In: Technical Report, volume 463. University of Wuerzburg, 2009.

An Extensible Architecture for Wiki-Enabled Semantic Data Mining

Martin Atzmueller, Florian Lemmerich, Jochen Reutelshoefer, and Frank Puppe

University of Würzburg,
Department of Computer Science,
Am Hubland, 97074 Würzburg, Germany
{atzmueller,lemmerich,reutelshoefer,puppe}@informatik.uni-wuerzburg.de

Abstract. Wikis provide powerful and flexible means for content creation, generation and management. Complementing semantic data mining systems by wikis and especially semantic wikis yields a flexible knowledge-rich approach. This paper presents an extensible architecture for accessing and embedding semantic data mining systems using wiki technology. We present the components and describe their interaction and application in detail.

1 Introduction

Wikis provide flexible ways for supporting the quick and simple creation, sharing, and management of content. Based upon the established wiki-technology, semantic wikis (e.g., [1,2]) enhance this by providing enriched content and features. For example, flexible inline queries and according results that are generated based on these dynamically are such prominent features. While the queries and answers (results) can be flexibly handled by the system, and can usually be formalized as textual content, the wiki system also provides appropriate means for the persistent storage, versioning management of content and elaborated access control for different user groups.

Semantic data mining systems enable the inclusion of a large set of background knowledge, for example, in order to access knowledge services, for selecting the applied data mining methods, or for postprocessing the obtained data mining results. Thus, integrating wikis is a convenient option for semantic data mining systems, since the semantic core components can support the semantic mining features, while the wiki component provides for a convenient front-end and user-management, enables the persistent storage of queries and mining results, and supports their extended annotation.

This paper presents a wiki-enabled approach for collaborative semantic data mining: The semantic data mining system VIKAMINE [3] is combined with the JSPWiki system (`http://www.jspwiki.org`) and its semantic extension KnowWE [1]. We describe the interaction and exchange of query and results data, and the integration of semantic information and knowledge. An exemplary implementation is given by the KNOWTA system (`http://www.knowta.de`).

The rest of the paper is structured as follows: Section 2 describes the basics of the data mining approach and discusses related work. After that, Section 3 provides an overview of the presented approach, while Section 4 describes its implementation. Finally, Section 5 concludes with a summary and interesting directions for future work.

2 Preliminaries

The presented approach can be considered on two layers: On the first layer, we present a general approach for wiki-enabled semantic data mining. The second (more detailed) layer concerns the instantiation of the approach using concrete implementation, specific methods, and systems. This section briefly introduces the necessary preliminaries: First, the general semantic data mining approach is outlined. After that, we introduce the necessary notation concerning the used knowledge representation and describe the applied core data mining method, that is, subgroup discovery.

2.1 Semantic Data Mining

Semantic data mining can be considered as an approach utilizing formal methods and techniques in order to explicitly integrate data semantics, background knowledge, or reasoning in the mining process. The knowledge is typically represented in a knowledge repository, such as an ontology, or a knowledge base. The main aspect of semantic data mining is the explicit integration of this knowledge into the data mining and knowledge discovery process, where the algorithms for data pre-processing, mining or post-processing make use of the formalized knowledge to improve the overall process. There has been growing interest in this issue, e.g., [4–6], in various domains, especially in the medical domain [4, 7, 8].

With the advent of the semantic web and standardized knowledge representations of semantic web techniques, e.g., the web ontology language *OWL*, utilizing these knowledge representation formalisms for data mining is a promising direction for task design, evaluation and refinement, as discussed below. In the following, we outline the different aspects of semantic data mining, and discuss their implications.

The general data mining process can be structured along the CRISP-DM process model (`http://www.crisp-dm.org`) and consists of the following phases:

1. Business understanding, i.e., understanding the application domain,
2. Data Understanding, i.e., considering the (potential) objects of analysis,
3. Data Preparation, e.g., pre-processing and schema-matching of the data elements,
4. Modeling, e.g., given by concrete mining sessions,
5. Evaluation, i.e., assessment of the mined models,
6. Deployment, i.e., putting the extracted knowledge into action.

The semantic data mining approach integrates ontologies in each of the six steps [7]. In the following, we provide examples for each of the phases structured along the dimensions of task design, evaluation, and refinement.

- **Task Design**:
  - In the **Business Understanding** phase ontologies help inexperienced users getting accustomed to the domain, by structuring the relations between the concepts, and explaining the concepts, e.g., in terms of their properties and/or their relations and connecting paths within the ontology.

  - In the **Data Understanding** phase, important data elements (contained in the ontology) need to be selected. Then, missing attributes, or redundant attributes can be added or removed from the data set. This can be accomplished by a *data-to-ontology mapping* step [5] where the data elements are mapped to concepts of the ontology, e.g., for integrating heterogenous data.
  - The **Data Preparation** phase is strongly connected to the *Modeling* phase. Depending on the latter, for example constraints on attributes or values can be derived. This concerns constraints on the relations between the attributes, as described in [4], for example, grouping constraints or exclusion constraints for certain attribute groups that should not be considered. A further possible inclusion of the ontology is given by a more abstract task composition phase, for which the modeling phase can be hierarchically decomposed along the generalization/specialization hierarchies modeled in the ontology. Then, more concise results can potentially be obtained on lower levels, but for efficiency reasons higher levels can be considered first and be used for filtering interesting hypotheses in an earlier stage, cf., [5].
- **Task Evaluation**:
  - During the **Evaluation** phase (of CRISP-DM), the discovered patterns can be interpreted and explained in a structured way using the concepts and/or contained patterns. Various post-processing options are available at this point, cf., [5]. Specifically, due to the data-to-ontology mapping, the discovered patterns can be matched to semantic relations or more complex relations between these. Additionally, such knowledge provides a potential (explaining) context for the discovered patterns. Furthermore, prior knowledge can be compared to the patterns, e.g., for confirming known relations, identifying new knowledge, and/or detecting exceptions and conflicts with formalized expectations. Concerning possible explanations, causal relations can often help in this respect, for validating and confirming discovered patterns, or for their analysis.
  - The **Deployment** phase concerns the integration of the discovered models into the business setting. It is easy to see, that for distributed processing and storage (e.g., on the semantic web) a shared ontology is inevitable. This is especially relevant for deploying results as *semantic analytic reports* (an extension of *analytic reports* [5]), described below. In a late evaluation step, the models/patterns can be tested during their practical application. In that case, the persistent sessions stored in the wiki provide direct access in a collaborative manner.
- **Task Refinement**: The task refinement step is activated after the evaluation step has been performed. It is accomplished either manually using the wiki system – by modifying the textual task description, or by applying formalized knowledge with respect to the applied data mining method. Then, parameters and/or the method itself can be adapted. Refinement is performed according to the results of the *evaluation* phase, so both steps are tightly coupled. Due to the application of the wiki, different persons can collaborate in separate sessions, such that previous results can be included in the refinement of other (related) sessions. Furthermore, previous experiences can be documented using the wiki, for example, explanations/comments by previous users. Furthermore, special refinement and/or evaluation knowledge can be formalized for further improving the respective steps.

2.2 Basic Definitions

Let Ω_A be the set of all nominal attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined. Furthermore, we assume $\mathcal{V}_A$ to be the (universal) set of attribute values of the form $(a = v)$, where $a \in \Omega_A$ is an attribute and $v \in dom(a)$ is an assignable value. Let CB be the case base (data set) containing all available cases, also often called instances. A case $c \in CB$ is given by the n-tuple $c = ((a_1 = v_1), (a_2 = v_2), \ldots, (a_n = v_n))$ of $n = |\Omega_A|$ attribute values, $v_i \in dom(a_i)$ for each a_i.

2.3 Subgroup Discovery Basics

The main application areas of subgroup discovery are exploration and descriptive induction, to obtain the k best relations (given by subgroups) between a (dependent) target variable and a set of explaining (independent) variables. Similar to the *MIDOS* approach [9] we try to identify subgroups that are, e.g., as large as possible, and have the most unusual (distributional) characteristics with respect to a given concept of interest represented by a (binary) target variable. The description language specifies the individuals belonging to the subgroup. The subgroup is thus given by all cases in the data set that satisfy its subgroup description. For a commonly applied single-relational propositional language a subgroup description can be defined as follows:

Definition 1 (Subgroup Description). *A subgroup description $sd = \{e_1, e_2, \ldots, e_n\}$ is defined by the conjunction of a set of selectors. These $e_i = (a_i, V_i)$ are selections on domains of attributes, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. We define Ω_E as the set of all selection expressions and Ω_{sd} as the set of all possible subgroup descriptions.*

A quality function ranks a subgroup by measuring its interestingness.

Definition 2 (Quality Function). *Given a particular target variable $t \in \Omega_E$, a quality function $q : \Omega_{sd} \times \Omega_E \rightarrow R$ is used in order to evaluate a subgroup description $sd \in \Omega_{sd}$, and to rank the discovered subgroups during search.*

In comparison to the strict support/confidence framework applied for association rule mining, e.g., [10], the subgroup quality functions can be flexibly defined. For binary target variables, examples for quality functions are given by

$$q_B = \frac{(p - p_0) \cdot \sqrt{n}}{\sqrt{p_0 \cdot (1 - p_0)}}\,, \quad q_{RG} = \frac{p - p_0}{p_0 \cdot (1 - p_0)}\,,$$

where p is the relative frequency of the target variable in the subgroup, p_0 is the relative frequency of the target variable in the total population, $N = |CB|$ is the size of the total population, and n denotes the size of the subgroup. Often, a minimal support threshold $n \geq \mathcal{T}_{Supp}$ is also applied.

Pattern rules are applied for formalizing subgroup patterns in a rule-like manner and an associated quantitative quality rating.

Definition 3 (Pattern Rule). *A pattern rule $r = B(r) \rightarrow H(r)\ [q(r)]$ is defined by the body $B(r)$ and the head $H(r)$ of the pattern rule, where $B(r) \subseteq \Omega_E, H(r) \subseteq \Omega_E$ for which the selectors are combined conjunctively, i.e., $e_1 \wedge \cdots \wedge e_k$, $(e_i \in \Omega_E, i = 1 \ldots k)$. A quality parameter $q(r) \in \mathbb{R}$ is assigned to the pattern r denoting its respective quality.*

3 Overview

This section provides an overview on the proposed approach by first introducing the problem statement. After that, an architectural overview on an exemplary application is given, which is subsequently discussed in an application example. Finally, we conclude this section with a general discussion and the review of related work.

3.1 General Overview

As discussed in the last section *semantic data mining* is concerned with the utilization of ontological knowledge and semantic annotations to be used throughout the data mining and knowledge discovery process, similar to *ontology-enhanced* [5] data mining. However, further semantic features are enabled by including a *semantic core* component, e.g., a RDF-Store: Using that, results can be incrementally formalized and provided to the store, while subsequent mining and semantic queries can make use of the collected knowledge. The data mining query, results, and additional knowledge can then be transparently integrated into a *semantic analytic report* [11]: The idea of such reports is based on *analytical reports* [5] that are simple text documents containing the mining results with additional text (which is created by humans). In the semantic setting, we can automatically transform the mining results into a format suitable for the report. Additionally, the content can be enriched using semantic annotations and links between the reports (and background information), and multi-modal information. The wiki also provides for flexible versioning which is especially useful in a collaborative setting.

The sketched scenario is especially suitable for inexperienced users that are mainly interested in reporting features of a data mining system. Such reports provide high-level access to pre-specified queries that can be evaluated routinely. However, using the wiki query mechanism, such queries can also be formalized in an ad-hoc fashion. Further more detailed reports, analyses and mining sessions can then be implemented using more advanced data mining tools, e.g., by applying the VIKAMINE [3] system.

On the application side, specialized sessions with domain experts, e.g., medical doctors, and data mining engineers can be easily implemented using the collaborative tool. In this context, the proposed approach provides, for example, flexible query formalization, versioning, a history of queries and results, and the potential for knowledge and experience management since the obtained semantic analytical reports can be commented on, and can be linked to other (similar) documents. Further sessions can thus easily build on results of previous sessions, with the same or new participants. For experience management, the wiki can also be complemented with a tagging system, e.g., [12].

3.2 Architectural Overview

The architecture consists of three core components: The basic wiki system (provided by JSPWiki (`http:/www.jspwiki.org`) is extended by the semantic wiki extension KnowWE [1]. The semantic core component is given by *Sesame* and OWLIM. For data mining component, i.e., the mining engine we utilize the mining kernel of the VIKAMINE [3] system (`http://www.vikamine.org`).

The wiki component provides basic features like editing, versioning, user management, access management and attachment management. Additionally, it directly supports the collaborative aspects of the sketched semantic mining approach. KnowWE itself is designed as a highly extensible minimal core providing basic semantic wiki features like formalization and reasoning. Therefore, for communication with the mining component we designed the connector plugin *KnowWE-RIP* (REST [13] Interface Plugin) that facilitates the connection to the mining web-service. The semantic core component for storage and reasoning is given by a combination of the *Sesame* (`http://www.openrdf.org`) framework and OWLIM. Sesame is a java-based framework with support for storing and analyzing RDF data. OWLIM is a semantic repository with reasoning capabilities that is packaged as a storage and inference layer for sesame. As such, KnowWE integrates a semantic component and contains a connector to the Sesame/OWLIM components for providing the semantic functionality.

We utilize the VIKAMINE [3] system (`http://www.vikamine.org`) for data mining. VIKAMINE features a web-service that can be queried using XML based on a specialized query language. The result (i.e., the answer) is also formulated as XML and can thus transparently be integrated with the wiki.

The semantic mining process is initiated by the user, that is, by formulating a query to the wiki system. Similar to other wiki-systems, the query is provided in the form of an *inline-query* (e.g., [2]): The query is directly entered in textual form. Whenever the wiki page is stored and/or reloaded with a new or modified query the result is requested. In addition, we provide 'extended' inline queries, such both the query and the result (i.e., the 'answer') can also be shown as required. Technically, the query is first transformed to an XML-representation (VPDL, the VIKAMINE *Pattern Description Language*, and then forwarded to the mining engine that produces an result in XML/VPDL format. Finally, this result is re-transformed into human-readable textual form to be displayed by the wiki. However, internally the 'raw' result can be retained by the versioning system of the wiki, such that always the latest result is available and can be cached for efficiency. Therefore, changes, for example, due to an updated dataset, can be easily extracted. The general architecture is shown in Figure 1. The seamless integration of the result presentation enables (inexperienced) users to quickly evaluate the obtained results by themselves and according to the formalized ontological knowledge.

3.3 Related Work and Discussion

Using ontologies for enhancing data mining has been discussed, e.g., by Svatek et al. [5] and by Antunes [6] in the context of mining association rules. Furthermore, Cespivova et al. [7] and Kuo et al. [8] describe applications in the medical domain. While the application of ontologies is also a focus of the presented approach, the proposed method aims at a more comprehensive integration of semantic information and knowledge. In contrast to the existing approaches, the proposed approach considers a comprehensive *two-way* integration of semantic and data mining methods, with feedback in both directions. In this way, prior knowledge can be transparently integrated. Using the wiki-support of the presented approach collaborative sessions can be implemented. Furthermore, semantic annotations linking unstructured, semi-structured and structured information of the wiki content is another novel issue with respect to the presented approach. Semantic

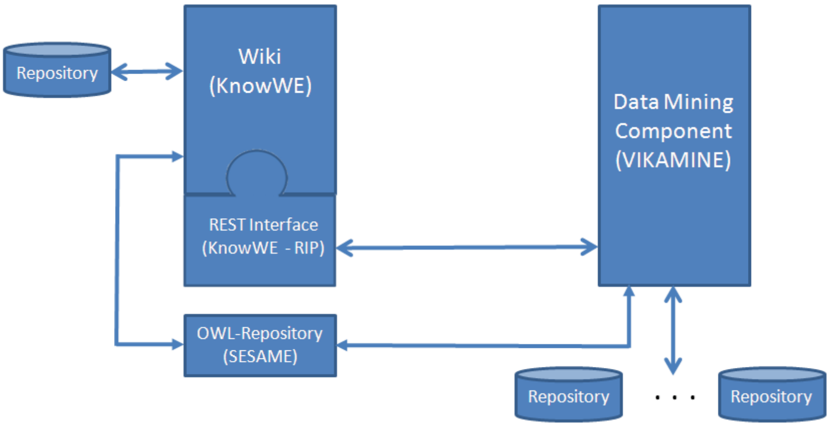


Fig. 1. Semantic Data Mining Architecture

analytical reports can include semantic annotations at the document level, global tagging, and associated query – data mining results are stored in the semantic store and thus provide powerful options for knowledge-rich applications.

4 Implementation

In the following, we describe the core components of the proposed architecture. First, we discuss the integration of JSPWiki/KnowWE, describe the semantic integration of Sesame and OWLIM, before we discuss VIKAMINE and the used description language VPDL (Vikamine Pattern Description Language). All used implementation components are available under open-source licenses.

4.1 JSPWiki/KnowWE

For the wiki component we are using the *JSPWiki* system that is complemented by the semantic wiki extension *KnowWE*. It provides a reliable basis with respect to basic wiki features like editing, versioning, user management, access management and attachment management. Different components like markup, editors, plugin-tags, renderers and compile-handlers can be introduced to the system or combined with already existing components. The main idea behind this architecture is to be able to support any semantic wiki application with a customized set of domain and user specific features at low engineering costs [1]. KnowWE is meant to support knowledge engineering by providing knowledge in various different formats and formalisms, and at different degrees of formality (e.g., plain text, tabular data, bullet lists, annotations, rules, models,...). KnowWE is designed as a toolkit that allows to introduce new syntactical formats and corresponding compilation hooks, and can be extended by customized markup for new types of knowledge. Also, the user-context, group-membership and access rights are transparently passed to any component in KnowWE for supporting the creation of user-specific web interfaces.

4.2 Sesame / OWLIM

For the management of RDF-data in KnowWE the *Sesame*[1] framework is used. Sesame is a java-based framework with support for storing and analyzing RDF data. OWLIM is a semantic repository with reasoning capabilities that is packaged as a storage and inference layer for sesame. It is developed by Ontotext and the profile *SwiftOWLIM* is available under LGPL license. Beyond RDFS reasoning OWLIM provides built-ins for rule-based inferencing (e.g., OWL-Horst [14]). At the time of writing OWLIM claims to be the fastest and most scalable RDF(S)/OWL reasoning engine available.

4.3 VIKAMINE

In general, the data mining technique of subgroup mining [15] is quite suitable for a variety of analytical questions. We used the VIKAMINE (Visual, Interactive and Knowledge-Intensive Analysis and Mining Environment) system [16] for interactive and automatic subgroup mining. VIKAMINE provides rich semantic data mining capabilities based on the technique of subgroup discovery. VIKAMINE provides both automatic and interactive mining and analysis techniques. The graphical interface of VIKAMINE is given by an Eclipse-based rich-client. Additionally, the system provides the mining kernel, which can be accessed as a web-service.

For the proposed approach, we utilize the latter for a REST-based approach using the *KnowWE-RIP* interface. Furthermore, often background knowledge can be utilized, since existing knowledge should not be rediscovered, but the available knowledge should be used to find new, often subtle correlations, to increase the interestingness of the discovered results. The VIKAMINE system enables powerful semantic mining options by utilizing background knowledge from the ontology. In addition, VIKAMINE offers an efficient exhaustive and various heuristic search options with constraints for automatic subgroup discovery and interactive visualizations for active user involvement. When the user discovers something unexpected/interesting in the data using standard tools, then these findings can be inspected and analyzed in detail using VIKAMINE. For more details, see [16].

4.4 VPDL

As communication language in our implementation we use the Vikamine Pattern Description Language (VPDL). VPDL is a concise XML-language, that is specialized on the task of subgroup discovery. In comparison to well known PMML language, that focuses on predictive models, VPDL is much less verbose, but also adds some additional information, which are specific for subgroup discovery. It consists of two main parts:

The first part of the VPDL language specifies one subgroup discovery task exactly by describing each of the following eight aspects in detail. An examplary XML-file is shown in figure 2.

[1] http://www.openrdf.org/

In the following, we describe the elements of the VPDL task specification:

- The unique *Id* of this mining task.
- The used *dataset* either by an Unified Resource Identifier (URI) or by the path in an implicitly given local repository.
- The *target concept* of the subgroup mining task
- The *initial subgroup*, that specifies the starting point for the search. Only refinements of the initial subgroup will be discovered.
- The *search space*. The attributes and attribute values to be included in the search can be specified.
- The *quality function*, that is used to measure the interestingness of a subgroup.
- The *subgroup discovery algorithm*, that will be applied for this task.
- *Constraints* on the set of discovered subgroups. Such constraints include the number of subgroups discovered, a maximum search depth, a minimum quality or minimum size threshold for a respective subgroup or the suppression of strictly irrelevant subgroups.

The second part of the VPDL specifies the *result* of a subgroup discovery task, that is a set of subgroup patterns, and an optional reference to this task. For each pattern the subgroup description and several statistics are denoted, e.g, the subgroup size, the target share of the respective subgroup, the lift of the target share and the chi-square value.

5 Conclusions

In this paper, we have presented an extensible architecture for wiki-driven semantic data mining. We have described an (abstract) overview on the approach, and have provided a detailed architectural view on the system. A prototypical implementation of the approach is given by the KNOWTA system (`http://www.knowta.de`). The latter utilizes the VIKAMINE system for semantic data mining and includes a connector to the KnowWE system for the Wiki-based capabilities. Using the KNOWWE-RIP plugin, we enable semantic integration into the wiki, and provide data and query interchange between the wiki and the data mining system (VIKAMINE). The prototypical implementation fulfills all the features of the sketched architecture; initial experiments already show the large potential of the presented approach.

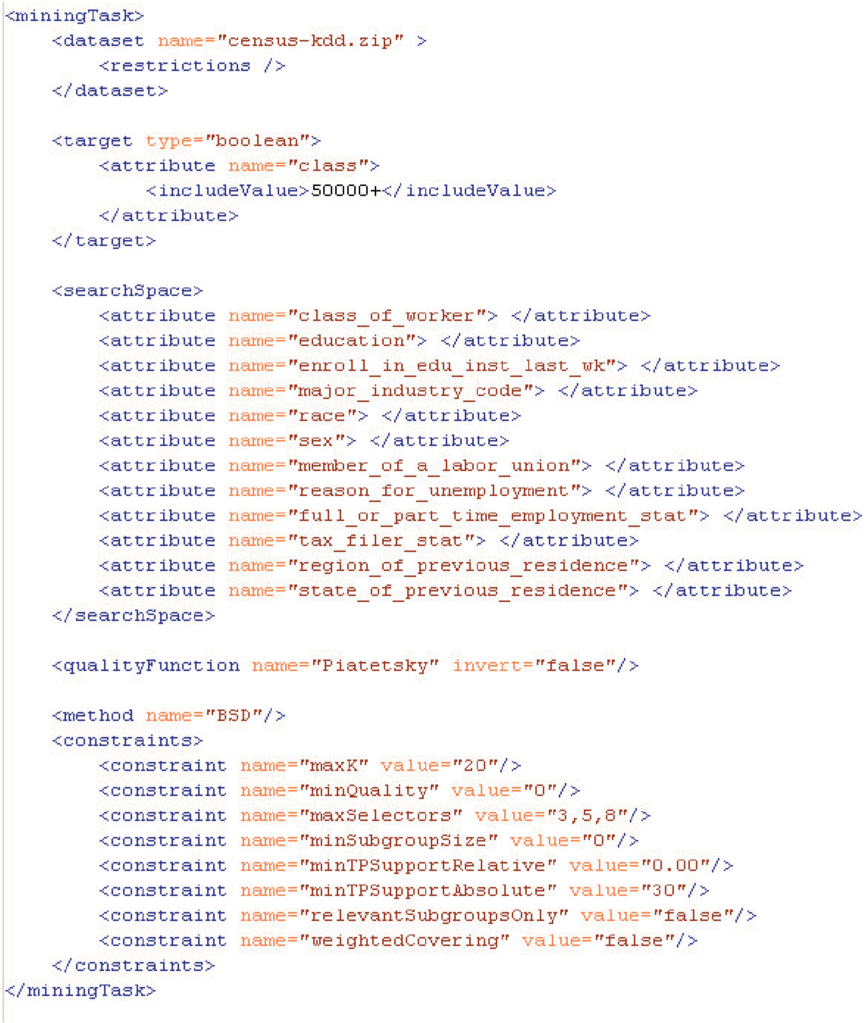
For future work, we aim to consider a knowledge-rich approach utilizing the semantic core capabilities of the semantic wiki: Since knowledge can be easily formalized, and semantic annotations for various types of knowledge can be incrementally added, the application of such knowledge elements for semantic data mining, e.g., preprocessing of mining input, post-processing of mining output, and configuration of the mining method itself, provide for a vast range of powerful mining options. Additionally, we also aim for an integration of other pattern description languages, e.g., PMML [17], or BKEF [11].

Acknowledgements

This work has been partially supported by the German Research Council (DFG) under grant Pu 129/8-2.

References

1. Reutelshoefer, J., Haupt, F., Lemmerich, F., Baumeister, J.: An Extensible Semantic Wiki Architecture. In: Proc. 4th Workshop on Semantic Wikis - The Semantic Wiki Web. (2009)
2. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. Web Semantics: Science, Services and Agents on the World Wide Web **5**(4) (2007) 251 – 261
3. Atzmueller, M., Puppe, F.: Semi-Automatic Visual Subgroup Mining using VIKAMINE. Journal of Universal Computer Science (JUCS), Special Issue on Visual Data Mining **11**(11) (2005) 1752–1765
4. Atzmueller, M., Seipel, D.: Using Declarative Specifications of Domain Knowledge for Descriptive Data Mining. In: Proc. 18th International Conference on Applications of Declarative Programming and Knowledge Management, Berlin, Springer Verlag (2008)
5. Svátek, V., Rauch, J., Ralbovský, M.: Ontology-Enhanced Association Mining. In: Semantics, Web and Mining. Volume 4289 of LNCS. (2005) 163–179
6. Antunes, C.: Onto4AR: A Framework for Mining Association Rules. In: International Workshop on Constraint-Based Mining and Learning (CMILE 2007), Warsaw, Poland, Warsaw University (september 2007)
7. Cespivova, H., Rauch, J., Svatek, V., Kejkula, M.: Roles of Medical Ontology in Association Mining CRISP-DM Cycle. In: Proc. ECML/PKDD 2004 Workshop on Knowledge Discovery and Ontologies, Pisa, Italy (2004)
8. Kuo, Y.T., Lonie, A., Sonenberg, L., Paizis, K.: Domain Ontology Driven Data Mining: A Medical Case Study. In: DDDM '07: Proceedings of the 2007 international workshop on Domain driven data mining, New York, NY, USA, ACM (2007) 11–17
9. Wrobel, S.: An Algorithm for Multi-Relational Discovery of Subgroups. In: Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97), Berlin, Springer Verlag (1997) 78–87
10. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: Proc. 20th Int. Conf. Very Large Data Bases, (VLDB), Morgan Kaufmann (1994) 487–499
11. Kliegr, T., Ralbovsky, M., Svatek, V., Simunuk, M., Jurkovsky, V., Nerava, J., Zemanek, J.: Semantic Analytical Reports: A Framework for Post-processing Data Mining Results. In: Proc. ISMIS 2009: Foundations of Intelligent Systems. Number 5722 in LNAI, Berlin (2009) 88–98
12. Atzmueller, M., Haupt, F., Puppe, F.: Knowta: Wiki-Enabled Social Tagging for Collaborative Knowledge and Experience Management. In: Proc. 2nd International Workshop on Design, Evaluation and Refinement of Intelligent Systems (DERIS). (2009)
13. Fielding, R.T., Taylor, R.N.: Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology **2**(2) (2002) 115–150
14. ter Horst, H.J.: Combining RDF and Part of OWL with Rules: Semantics, Decidability, Complexity. In Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A., eds.: The Semantic Web - ISWC 2005. Number 3729 in Lecture Notes in Computer Science, Heidelberg, Springer Verlag (2005) 668–684
15. Klösgen, W.: 16.3: Subgroup Discovery. In: Handbook of Data Mining and Knowledge Discovery. Oxford Univ. Press, NY (2002)
16. Atzmueller, M.: Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery. Volume 307 of Dissertations in Artificial Intelligence-Infix (Diski). IOS Press (March 2007)
17. Guazzelli, A., Zeller, M., Lin, W., Williams, G.: PMML: An Open Standard for Sharing Models. The R Journal **1**(1) (2009)



```xml
<miningTask>
    <dataset name="census-kdd.zip" >
        <restrictions />
    </dataset>

    <target type="boolean">
        <attribute name="class">
            <includeValue>50000+</includeValue>
        </attribute>
    </target>

    <searchSpace>
        <attribute name="class_of_worker"> </attribute>
        <attribute name="education"> </attribute>
        <attribute name="enroll_in_edu_inst_last_wk"> </attribute>
        <attribute name="major_industry_code"> </attribute>
        <attribute name="race"> </attribute>
        <attribute name="sex"> </attribute>
        <attribute name="member_of_a_labor_union"> </attribute>
        <attribute name="reason_for_unemployment"> </attribute>
        <attribute name="full_or_part_time_employment_stat"> </attribute>
        <attribute name="tax_filer_stat"> </attribute>
        <attribute name="region_of_previous_residence"> </attribute>
        <attribute name="state_of_previous_residence"> </attribute>
    </searchSpace>

    <qualityFunction name="Piatetsky" invert="false"/>

    <method name="BSD"/>
    <constraints>
        <constraint name="maxK" value="20"/>
        <constraint name="minQuality" value="0"/>
        <constraint name="maxSelectors" value="3,5,8"/>
        <constraint name="minSubgroupSize" value="0"/>
        <constraint name="minTPSupportRelative" value="0.00"/>
        <constraint name="minTPSupportAbsolute" value="30"/>
        <constraint name="relevantSubgroupsOnly" value="false"/>
        <constraint name="weightedCovering" value="false"/>
    </constraints>
</miningTask>
```

Fig. 2. An Example of a VPDL task file