule-based system for classifying patients according the atherosclerosis risk. We build the set of rules in two steps. At first, we create the initial set of rules from data (described in section 2) using machine learning algorithm KEX (section 3). The machine learning experiments are reported in section 4. Then we refine this set of rules according to suggestions of domain expert and according to further testing (see section 5). Section 6 describes the rule based expert system shell NEST we use to implement the front-end for classification of new patients and section 7 gives a comparison with cardiovascular diseases risk calculators.

2 The STULONG Study

In the early seventies of the twentieth century, a project of extensive epidemiological study of atherosclerosis primary prevention was developed under the name "National Preventive Multifactor Study of Hard Attacks and Strokes" in the former Czechoslovakia. The aims of the study were:

1. to identify atherosclerosis risk factors prevalence in a population considered to be the most endangered by possible atherosclerosis complications (i.e. middle-aged men),
2. to follow the development of these risk factors and their impact on the examined men health, especially with respect to atherosclerotic cardiovascular diseases, (CVD),
3. to study the impact of complex risk factors intervention on their development and cardiovascular morbidity and mortality,
4. 10-12 years into the study, to compare risk factors profile and health of the selected men, who originally did not show any atherosclerosis risk factors with a group of men showing risk factors from the beginning of the study.

Following risk factors were defined at the beginning of the study: arterial hypertension (BP $\geq$ 160/95 mm Hg), cholesterol (level $\geq$ 260mg%) triglycerides (level $\geq$ 200mg%), smoking ($\geq$ 15 cig./day), overweight (Brocka index $>$ 115%), positive family case history. Later, further laboratory examinations were included: blood sugar level, high density cholesterol, low density cholesterol and uric acid.

Table 1 Prevalence of risk factors

Risk factor	n	%
hypercholesteromia	290	34.2
hypertension	287	34.0
smoking	543	63.3
obesity	196	23.0
positive family history	216	25.3

Table 2 Numbers of men in different groups

Group	n	%
normal	277	19.5
risk	861	60.8
pathological	114	8.0
non classifiable	165	11.6
total	1417	100

The study included data of more than 1400 men born between 1926-1937 and living in centre of Prague. The men were divided according to presence of risk factors (RF), overall health conditions and ECG result into following three groups: normal (a group of men showing no RF defined above), risk (group of men with at least one RF defined above - the prevalence of risk factors for this group is shown in Table 1) and pathological (group of men with a manifested cardio-vascular disease). Long-term observation of patients was based on following the men from normal group and risk group (randomly divided into intervened risk group - RGI and control risk group - RGC). The men from the pathological group were excluded from further observation. Table 2 shows the distribution of men in the initial groups.

STULONG is the data set concerning this longitudinal study of the risk factors of the atherosclerosis. Four data files have been created when transforming the collected data into electronic form [1]:

- the file ENTRY contains values of 224 attributes obtained from entry examinations; these attributes are either codes or results of measurements of different variables or results of transformations of the rest of the 244 attributes actually surveyed for each patient,

[1] The study was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital, U nemocnice 2, Prague 2 (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudk, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvárová, DrSc).

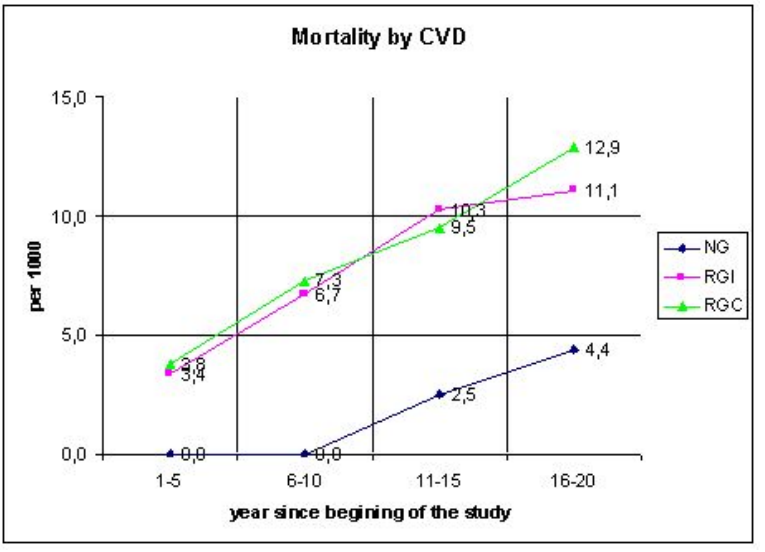


Fig. 1 Mortality in different groups

- risk factors and clinical complications of atherosclerosis have been followed during the control examination for the duration of 20 years. The file CONTROL contains results of observation of 66 attributes recorded during these control examinations (10572 records),
- additional information about health status of 403 men was collected by the postal questionnaire. Resulting values of 62 attributes are stored in the file LETTER,
- there are 5 attributes concerning death of 389 patients who died during the study. Values of these attributes are stored in the file DEATH.

The STULONG data were analyzed using some statistical methods: descriptive statistics, logistic regression and survival analysis. The long term observation shows clear distinctions between the three groups (Figure 1, Figure 2).

Anyway, the domain experts were curious about applying data mining methods to this data. They made therefore the data available for the Discovery Challenge workshops held at the ECML/PKDD Conferences 2002, 2003, 2004. A number of analyses of the STLULONG data has been performed, focused on:

- analytic questions related to the entry examination (what are the relations between social factors, or physical activity, or alcohol consumption and the risk factors),
- analytic questions related to the long-term observation (are there any differences between men of the two risk subgroups RGI, RGC, who came down with the observed cardiovascular diseases in the course of 20 years and those who stayed healthy),

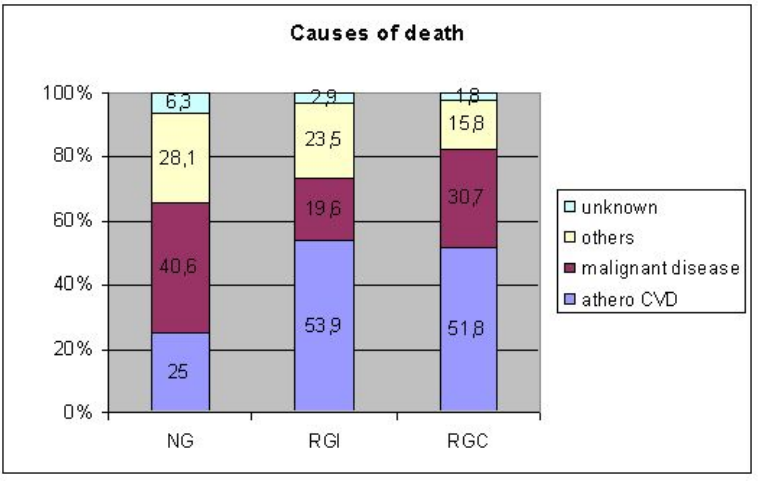


Fig. 2 Causes of death

- analytic questions concerning postal questionnaire,
- analytic questions concerning entry examination, long-term observation and death.

For details see the Discovery Challenge web site (http://sorry.vse.cz/ berka/challenge) or the summarization paper [3]. The STULONG data are also the basics for the work reported in this chapter. We used these data to build a set of rules that can be used to classify persons according to their atherosclerosis risk. To do so, we applied our machine learning algorithm KEX.

3 The KEX Algorithm

KEX [11] performs symbolic empirical multiple concept learning from examples, where the induced concept description is represented as a set of weighted decision rules in the form

$$Ant \Rightarrow C(w),$$

where *Ant* is a conjunction of attribute-value pairs, *C* is the class attribute, and *w* is weight of the rule (from the interval [0,1]).

This set of rules is *compositional*, i.e. more rules can be applied simultaneously to classify an unseen example. This example is thus classified by combining weights of rules that cover the example. To combine weights we use a pseudobayesian (Prospector-like) combination function [4]:

$$w^{\oplus} = w_1 \oplus w_2 = \frac{w_1 \cdot w_2}{w_1 \cdot w_2 + (1 - w_1) \cdot (1 - w_2)}.$$

KEX algorithm

Initialization

1. forall category (attribute-value pair) A add $A \Rightarrow C$ to *OPEN*
2. add empty rule to the rule set *KB*

Main loop

while *OPEN* is not empty do

1. **select** the first implication $Ant \Rightarrow C$ from *OPEN*
2. **test** if this implication significantly improves the set of rules *KB* build so far (we test using the χ^2 test the difference between the rule validity and the result of classification of an example covered by *Ant*) then add it as a new rule to *KB*
3. for all possible categories A
   a. **expand** the implication $Ant \Rightarrow C$ by adding A to *Ant*
   b. **add** $Ant \wedge A \Rightarrow C$ to *OPEN* so that *OPEN* remains ordered according to decreasing frequency of the condition of rules
4. **remove** $Ant \Rightarrow C$ from *OPEN*

Fig. 3 Simplified sketch of the KEX rule learning algorithm

The resulting (soft) classification is done by assigning the example to the class with highest value of $w^{\oplus}$.

The basic idea of the KEX machine learning algorithm is a top-down refinement of set of rules. A simplified description of the algorithm is shown in Figure 3. KEX works in an iterative way, in each iteration testing and expanding an implication $Ant \Rightarrow C(w)$. This process starts with an "empty rule" weighted with the relative frequency of the class C and stops after testing all implications $Ant \Rightarrow C(w)$ created according to the user given values[2] for maximal length of *Ant*, minimal frequency of *Ant*, and minimal validity of $Ant \Rightarrow C(w)$.

During testing, the validity i.e. the conditional probability

$$P(C|Ant) = \frac{||Ant \wedge C||}{||Ant||}$$

of an implication $Ant \Rightarrow C$ is computed (in the formula above, $||A||$ denotes number of examples covered by A). If this validity significantly differs from the composed weight $w^{\oplus}(Ant)$ (value obtained when composing weights of all sub-rules of the implication $Ant \Rightarrow C$), then this implication is added to the knowledge base. To test the difference between validity and composed weight, we use the chi-square goodness-of-fit test. We thus compute the value

[2] KEX offers several standard settings for these parameters. In the *default* setting, all implications with single category in the *Ant* are evaluated. In the *maximal* setting, all implications up to the maximal length of *Ant* are evaluated. In the *strong* setting, only the implications with minimal validity close to 1 are evaluated.

$$\chi^2 = \sum_{i=1}^{T} \frac{(||Ant||_i - ||Ant|| \times w^{\oplus}(Ant))^2}{||Ant|| \times w^{\oplus}(Ant)}$$

and test it against the value of χ^2 distribution at given significance value α (by default, $\alpha = 0.05$) with $T-1$ degrees of freedom. T denotes the number of classes (remind, that KEX performs multiple class learning), and $||Ant||_i$ denotes the number of examples of class i covered by Ant. If the computed value is greater than the value of χ^2 distribution, we add the implication $Ant \Rightarrow C$ as a new rule into the resulting rule set. The weight w of this rule is computed from the validity $P(C|Ant)$ and from the composed weight $w^{\oplus}(Ant)$ in such a way, that

$$w^{\oplus}(Ant) \oplus w = P(C|Ant).$$

We compute the weight w using inverse composing function, so

$$w = \frac{u}{1-u}$$

where

$$u = \frac{\frac{P(C|Ant)}{1-P(C|Ant)}}{\frac{w^{\oplus}(Ant)}{1-w^{\oplus}(Ant)}}$$

When expanding, new implications are created by adding single attribute-value pair to Ant. New implications are stored according to the frequencies of Ant in an ordered list. Thus, for any implication in question during testing, all its sub-implications have been already tested.

We will clarify the step 2 of the main loop of the KEX algorithm using the following simple example. Let the implication in question be $7a11a \Rightarrow 1+$ with the four-fold contingency table shown in Table 3.

Table 3 Contingency table for $7a11a \Rightarrow 1+$

	C	non C
Ant	11	14
non Ant	c	d

So, the validity of this implication is 11/(11+14) = 0.44. Suppose, there are the following rules in the KB, which are applicable for the Ant combination:

```
      ==>  1+ (0.6800)
  11a ==>  1+ (0.2720)
   7a ==>  1+ (0.3052)
```

From these three rules, we can compute the composed weight $w^{\oplus} = 0.6800 \oplus 0.2720 \oplus 0.3052 = 0.2586$. Since this composed weight significantly (according to

the chi-square test) differs from the validity, we must add the implication $7a11a \Rightarrow 1+$ into KB with weight w such, that $w \oplus 0.2586 = 0.44$. So $w = 0.6926$.

When comparing KEX with divide-and-conquer algorithms (like C4.5) or set covering algorithms (like CN2), we can observe, that:

- KEX creates more rules (because KEX does not remove covered examples),
- the set of rules can contain both a rule and its sub-rule (the redundancy of rules is evaluated using statistical test),
- examples are assigned to class with uncertainty.

KEX thus creates more rules which allows different views on the given task and enables to classify new example even if not all values of the input attributes are known.

4 KEX Applied to the STULONG Study

Using KEX we analyzed the data concerning examination of patients when entering the STULONG study - the ENTRY file. These data contain the information about life style, personal history, family history, some laboratory tests and about classification w.r.t atherosclerosis risk (non risk, risky, pathological group). Table 4 shows summary of the attributes in this file. We performed several analyses for different subsets S1 - S4 of input attributes.

1. S1: classification based only on already known risk factors (this rule base should confirm the initial classification of patients in the analyzed data),
2. S2: classification based on attributes concerning life style, personal and family history (but without the laboratory tests),
3. S3: classification based on attributes concerning life style and family history,
4. S4: classification based only on attributes concerning life style.

Table 4 Summary of the ENTRY data table

Group of attributes	no. of attributes
identification data	2
social characteristics	5
physical activity	4
smoking	3
drinking of alcohol	10
sugar, coffee, tea	3
family history	160
personal history	18
chest pain, lower limbs pain, asthma	3
physical examination	8
biochemical examination	3
risk factors	5

Table 5 Rule bases created from the STULONG data

Rule base	no.input attributes	no.rules	overall accuracy	accuracy for non-risk group	accuracy for other groups
S1	13	19	0.87	0.83	0.88
S2	35	35	0.83	0.70	0.86
S3	28	32	0.77	0.63	0.83
S4	18	27	0.73	0.48	0.83

Table 6 Classification accuracies for Weka algorithms

System	overall accuracy	accuracy for non-risk group	accuracy for other groups
C4.5	0.79	0.64	0.87
Random Forrest	0.76	0.64	0.80
JRip	0.80	0.64	0.87
naiveBayes	0.78	0.65	0.84
Bayes net	0.81	0.70	0.86
multilayer perceptron	0.77	0.62	0.82
logistic regression	0.78	0.65	0.83
SVM	0.81	0.67	0.86
k-NN	0.70	0.50	0.77

The classification accuracies (computed using 10 fold cross-validation) of the rule bases resulting from these analyses are summarized in table 5. As a final output from this first (machine learning) step of building the knowledge base, we selected the result of the second type of analyses. The reason for this choice was twofold: the rules have reasonable high classification accuracy and they do not use any ”special” attributes concerning laboratory tests.

Table 7 shows the created rules. Notice, that the rules are created only for the patients belonging to non risky group, as the rules for patients belonging to risky or pathological groups are complementary to these rules (there is always a pair of rules with the same *Ant*, one rule for each of classes ”no-risk” and ”others”, and the weights of these two rules sums up to 1). If the weight of a shown rule exceeds 0.5, then this rule contributes to classification to the class ”non risky group”, otherwise it contributes to classification to the class ”other groups”.

Similar classification accuracies can be reached by other algorithms as well. Table 6 shows the classification results for the S2 subset obtained by some algorithms from the Weka system. Again, the testing is based on 10 fold cross-validation. We applied a variety of methods including decision trees (C4.5, random forest), decision rules (JRip), naive Bayes, bayesian network, multilayer perceptron, logistic

Table 7 Rules created by KEX

no.	rule weight
1	⇒ Class(norisk) 0.2889
2	Hypertension(no) ⇒ Class(norisk) 0.5566
3	Ictus-Dead-parents(no) ⇒ Class(norisk) 0.5352
4	Tea(1 to 2 cups/day) ⇒ Class(norisk) 0.5447
5	Bmi(21.000-26.000)) ⇒ Class(norisk) 0.5433
6	Years-smoking(10.000) ⇒ Class(norisk) 0.2659
7	Tea(no) ⇒ Class(norisk) 0.4039
8	Coffee(no) ⇒ Class(norisk) 0.6046
9	Education(apprentice school) ⇒ Class(norisk) 0.4299
10	Education(university) ⇒ Class(norisk) 0.6099
11	Smoking(15 to 20 cig/day) ⇒ Class(norisk) 0.1948
13	Years-smoking(0.000) ⇒ Class(norisk) 0.7206
12	Smoking(non-smoker) ⇒ Class(norisk) 0.7206
14	Smoking(21 and more cig/day) ⇒ Class(norisk) 0.1995
15	Coffee(3 and more cups) ⇒ Class(norisk) 0.3288
16	Myocardial-infarction-Dead-parents(yes) ⇒ Class(norisk) 0.3232
17	Hypertension(yes) ⇒ Class(norisk) 0.0434
18	Ictus-Dead-parents(yes) ⇒ Class(norisk) 0.2591
19	Smoking(5 to 14 cig/day) ⇒ Class(norisk) 0.7071
20	Asthma(grade I.) ⇒ Class(norisk) 0.2909
21	Hypertension-Dead-parents(yes) ⇒ Class(norisk) 0.2652
22	Beer(more than 1 litre/day) ⇒ Class(norisk) 0.3282
23	Education(basic school) ⇒ Class(norisk) 0.3066
24	Physical-activity-after-job(great) ⇒ Class(norisk) 0.6196
25	Marital-status(divorced) ⇒ Class(norisk) 0.3695
26	Physical-activity-in-job(carries heavy loads) ⇒ Class(norisk) 0.3507
27	Bmi(31.000-46.000) ⇒ Class(norisk) 0.3745
28	Angina-pectoris-Dead-parents(yes) ⇒ Class(norisk) 0.2007
29	Chest pain(angina pectoris) ⇒ Class(norisk) 0.0233
30	Years-smoking(8.000) ⇒ Class(norisk) 0.7111
31	Smoking(1 to 4 cig/day) ⇒ Class(norisk) 0.7547
32	Myocardial-infarction(yes) ⇒ Class(norisk) 0.1333
33	Diabetes(yes) ⇒ Class(norisk) 0.0782
34	Lower-limbs-pain(claudication) ⇒ Class(norisk) 0.1333
35	Transport-to-work(by bike) ⇒ Class(norisk) 0.8807

regression, SVM, and instance based methods (3-NN). As can be seen from the table, the results are roughly comparable with k-NN slightly worse than the other algorithms.

The fact, that KEX performs soft classification allows us to analyze the relation between the weight of class assigned to an example and the correct (according to the testing data) classification. As expected, the higher the weights assigned to the class, the more reliable results we get. But, of course, if we classify only examples

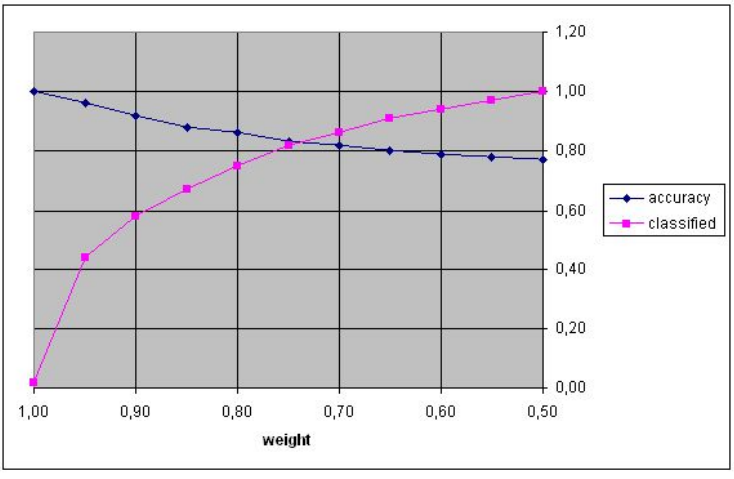


Fig. 4 KEX reliability

weight of which exceeds some threshold, then with increasing threshold the number of classified examples decreases. This is illustrated on Figure 4.

5 Rule Base Modifications

The set of rules obtained using KEX has been revised by the domain expert who suggested following improvements:

- add the attribute ”total cholesterol” and respective rules,
- add rules for remaining values of an attribute, if at least one value of this attribute occur in rules obtained from data,
- use the goals ”no risk”, ”low risk”, ”medium risk” and ”high risk” instead of original groups taken from data.

Further testing has then been performed on new patients’ data currently collected for the purpose of evaluating the Minimal Data Model for Cardiology. This model has been developed in cooperation between the European Centre for Medical Informatics, Statistics and Epidemiology in Prague (EuroMISE Centre), General University Hospital in Prague and Municipal Hospital in Caslav. We obtained the total accuracy 0.70, the accuracy for normal group 0.54 and the accuracy for other groups 0.88 in these tests. The most severe errors (that cause the relative small accuracy for normal group) were misclassifications of patients with high risk as normal ones. When analyzing these errors we found, that this usually happened for patients that were out of the scope of the original STULONG study (they were either too old or too young). After taking this into account by adding new rules (considering the age of the patients), we further improved the classification accuracy.

6 Implementation Using NEST

To allow user friendly access to consultations with the rule base, we use our rule-based expert system shell NEST [1]. This system follows the compositional paradigm of the early expert systems like MYCIN [13] and PROSPECTOR [4], but in its design we attempted to partially overcome the problem that represented the most severe hindrance to compositional system deployment: limited expressiveness of proposition-rule networks for real-world modeling purposes.

6.1 Basic Principles of NEST

NEST uses attributes and propositions, rules, integrity constraints and contexts to express the task-specific (domain) knowledge.

Four types of attributes can be used in the system: binary, single nominal, multiple nominal, and numeric. According to the type of attribute, the derived propositions correspond to:

- values `True` and `False` for a *binary* attribute,
- each value for a *nominal* attribute (the difference between single and multiple nominal attribute is apparent only when answering the question about value of the attribute - single attribute can have only one value, multiple attribute can have more values),
- fuzzy intervals for a *numeric* attribute. Each interval is defined using four points; fuzzy lower bound (FL), crisp lower bound (CL), crisp upper bound (CU), fuzzy upper bound (FU). These values need not to be distinct; this allows to create rectangular, trapezoidal and triangular fuzzy intervals.

Rules are defined in the form

$$condition \Rightarrow conclusion(weight),$$

where *condition* is disjunctive form (disjunction of conjunctions) of literals (propositions or their negations), *conclusion* is a list of literals, and *weight* from the interval $[-1,1]$ expresses the uncertainty of the rule. We distinguish three types of rules:

- *compositional* - each literal in conclusion has a weight which expresses the uncertainty of the conclusion if the condition holds with certainty. The term compositional denotes the fact, that to evaluate the weight of a proposition, **all** rules with this proposition in the conclusion are evaluated and combined.
- *apriori* - compositional rules without condition; these rules can be used to assign implicit weights to goals or intermediate propositions,
- *logical* - non-compositional rules without weights; only these rules can infer the conclusion with the weight `true` or `false`. **One** activated rule thus fully evaluates the proposition in conclusion.

When comparing this syntax with the syntax of rules created by KEX, we can see that the rules in KEX are subsumed by rules in NEST. This allows us to easily import rules from KEX to NEST.

During consultation, the system uses the rules to compute weights of goals from the weights of questions. This is accomplished by (1) selecting relevant rule during current state of consultation, and (2) applying the selected rule to infer the weight of its conclusion.

1. The selection of relevant rule can be done using either backward or forward chaining. The actual direction is determined by the user when selecting the consultation mode (see later).
2. For rules with weights (compositional and apriori ones), the system combines uncertain contributions of rules using compositional approach described bellow. For rules without weights, the system uses non-compositional approach based on (crisp) modus ponens – to evaluate the weight of a conclusion, and (crisp) disjunction – to evaluate a set of rules with the same conclusion.

Uncertainty processing in NEST is based on the algebraic theory of P. Hájek [6]. This theory generalizes the methods of uncertainty processing used in the early expert systems like MYCIN and PROSPECTOR. Algebraic theory assumes that the knowledge base is created by a set of rules in the form shown above. During a consultation, all relevant rules are evaluated by combining their weights with the weights of conditions. Weights of questions are obtained from the user, weights of all other propositions are computed by the inference mechanism. Five combination functions are defined to process the uncertainty in such knowledge base:

1. $NEG(w)$ - to compute the weight of negation of a proposition,
2. $CONJ(w_1, w_2, ..., w_n)$ - to compute the weight of conjunction of literals,
3. $DISJ(w_1, w_2, ..., w_n)$ - to compute the weight of disjunction of literals,
4. $CTR(a, w)$ - to compute the contribution of the rule to the weight of the conclusion (this is computed from the weight of the rule w and the weight of the condition a),
5. $GLOB(w'_1, w'_2, ..., w'_n)$ - to compose the contributions of more rules with the same conclusion.

Algebraic theory defines a set of axioms, the combination functions must fulfill. Different sets of combination functions can thus be implemented. NEST uses three such sets:

1. *standard*, where the combination functions are based on classical approach of MYCIN and PROSPECTOR,
2. *logical*, where the combination functions are based on an application of the completeness theorem for Lukasiewicz's many-valued logic; the modus ponens inference rule in this logic assigns the degree of truth to a conclusion of a rule according to the formula

$$\frac{\alpha, \alpha \Longrightarrow \beta}{\beta} \quad \left(\frac{x, y}{\max(0, x + y - 1)} \right)$$

3. *neural*, where the combination functions are inspired by active dynamics of artificial neural networks.

The respective formulas for computing values of CTR and $GLOB$ for these different approaches are shown in table 8, the values for the remaining functions are the same, namely:

- $NEG(w) = -w$
- $CONJ(w_1, w_2) = min(w_1, w_2)$
- $DISJ(w_1, w_2) = max(w_1, w_2)$

Table 8 Functions CTR and $GLOB$ for different inference mechanisms

inference mechanism	$CTR(a,w)$ for $a > 0$	$GLOB(w'_1, w'_2, ..., w'_n)$
standard	$a \cdot w$	$\frac{w'_1 + w'_2}{1 + w'_1 \cdot w'_2}$
logical	$sign(w) \cdot \max(0, a + \lvert w \rvert - 1)$	$\min(1, \sum_{w'>0} w') - \min(1, \sum_{w'<0} \lvert w' \rvert)$
neural	$a \cdot w$	$\min(1, \max(-1, \sum_{i=1}^{n} w'_i))$

Again, the method of combining rules in KEX is a special case of uncertainty processing available in NEST.

During consultation with the system, the user answers the questions concerning the input attributes. According to the type of attribute, the user gives the weight (for binary attributes), the value and its weight (for single nominal attributes), list of values and their weights (for multiple nominal attributes), or the value (for numeric attributes).

Questions not answered during consultation are not known. Two different notions of this answer are introduced in NEST. First notion, "irrelevant", is expressed by the weight 0; this weight will prevent a rule having either a proposition or its negation in conditional part from being applied. Second notion, "unknown", is expressed by the weight interval $[-1, 1]$; this weight interval is interpreted as "any weight". Uncertainty processing has thus been extended to work with intervals of weights. The idea behind is to take into account all values from the interval in parallel. Due to the monotonicity of the combination functions, this can be done by taking into account the boundaries of intervals only.

Two versions of NEST have been implemented, stand-alone and web-based client server one. Stand-alone version is implemented as one program running under MS Windows, client-server version is implemented as web server that uses any web browser as the client. In the client-server version, different page layouts can be defined for different knowledge bases to customize the system.

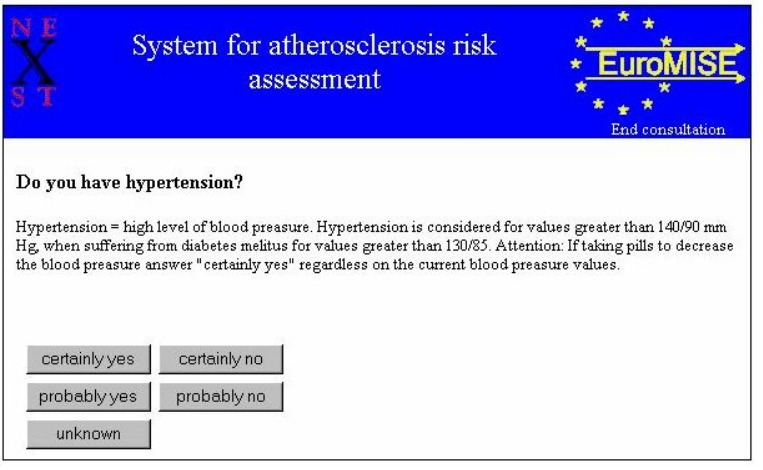


Fig. 5 Screenshot of the system

6.2 Implemented Consultation Module

We used the client/server version of NEST to implement the consultation module (we call AtherEx). To make the consultation module user-friendly for users who are neither experts in expert systems, nor experts in medicine, we built a front-end, that hides the details about inference and uncertainty processing. The system works in dialogue mode, showing one question on a single page. The questions (their number is 22) are grouped into following groups:

- questions concerning personal data (marital status, education, body mass index, cholesterol),
- questions concerning life style (smoking, physical activity in job and in leisure time, consumption of alcohol, coffee or tea),
- questions concerning personal history (hypertension, diabetes, myocardial infarction, stroke),
- questions concerning family history (hypertension, diabetes, myocardial infarction, stroke, angina pectoris for parents).

The user can answer the questions using predefined values (buttons) ”certainly yes”, ”maybe yes”, ”maybe no”, ”certainly no”, or ”unknown” (i.e. ”any value”). Figure 4 shows as an example the question about hypertension.

7 Comparison with Atherosclerosis Risk Calculators

A number of calculators that compute the risk of atherosclerosis, cardio-vascular disease (CVD) or myocardial infarction (IM) has been developed and made available on the Internet. These systems usually ask questions about life style (typically

Table 9 Calculators of CVD Risk

system	knowledge source	no. of questions	suitable for	results
NCEP ATP III	ATP III Guidelines	11 + 2	all patients	CVD risk in 10 years
Risk assesment tool	Framingham study	4 + 2	all patients	IM risk in 10 years
Framingham Risk Assessment	Framingham study	5 + 2	all patients	IM risk in 10 years
PROCAM Risk Calculator	PROCAM study	6 + 3	middle-aged men	IM risk in 10 years
PROCAM Risk Score	PROCAM study	7 + 4	middle-aged men	IM risk or death on CVD in 10 years
PROCAM Neural Net	PROCAM study	11 + 5	middle-aged men	IM risk in 10 years
Heart Score	European Society of Cardiology	4 + 2	middle-aged patients	death on CVD in 10 years

about smoking habits) and about results of examination and laboratory tests (typically about blood pressure and cholesterol level) and then compute a risk (in percentage) that given person will suffer from cardiovascular disease (CVD) in 10 years. The computation has a form of weighted sum of used risk factors.

The exact formula is based on different knowledge sources: the NCEP ATP III system [8] is based on the Adult Treatment Program III guidelines issued by the US National Heart, Lunge and Blood Institute (NHLBI) within the National Cholesterol Education Program (NCEP), the Risk Assessment Tool [12] also from NHLBI is based on the data collected within the Framingham Heart Study performed in the U.S.A. - the same study is behind the Framingham Risk Assessment calculator [5]. The Prospective Cardiovascular Muenster Study (PROCAM) is the background for the PROCAM Risk calculator [9] and the PROCAM Risk score [10] systems developed in Germany. The Heart Score system [7] developed by the European Society of Cardiology is based on data from 12 European cohort studies covering a wide geographic spread of countries at different levels of cardiovascular risks. Table 9 summarizes some further information about these systems (the column no. of questions gives the number of questions on life style (first number) and the number of lab. tests (second number)).

We empirically compared the Risk Assessment Tool developed by NIH [12] (further reffered to as NIH), PROCAM Risk calculator [9] (further reffered to as ProCam) and Heart Score [7] (further reffered to as Heart) on the data used to test our rule-based approach (see section 5). Figure 6 shows the results of evaluating CVD risk of the testing set of patients. We can observe, that Heart Score systematically assigns lower values, while the values assigned by the remaining two systems were roughly the same.

To compare results of the CVD risk calculators with our system as well with the opinion of domain expert, we turned the numerical risk score into binary values Risk

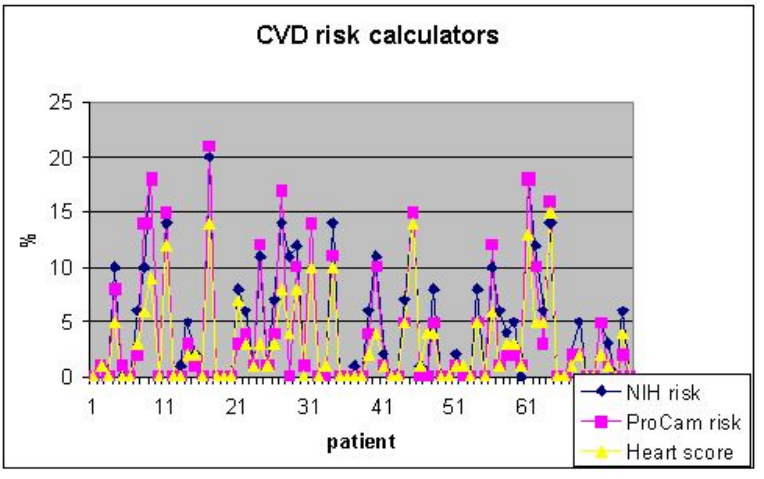


Fig. 6 CVD risk computed by different risc calculators

Table 10 Classification accuracy

system	overall accuracy	accuracy Risk	accuracy NoRisk
NIH	0.76	0.84	0.63
ProCam	0.69	0.95	0.52
Heart	0.67	1.00	0.50
AtherEx	0.70	0.88	0.54

or NoRisk respectively. The threshold 5% was taken from the Heart Score system. This allows us to express the performance of the calculators in the terms of classification accuracy (Table 10). The NIH system outperforms all the other systems in both overall accuracy and in accuracy for non risk patients (thus making less errors by classifying risky patients as non risky ones), our approach was comparable with the remaining two calculators. Anyway, none of the systems makes reliable classifications of non risky patients and an interesting trade-of between classification accuracies of both groups can be observed.

8 Conclusions

The described system AtherEx should help non-expert users to determine their atherosclerosis risk. We acquired knowledge for this system in two steps: at first, we created an initial set of rules from data using machine learning techniques. In the next step, the set of rules was refined according to suggestions of domain expert and results of further experimental evaluation.

We see the main advantages of our system (when compared with the CVD risk calculators) in its ability to infer a conclusion from incomplete and/or uncertain input information (the user need not to answer all questions). This is especially important for an non-expert user, who do not know the results of his laboratory tests.

Our experiments have shown that the information about life style can be used instead of laboratory tests. AtherEx is now tested by domain expert and other physicians from the EuroMISE center in Prague with similar results (system is available at http://www.euromise.cz). Anyway, the resulting classification does not substitute a diagnosis done by a specialist, it is rather a recommendation that should by consulted with a physician.

Acknowledgements. The work reported in this chapter is supported by the grant MSMT 1M06014 (from the Ministry of Education of the Czech Republic) and the grant GACR 201/08/0802 (from the Grant Agency of the Czech Republic).

References

1. Berka, P., Laš, V., Svátek, V.: NEST: re-engineering the compositional approach to rule-based inference. Neural Network World, 367–379 (5/04, 2004)
2. Berka, P., Laš, V., Tomečková, M.: AtherEx: an Expert System for Atherosclerosis Risk Assessment. In: Miksch, S., Hunter, J., Keravnou, E.T. (eds.) AIME 2005. LNCS (LNAI), vol. 3581, pp. 79–88. Springer, Heidelberg (2005)
3. Berka, P., Rauch, J., Tomečková, M.: Lessons Learned from the ECML/PKDD Discovery Challenge on the Atherosclerosis Risk Factors Data. Computing and Informatics 26(3), 329–344 (2007)
4. Duda, R.O., Gasching, J.E.: Model Design in the Prospector Consultant System for Mineral Exploration. In: Michie, D. (ed.) Expert Systems in the Micro Electronic Age. Edinburgh University Press, UK (1979)
5. Framingham Risk Assessment Calculator, chd.uni-muenster.de/Framingham.php
6. Hájek, P.: Combining Functions for Certainty Factors in Consulting Systems. Int. J. Man-Machine Studies 22 (1985)
7. Heart Score, http://www.escardio.org/knowledge/decision_tools/heartscore
8. NCEP APT III system, http://www.incirculation.net/index.asp?did=23849
9. PROCAM Risk Calculator, http://www.chd-taskforce.de/calculator/calculator.htm
10. PROCAM Risk Score, http://www.chd-taskforce.de/risk-english.htm
11. Rauch, J.,, Berka, P.: Knowledge Discovery in Financial Data - a Case Study. Neural Network World 7(4-5) (1997)
12. Risk Assessment Tool, http://www.nhlbi.nih.gov/guidelines/cholesterol/risk_tbl.htm
13. Shortliffe, E.H.: Computer-Based Medical Consultations: MYCIN. Elsevier, New York (1976)

Interpretation of Imprecision in Medical Data

Mila Kwiatkowska, Peter Riben, and Krzysztof Kielan

Abstract. Imprecision is an intrinsic part of all data types and even more so of medical data. In this paper, we revisit the definition of imprecision as well as closely related concepts of incompleteness, uncertainty, inaccuracy, and, in general, imperfection of data. We examine the traditional hierarchical approach to data, information, and knowledge in the context of medical data, which is characterized by heterogeneity, variable granularity and time-dependency. We observe that (1) imprecision has syntactic, semantic, and pragmatic aspects and (2) imprecision has its spectrum from most precise to most imprecise and unknown. We argue that interpretation of imprecision is highly contextual, and, furthermore, that medical data cannot be decoupled from their meanings and their intended usage. To address the contextual interpretation of imprecision, we present a framework for knowledge-based modeling of medical data, which comprises a semiotic approach, a fuzzy-logic approach, and a multidimensional approach.

Keywords: imprecision, medical data, fuzzy logic, semiotics.

1 Introduction

Vagueness, inexactness and imprecision, as well as imperfection of information in general, have been studied for many years in the context of computer-supported decision making, knowledge engineering, and artificial intelligence. Although imprecision is intrinsic to information and knowledge in the real world, oftentimes, the models of reality created for computational purposes are oversimplified, and they represent isolated fragments of the complex systems present in our lives. A simplified representation of reality is often necessary in order to design and

Mila Kwiatkowska
Department of Computing Science, Thompson Rivers University, 900 McGill Road, Kamloops, BC, V2C 5N3, Canada
e-mail: mkwiatkowska@tru.ca

Peter Riben
Inc., 280 Morrisey Pl., Kamloops, BC, V2C 1M5, Canada
e-mail: priben@telus.net

Krzysztof Kielan
North East Lincolnshire Care Trust Plus, Prince Albert Gardens 1, DN31 3HT Grimsby, UK
e-mail: Krystof.Kielan@nelctp.nhs.uk

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 351–369.
springerlink.com

develop feasible information systems. On the other hand, such simplified models may create a false assurance that they are themselves complete and precise and that they reflect a complete and precise reality. This caution is especially important in complex disciplines such as medicine and health care. Often, in medical care, the decisions are made based on subjective, uncertain, multidimensional, and imprecise information. Thus, the computerized models in order to represent real life data, information, and knowledge used in diagnosis, prognosis, and treatment, must represent various forms of imprecision and must provide reasoning methods which tolerate imprecision. As it was emphasized by Zadeh [32] and Parsons [19] imperfections must be studied and accounted for in the models of reality. In this paper, the authors concentrate on the term imprecision, its definition, classification, and interpretation in context of medical data and medical decision making.

This paper is structured as follows. Section 2 surveys various classifications of imprecision, and it makes distinctions between imprecision and other aspects of imperfect information such as uncertainty, incompleteness, inconsistency, and vagueness. Section 3 presents the characteristics and structure of medical data. Section 4 presents a framework for analysis of imprecision in medical data. This framework considers the nature and the sources of imprecision and uses three approaches for modeling and reasoning with imprecise information: fuzzy logic, semiotics, and multidimensional representation.

2 Definition and Classification of Imprecision

For many years, mathematicians, linguists, and philosophers have studied imprecision and have recognized its importance in the modeling of real-world concepts and decision-making processes. They have presented a number of definitions, classifications, and formal representations [15,16,30]. In this section, we limit our discussion to three aspects: the most significant works pertaining to formal representation of imprecision, an overview of different approaches to defining imprecision, and a short description of our interpretation of imprecision.

2.1 Formal Representation of Imprecision

The first step towards creation of a formal mathematical representation of vagueness was the work of Jan Łukasiewicz, who in the 1920's introduced multi-valued logic. Łukasiewicz extended the traditional two-valued logic (values: true and false) to a closed real interval [0,1] representing the possibility that a given value is true or false. While in traditional logic and set theory, an element either belongs to a set or not, in fuzzy set, an element may belong to a set "partially" with some degree of membership. At the same time, Emil Post introduced similar ideas in logics which are more general than two-valued logic. In 1937, Max Black published the paper, "Vagueness: an exercise in logical analysis," in which he introduced "vague sets" and operations. In 1965, Lotfi Zadeh published a paper "Fuzzy sets" [32]. Zadeh introduced the term "fuzzy set," extended the fuzzy set theory, and created fuzzy logic as a new field of study. In 1982, Zdzislaw Pawlak

presented a rough set theory [20] for the mathematical representation of vagueness. Whereas Zadeh's theory is based on a "fuzzification" of quantitative measures, Pawlak's approach is based on a more "qualitative" concept of a set approximation by a pair of sets: the lower approximation (set of concepts that belong to the approximated set) and an upper approximation (set of concepts that probably belong to the approximated set) [20]. These approximations are the positive and negative extensions of a vague concept. However, the fuzzy set theory and the rough set theory are not mutually exclusive; they can be combined together to create a fuzzy-rough set representation. Fuzzy set theory has been used for the representation of imprecision in the fuzzy databases. Other approaches to the explicit representation of imprecision in databases have been used. For example, Barga and Pu extended traditional relational database model to handle quantitative imprecision and proposed an interval-based model for imprecise data [2].

The concept of imprecision has been also studied in the context of imprecise probability which is concerned with mathematical models of chance and uncertainty without sharp numerical probabilities.

2.2 An Overview of Approaches to the Definition of Imprecision

Imprecision has been defined using various approaches and classifications. However, there is no unanimous definition. We organize the many approaches to the definition of imprecision and its classification into five groups as follows:

1. **Conceptual vagueness and imprecision:** Skala [28] distinguishes between two sources of imprecision: conceptual vagueness (fuzziness) and imprecision due to inexact measurement. The vague concepts are represented using Zadeh's fuzzy logic approach. The inexact measurements are modeled by Skala using subjective probabilities.
2. **Imperfect information and imprecision:** Bonnissone and Tong [3] describe the imperfect information using three characteristics: *uncertainty*, *incompleteness*, and *imprecision*. Uncertainty is defined as inherently subjective – an individual estimate of the truth of some fact. On the other hand, incompleteness and imprecision are defined as having a more objective quality. Incompleteness arises from the nonexistence of a value, whereas imprecision arises from the existence of a value which cannot be measured with suitable precision. Furthermore Bonnissone and Tong describe four types of imprecision according to specific categories of values: interval (age between 25 and 30), fuzzy ("quite young"), disjunctive (age is either 25 or 30), and induced (value induced from a negation, for example, "not old"). Bosc and Prade [4] describe imperfection using four categories: *uncertainty*, *imprecision*, *vagueness*, and *inconsistency*. Their classification is somewhat different from classification presented in [3]. Inaccuracy arises from a lack of information about specific value. Imprecision, on the other hand, arises from the granularity of the language and is viewed as having a more subjective quality. Vagueness is a category similar to the fuzzy-valued imprecision in [3]. Inconsistency arises from the existence of two contradicting values.

3. **Inaccuracy and imprecision:** Smets [29] makes a distinction between imprecision (data with errors and data without errors) and data without errors (fuzziness, deficient, vagueness). A measurement may be expressed precisely but may not be accurate. Accuracy is conforming to a standard or a true value. Accuracy is distinguished from precision in this way: a measurement or statement can reflect or represent a true value without detail. The temperature reading of 37°C is accurate, but it is not precise if a more refined thermometer registers the temperature of 37.543°C.
4. **Incompleteness and imprecision:** Medical decisions often involve incomplete data. The records may have missing values for several reasons: limited number of tests required for diagnoses, logical exclusion of not applicable data (e.g., data specific to female gender is omitted from a record of a male patient), intentional omission of sensitive data, lack of information omission, or discontinuation (drop-out) of study. In our discussion, we make a distinction between incomplete or missing data and imprecise data.
5. **Uncertainty and imprecision:** Niskanen [17] makes a clear distinction between imprecision and uncertainty. Uncertainty is associated with probability, whereas imprecision is described as being independent from uncertainty. Niskanen defines imprecision in a broad context of human sciences, and classifies imprecision into three types: *ontological*, *epistemological* and *linguistic* imprecision. Ontological imprecision relates to imprecise object of the reality. Epistemological imprecision relates to imprecision of human (agent's) knowledge about the object. Linguistic imprecision relates to verbal expression, which inherently is imprecise and ambiguous.

2.3 Interpretation of Imprecision

In this subsection, we present two approaches to the term "imprecision." First, we examine the word "imprecision" from a linguistic perspective: its etymology and semantics. Then, we define various aspects of imprecision.

In the Oxford English Dictionary (OED) [18] the noun "imprecision" is defined as "want of precision, inexactness" and the adjective "imprecise" as "not precise." Imprecision, thus, has the opposite meaning to "precision:" -im is an assimilated form of the Latin negative prefix –in and "precision" has its origin in French noun précision (action of cutting off) and Latin praecision-, praecisio (act of cutting off). The OED describes several uses of the word "precision." In the general use, "precision" denotes "the fact, condition, or quality of being precise." In philosophy, "precision" denotes "The action or an act of separating or cutting off" (mental separation). In sciences, especially when referring to measurements, the word "precision" denotes "The degree of refinement in a measurement, calculation, or specification, esp. as represented by the number of digits given." In statistics, the word "precision" is used to describe "the reproducibility or reliability of a measurement or numerical result."

In medical dictionaries, the word "precision" is defined, for example, in two ways. Rothman and Greenland [23] define precision in epidemiology as the reduction of random error in measurement and estimation. Precision can be improved in two ways: by increasing the number of the subjects and by modifying the design

of the study. In the Dictionary of Epidemiology [12], the term precision is defined as "the quality of being sharply defined or stated. One measure of precision it is the number of distinguishable alternatives from which a measurement was selected, sometimes indicated by the number of significant digits in the measurement. Another measure of precision is the standard error measurement, the standard deviation of a series of replicate determinations of the same quantity."

We describe imprecision as a concept with the following characteristics:

1. Imprecision is distinct from incompleteness (absence of value), inaccuracy (value is not close to the "true" value), inconsistency (dissimilar values from several sources), and uncertainty (probability or belief that the value is the "right" value).
2. Imprecision is highly contextual and interpretative, i.e., a statement "high body temperature" may be sufficiently precise in a specific situation or a more precise value such as "40.5°C rectal" is needed. Thus, imprecision is a quality of specific value used in a reference to a concept in a specific context. Often, imprecise values are sufficient, since the precision may be not possible, impractical, expensive, or not needed. Feinstein [10] describes a situation in which, patients with cancer should not be given "precise" prognosis, without considering specific clinical situation.
3. Imprecision is not a binary concept. Each concept, its representation, and its interpretation have certain degree of imprecision, which can be ordered from the lowest level to the highest level. For example, the following values can be ordered in an order of an increasing precision: "high body temperature," "body temperature above 38°C," and "40.5°C rectal."

We define two aspects of imprecision: *qualitative* and *quantitative*. The qualitative imprecision is a result of a vagueness of the concept (e.g., quality of life, health) and the lack of precise measures of the concept (e.g., measures of sleepiness). The quantitative imprecision is a result of a lack of precision in a measurement. We view these aspects of imprecision as *pragmatic* (vagueness of the concept), *semantic* (lack of precise measures), and *syntactic* (lack of precision in a measurement).

3 Medical Data Definition

The word "data" is used by many disciplines in different contexts and with different meanings, and sometimes it is used interchangeably with the word "information." Thus, the definition of the term "data" varies and often hinges upon the distinction or lack of distinction between data and information. Although a universal definition of data would be interesting, our purpose is not to provide a universal definition – such a definition probably is not attainable without oversimplification – but to present important characteristics of data, in particular, medical data. First, we discuss the meaning of "data" from the perspective of database systems. Second, we present the various aspects of medical data.

3.1 General Data Definition

The term "data" carries many meanings from general usage to specialized use in science and computing science. The word "datum" (from Latin past participle of "dare" meaning "to give") and its plural form "data" have been used in English and other languages for hundreds of years. The term has also been used in a more specific meaning "given values" in context of mathematics and engineering, where calculations are performed based on given values. With the emergence of calculating machines (computers), the word data in its scientific meaning "given values" has become a part of computing science language. On the other hand, various forms of permanent computer-readable storage have been invented – from punched Hollerith cards and paper tapes through to file systems on magnetic tapes (in mid 1950s), to integrated indexed files and hierarchical database structures on magnetic disks (in 1960s) and to large multimedia databases stored on mass storage servers (1990s). Furthermore, with the development of the Internet and availability of Web servers, the data are distributed across the Web. This growth in usage of automated storage of data, data retrieval, and processing has been unmatched in history. With higher capacity for storage and developments in digitization techniques, it became possible to store and process new types of data: text data (semi-structured or unstructured documents), spatial data, audio data, images, biomedical signals, and digital-video data. Thus, the data, in context of data management systems, encompass numerous sources of information available to humankind. Moreover, with the development of the Web, we are presented not only with a multitude of data modalities, but also with multitude of data sources. These different types of data have various granularities and abstraction levels. Thus, often it is difficult to distinguish between data and information.

Traditionally, in the context of database and information systems, data have been distinguished from information. While data have been defined as numbers, characters, and recorded facts, information has been defined as structured, processed data used for decision making. In addition, some authors differentiate between raw data and processed data; however, this distinction is relative since processed data could be raw data for another level of analysis. Furthermore, the concept of data has been used as a base for a hierarchical structure built of *data*, *information*, and *knowledge*. This traditional hierarchy has been extended by Ackoff with an addition of two layers: *understanding*, and *wisdom* [1]. Ackoff's hierarchical system is often referred to as DIKW or the Pyramid of Knowledge. At the lowest level, Ackoff defines data as raw data which have no significance beyond their existence, and have no meaning in themselves. At the higher level, Ackoff defines information as data that have been processed and that have meaning. Although the DIKW hierarchy has been widely accepted in the field of database systems, informatics, and in data mining, Fricke recently critically revised the Knowledge Pyramid from the perspective of information science [11]. Fricke pointed out that the data are contextual, thus the data must be described within their context.

3.2 Medical Data

Data are crucial in a day to day medical practice. They are gathered in the form of the patient history, physiological data from a physical examination, biochemical data from tests, records of biosignals, medical images, and epidemiological data. Data are used in diagnosis, treatment, and prognosis, as well as in epidemiological studies, control trials, and medical research in general. The type and quantity of data stored depend on the overall purpose. Thus, medical data are often defined in terms of their functionality – Merriam-Webster's Medical Desk Dictionary defines data as "factual information (as measurements and statistics) used as a basis for reasoning, discussion, or calculation" [13].

We define medical data from the perspective of the data modeling for computer-supported medical decision systems. In accordance with Shortliffe and Barnett [27], we define a medical datum as a single observation about a patient and medical data as a set of multiple observations. A medical datum is composed of four elements: the reference to the patient, the parameter being observed, the value of the parameter, and the time of the observation. Moreover, a medical datum (observation) must record additional information called *modifiers*, such as the type of instrument, the type of measurement, and any additional relevant information. For example, the recording of a blood pressure should also include information about the instrument (manual sphygmomanometer or automatic cuff), placement (left or right arm, leg), patient's position (standing, lying), and information about medication (antihypertensive medication), activity prior to the measurement, food and drink intake (alcohol, caffeinated drinks, etc.).

Thus, we define medical datum (observation) as a tuple:

$$MedicalDatum = < R, P, V, T, \{M\}_{i=1}^{n} >$$

Where R represents a finite set of references to the patients, P represents a finite set of the parameters being measured, V represents a finite set of the values for the parameters, T represents a finite set of time points, and M represents a finite is a set of modifiers. The modifiers are represented by a set of attribute-value pairs.

We define medical data as a set of observations:

$$MedicalData = \{MedicalDatum\}_{i=1}^{n}$$

For example, a single recording of the blood pressure for the patient p_1 is represented by the following tuple: *(p_1, arterial blood pressure, 160/90 systolic/diastolic in mmHg, 2008/02/01 10:00, {instrument = automatic cuff, placement = left arm, position = sitting, food intake = no prior caffeine intake, antihypertensive medication = no})*. A set of blood pressure recordings repeated over a period of time will constitute medical data for the patient p_1 .

3.3 Medical Data Characteristics

Medical data are characterized by several aspects: heterogeneity, mixed granularity, imprecision, uncertainty, incompleteness, time-dependency, problem orientation,

standardization, acquisition cost, and confidentiality. We concentrate on three aspects relevant to imprecision: heterogeneity, granularity, and standardization.

3.3.1 Medical Data Heterogeneity

The heterogeneity of medical data can be described from two perspectives: data types and data sources. Medical practice uses wide-ranging data types: numerical measurements, recorded biosignals, qualitative and quantitative responses to questionnaires, images (MRI, X-Ray), coded data, narrative text data (doctors' and nurses' notes, referrals, and textual diagnosis), drawings (physicians' hand-drawn sketches), medical history, clinical assessment, and, possibly, genetic information. With respect to the source, medical data can be divided roughly into objective measurements and subjective measurements. For example, snoring, one of the important predictors of obstructive sleep apnea, can be self-reported by the patient (subjective) or recorded as a sound signal during an overnight study (objective).

3.3.2 Medical Data Granularity

Medical data vary in their granularity and representational level from low level raw data (numerical values, biosignals) through processed data (features and patterns) to knowledge represented by facts, rules and cases. Figure 1 illustrates three granularity types: data, information, and knowledge.

Medical experts operate at several levels of granularity; they aggregate raw data into information, integrate information from several sources, and use knowledge to gain information. Whereas switching between levels of abstractions is typical for

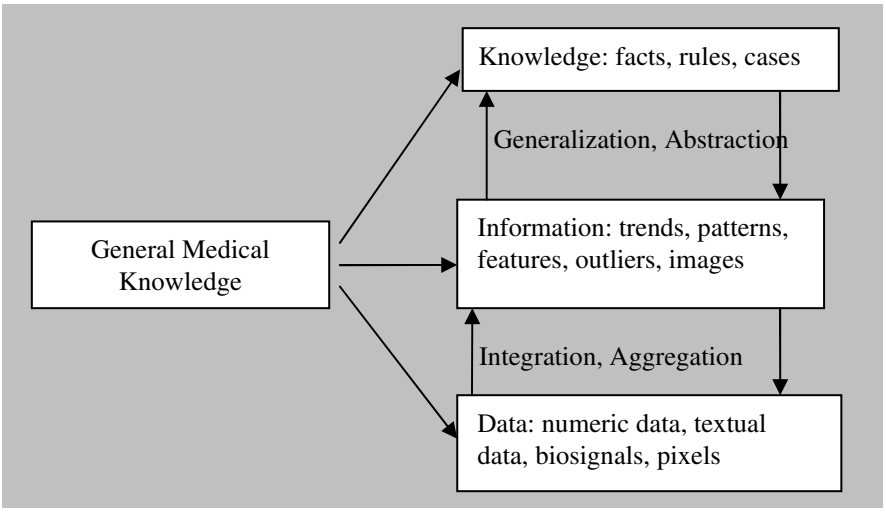


Fig. 1 The granularity spectrum for data, information, and knowledge

human reasoning, the same ability is extremely difficult for a computer-based system, which typically operates at one level of granularity. Thus, granularity and specific knowledge-based rules for data integration, aggregation, generalization and abstraction must be explicitly represented in the medical data models. For example, several blood pressure measurements (data) can be aggregated into information that a person has a hypertension. Furthermore, the evidence of hypertension can be used to determine that a patient is more likely to have a heart attack.

3.3.3 Medical Data Standardization

One of the main sources of vagueness in medical data is the lack of a standard and well defined terminology. On one hand, multiple systems have been created to capture and share clinical data, for example, Unified Medical Language System (UMLS), Generalized Architecture for Languages, Encyclopedias and Nomenclatures in Medicine (GALEN), systemized nomenclature of medicine (SNOMED CT), and ICD [22]. On the other hand, several medical concepts remain imprecise, for example, the concept of health [14]. The WHO, in 1946, defined health as "a state of complete physical, mental, and social well-being and not merely the absence of disease or infirmity." While this definition appears to be generally accepted, there certainly is a lack of agreement of what might be considered complete physical, mental and social well-being. Since health cannot be measured directly, the measuring process builds a number of variables used as the indicators of health. The WHO has developed over 70 indicators to measure the state of the health of a population as well as individuals. Moreover, most countries have their own lists of health indicators. This multiplicity of indicators indicates that there is no unanimous definition of health or adequate set of health measures. Furthermore, some indicators, for example, consultations rates with physicians are ambiguous, since they may indicate better health in a population or they may indicate more illness requiring treatment.

3.4 Medical Data and the Data-Information-Knowledge Spectrum

In this section, we discuss the problems with the separation between data, information, and knowledge in such a complex discipline as medicine. These three items of the traditional data-information-knowledge (DIK) hierarchy [1] are closely interrelated and require special modeling approaches. We identify three groups of problems: problems with the definition of terms, problems with the representation of DIK in data models, and problems with the "informational approach" as a paradigm used for the representation and reasoning in medicine.

3.4.1 Problems with the Terminology

The traditional DIK hierarchy two inherent problems: the definition of the terms and the hierarchical (sequential) structure of the concepts. First, there is a lack of consensus among the researchers on the meaning of each of the layers: data, information, and knowledge. In the data management field, data is typically defined as

raw facts, information is defined as data processed into a meaningful form, and knowledge is described as the capacity to use information [31]. However, we argue that each layer is not an isolated entity, but it is an entity created in the context of specific meanings and for a specific purpose. Second, the linear progression from data through information to knowledge in the DIK hierarchy is an oversimplified model for an iterative and bidirectional process of interpretation of data, creation of patterns, and synthesis of information. We emphasize that data as well as information and knowledge are acquired, organized, stored, retrieved, and processed using pre-existing knowledge, information, and data. Thus, we claim that medical data acquisition, retrieval, and processing are knowledge-intensive processes.

3.4.2 Problems with the Representation

The definition of data as numbers, characters, and recorded facts separates the data from their contextual meaning. This decoupling of the data and the meaning could be only theoretical, since in practice in all database systems data are intrinsically connected with the *metadata*, which define the data at least in terms of their syntactical properties – their domain (set of possible values). More realistically, a data model and its physical implementation must represent contextual information, for example, units of the measurement, time of the measurement, and the relationships with other characteristics of the object. For example, a number 150 used as a value for a weight of a human being has no meaning (cannot be interpreted) without the specification of the units: kilograms, grams, pounds, stones, or ounces. Even with the specified units, the value 150 is difficult to interpret without the contextual data, such as an age of the human being. While the weight of 150 kilograms is possible for an adult, it is impossible for a newborn baby. On the other hand, the weight of 150 stones is impossible for all human beings. In addition, the weight of a newborn baby requires more precision than the weight of a healthy adult. Depending on the clinical practice, the weight of a newborn is expressed in grams or pounds and ounces, and even small changes of 20 grams can be significant.

Thus, the data collection presupposes en existence of information and knowledge. The traditional database systems store the metadata in data dictionaries; however, data dictionaries are used to define explicitly only rudimentary knowledge, but most of the knowledge is not represented and is used implicitly (as a part of the procedural specification) in the data processing. With the emergence of the distributed databases and the Web, the data and their metadata are stored in the stand-alone databases, in the semi-structured documents as a part of the XML, and in various specifications for the exchange of the data.

The rapid development of vast repositories of data on the Web created a need for a standardized data description and universal description of Web resources. The XML schemas provide syntactical rules for the semi-structured data; however, the schemas are very limited in terms of semantics. Two Web languages provide a framework for the semantics: Resource Description Framework (RDF) and Web Ontology Language (OWL). Recently, the Semantic Web Health Care and Life Sciences Interest Group (HCLS) has been working on a RDF-based semantic description of the medical data and medical ontologies [22, 24]. The HCLS efforts are directed towards building universal models for the semantic exchange of data.

The structural specification (meta-data, abstract data types, objects, XML) and semantic specifications (RDF, semantic data modeling) are important steps towards data sharing. However, we believe that the meaning of the data and information is created as a part of the interpretation by the users and their usage for specific purposes.

3.4.3 Problems with the Methodology

The models of reality created by computing science and informatics are based on the information-processing paradigm, which assumes the existence of objective information and computational methods for processing data into information and measuring the amount of information in messages. However, the informational paradigm disregards the actual meaning of the message. Therefore, we argue that the quantitative approach must be balanced with a qualitative approach – a semiotic approach, which focuses on meaning and contextual interpretation.

3.5 Imprecision in Medical Data

In subsection 3.2, we have defined a medical datum as a structure (tuple) including a reference to the patient, a parameter measured, a value for the parameter, time, and a set of modifiers. Thus, imprecision may apply to each of the components: reference, parameter, value, time, and modifiers.

For example, the following blood pressure (BP) recordings, Rec_1 and Rec_2, considerably differ in their levels of precision: Rec_1 = *(p_1, arterial blood pressure, 160/90 systolic/diastolic in mmHg, 2008/02/01 10:00, {instrument = automatic cuff, placement = left arm, position = sitting, food intake = no prior caffeine intake within 3 hours, antihypertensive medication = no})*; Rec_2 = *(adult patient, blood pressure, above normal, morning, {food intake = some coffee}).* The first recording refers to a specific patient p_1, gives the systolic and diastolic values, describes precise time, and defines four modifiers. The second recording refers to an adult patient (without any additional information, we can assume age > 18), gives the nominal value for BP "above normal," (typically systolic BP > 119 and diastolic BP > 79; however, this value could be modified by age and gender), describes time as "morning" (we can assume early morning or any time before noon), and defines one modifier for prior coffee intake (it could be decaffeinated coffee).

While the concept of "arterial blood pressure" is well defined and has its quantitative measures, many other medical concepts, for example, quality of life, health, sleepiness, and depression are difficult to define, measure, or quantify. Thus, the modeling of imprecision requires both: qualitative description and quantitative description.

In addition, imprecision (or a specific level of precision) is an intrinsic part of all data models. The data is stored using a specific level of precision; however, it could be retrieved or processed using less precise (or nominal) values.

4 A Framework for Modeling Imprecision

In this section, we present a conceptual framework for modeling imprecision in medical data. Our framework is based on semiotics, fuzzy logic, and multi-dimensional data model. The semiotic approach provides a model for context-dependent interpretation of imprecision. The fuzzy-logic approach provides explicit representation for fuzzy (imprecise) measurements, and the multi-dimensional approach provides modeling constructs for defining several dimensions.

4.1 Semiotic Approach

Originally, the term 'semiotics' (from a Greek word for sign "semainon") was introduced in the second century by the famous physician and philosopher Galen (129-199), who classified semiotics (the contemporary symptomatology) as a branch of medicine [25]. The use of term semiotics to describe the study of signs was developed by the Swiss linguist Ferdinand de Saussure (1857-1913) and the American logician and philosopher Charles Sanders Peirce (1839-1914). Originally, Saussure used the term "semiology" and Peirce "semeiotic," but both terms correspond to today's usage of the word "semiotics."

Semiotics is a discipline which can be broadly defined as *the study of signs.* Since signs, meaning-making, and representations are all present in every part of human life, the semiotic approach has been used in almost all disciplines, from mathematics through literary studies to ethnography, including information systems, and library and information sciences [26]. A semiotic paradigm is, on one hand, characterized by its universality and transdisciplinary nature, but, on the other hand, it is associated with different traditions and with a variety of empirical methodologies. The semiotic-based approach has inspired specialized fields such as cybersemiotics, biosemiotics, and computational semiotics.

In this section, we briefly discuss the Peircean model of sign and semiosis as a process. Our intention is not to give an exhaustive history of semiotics; rather, our goal is to define the basic terminology needed to present two examples of the semiotic approach to modeling of medical concepts. The first example is an application of Peircean Semiotics to medical interpretation of radiological images. This model, called Roentgen Semiotics, has been introduced by Cantor [5,6]. Roentgen Semiotics is a systematic approach to interpretation of medical images and can be generalized to visual diagnosis from other modalities. The second example is an application of Peircean Semiotics to the interpretation of vague concepts such as sleepiness. These examples illustrate that the meaning of a sign arises in its interpretation or even multiple possible interpretations. Thus, the notion of imprecision is not universal and absolute, but should be studied in context of the interpretations of the sign.

4.1.1 Peircean Semiotics

Peirce defined "sign" as any entity carrying some information and used in a communication process. Peirce, and later Charles Morris, divided semiotics into three

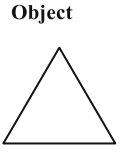


Fig. 2 Peircean semiotic triangle

categories [25]: syntax (the study of relations between signs), semantics (the study of relations between signs and the referred objects), and pragmatics (the study of relations between the signs and the agents who use the signs to refer to objects in the world). This triadic distinction is represented by a Peirce's *semiotic triangle:* the *representamen* (the form which the sign takes), *object*, and *interpretant*. The notion of "interpretant," is represented in this paper by a set of pragmatic modifiers: agents (for example: patients, health professionals, medical sensors, computer systems), perspectives (e.g., health care costs, accessibility, ethics), biases (e.g., specific subgroups of agents), and views (e.g., variations in the diagnostic criteria used by individual experts or clinics). Peirce's semiotic triangle is illustrated in Figure 2.

In the Peircean model, the relation between an object and its representation has three possible modes: symbolic, iconic, and indexical. In a symbolic relation, the sign does not resemble the object and the relation is conventional or arbitrary, for example an answer in natural language to questions about the level of sleepiness. In an iconic relation, the sign is perceived as resembling the object, for example a recorded sound of snoring or a picture of a sleeping patient. In an indexical relation, the sign is directly connected with the object, for example, patient's temperature or blood pressure. However, these modes may co-exist in the same sign, and the dominating mode is determined by the usage.

4.1.2 Example 1: Interpretation of the Roentgen Images

The Peircean model has been used by Cantor in diagnostic radiology [5,6]. Cantor's Roentgen Semiotics models the interpretation of clinical x-ray images. A radiographic image represents a three-dimensional region of the body in two dimensions. Furthermore, a radiographic image is a transmission image formed by an x-ray beam, which has been transformed by different absorption levels of the human anatomy. In two-dimensional transmission images, the localization is based on differential brightness, sharpness, magnification, projection or displacement. Cantor defines a Roentgen sign as a Peircean triad comprised of an image (the representamen), an anatomic event (the object) and an interpretation by an image reader (the interpretant). The interpretation of a Roentgen sign requires prior knowledge of the object: knowledge of normal anatomy, knowledge of human pathology, and knowledge of imaging conventions.

Fig. 3 Peircean semiotic triangle for sleepiness

The imprecision in the interpretation of an x-ray image (e.g., lack of precise localization of an abnormal event) may result from three sources: (1) inadequate quality of the attributes of the image: brightness, sharpness, magnification, projection, and displacement, (2) the imperfection of the human interpreter's knowledge and skills, and (3) plurality and mutability of interpreters (e.g., lack of consensus between several interpreters of the same image).

4.1.3 Example 2: Concept of Sleepiness

The symptom of excessive sleepiness is not easy to describe and to quantify since sleepiness can be measured only indirectly. However, the excessive daytime sleepiness is considered to be the most important indicator of sleep disorders. The measuring of sleepiness involves three aspects: conceptualization (what to measure), operationalization (how to measure), and utilization (how the measure is used). We map these three aspects to the semiotic triangle shown in Figure 3.

In specialized medical usage, "sleepiness" is often called "somnolence" and is defined as the inability to maintain wakefulness or a strong sleep propensity. In sleep medicine, the concept of sleepiness is viewed from three perspectives: biological, behavioral, and psychological [9]. Thus, there are three categories of sleepiness: *physiologic sleepiness* (biological drive to sleep), *manifested sleepiness* (decreased performance in motor activity, memory, cognition and observable behaviors such as head nodding, facial expressions, eye movement, blinking, and yawning), and *introspective sleepiness* (subjective feeling of being not alert and falling asleep). In addition, sleepiness has a temporal dimension – it can be categorized as transient (state sleepiness) or persistent (trait sleepiness). The *state sleepiness* is defined as an occasional sleepiness lasting for one or two days, as a result of occasional sleep deprivation or circadian rhythm disruptions (shift work or jet lag). The *trait sleepiness* is defined as a permanent sleepiness resulting from chronic sleep deprivation, sleep disorders, or other medical conditions.

Sleep medicine has developed several measures for sleepiness [7]. The most often used are Epworth Sleepiness Scale (ESS), Stanford Sleepiness Scale (SSS), Multiple Sleep Latency Test (MSLT), and Maintenance of Wakefulness Test (MWT). We describe these tests in detail to illustrate the various levels of their precision or imprecision.

In clinical practice, the most often used sleepiness measure is a self-administrated questionnaire, the Epworth Sleepiness Scale (ESS). This subjective

questionnaire is composed of eight questions to measure the general level of daytime sleepiness in terms of the probability of falling asleep during daily activities: (1) sitting and reading, (2) watching TV, (3) sitting inactive in public place (e.g. a theatre or a meeting), (4) riding as a passenger in a car for an hour without a break, (5) lying down to rest in the afternoon when circumstances permit, (6) sitting and talking to someone, (7) sitting quietly after lunch without alcohol, and (8) sitting in a car, while stopped for a few minutes in traffic. Each item has a score between 0 – 3. The answers are *never*, *slight chance*, *moderate chance*, and *high chance*. The maximum score is 24. Typically a score of 11 and above is recognized as excessive daytime sleepiness.

The Stanford Sleepiness Scale (SSS) is a self-reporting instrument measuring state sleepiness. Patients grade their state of alertness on a scale from 1-7; 1 corresponding to alert and 7 to falling asleep. The score above 3 indicates sleepiness.

The objective measures, such as MSLT or MWT, are expensive and time consuming. Typically, they are used in cases of narcolepsy and unexplained daytime sleepiness. The MSLT and MWT tests are performed in a sleep disorders' clinic after an overnight polysomnography (PSG). They last about 10 hours. In the MLST test, the patient is asked to have 4-5 naps every 2 hours in a quiet place. In the MWT test, the patient is asked to stay awake. In both tests, the sleep latency (the time a person takes to fall asleep) is measured using the PSG equipment. The average latency time is used for grading: 10-15 minutes ("mild degree"), 5-10 minutes ("moderate"), and less than 5 minutes ("severe"). Although the MSLT and MWT are considered gold standards, their results may be influenced by a patient's motivation, prior activity, or natural ability to fall asleep quickly. Table 1 summarizes the four sleepiness measures.

The imprecision in the representation of a vague concept such as sleepiness is related to three aspects of the semiotic triangle: the level of the precision in the definition of the object as trait, state, introspective, or physiologic sleepiness, the use of appropriate measure, and the interpretation of the measure for specific diagnostic purpose.

Table 1 Sleepiness measures used in clinical setting

Object (concept)	*Operationalization*	*Measure*	*Instrument*
Trait Sleepiness Introspective	Propensity to fall asleep in everyday situations	Subjective	ESS Scale: 0–24 Abnormal: >10
State Sleepiness Introspective	Current level of conscious	Subjective	SSS Scale: 1–7 Abnormal: >3
State/Trait Sleepiness Physiologic	Ability (time) to fall asleep in a soporific environment	Objective	MSLT Scale: 0–20 Abnormal: < 5 min
State/Trait Alertness Physiologic	Ability to stay awake (time) in a soporific environment	Objective	MWT Scale: 0–20 Abnormal: < 5 min

4.2 Fuzzy Logic Approach

One of the key concepts in fuzzy logic is the linguistic variable (fuzzy variable). A linguistic variable may be qualitative, for example sleepiness, fatigue, quality of life, or quantitative, for example, blood pressure, heart rate, and total sleep time. A linguistic variable is associated with terms. A set of terms describes the possible states of the variable. A linguistic variable can be formally represented as a quintuple: $L = <X, T(X), U, G, M>$, where X is the name of the variable, $T(X)$ is the set of terms for X, U is the universe of discourse (the set of all possible values of a linguistic variable), G is the set of grammar rules to generate $T(X)$, and M is the set of semantic rules $M(X)$.

We use the fuzzy-logic approach to define sleepiness is terms of diagnostic grades: "normal," "excessive," and "severe" based on the ESS scale. We define sleepiness as a linguistic variable represented by a quadruple: <*sleepiness*, {*normal*, *excessive*, *severe*}, [0, 24], *M*>. Where sleepiness is the name of the variable, the set {*normal*, *excessive*, *severe*} represents the three terms, the scale [0, 24] is the universe of discourse corresponding to ESS scale, and M is a set of membership functions defining the terms. The membership functions, shown in Figure 4, have been constructed based on the typically assumed values: an ESS score of 11 and above is recognized as "excessive" daytime sleepiness, and a score above 20 as "severe" sleepiness.

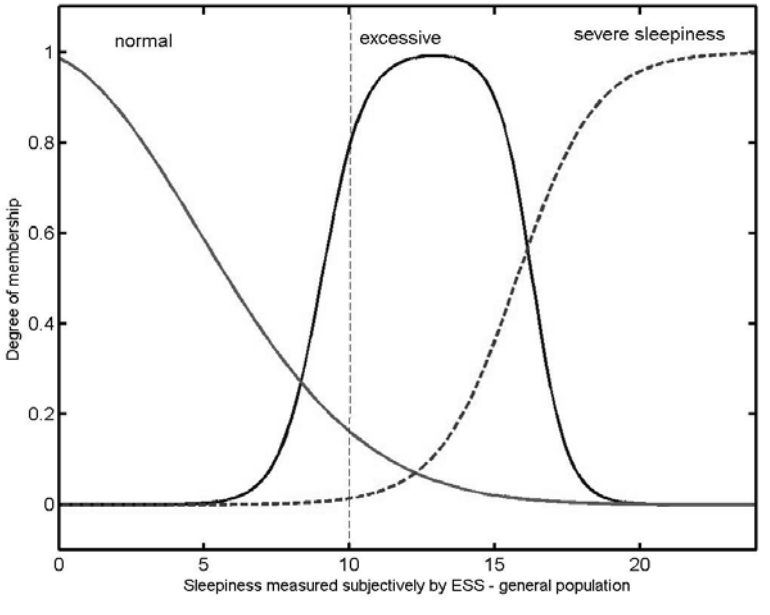


Fig. 4 Membership functions for sleepiness

4.3 Multidimensional Approach

The traditional relational data model does not provide a good support for the medical data multidimensionality and use of the Online Analytical Processing (OLAP) tools. To address these issues, a multidimensional data model has been used to represent the dimensions and, furthermore, to address various granularities and imprecision of the values [21, 8].

Time is an inherent dimension in medical data. It represents the frequency of observations (time granularity). Frequency of observations depends on particular circumstances. Some observations may be collected on daily basis, some minute-to-minute (patient in diabetic ketoacidosis), some continuous (continuous ECG monitoring of patients in clinical and ambulatory settings). The time dimension can be modeled by three temporal abstracts: a *time-point*, a *time-measure*, and a *time-interval*. A time-point describes a specific time, e.g., 2008/01/01 10:00. A time-measure described an amount (length of time), e.g., 3 hours. A time-interval denotes a segment of time, e.g., from 10:00 to 13:00.

5 Conclusions and Future Work

In this paper, we examined the definition of imprecision in the context of medical data. We demonstrated that (1) imprecision is intrinsic to medical data, (2) imprecision applies to all components of medical data: the reference to the patient, the observed parameter, the value for the parameter, time, and the modifiers; and (3) imprecision has qualitative and quantitative aspects, which depend on the knowledge-based interpretation of data. To address these issues, we presented a conceptual framework for explicit modeling of imprecision in medical data. Our framework has its theoretical foundations in semiotics, fuzzy logic, and multi-dimensional approach to data modeling. We observed that imprecision is highly contextual and has several interpretations, which led us to the application of a semiotic approach. Semiotics provides the modeling constructs for the description of the concept, its representation, and its interpretation. Furthermore, we required a formal framework to explicitly represent the imprecision and vagueness of various measures. To address this problem, we applied a fuzzy logic approach. Fuzzy logic provides representational constructs for transforming crisp numeric values into grades of membership functions corresponding to nominal values (fuzzification), reasoning with fuzzy values, and producing quantifiable results (defuzzification). To address the multidimensionality of medical data, we used a multi-dimensional data model and applied it to the time dimension. We applied the semiotic and fuzzy logic approach to define a vague concept of "sleepiness." We used the classical Peircean triangle to represent the concept of sleepiness, its measurements, and its interpretations.

We are planning to integrate and formalize the proposed framework and to build a comprehensive data model for the medical concept of "depression." We will apply the model of excessive daytime sleepiness and the model of depression

in a computer-supported clinical decisions system for the diagnosis and treatment of obstructive sleep apnea. The explicit modeling of imprecision will allow us to analyze and integrate patients' data of varied granularity and heterogeneity.

References

1. Ackoff, R.L.: From Data to Wisdom. Journal of Applied Systems Analysis 16, 3–9 (1989)
2. Barga, R.S., Pu, C.: Accessing imprecise data: An approach based on intervals. IEEE Data Engineering Bulletin 16, 12–15 (1993)
3. Bonissone, P.P., Tong, R.M.: Reasoning with uncertainty in expert systems. International Journal of Man Machine Studies 22, 241–250 (1985)
4. Bosc, P., Prade, H.: An introduction to fuzzy set and possibility theory based approaches to the treatment of uncertainty and imprecision in database management systems. In: Proceedings of the 2nd Workshop on Uncertainty Management in Information Systems, Catalina, CA (1993)
5. Cantor, R.M.: Foundations of Roentgen semiotics. Semiotica: Journal of the International Association for Semiotic Studies 131, 1–18 (2000)
6. Cantor, R.M.: Diagnostic logic in Roentgen semiotics. Semiotica: Journal of the International Association for Semiotic Studies 149, 361–376 (2004)
7. Carskadon, M.A., Dement, W.C., Mitler, M., et al.: Guidelines for the Multiple Sleep Latency Test (MSLT): A standard measure of sleepiness. Sleep 9, 519–524 (1986)
8. Delgado, M., Molina, C., Rodriguez-Ariza, L.: F-Cube factory: A fuzzy OLAP system for supporting imprecision. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 15, 59–81 (2007)
9. De Valck, E., Cluydts, R.: Sleepiness as a state-trait phenomenon, comprising both a sleep drive and a wake drive. Medical Hypotheses 60(4), 509–512 (2003)
10. Feinstein, A.R.: Principles of Medical Statistics. CRC Press, Boca Raton (2001)
11. Fricke, M.: The Knowledge Pyramid: A Critique of the DIKW Hierarchy available as electronic pre-print from Digital Library of Information Science and Technology (2008), `http://dlist.sir.arizona.edu/2327/` (accessed November 10, 2008)
12. Last, J. (ed.): A Dictionary of Epidemiology. Oxford University Press, Oxford (2001)
13. Merriam-Webster Medical Desk Dictionary (1996)
14. McDowell, I., Newell, C.: Measuring health. A guide to rating scales and questionnaires. Oxford University Press, New York (1996)
15. Motro, A.: VAGUE: A User Interface to Relational Database to Permit Vague Queries. ACM Trans. on Office and Info. Systems 6(3), 187–214 (1988)
16. Motro, A.: Accommodating imprecision in database systems: issues and solutions. SIGMOD Rec. 19(4), 69–74 (1990)
17. Niskanen, V.A.: Soft Computing Methods in Human Sciences. Studies in Fuzziness and Soft Computing, vol. 134. Springer, New York (2003)
18. Oxford English Dictionary Online. Oxford University Press (2003), `http://dictionary.oed.com.ezproxy.tru.ca/cgi/entry/00299371` (accessed December 10, 2008)

19. Parsons, S.: Current approaches to handling imperfect information in data and knowledge bases. IEEE Transactions on Knowledge and Data Engineering 8(3), 353–372 (1996)
20. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
21. Pedersen, T.B., Jensen, C.S., Dyreson, C.E.: Supporting Imprecision in Multidimensional Databases Using Granularities. In: Proceedings of the 11th international Conference on Scientific on Scientific and Statistical Database Management, Washington, DC (1999)
22. Robu, I., Robu, V., Thirion, B.: An introduction to the Semantic Web for health sciences librarians. Journal of the Medical Library Association 94(2), 198–205 (2006)
23. Rothman, K.J., Greenland, S.: Modern epidemiology. Lippincott-Raven, Philadelphia (1998)
24. Ruttenberg, A., Clark, T., Bug, W., et al.: Advancing translational research with the Semantic Web. BMC Bioinformatics (2007), doi:10.1186/1471-2105-8-S3-S2
25. Sebeok, T.A.: Signs: An introduction to semiotics. University of Toronto Press (1999)
26. Sheng-Cheng, H.: A semiotic view of information: Semiotics as a foundation of LIS research in information behaviour. Proceedings of the American Society for Information Science and Technology 43(1), 66 (2006)
27. Shortliffe, E.H., Barnett, G.O.: Biomedical data: their acquisition, storage, and use. In: Shortliffe, E.H., Cimino, J.J. (eds.) Biomedical informatics. Springer, Heidelberg (2006)
28. Skala, H.J.: On the problem of imprecision. Theory and Decision 7(3), 159–170 (1976)
29. Smets, P.: Imperfect Information: Imprecision and Uncertainty. In: Uncertainty Management in Information Systems 1996, pp. 225–254 (1996)
30. Straszecka, E.: Combining uncertainty and imprecision in models of medical diagnosis. Information Sciences 176, 3026–3059 (2006)
31. Watson, R.T.: Data Management: Databases and Organizations. John Wiley & Sons, Chichester (2006)
32. Zadeh, L.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Promoting Diversity in Top Hits for Biomedical Passage Retrieval

Bill Andreopoulos, Xiangji Huang, Aijun An, Dirk Labudde, and Qinmin Hu

Abstract. With the volume of biomedical literature exploding, such as BMC or PubMed, it is of paramount importance to have scalable passage retrieval systems that allow researchers to quickly find desired information. While topical relevance is the most important factor in biomedical text retrieval, an effective retrieval system needs to also cover diverse aspects of the topic. Aspect-level performance means that top-ranked passages for a topic should cover diverse aspects. Aspect-level retrieval methods often involve clustering the retrieved passages on the basis of textual similarity. We propose the HIERDENC text retrieval system that ranks the retrieved passages, achieving scalability and improved aspect-level performance over other clustering methods. HIERDENC runtimes scale on large datasets, such as PubMed and BMC. The HIERDENC aspect-level performance is consistently better than cosine similarity and Hamming Distance-based clustering methods. HIERDENC is comparable to biclustering separation of relevant passages, and improves on topics where many aspects are involved. Converting textual passages to GO/MeSH ontological terms improves the HIERDENC aspect-level performance.

1 Introduction

The body of biomedical literature is growing rapidly. PubMed, the main biomedical literature database, holds over 17 million abstracts and over 2000 new abstracts are added a day (1). Information retrieval (IR) technology

Bill Andreopoulos, Xiangji Huang, Aijun An, and Qinmin Hu
Dept. of Computer Science and Engineering, York University, Toronto, Canada
e-mail: {billa,jhuang}@cse.yorku.ca

Bill Andreopoulos
Biotechnological Centre, Technische Universität Dresden, Germany

Dirk Labudde
Bioinformatics group, University of Applied Sciences, Mittweida, Germany

Z.W. Ras and A. Dardzinska (Eds.): Advances in Data Management, SCI 223, pp. 371–393.
springerlink.com

plays a vital role in biomedical data management, especially for users who desire passages that are most relevant to a topic or question. A biomedical information retrieval system is a computer system for browsing, searching and retrieving passages from a large collection of biomedical literature. Methods of information retrieval may utilize some method of adding metadata to the text, such as ontological term annotations extracted via text mining, keywords or descriptions; then retrieval is performed over the textual annotations (2). Information retrieval is required to scale up efficiently to the quickly growing body of biomedical literature.

1.1 Aspect-Level Performance: Promoting Diversity in the Top Hits

For addressing users' questions on a topic in competitions, such as in the TREC 2007 Genomics Track (3; 4) or the ImageCLEF 2008 Photo Retrieval Task (5), one of the main tasks is to extract ranked textual snippets from documents (6; 7). The performance is considered better if the top-ranked textual snippets are not only relevant to the topic, but also cover diverse *aspects*. A biomedical researcher would like to avoid seeing similar or duplicated passages in the top hits, and redundant information is removed by covering diverse aspects. A search engine that retrieves a diverse, yet relevant set of textual passages at the top of a ranked list is more likely to satisfy its users. Another reason why it's a good idea to promote diversity is because often different people type in the same query but wish to see different results. Aspect-level retrieval performance was previously studied in the context of competitions such as TREC and ImageCLEF. Text-based clustering was used to group passages, consequently promoting diverse topics in the top retrieved hits.

The main difference of our work from previous work is to take a more practical approach to the problem of aspect-level retrieval, making it scalable to large and quickly expanding biomedical literature. Our system promotes diversity in the top hits through scalable text-based clustering. The main contributions of our work include: *a*. We propose scalable aspect-level retrieval that works with millions of documents as well as thousands of documents, and *b*. We convert passages to vectors of ontological terms, improving aspect-level retrieval performance. Further benefits of our methodology as far as the clustering method is concerned include: no re-clustering needed when new text is presented, no user-specified input parameters required, and insensitivity to ordering of passages.

This chapter is organised as follows. Section 2 discusses related work, including document clustering algorithms for separating the relevant passages, and ontology-based integration of biomedical information. Section 3 presents the HIERDENC algorithm that ranks the passages through text-based clustering. Section 4 presents the evaluation methods, experimental results and

the corresponding discussions, demonstrating scalable HIERDENC runtimes on large real world biomedical datasets. This section also demonstrates reasonable aspect-level performance with ranking and after converting passages to GO/MeSH ontological term vectors. Finally, section 5.1 gives our conclusions and Section 5.2 describes future research directions.

2 Related Work

2.1 Clustering in Information Retrieval

Clustering is a common technique for statistical data analysis, which has been used in biomedical question answering and aspect-level retrieval.

Goldberg et al. used a naive clustering for reranking passages; the results were discouraging, resulting in worse aspect-level performance than the original ranking, as well as lower document- and passage-level performance scores (6). In particular, they used bag-of-words vector representations and cosine-similarity based clustering. This differs from our work where we convert passages to ontological term vectors. While they interleaved results from clusters to achieve aspect diversity, our method ranks the clusters by their coverage in the entire dataset and keeps the most representative passage from each cluster. They also used random walks on a graph over passages to promote diversity, but our method considers which clusters are the most prominent in the dataset and likely to represent different aspects of the topic.

Si et al. derive the MeSH representations for the top-ranked passages for a user query, reflecting the topical aspects of passages. Then, they rerank the passage retrieval result to construct a new ranked list. A document is selected and added to the bottom of the current reranked list, by considering the novelty information of the topical aspects with respect to the current reranked list (8). While they extract representative MeSH terms for each passage, we also extract Gene Ontology terms. Another difference from our work is that they adopt a gradient-based search approach, while we consider globally significant clusters in the entire dataset. Therefore, while their method is sensitive to ordering of passage input, our method is not.

In this work, we will compare our HIERDENC clustering method to three other clustering methods, presented next. We will evaluate these methods' aspect-level retrieval performance: biclustering, cosine similarity and hamming distance-based clustering. We will cluster both the original text passages and the extracted ontological term vectors.

2.2 Biclustering of Passages

Biclustering allows simultaneous clustering of the rows and columns of a matrix, where the columns are textual passages and the rows correspond to

words ([9]; [10]). In our biclustering approach, we produce only two clusters, since we want to separate the passages that are relevant to the topic from the rest. Another reason for producing only two clusters is that biclustering performance is known to deteriorate for more clusters. We select the smallest cluster as more likely to contain relevant passages, since usually the majority of passages retrieved are irrelevant. Biclustering differs from our proposed HIERDENC method that produces many clusters and then ranks them, keeping a representative passage from each cluster. Given an $m \times n$ word-by-document matrix, the biclustering algorithm generates biclusters - a subset of rows which exhibit similar behavior across a subset of columns, or vice versa. We find subgroups in a binary matrix where entries are one or zero.

Let A denote the $m \times n$ word-by-document matrix, and D_1 and D_2 denote diagonal matrices such that $D_1(i,i) = \Sigma_j A_{ij}$, $D_2(j,j) = \Sigma_i A_{ij}$. Then, the following equations define the singular value decomposition (SVD) of the normalized matrix $A_n = D_1^{-1/2} A D_2^{-1/2}$:

$$D_1^{-1/2} A D_2^{-1/2} v = (1-\lambda)u, \; and \; D_2^{-1/2} A^T D_1^{-1/2} u = (1-\lambda)v.$$

In particular, u and v are the left and right singular vectors respectively, while $(1-\lambda)$ is the corresponding singular value σ. We compute the left and right singular vectors corresponding to the second (largest) singular value of A_n, $A_n v_2 = \sigma_2 u_2$, $A_n^T u_2 = \sigma_2 v_2$, where $\sigma_2 = 1 - \lambda_2$. The right singular vector v_2 will give a bipartitioning of documents while the left singular vector u_2 will give a bipartitioning of the words. Given the singular vectors u_2 and v_2 the key task is to extract the optimal partition from these vectors. Biclustering looks for a bi-modal distribution in the values of u_2 and v_2. Let m_1 and m_2 denote the bi-modal values that we are looking for. The second eigenvector of the Laplacian matrix is given by $z_2 = \left(D_1^{-1/2} u_2 \;\; D_2^{-1/2} v_2\right)$. One way to approximate the optimal bipartitioning is by the assignment of $z_2(i)$ to the bi-modal values $m_j (j = 1, 2)$ via the classical k-means algorithm:

1. Given A, form $A_n = D_1^{-1/2} A D_2^{-1/2}$.
2. Compute the second singular vectors of A_n, u_2 and v_2; form the vector z_2.
3. Run the k-means algorithm on the 1-dimensional data z_2 to obtain the desired bipartitioning.

This algorithm runs k-means simultaneously on the reduced representations of both words and documents to get the co-clustering. Thus, the biclustering algorithm co-clusters words and documents.

2.3 Cosine Similarity Textual Clustering

Cosine similarity is a measure of similarity between two vectors of words by finding the angle between them, often used to compare documents (or passages)

in text mining (11). Cosine similarity is typically used for bags-of-words representations of textual passages, and is significantly slower than our proposed method depending on all-by-all comparisons. Given two vectors of words, A and B, the cosine similarity, θ, is represented using a dot product and magnitude as $\theta = arccos\frac{AB}{|A||B|}$. In the cosine similarity based approach that we used in our experiments, each passage is matched to its nearest passage according to θ; in graph terms this is conceptualized as a directed edge from the former passage to the latter. Then, every connected component is considered as a cluster. Passages that are not connected via a path are separate clusters.

2.4 Hamming Distance Textual Clustering

The Hamming Distance is used as a measure of dissimilarity between two vectors of words, by counting the number of words that are contained in one vector but not the other (12; 13). The HD-based clustering depends on the ordering of passage input, and exhibits quadratic complexity unlike our proposed method. Given two vectors of words, A and B, the Hamming Distance, HD, is computed as $HD = |(A - B) \cup (B - A)|$. The clustering iterates over all passages from the smallest to the largest; a passage π is matched to cluster c_π with which it has the most words in common, considering the union of all words appearing in the cluster. The passage π is clustered in c_π if the HD between them does not exceed a threshold ϕ. Threshold ϕ represents the maximum HD, determining if π is clustered or not; the ϕ value starts from 1 and is progressively relaxed, thus producing layers in clusters. Layered clusters have an "onion"-layered structure, such that the least dissimilar passages are placed in the initial-created layers and affect subsequent clustering decisions. The iteration through passages continues until all passages have been clustered.

2.5 Query Term Expansion

Query expansion is a popular and commonly used strategy to improve the passage-retrieval performance. Our ontological term extraction on retrieved passages resembles query term expansion, in the sense that ontological terms and potentially their ancestors are also associated with passages. In the past, expansion was done on queries, but nobody tried expansion on retrieved passages. Through extracting ontological terms from passages, our proposed method has potential to outperform methods that expand queries for improved retrieval performance. Moreover, previous work which applied query expansion based on hand-crafted thesaurus is often limited in improving the performance (14; 15). For example, Voorhees (15) expanded

queries with synonyms manually selected from WordNet and achieved only limited improvements (around −2% to +2%) on some queries. Recently a lot of work on biomedical information retrieval appeared in the TREC Genomics Track (16; 17; 3). Huang and others (18) achieved notable performance improvements by manually processing the gene name variants from gene databases. Zhou et al (19) proposed their effective conceptual retrieval model by incorporating five types of domain knowledge including synonyms.

2.6 Ontology-Based Data Integration in Bioinformatics

Individually developed ontologies often support the annotation of online databases for information retrieval purposes. Significant work has been done in the past two years to make the ontologies interoperable and support integration of information from different sources. These efforts aim to facilitate ontology interoperability and automated reasoning. We leverage our ontological term extraction for ontology linking, which differs from other ontology-based data integration methods through its automation and simplicity of use. We rely solely on the notion of term extraction from documents and co-occurring terms in the same passage (or image caption). Our ontology linking method is likely to appeal to biomedical practitioners and researchers better than RDF and semantic web-based methods that exhibit low usability and appeal.

Burek et al. (2006) (20) present a top-level ontological framework for representing knowledge about biological functions. This framework provides a means to capture existing functional knowledge in a principled way.

Garcia-Sanchez et al. (2008) (21) propose an ontology-based framework for seamlessly integrating intelligent agents and semantic web services. Agent technology can assist users in discovering services available on the Internet. This allows integrated access to biomedical information.

Smith et al. (2007) (22; 23) leverage the structure of the semantic web to enhance information retrieval for proteomics. They use an RDF graph that inter-relates documents through their associated biological identifiers (e.g., protein ID). In related work, they built a software system called LinkHub using semantic web RDF that manages the graph of identifier relationships. LinkHub facilitates cross-database queries and information retrieval in proteomics.

Ruttenberg et al. (2007) (24) discuss advancing translational research with the semantic web. They present a scenario that shows the value of the semantic web technologies for aiding biomedicine researchers. They conclude that semantic web technologies present promise and current tools and standards are already adequate for translational research.

3 Methods

The previous section discussed how clustering is used in the information retrieval process. In this section we will examine our HIERDENC system, which differs from previous work as follows: *a*. For aspect-level retrieval performance, it provides a scalable clustering method for ranking textual passages, and *b*. We use Go/MeSH ontological term vectors as caption-based term expansion. In this section we first present our textual retrieval system. Then, we present our test datasets, and the TREC evaluation measures used to compare performances of all methods.

3.1 Workflow of HIERDENC Text Retrieval System

Figure 1 shows the workflow of HIERDENC text retrieval. The objects to be clustered are the textual passages and snippets, which may be captions of images. HIERDENC applies text-based clustering in combination with ontological term extraction on text. Each passage is represented as a "word vector", whether it is the original passage or the one converted to ontological terms.

Automatic annotation of biomedical passages can be an important step when searching for information from a database. We used the GoPubMed term extraction algorithm for converting each passage to a vector of ontological terms. This vector describes each passage on an ontological basis.

Users search via keywords and the retrieved passages are clustered into groups of topics. Retrieved passages are clustered based on the original text, or extracted ontological term vectors. The HIERDENC retrieval system clusters the passages to achieve good *aspect-level* performance. The clustering imposes a ranking of retrieved passages, such that top ranked passages reflect different topics for the query. Top ranked passages are similar to many other passages in the database, but any two top ranked passages are likely to be different.

Text-based clustering can also be applied to biomedical image databanks, where images have textual captions or comments associated with them. Figure 2 shows a snapshot of HIERDENC retrieval as applied to image captions; caption-based clusters are represented on the right-hand side bar and clicking on a cluster takes the user to the corresponding cluster of images.

A final capability of our system is to link different ontologies (or vocabularies) if two extracted ontological terms co-occur in the same passage or caption. Linked ontologies support reasoning over the vast biomedical knowledge.

3.2 Ontological Term Extraction from Passages

After standard stop word removal in the data preparation step, we converted each passage to a vector of ontological terms extracted via the GOPubMed

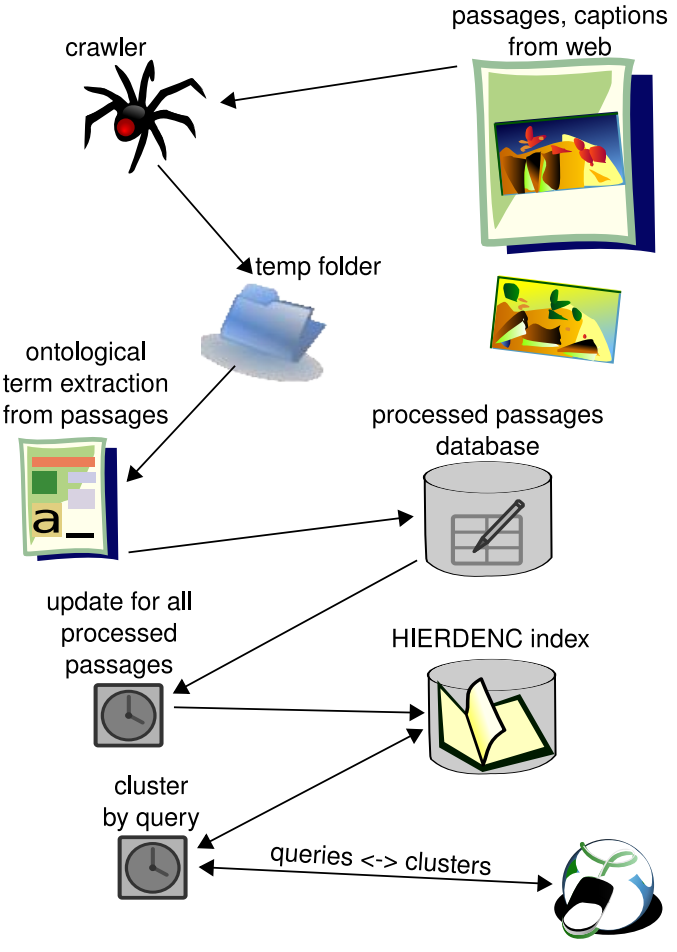
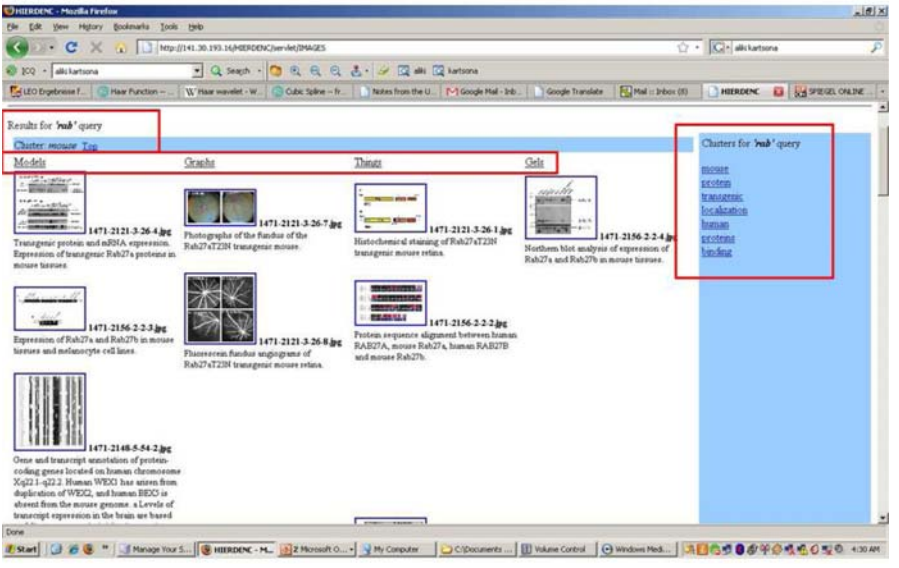


Fig. 1 The workflow of the HIERDENC image retrieval system. The system starts by converting passages (or image captions) to vectors of ontological terms. In continuation, the HIERDENC index is updated with the processed passages. User queries in the form of one or more keywords result in clusters of passages

text mining algorithms (2). The ontologies used for this purpose included MeSH and Gene Ontology (GO). The term extraction algorithm uses local sequence alignment of words of the passage and the words of GO terms. First we applied a tokenizer to the GO terms. The words of each term are then aligned against the passage text. Figure 3*a* shows an example of how we might extract ontological terms from a passage, in this case an image caption. The image caption contains terms from the Gene Ontology, MeSH, as well as a gene name. The ontological terms extracted from passages can be useful for

a)



b)

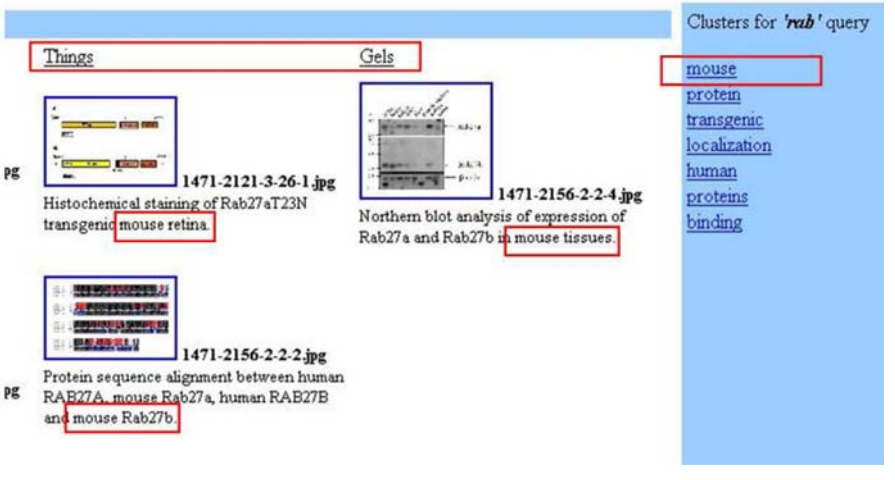


Fig. 2 *a*. A screenshot shows the results of passage retrieval for protein "rab" query. The HIERDENC system allows a user to search passages and retrieve clusters. In this example, the passages are image captions converted to ontological terms. The blue right-hand sidebar denotes clusters derived by text-based clustering, and clicking on a cluster name takes the user to the corresponding images. In this example, the clusters are defined by Gene Ontology and MeSH terms such as "mouse", "protein", "humans", "localization" and "binding". *b*. A zoom-in shows the clusters better. Columns denote classifications of images via image feature analysis done by an expert

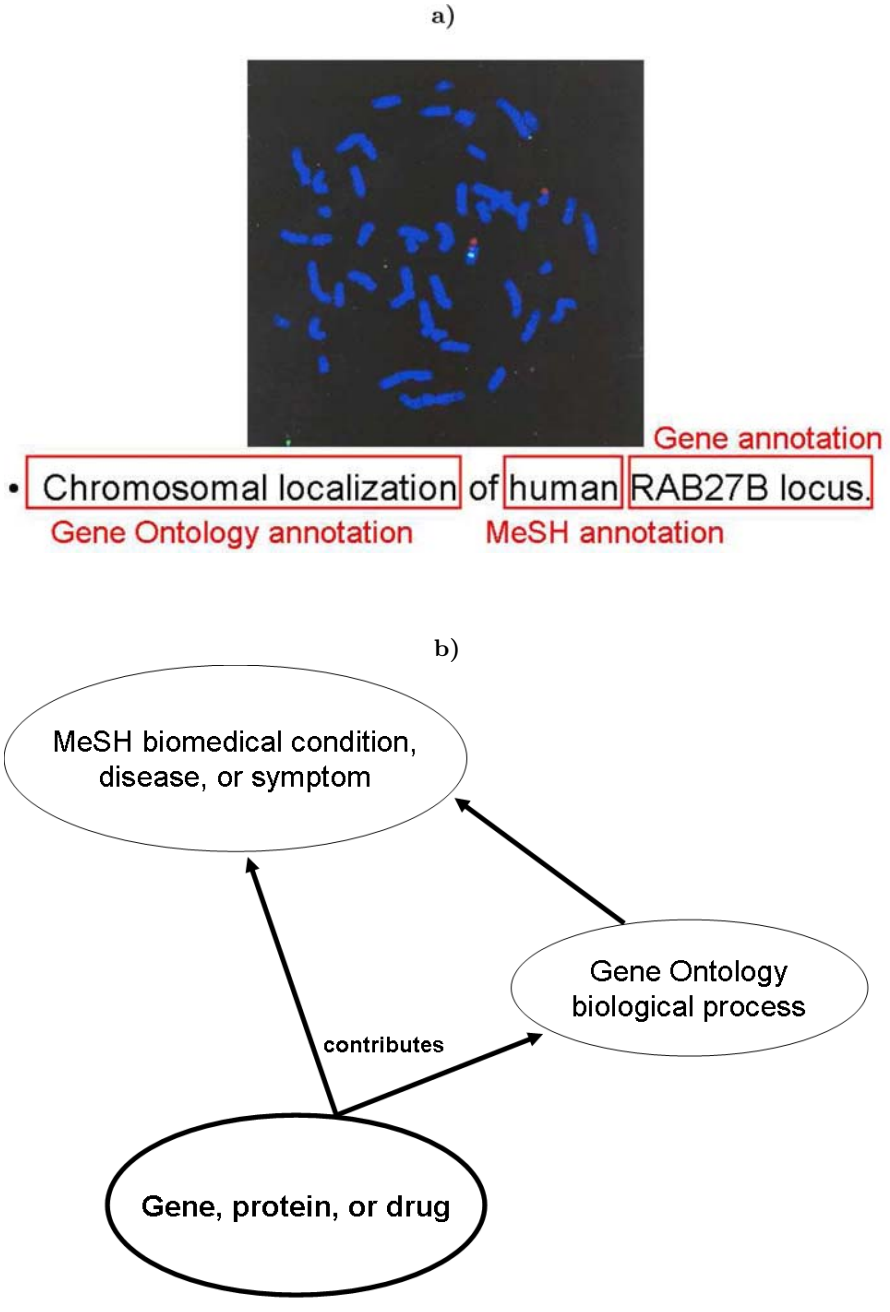


Fig. 3 Image caption ontological term extraction via text mining. *a*. An image caption on the Rab protein contains ontological terms from Gene Ontology, MeSH, and gene names. *b*. The co-occurrences of terms in the same caption can imply links between ontologies

clustering the passages more effectively and improving aspect-level retrieval performance. Figure 3*b* shows further how the ontological term extraction can be used to link different ontologies and vocabularies; terms that co-occur in the same passage imply a link between different ontologies, such as a gene contributing to a GO biological process which contributes to a MeSH medical condition.

For biclustering we used only the Gene Ontology/MeSH annotations of passages, transformed to boolean matrices, since the original passages would result in huge matrices. We tested the other clustering methods on both the original passages, as well as the ones converted to GO/MeSH ontological terms.

3.3 HIERDENC: Density-Based Clustering for Reranking Passages

We adapt the HIERDENC algorithm which was previously presented for *efficient density-based clustering of categorical data* (13; 25). The goal of this clustering adaptation is to rank the passages, such that: *a*. A highly ranked passage is representative of an aspect, identified in clustering as a relatively large group of similar passages. *b*. Top ranked passages cover many different aspects of the topic.

Basics. Let Π denote the set of all passages in our dataset. We define the cluster $\Pi_0(\pi_\mathbf{0}, \sigma) \subset \Pi$, centered at passage $\pi_\mathbf{0}$ with radius σ, as follows:

$$\Pi_0(\pi_\mathbf{0}, \sigma) = \{\pi : \pi \in \Pi \; and \; sim(\pi, \pi_\mathbf{0}) = \sigma\}.$$

The $sim(\cdot)$ is a similarity function representing the number of common words in two passages, defined as follows:

$$sim(\pi_\alpha, \pi_\beta) = |\pi_\alpha \cap \pi_\beta|$$

The *density* of a cluster $\Pi_X \subset \Pi$, where Π_X equals $\Pi_X(\pi_\mathbf{X}, \sigma) \subset \Pi$, involves the number of passages that are included in Π_X: $density(\Pi_X) = |\Pi_X|$, where $|\Pi_X|$ is the size of Π_X. This density can also be viewed as the likelihood that cluster $\Pi_X \subset \Pi$ contains a random passage from Π.

HIERDENC seeks the densest cluster $\Pi_0(\pi_\mathbf{0}, \sigma) \subset \Pi$. This is the cluster centered at $\pi_\mathbf{0}$ that has the most other passages from Π with a similarity of σ.

HIERDENC Ranking Algorithm and Discussion. The ranking is performed on the representative central passages of clusters; our goal is to rank higher passages that are centers of larger and more dense clusters, representing big distinct groups of passages. Every passage $\pi \in \Pi$ is the center

of a cluster with the maximum radius for which at least one other passage exists, $MaxSim_\pi$. We retrieve the clusters in order using the HIERDENC index, which supports finding the densest cluster of passages efficiently. The HIERDENC index is updated fast when a new passage is introduced.

For each passage π, the HIERDENC index stores three values determining the rank of the cluster centered at π: $MaxSim_\pi$ is the maximum similarity (cluster radius) found between π and any other passage; $PassSize_\pi$ is the length of π in terms of words; $NumSimPass_\pi$ is the number of passages that are cluster members with $MaxSim_\pi$ similarity to π, i.e., the size of the cluster centered at π. Figure 4 shows two clusters, which differ in terms of $MaxSim$, $PassSize$, and $NumSimPass$. The retrieved passages are ranked by decreasing $MaxSim$, increasing $PassSize$, and decreasing $NumSimPass$. The top-ranked passages are those retrieved for the highest value of radius $MaxSim$, the lowest $PassSize$ value, and the highest $NumSimPass$, capturing the centers of large clusters with similar passages. The decreasing $NumSimPass$ will give priority to larger clusters. The decreasing $MaxSim$ and increasing $PassSize$ are motivated by the Jaccard Index similarity measure; the Jaccard Index of two word vectors, π_α and π_β, results in a higher similarity for more common words and fewer overall words: $Jaccard\ Index(\pi_\alpha, \pi_\beta) = \frac{|\pi_\alpha \cap \pi_\beta|}{|\pi_\alpha \cup \pi_\beta|}$.

Figure 5 shows the pseudocode of the HIERDENC ranking process. For ranking the passages, we retrieve the passages using the HIERDENC index in the order described above. Then, we maintain a set Υ of the central and member passages of all clusters that were considered previously. We print the central passage under consideration if there is null intersection between its cluster members and Υ. Therefore, we print the central passages for the densest clusters, which are most likely to be representative of different aspects of the topic. If there is non-null intersection, then we put the central passage instead in a special ordered list Λ, which can be printed out after all passages have been iterated through if a user desires further results.

The HIERDENC index updating and cluster retrieval are efficient, achieving runtime scalability on the number of passages. For a new passage π, its most similar previous passages are found; this is done fast by maintaining each non-stop word's previous passage occurrences. Then the HIERDENC index is updated with the passages' $MaxSim$, $NumSimPass$, and $PassSize$ information. The first time the HIERDENC index is updated with N passages, the average runtime is $O(Nm)$, where m is the number of words (usually $m << N$). When n new passages are introduced, the updating of the index has a runtime of $O(nm)$. For the passages to be ranked by retrieving the centers of densest clusters, the worst-case runtime is $O(N)$; the ranking iterates until a maximum of N passages that are cluster centers are retrieved. The worst-case space complexity is $O(N^2)$, since for each passage information regarding the maximum similarity $MaxSim$ found to any other passage is stored; however, for large datasets, most pairs of passages have little similarity, significantly reducing the space requirement.

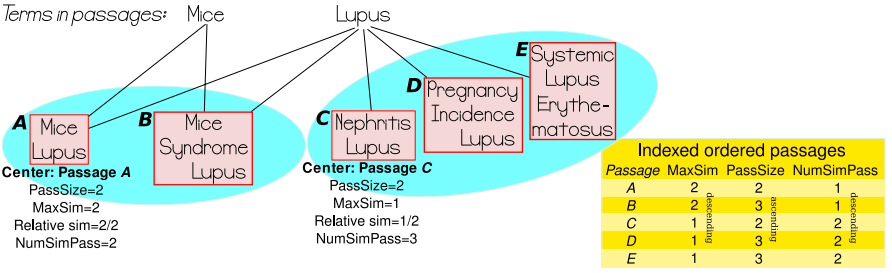


Fig. 4 HIERDENC indexing overview. Newly introduced passages involve updating the index. There are three fields involved in determining the rank for each passage: $MaxSim$ (descending), $PassSize$ (ascending), $NumSimPass$ (descending). The passages are records that are ordered accordingly in the HIERDENC index

```
function HIERDENC() {
    Π = retrieve list of ordered passages from HIERDENC index;
    Υ = empty set; //Holds passages already considered;
    Λ = empty list; //Holds remaining ordered passages;

    for passage π in Π:
        C = clusterCenteredAt(π, Π, MaxSim_π);
        if |C ∩ Υ| = 0:
            print π;
            Υ = Υ ∪ C;
        else: Λ = Λ ∪ π;
    print Λ;
}

function clusterCenteredAt(π, Π, MaxSim_π) {
    return {π_β ∈ Π|sim(π, π_β) = MaxSim_π};
}
```

Fig. 5 HIERDENC algorithm for retrieving passages in a ranked ordering, such that top-ranked passages include different aspects. Clusters C are retrieved in sequence based on their density, and the center π of each cluster C is printed if it does not overlap with any previous clusters. If there is overlap, then the center π is added to list Λ, which is printed out in the end after all passages have been considered

4 Evaluation: Results and Discussions

In the remainder of this chapter, we will discuss our evaluation: *a*. Passage retrieval scalability to very large datasets, *b*. Aspect-level retrieval performance, or how many aspects are represented by the top ranked passages.

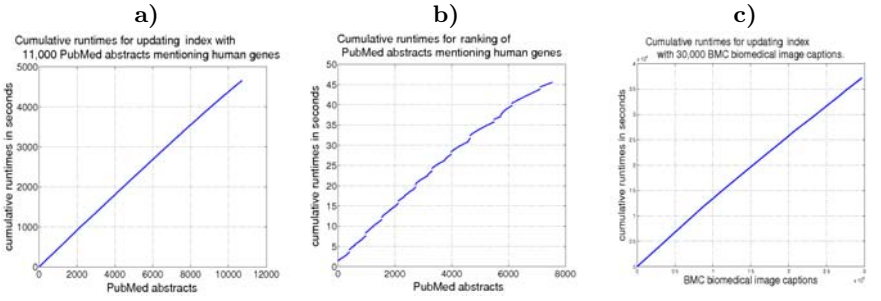


Fig. 6 HIERDENC index updating and passage reranking runtimes: *a*. HIERDENC index updating cumulative runtimes for a set of 11,000 PubMed passages containing human gene mentions, and *b*. Ranking cumulative runtimes, *c*. HIERDENC index updating cumulative runtimes for a set of 30,000 BMC biomedical image captions. In comparison, *k*-Means did not finish on clustering such a large image captions dataset

4.1 Scalability of HIERDENC to Large Image Datasets

Figure 6*a* shows the cumulative runtimes for updating HIERDENC with 11,000 documents that mention human genes in PubMed. Figure 6*b* shows the cumulative runtimes for clustering and returning all of the documents in a ranked ordering. To assess scalability further, we clustered the image captions of 30,000 BMC images published in the period 2000-2007. Figure 6*c* shows that the HIERDENC index updating runtimes scale with the number of image captions. These runtimes highlight the potential of HIERDENC as a text retrieval system that can deal with growing biomedical databanks.

4.2 Aspect-Level Retrieval on TREC 2007 Genomics Track

We evaluated HIERDENC's aspect-level retrieval performance on textual passages from the TREC 2007 Genomics track. To evaluate the HIERDENC aspect-level retrieval performance, we compare to separating the most relevant images using document clustering approaches: biclustering (9; 10), Hamming-distance (12; 13) and cosine-similarity (11) clustering.

TREC 2007 Genomics Track Topics and Evaluation. To evaluate aspect-level retrieval performance, we used the 36 topics from the TREC 2007 Genomics track. The topics are in the form of questions asking for lists of specific entities; these entities are based on controlled terminologies from different sources, with the source of the terms depending on the entity type.

Given a question, we initially retrieved 1,000 passages using the well-known OKAPI question answering system (26; 27; 28). Then, we updated the HIERDENC index with the passages and we retrieved clusters. Suppose that the information needed is: "What is the genetic component of alcoholism?" This is transformed into a question of the form: "What [GENES] are genetically linked to alcoholism?" Answers to this question will be passages that relate one or more entities of type GENE from MeSH terminology to alcoholism. For example, the following would be a relevant answer: "The DRD4 VNTR polymorphism moderates craving after alcohol consumption." The GENE entity supported by this statement would be DRD4.

The TREC results are evaluated based on how well they provide relevant information at three levels for a user trying to answer the given topic questions: passage retrieval, aspect retrieval, and document retrieval. The TREC statistic of Mean Average Precision (MAP) is the average precision at each point a relevant document or passage is retrieved. The evaluation measures for the TREC 2007 Genomics track are also called gold standard measures and have the following levels of retrieval performance (16):

Aspect-level MAP: A question could be addressed from different aspects. For example, the question "what is the role of gene PRNP in the Mad cow disease?" could be answered from aspects like "Diagnosis", "Neurologic manifestations", or "Prions/Genetics". This measure indicates how comprehensively the question is answered. Aspect retrieval was measured using the average precision for the aspects of a topic, averaged across all topics given their retrieved passages. The precision for the retrieval of each aspect was the fraction of relevant passages for the retrieved passages of a topic, up to the first passage in the ranked list that has the aspect assigned. These fractions at each point of first aspect retrieval were then averaged together to compute the average aspect precision. A relevant passage may have associated with it multiple aspects. Relevant passages that did not contribute any new aspects to the aspects of higher ranked passages were removed from the ranking, since the utility for a user of the same aspect occurring again further down the list is uncertain. Taking the mean over all topics produced the final aspect-based MAP (3).

Document-level MAP: This is the standard IR measure. The precision is measured at every point where a relevant document is obtained and then averaged over all relevant documents to obtain the average precision for a given query. For a set of queries, the mean of the average precision for all queries is the MAP of that IR system.

Passage-level MAP: As described in (17), this is a character-based precision calculated as follows: for each relevant retrieved passage, precision will be computed as the fraction of characters overlapping with the gold standard passages divided by the total number of characters included in all nominated passages from this system for the topic up until that point. Similar to regular MAP, relevant passages that were not retrieved will be

Table 1 2007 topic#1: "What serum [PROTEINS] change expression in association with high disease activity in lupus?". Frequent words and two-word phrases for the top 100 HIERDENC-ranked articles in increments of 10. These show different frequent contents for every 10 passages, and the contents indicate different aspects

	Top word frequency		Two-word phrases frequency	
Ranked	Word	Occ.	Expr.	Count
1-10	plasminogen	25	peptide elongation	10
11-20	purpura	9	anemia hemolytic	6
21-30	rickettsia	6	myocardial infarction	2
31-40	hepatitis	6	bone marrow	2
41-50	hepatitis	9	hepatitis evaluation	3
51-60	nervous	6	nervous system	6
61-70	thrombosis	7	thrombosis arteries	3
71-80	immune	10	immune response	10
81-90	contraceptive	3	postmenopause contraceptives	3
91-100	system	13	system development	2

added into the calculation as well, with precision set to 0 for relevant passages not retrieved. Then the mean of these average precisions over all topics will be calculated to compute the mean average passage precision".

Table 1 shows for the 2007 topic#1 the most significant words in the top 100 ranked passages, examined in increments of 10 passages. The significant words change between increments, showing that the top ranked passages cover several different aspects of the topic. Next, we compare HIERDENC's passage ranking to other clustering-based separation of relevant passages. To make the comparison meaningful, we evaluate all methods with the same TREC performance measures.

HIERDENC vs. Other Clustering Aspect-level Performance. We compare the aspect-level performance of HIERDENC on each topic to Hamming-distance and cosine-similarity based clustering methods that return two or more clusters. For Hamming-distance and cosine-similarity clustering, one can select the relevant passages based on cluster sizes: we prefer the smallest clusters (size ≤ 4) because they are more likely to correspond to diverse aspects of the topic in question. We produce results for the GO/MeSH ontology-converted passages and the original passages.

Table 2 shows the results. For all topics, HIERDENC ranking of retrieved passages improved the aspect-level performance over the Hamming-distance and cosine-similarity clusterings. The main reason for this is that HIERDENC considers all similarities found between passages in the dataset, and therefore can separate groups of passages considering whether their similarities are significant relative to the other similarities found. On the other hand, Hamming-distance and cosine-similarity clusterings do not consider the significance of a similarity relative to other similarities found elsewhere in the dataset. HIERDENC ranks all passages retrieved for a topic, while the latter

Table 2 Average aspect-level performance results over 36 TREC 2007 topics. We used HIERDENC clustering of passages for aspect-level performance, and three clustering methods to separate the relevant passages. We used the original passages as well as those converted to extracted GO/MeSH terms, except for biclustering where the original passages were too large for matrix computations. Small clusters have size ≤ 4 and are more likely to contain relevant passages than large clusters. HIERDENC ranking gives overall better aspect-level performance than Hamming-distance and cosine-similarity based clustering. Extracting ontological terms from passages results in improved apsect-level performance over using the original passages. Biclustering for separating the relevant passages gives better results on some topics. These results are also consistent across the other TREC evaluation measures that are summarised here: document-level, passage-level and passage2-level performance

Clustering	**Aspect**	**Document**	**Passage**	**Passage2**
HIERDENC GO/MeSH terms	**0.073**	**0.129**	**0.054**	**0.019**
HIERDENC original passages	0.034	0.127	0.049	0.013
Cosine-sim. GO/MeSH terms - LARGEST cluster	0.0167	0.0287	0.0012	0.00046
Cosine-sim. GO/MeSH terms - SMALL clusters	0.0449	0.0906	0.026	0.0098
Cosine-sim. original passages - LARGEST cluster	0.00562	0.01194	0.00084	0.00029
Cosine-sim. original passages - SMALL clusters	0.0458	0.1012	0.0303	0.0119
Hamming-distance GO/MeSH terms - SMALL clusters	0.02960	0.05902	0.01614	0.00565
Hamming-distance original passages - SMALL clusters	0.0231	0.07093	0.0206	0.00401
Biclustering GO/MeSH terms - SMALL clusters	**0.0749**	**0.1238**	**0.0591**	**0.0225**

consider the smallest clusters (size ≤ 4) to be the most relevant passages; however the smallest clusters may still exhibit insignificant similarity relative to the similarities found between other passages in the dataset. This suggests that one should consider a similarity in relation to the overall similarities found in a dataset.

Table 2 shows that for the Hamming-distance and cosine-similarity clusterings, taking the small clusters as relevant passages improves the results over taking the larger clusters. The reason is that in the smallest clusters the prominent words are more relevant to the topic than in the largest cluster. This especially holds true for topics with many aspects, where the smallest clusters are more likely to cover different aspects. For Hamming-distance clustering we notice a better result than for cosine similarity clustering, which can be explained by the gradually relaxing threshold making objects in small clusters to be similar to one another. Nevertheless, Hamming-distance clustering performance was not better than HIERDENC passage ranking.

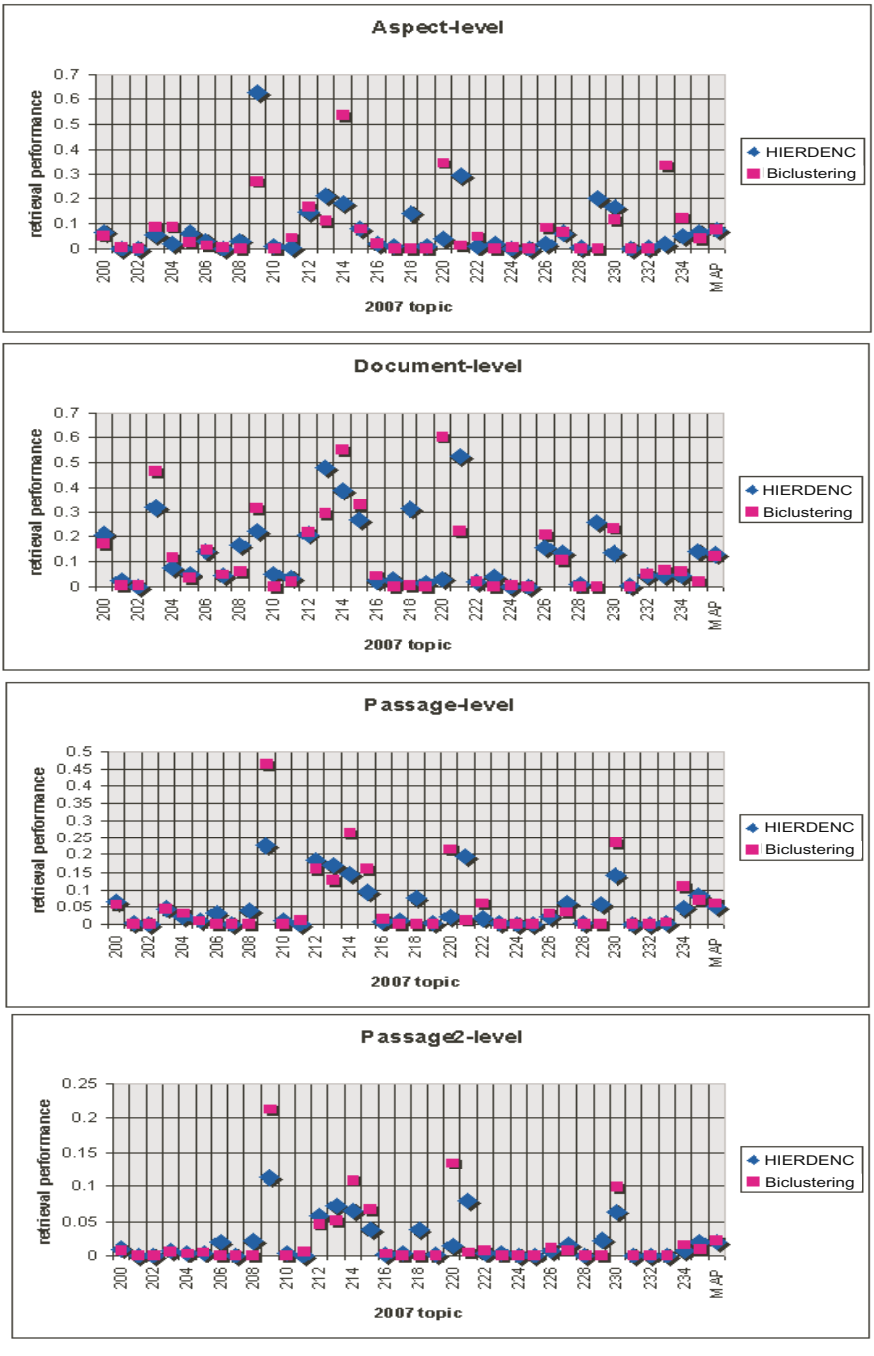


Fig. 7 All results for HIERDENC and biclustering across the 2007 topics (x-axis); *a*. Aspect-level, *b*. Document-level, *c*. Passage-level, and *d*. Passage2-level retrieval performance

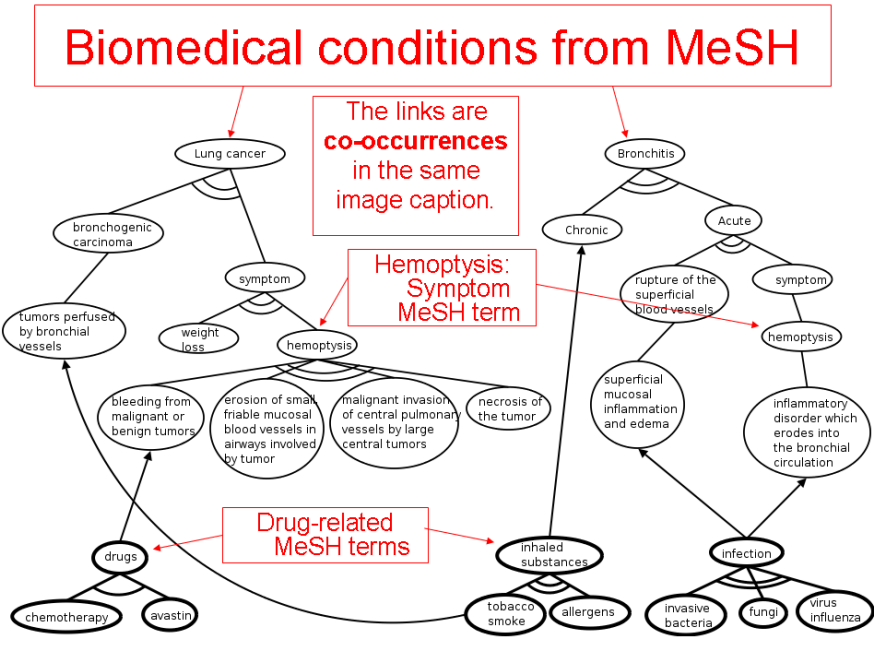


Fig. 8 Ontological terms that co-occur in the same passage (or image caption) indicate that both "Lung cancer" and "Bronchitis" involve the "Hemoptysis" symptom. Hemoptysis could be a symptom of lung cancer or bronchitis and in the former case there could be several causes (29). Serious hemoptysis often occurs in patients with lung cancer when treated with chemotherapy and Avastin. In this case, the incidence of hemoptysis is relatively high in patients receiving chemotherapy and Avastin, as compared to no cases in patients treated with chemotherapy alone. This figure shows that hemoptysis could also be a symptom of bronchitis; in this case it is mild and self-limited. Bronchitis is often a *viral* or *bacterial* disease which follows a cold or infection. A physician who diagnoses hemoptysis as a symptom of bronchitis may be wrong, since the patient could suffer from lung cancer instead. In fact, physicians who work long hours frequently make such errors. How can a physician tell which of all possible conditions holds for a patient with hemoptysis? With a unifying framework to integrate hemoptysis information online for fast lookup and analysis, a physician could make more informed decisions concerning the underlying cause of hemoptysis in a patient

Table 2 shows that in all cases converting passages to GO/MeSH ontological terms improved the result. Clustering the GO/MeSH terms extracted from passages has a significant effect, since stop word removal eliminated phrases like "of the" and "in the" that were considered prominent in the original top-ranked passages. Extracting ontological terms helps us to keep the semantically meaningful words for each passage that are more representative of the aspects.

We noticed more meaningful phrases in the top-ranked passages for GO/MeSH extracted terms than for the original passages.

HIERDENC vs. Biclustering Aspect-level Performance. We compare the performance of HIERDENC on each topic to biclustering that returns two clusters separating the relevant passages; for biclustering we use only extracted ontological terms from passages because of the huge sizes of the resulting matrices for the original passages.

Figure 7 shows all results for HIERDENC and biclustering on the various topics. The results often differ by topic. HIERDENC outperforms biclustering on topics with many aspects of retrieved passages. For topics with few (one or two) aspects, biclustering often outperforms HIERDENC, because of its focus on returning two clusters that are dissimilar. Therefore, biclustering succeeds in separating the relevant from the irrelevant passages, resulting in higher aspect-level retrieval performance values. The tradeoff between using HIERDENC vs. biclustering is that the former ranks all passages, while the latter returns only a subset of the passages predicted to be relevant; for some topics biclustering resulted in < 50 passages predicted to be relevant. For example, HIERDENC gave high aspect-level performance on the 2007 topic#10 (209) "What [BIOLOGICAL SUBSTANCES] have been used to measure toxicity in response to etidronate?". For this topic, the top 5 ranked passages by HIERDENC covered the diverse aspects of women, tumors, surgery, responses, and detoxification.

For some 2007 topics, HIERDENC gave significantly improved aspect-level performance over the original passage retrieval without using any clustering (4). An example is the topic#25 (224) "What [GENES] are involved in the melanogenesis of human lung cancers?". These topics request identifiers, allowing HIERDENC to use the surrounding terms of the identifiers to find the aspects.

5 Conclusions and Future Work

5.1 Conclusions

We complemented the HIERDENC clustering algorithm with an index that supports scalable ranking of textual passages, such as image captions. The top-ranked passages cover diverse aspects of a question on a topic. HIERDENC is useful for ranking passages for presentation to the user assuming several aspects exist, improving upon the results of other clustering methods. For topics with diverse aspects, HIERDENC results in improved aspect-level performance. HIERDENC is scalable to large and quickly growing datasets, such as the PubMed biomedical literature databank and biomedical image captions. Further benefits of HIERDENC include: no re-clustering needed when new text is presented, no user-specified input parameters required, and insensitivity to

ordering of passages. Other methods such as biclustering may be more useful for separating a subset of passages that are believed to be relevant for a topic. For all methods, using the GO/MeSH ontological term vectors extracted from text is likely to improve the aspect-level performance. This work provides guidelines for using clustering to improve aspect-level performance.

5.2 Future Work

Our methodology has potential to be extended to large biomedical image databanks, by using the textual image captions as passages. We are currently putting online a system for retrieving BMC biomedical images based on their captions `http://www.hierdenc.com` or `http://141.30.193.12/HIERDENC/images.html`.

Figure 3*b* showed that the ontological term extraction can also serve another purpose besides clustering and aspect-level performance: ontology linking if a pair of ontological terms from two different ontologies that co-occur in the same passage (or image caption). The main idea is to link ontological terms α and β from different ontologies or vocabularies if α and β co-occur in the same passage. We capture the following relations described in biomedical text, as shown in Figure 3*b*: a protein or drug contributes to a Gene Ontology biological process occurring over time; the GO biological process contributes to a MeSH medical condition; consequently the proteins contribute indirectly to the MeSH medical condition. The ultimate purpose of linking ontologies on the basis of co-occurring terms in passages is to reason over the information contained in biomedical text in a simpler manner than current semantic web-based integration frameworks would allow (22; 23; 24). Figure 8 shows a case of finding which medical condition is the most probable cause of a symptom. Suppose a patient is observed with the symptom *hemoptysis*, the act of coughing up blood (Figure 8). Hemoptysis is often a sign of lung cancer, but it may be caused by different underlying events in lung cancer patients. Hemoptysis also occurs in patients with acute or chronic bronchitis, as well as tuberculosis and pneumonia (29). Determining the cause of hemoptysis is often not a trivial matter. Figure 8 shows that a physician could use the linked ontologies over passages to find if the cause of hemoptysis in a patient is likely to be bronchitis or lung cancer. Linking ontologies is a step towards reasoning over existing medical knowledge, which may allow a physician to relate observed symptoms to a known medical condition, or find likely side-effects of a drug (30; 31; 32).

Acknowledgements. We are grateful for the financial support of the Natural Science and Engineering Research Council (NSERC), the Ontario Graduate Scholarship (OGS), the EU Sealife project, Dresden-exists, and the Nanobrain project.

References

[1] Vanteru, B.C., Shaik, J.S., Yeasin, M.: Semantically linking and browsing pubmed abstracts with gene ontology. BMC Genomics 9(suppl. 1), S10 (2008)

[2] Doms, A., Schroeder, M.: Gopubmed: exploring pubmed with the gene ontology. Nucleic Acids Res. 33(web server issue), W783–W786 (2005)

[3] Hersh, W., Cohen, A.M., Roberts, P.: TREC 2007 Genomics Track Overview. In: Proceedings of 16th Text REtrieval Conference. NIST Special Publication (2007)

[4] Huang, X., Hu, B., Rohian, H.: York University at TREC 2006: Genomics Track. In: Proceedings of 15th Text REtrieval Conference (2006)

[5] Deselaers, T., Mueller, H., Clogh, P., Ney, H., Lehmann, T.: The clef 2005 automatic medical image annotation task. International Journal of Computer Vision 74(1), 51–58 (2007)

[6] Goldberg, A., Andrzejewski, D., Van Gael, J., Settles, B., Zhu, X., Craven, M.: Ranking biomedical passages for relevance and diversity, uw-madison at trec genomics 2006. In: 15th Text Retrieval Conference, TREC 2006 (2006)

[7] Zhong, M., Huang, X.: Concept-based biomedical text retrieval. In: Proceedings of ACM SIGIR 2006 Conference (2006)

[8] Si, L., Lu, J., Callan, J.: Combining multiple resources, evidence and criteria for genomic information retrieval. In: 15th TREC Conference (2006)

[9] Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partitioning. In: 7th ACM SIGKDD, pp. 269–274 (2001)

[10] Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)

[11] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)

[12] Andreopoulos, B., An, A., Wang, X., Faloutsos, M., Schroeder, M.: Clustering by common friends finds locally significant proteins mediating modules. Bioinformatics (Feburary 2007)

[13] Andreopoulos, B., An, A., Wang, X.: Hierarchical density-based clustering of categorical data and a simplification. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 11–22. Springer, Heidelberg (2007)

[14] Stairmand, M.A.: Textual Conext Analysis for Information Retrieval. In: Proceedings of the 1997 ACM SIGIR Conference (1997)

[15] Voorhees, E.M.: Query Expansion Using Lexical-Semantic Relations. In: Proceedings of the 1994 ACM SIGIR Conference (1994)

[16] Hersh, W., Cohen, A., Yang, J.: TREC 2005 Genomics Track Overview. In: Proceedings of 14th Text REtrieval Conference. NIST Special Publication (2005)

[17] Hersh, W., Cohen, A.M., Roberts, P.: TREC 2006 Genomics Track Overview. In: Proceedings of 15th Text REtrieval Conference. NIST Special Publication (2006)

[18] Huang, X., Zhong, M., Si, L.: York University at TREC 2005: Genomics Track.. In: Proceedings of the 14th Text Retrieval Conference (2005)

[19] Zhou, W., Yu, C., Neil, S., Vetle, T., Jie, H.: Knowledge-Intensive Conceptual Retrieval and Passage Extraction of Biomedical Literature. In: Proceedings of the 30th ACM SIGIR Conference (2007)

[20] Burek, P., Hoehndorf, R., Loebe, F., Visagie, J., Herre, H., Kelso, J.: A top-level ontology of functions and its application in the open biomedical ontologies. Bioinformatics 22(14), e66–e73 (2006)
[21] Garcia-Sanchez, F., Fernandez-Breis, J., Valencia-Garcia, R., Gomez, J., Martinez-Bejar, R.: Combining semantic web technologies with multi-agent systems for integrated access to biological resources. J. Biomed. Inform. 41(5), 848–859 (2008)
[22] Smith, A., Cheung, K., Krauthammer, M., Schultz, M., Gerstein, M.: Leveraging the structure of the semantic web to enhance information retrieval for proteomics. Bioinformatics 23(22), 3073–3079 (2007)
[23] Smith, A.K., Cheung, K.H., Yip, K.Y., Schultz, M., Gerstein, M.K.: Linkhub: a semantic web system that facilitates cross-database queries and information retrieval in proteomics. BMC Bioinformatics 8(suppl. 3), S5 (2007)
[24] Ruttenberg, A., Clark, T., Bug, W., Samwald, M., Bodenreider, O., Chen, H., Doherty, D., Forsberg, K., Gao, Y., Kashyap, V., Kinoshita, J., Luciano, J., Marshall, M.S., Ogbuji, C., Rees, J., Stephens, S., Wong, G.T., Wu, E., Zaccagnini, D., Hongsermeier, T., Neumann, E., Herman, I., Cheung, K.H.: Advancing translational research with the semantic web. BMC Bioinformatics 8(suppl. 3), S2 (2007)
[25] Andreopoulos, B., An, A., Wang, X., Labudde, D.: Efficient layered density-based clustering of categorical data. Elsevier Journal of Biomedical Informatics (2009) (in press)
[26] Beaulieu, M., Gatford, M., Huang, X., Robertson, S., Walker, S., Williams, P.: Okapi at TREC-5. In: Proc. of TREC-5. NIST Special Publication (1997)
[27] Huang, X., Peng, F., Schuurmans, D., Cercone, N., Robertson, S.: Applying machine learning to text segmentation for information retrieval. Information Retrieval Journal 6(4), 333–362 (2003)
[28] Huang, X., Huang, Y., Wen, M., Zhong, M.: York Univeristy at TREC 2004: Genomics and HARD Tracks. In: Proceedings of TREC-13. NIST Spec. Publ. (2004)
[29] Fartoukh, M., Khalil, A., Louis, L., Carette, M.F., Bazelly, B., Cadranel, J., Mayaud, C., Parrot, A.: An integrated approach to diagnosis and management of severe haemoptysis in patients admitted to the intensive care unit: a case series from a referral centre. Respir Res. 8, 11 (2007)
[30] Badea, L., Tilivea, D., Hotaran, A.: Semantic web reasoning for ontology-based integration of resources. In: Ohlbach, H.J., Schaffert, S. (eds.) PPSWR 2004. LNCS, vol. 3208, pp. 61–75. Springer, Heidelberg (2004)
[31] Brazhnik, O., Jones, J.: Anatomy of data integration. J. Biomed. Inform. 40(3), 252–269 (2007)
[32] Koehler, J., Philippi, S., Lange, M.: Semeda: ontology based semantic integration of biological databases. Bioinformatics 19(18), 2420–2427 (2003)

Author Index