1

Service-Oriented Data Mining

Derya Birant
Dokuz Eylul University,
Turkey

1. Introduction

A *service* is a software building block capable of fulfilling a given task or a distinct business function through a well-defined interface, loosely-coupled interface. Services are like "black boxes", since they operate independently within the system, external components are not aware of how they perform their function, they only care that they return the expected result.

The *Service Oriented Architecture* (SOA) is a flexible set of design principles used for building flexible, modular, and interoperable software applications. SOA represents a standard model for resource sharing in distributed systems and offers a generic framework towards the integration of diverse systems. Thus, information technology strategy is turning to SOA in order to make better use of current resources, adapt to more rapidly changes and larger development. Another principle of SOA is the reusable software components within different applications and processes.

A *Web Service* (WS) is a collection of functions that are packaged as a single entity and published to the network for use by other applications through a standard protocol. It offers the possibility of transparent integration between heterogeneous platforms and applications. The popularity of web services is mainly due to the availability of web service standards and the adoption of universally accepted technologies, including XML, SOAP, WSDL and UDDI.

The most important implementation of SOA is represented by web services. *Web service-based SOAs* are now widely accepted for on-demand computing as well as for developing more interoperable systems. They provide integration of computational services that can communicate and coordinate with each other to perform goal-directed activities.

Among intelligent systems, *Data Mining* (DM) has been the center of much attention, because it focuses on extracting useful information from large volumes of data. However, building scalable, extensible, interoperable, modular and easy-to-use data mining systems has proved to be difficult. In response, we propose SOMiner (Service Oriented Miner), a service-oriented architecture for data mining that relies on web services to achieve extensibility and interoperability, offers simple abstractions for users, provides scalability by cutting down overhead on the number of web services ported to the platform and supports computationally intensive processing on large amounts of data.

This chapter proposes SOMiner, a flexible service-oriented data mining architecture that incorporates the main phases of knowledge discovery process: data preprocessing, data mining (model construction), result filtering, model validation and model visualization. This

architecture is composed of generic and specific web services that provide a large collection of machine learning algorithms written for knowledge discovery tasks such as classification, clustering, and association rules, which can be invoked through a common GUI. We developed a platform-independent interface that users are able to browse the available data mining methods provided, and generate models using the chosen method via this interface. SOMiner is designed to handle large volumes of data, high computational demands, and to be able to serve a very high user population.

The main purpose of this chapter is to resolve the problems that appear widely in the current data mining applications, such as low level of resource sharing, difficult to use data mining algorithms one after another and so on. It explores the advantages of service-oriented data mining and proposes a novel system named SOMiner. SOMiner offers the necessary support for the implementation of knowledge discovery workflows and has a workflow engine to enable users to compose KDD services for the solution of a particular problem. One important characteristic separates the SOMiner from its predecessors: it also proposes Semantic Web Services for building a comprehensive high-level framework for distributed knowledge discovery in SOA models.

In this chapter, proposed system has also been illustrated with a case study that data mining algorithms have been used in a service-based architecture by utilizing web services and a knowledge workflow has been constructed to represent potentially repeatable sequences of data mining steps. On the basis of the experimental results, we can conclude that a service-oriented data mining architecture can be effectively used to develop KDD applications.

The remainder of the chapter is organized as follows. Section 2 reviews the literature, discusses the results in the context of related work, presents a background about SOA+Data Mining approach and describes how related work supports the integrated process. Section 3 presents a detailed description of our system, its features and components, then, describes how a client interface interacts with the designed services and specifies the advantages of the new system. Section 4 demonstrates how the proposed model can be used to analyze a real world data, illustrates all levels of system design in details based on a case study and presents the results obtained from experimental studies. Furthermore, it also describes an evaluation of the system based on the case study and discusses preliminary considerations regarding system implementation and performance. Finally, Section 5 provides a short summary, some concluding remarks and possible future works.

2. Background

2.1 Related work

The Web is not the only area that has been mentioned by the SOA paradigm. Also the Grid can provide a framework whereby a great number of services can be dynamically located, managed and securely executed according to the principles of on-demand computing. Since Grids proved effective as platforms for data-intensive computing, some grid-based data mining systems have been proposed such as DataMiningGrid (Stankovski et al., 2008), KnowledgeGrid (K-Grid) (Congiusta et al., 2007), Data Mining Grid Architecture (DMGA) (Perez et al., 2007), GridMiner (Brezany et al., 2005), and Federated Analysis Environment for Heterogeneous Intelligent Mining (FAEHIM) (Ali et al., 2005). A significant difference of these systems from our system (SOMiner) is that they use grid-based solutions and focus on grid-related topics and grid-based aspects such as resource brokering, resource discovery, resource selection, job scheduling and grid security.

Another grid-based and service-based data mining approaches are ChinaGrid (Wu et al., 2009) and Weka4WS (Talia and Trunfio, 2007). A grid middleware ChinaGrid consists of services (data management service, storage resource management service, replication management service, etc.) to offers the fundamental support for data mining applications. Another framework Weka4WS extends the Weka toolkit for supporting distributed data mining on grid environments and for supporting mobile data mining services. Weka4WS adopts the emerging Web Services Resource Framework (WSRF) for accessing remote data mining algorithms and managing distributed computations. In comparison, SOMiner tackles scalability and extensibility problems with availability of web services, without using a grid platform.

Some systems distribute the execution within grid computing environments based on the resource allocation and management provided by a resource broker. For example, Congiusta et al. (2008) introduced a general approach for exploiting grid computing to support distributed data mining by using grids as decentralized high performance platforms where to execute data mining tasks and knowledge discovery algorithms and applications. Talia 2009 discussed a strategy based on the use of services for the design of open distributed knowledge discovery tasks and applications on grids and distributed systems. On the contrary, SOMiner exposes all its functionalities as Web Services, which enable important benefits, such as dynamic service discovery and composition, standard support for authorization and cryptography, and so on.

A few research frameworks currently exist for deploying specific data mining applications on application-specific data. For example, Swain et al. (2010) proposed a distributed system (P-found) that allows scientists to share large volume of protein data i.e. consisting of terabytes and to perform distributed data mining on this dataset. Another example, Jackson et al. (2007) described the development of a Virtual Organisation (VO) to support distributed diagnostics and to address the complex data mining challenges in the condition health monitoring applications. Similarly, Yamany et al. (2010) proposed services (for providing intelligent security), which use three different data mining techniques: the association rules, which helps to predict security attacks, the OnLine Analytical Processing (OLAP) cube, for authorization, and clustering algorithms, which facilitate access control rights representation and automation. However, differently from SOMiner, these works include application-specific services i.e. related to protein folding simulations or condition health monitoring or security attacks.

Research projects such as the Anteater (Guedes et al., 2006) and the DisDaMin (Distributed Data Mining) (Olejnik et al., 2009) have built distributed data mining environments, mainly focusing on parallelism. Anteater uses parallel algorithms for data mining such as parallel implementations of Apriori (for frequent item set mining), ID3 (for building classifiers) and K-Means (for clustering). DisDaMin project was addressed distributed discovery and knowledge discovery through parallelization of data mining tasks. However it is difficult to implement the parallel versions of some data mining algorithms. Thus, SOMiner provides parallelism through the execution of traditional data mining algorithms in parallel with different web services on different nodes.

Several studies mainly related to the implementation details of data mining services on different software development platforms. For example, Du et al. (2008) presented a way to set up a framework for designing the data mining system based on SOA by the use of WCF (Windows Communication Foundation). Similarly, Chen et al. (2006) presented architecture

for data mining metadata web services based on Java Data Ming (JDM) in a grid environment.
Several previous works proposed a service-oriented computing model for data mining by providing a markup language. For example, Discovery Net (Sairafi et al., 2003) provided a Discovery Process Markup Language (DPML) which is an XML-based representation of the workflows. Tsai & Tsai (2005) introduced a Dynamic Data Mining Process (DDMP) system in which web services are dynamically linked using Business Process Execution Language for Web Service (BPEL4WS) to construct a desired data mining process. Their model was described by Predictive Model Markup Language (PMML) for data analysis.
A few works have been done in developing service-based data mining systems for general purposes. On the other side, Ari et al., 2008 integrated data mining models with business services using a SOA to provide real-time Business Intelligence (BI), instead of traditional BI. They accessed and used data mining model predictions via web services from their platform. Their purposes were managing data mining models and making business-critical decisions. While some existing systems such as (Chen et al., 2003) only provide the specialized data mining functionality, SOMiner includes functionality for designing complete knowledge discovery processes such as data preprocessing, pattern evaluation, result filtering and visualization.
Our approach is not similar in many aspects to other studies that provided a service-based middleware for data mining. First, SOMiner has no any restriction with regard to data mining domains, applications, techniques or technology. It supports a simple interface and a service composition mechanism to realize customized data mining processes and to execute a multi-step data mining application, while some systems seem to lack a proper workflow editing and management facility. SOMiner tackles scalability and extensibility problems with availability of web services, without using a grid platform. Besides data mining services, SOMiner provides services implementing the main steps of a KDD process such as data preprocessing, pattern evaluation, result filtering and visualization. Most existing systems don't adequately address all these concerns together.
To the best of our knowledge, none of the existing systems makes use of Semantic Web Services as a technology. Therefore, SOMiner is the first system leveraging Semantic Web Services for building a comprehensive high-level framework for distributed knowledge discovery in SOA models, supporting also the integration of data mining algorithms exposed through an interface that abstracts the technical details of data mining algorithms.

2.2 SOA + data mining

Simple client-server data mining solutions have scalability limitations that are obvious when we consider both multiple large databases and large numbers of users. Furthermore, these solutions require significant computational resources, which might not be widely available. For these reasons, in this study, we propose service-oriented data mining solutions to be able to expand the computing capacity simply and transparently, by just advertising new services through an interface.
On the other side, while traditional Grid systems are rather monolithic, characterized by a rigid structure; the SOA offers a generic approach towards the integration of diverse systems. Additional features of SOA, such as interoperability, self-containment of services, and stateless services, bring more value than a grid-based solution.
In SOA+Data Mining model, SOA enables the assembly of web services through parts of the data mining applications, regardless of their implementation details, deployment location,

and initial objective of their development. In other words, SOA can be viewed as architecture that provides the ability to build data mining applications that can be composed at runtime using already existing web services which can be invoked over a network.

3. Mining in a service-oriented architecture

3.1 SOMiner architecture

This chapter proposes a new system SOMiner (Service Oriented Miner) that offers to users high-level abstractions and a set of web services by which it is possible to integrate resources in a SOA model to support all phases of the knowledge discovery process such as data management, data mining, and knowledge representation. SOMiner is easily extensible due to its use of web services and the natural structure of SOA - just adding new resources (data sets, servers, interfaces and algorithms) by simply advertising them to the application servers.

The SOMiner architecture is based on the standard life cycle of knowledge discovery process. In short, users of the system can be able to understand what data is in which database as well as their meaning, select the data on which they want to work, choose and apply data mining algorithms to the data, have the patterns represented in an intuitive way, receive the evaluation results of patterns mined, and possibly return to any of the previous steps for new tries.

SOMiner is composed of six layers: data layer, application layer, user layer, data mining service layer, semantic layer and complementary service layer. A high speed enterprise service bus integrates all these layers, including data warehouses, web services, users, and business applications.

The SOMiner architecture is depicted in the diagram of Fig. 1. It is an execution environment that is designed and implemented according to a multi-layer structure. All interaction during the processing of a user request happens over the Web, based on a user interface that controls access to the individual services. An example knowledge discovery workflow is as follows: when the *business application* gets a request from a *user*, it firstly calls *data preparation web service* to make dataset ready for data mining task(s), and then related *data mining service(s)* is activated for analyzing data. After that, *evaluation service* is invoked as a complementary service to validate data mining results. Finally, *presentation service* is called to represent knowledge in a manner (i.e. drawing conclusions) as to facilitate inference from data mining results.

Data Layer: The *Data Layer (DL)* is responsible for the publication and searching of data to be mined (data sources), as well as handling metadata describing data sources. In other words, they are responsible for the access interface to data sets and all associated metadata. The metadata are in XML and describes each attribute's type, whether they represent continuous or categorized entities, and other things.

DL includes services: *Data Access Service (DAS)*, *Data Replication Service (DRS)*, and *Data Discovery Service (DDS)*. Additional specific services can also be defined for the data management without changes in the rest of the framework. The *DAS* can retrieve descriptions of the data, transfer bases from one node to another, and execute SQL-based queries on the data. Data can be fed into the DAS from existing data warehouses or from other sources (flat files, data marts, web documents etc.) when it has already been preprocessed, cleaned, and organized. The *DRS* deals with data replication task which is one important aspect related to SOA model. *DDS* improves the discovery phase in SOA for mining applications.

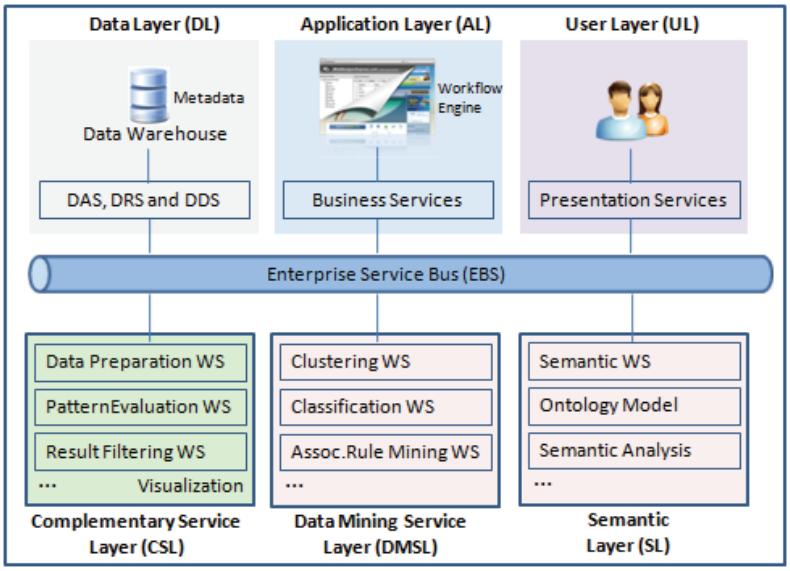


Fig. 1. SOMiner: a service-oriented architecture (SOA) for data mining

Application Layer: *Application Layer (AL)* is responsible for business services related to the application. Users don't interact directly with all services or servers - that's also the responsibility of the AL. It controls user interaction and returns the results to any user action. When a user starts building a data mining application, the AL looks for available data warehouses, queries them about their data and presents that information back to the user along with metadata. The user then selects a dataset, perhaps even further defines data preprocessing operations according to certain criteria. The AL then identifies which data mining services are available, along with their algorithms. When the user chooses the data mining algorithm and defines the arguments of it, the task is then ready to be processed. For the latter task, the AL informs the result filtering, pattern evaluation and visualization services. Complementary service layer builds these operations and sends the results back to the AL for presentation. SOMiner saves all these tasks to the user's list from which it can be scheduled for execution, edited for updates, or selected for visualization again.

User Layer: *User Layer (UL)* provides the user interaction with the system. The *Results Presentation Services (RPS)* offer facilities for presenting and visualizing the extracted knowledge models (e.g., association rules, classification rules, and clustering models). As mentioned before, a user can publish and search resources and services, design and submit data mining applications, and visualize results. Such users may want to make specific choices in terms of defining and configuring a data mining process such as algorithm selection, parameter setting, and preference specification for web services used to execute a particular data mining application. However, with the transparency advantage, end users have limited knowledge of the underlying data mining and web service technologies.

Data Mining Service Layer: *Data Mining Service Layer (DMSL)* is the fundamental layer in the SOMiner system. This layer is composed of generic and specific web services that provide a large collection of machine learning algorithms written for knowledge discovery tasks. In DMSL, each web service provides a different data mining task such as classification,

clustering and association rule mining (ARM). They can be published, searched and invoked separately or consecutively through a common GUI. Enabling these web services for running on large-scale SOA systems facilitates the development of flexible, scalable and distributed data mining applications.

This layer processes datasets and produces data mining results as output. To handle very huge datasets and the associated computational costs, the DMSL can be distributed over more than one node. The drawback related to this layer, however, is that it is now necessary to implement a web service for each data mining algorithm. This is a time consuming process, and requires the scientist to have some understanding of web services.

Complementary Service Layer: *Complementary Service Layer (CSL)* provides knowledge discovery processes such as data preparation, pattern evaluation, result filtering, visualization, except data mining process. *Data Preparation Service* provides data preprocessing operations such as data collection, data integration, data cleaning, data transformation, and data reduction. *Pattern Evaluation Service* performs the validation of data mining results to ensure the correctness of the output and the accuracy of the model. This service provides validation methods such as Simple Validation, Cross Validation, n-Fold Cross Validation, Sum of Square Errors (SSE), Mean Square Error (MSE), Entropy and Purity. If validation results are not satisfactory, data mining services can be re-executed with different parameters more than one times until finding an accurate model and result set. *Result Filtering Service* allows users to consider only some part of results set in visualization or to highlight particular subsets of patterns mined. Users may use this service to find the most interesting rules in the set or to indicate rules that have a given item in the rule consequent. Similarly, in ARM, users may want to observe only association rules with k-itemsets, where *k* is number of items provided by user. *Visualization* is often seen as a key component within many data mining applications. An important aspect of SOMiner is its visualization capability, which helps users from other areas of expertise easily understand the output of data mining algorithms. For example, a graph can be plotted using an appropriate visualize for displaying clustering results or a tree can be plotted to visualize classification (decision tree) results. Visualization capability can be provided by using different drawing libraries.

Semantic Layer: On the basis of those previous experiences we argue that it is necessary to design and implement semantic web services that will be provided by the *Semantic Layer (SL)*, i.e. ontology model, to offer the semantic description of the functionalities.

Enterprise Service Bus: The *Enterprise Service Bus (ESB)* is a middleware technology providing the necessary characteristics in order to support SOA. ESB can be sometimes considered as being the seventh layer of the architecture. The ESB layer offers the necessary support for transport interconnections. Translation specifications are provided to the ESB in a standard format and the ESB provides translation facilities. In other words, the ESB is used as a means to integrate and deploy a dynamic workbench for the web service collaboration. With the help of the ESB, services are exposed in a uniform manner, such that any user, who is able to consume web services over a generic or specific transport, is able to access them. The ESB keeps a registry of all connected parts, and routes messages between these parts. Since the ESB is solving all integration issues, each layer only focuses on its own functionalities.

SOMiner is easily extensible, as such; administrators easily add new servers or web services or databases as long as they have an interface; they can increase computing power by adding services or databases to independent mining servers or nodes. Similarly, end users can use any server or service for their task, as long as the application server allows it.

3.2 Application modeling and representation

SOMiner has the capability of composition of services, that is, the ability to create workflows, which allows several services to be scheduled in a flexible manner to build a solution for a problem. As shown in Fig. 2, a service composition can be made in three ways: horizontal, vertical and hybrid. *Horizontal composition* refers to a chain-like combination of different functional services; typically the output of one service corresponds to the input of another service, and so on. One common example of horizontal composition is the combination of pre-processing, data mining and post-processing functions for completing KDD process. In v*ertical composition*, several services, which carry out the same or different functionalities, can be executed at the same time on different datasets or on different data portions. By using vertical composition, it is possible to improve the performance in a parallel way. *Hybrid composition* combines horizontal and vertical compositions, and provides one-to-many cardinality, typically the output of one service corresponds to the input of more than one services or vice versa.

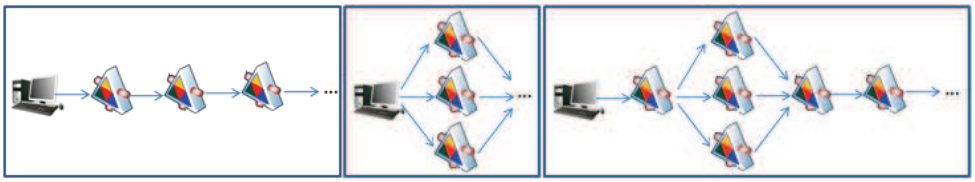


Fig. 2. Workflow types: horizontal composition, vertical composition, and hybrid

A workflow in SOMiner consists of a set of KDD services exposed via an *interface* and a *toolbox* which contains set of tools to interact with web services. The *interface* provides the users a simple way to design and execute complex data mining applications by exploiting the advantages coming from a SOA environment. In particular, it offers a set of facilities to design data mining applications starting from a view of available data, web services, and data mining algorithms to different steps for displaying results. A user needs only a browser to access SOMiner resources. The *toolbox* lets users choose from different visual components to perform KDD tasks, reducing the need for training users in data mining specifics, since many details of the application, such as the data mining algorithms, are hidden behind this visual notation.

Designing and executing a data mining application over the SOMiner is a multi-step task that involves interactions and information flows between services at the different levels of the architecture. We designed toolbox as a set of components that offer services through well defined interfaces, so that users can employ them as needed to meet the application needs. SOMiners's components are based on major points of the KDD problem that the architecture should address, such as accessing to a database, executing a mining task(s), and visualizing the results.

Fig. 3 shows a screenshot from the interface which allows the construction of knowledge discovery flows in SOMiner. While, on the left hand side, the user is provided with a collection of tools (toolbox) to perform KDD tasks, on the right hand side, the user is provided with workspace for composing services to build an application. Tasks are visual components that can be graphically connected to create a particular knowledge workflow. The connection between tasks is made by dragging an arrow from the output node of the sending task to the input node of the receiving task. Sample workflow in Fig. 3 was composed of seven services: data preparation, clustering, evaluation of clustering results,

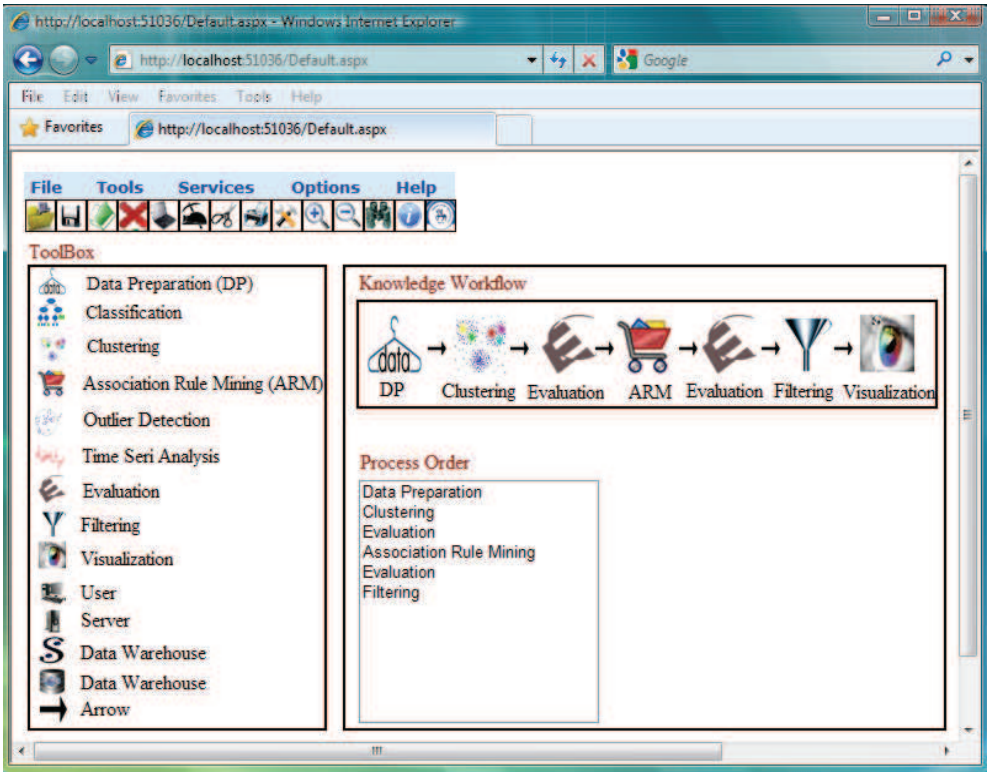


Fig. 3. Screenshot from the interface used for the construction of knowledge workflows

ARM, evaluation of association rules, filtering results according to user requests, and visualization.

Interaction between the workflow engine and each web service instance is supported through pre-defined SOAP messages. If a user chooses a particular web service from the place on the composition area, a URL specifying the location of the WSDL document can be seen, along with the data types that are necessary to invoke the particular web service.

3.3 Advantages of service-oriented data mining

Adopting SOA for data mining has at least three advantages: (i) implementing data mining services without having to deal with interfacing details such as the messaging protocol, (ii) extending and modifying data mining applications by simply creating or discovering new services, and (iii) focusing on business or science problems without having to worry about data mining implementations. (Cheung et al., 2006)

Some key advantages of service-oriented data mining system (SOMiner) include the following:

1. *Transparency*: End-users can be able to carry out the data mining tasks without needing to understand detailed aspects of the underlying data mining algorithms. Furthermore, end-users can be able to concentrate on the knowledge discovery application they must develop, without worrying about the SOA infrastructure and its low-level details.

2. *Application development support*: Developers of data mining solutions can be able to enable existing data mining applications, techniques and resources with little or no intervention in existing application code.
3. *Interoperability*: The system will be based on widely used web service technology. As a key feature, web services are the elementary facilitators of interoperability in the case of SOAs.
4. *Extensibility*: System provides extensibility by allowing existing systems to integrate with new tasks, just adding new resources (data sets, servers, interfaces and algorithms) by simply advertising them to the system.
5. *Parallelism*: System supports processing on large amounts of data through parallelism. Different parts of the computation are executed in parallel on different nodes, taking advantage at the same time of data distribution and web service distribution.
6. *Workflow capabilities*: The system facilitates the construction of knowledge discovery workflows. Thus, users can reuse some parts of the previously composed service flows to further strengthen the data mining application development's agility.
7. *Maintainability*: System provides maintainability by allowing existing systems to change only a partial task(s) and thus to adapt more rapidly to changing in data mining applications.
8. *Visual abilities*: An important aspect of the system is its visual components, since many details of the application are hidden behind this visual notation.
9. *Fault tolerance:* The application can continue to operation without interruption in the presence of partial network failures, or failures of the some software components, taking advantage of data distribution and web service distribution.
10. *Collaborative*: A number of science and engineering projects can be performed in collaborative mode with physically distributed participants.

A significant advantage of SOMiner over previous systems is that SOMiner is intended for using semantic web services to the semantic level, i.e. ontology model and offer the semantic description of the functionalities. For example, it allows integration of data mining tasks with ontology information available from the web.

Overall, we believe the collection of advantages and features of SOMiner make it a unique and competitive contender for developing new data mining applications on service-oriented computing environments.

4. Case study

4.1 SOMiner at work

In this section, we describe a case study and experimental results obtained by the construction of a knowledge workflow in which data in a data warehouse was analyzed by using clustering and ARM algorithms, with the goal of evaluating the performance of the system. In the case study, interface within the SOMiner framework has been used to implement a data mining application related to dried fruit industry, and obtained significant results in terms of performance. The analyzed data provided by a dried fruits company in Turkey consists of about three years of sales data collected within the period January 2005 and April 2008. The complete data that consists of five tables (customers, products, sales, sales details, and branches) included about 56,000 customers, 325 products, 721,000 sales and 3,420,000 sales details. Fig. 4 shows the star schema of the data warehouse that consists of a fact table with many dimensions.

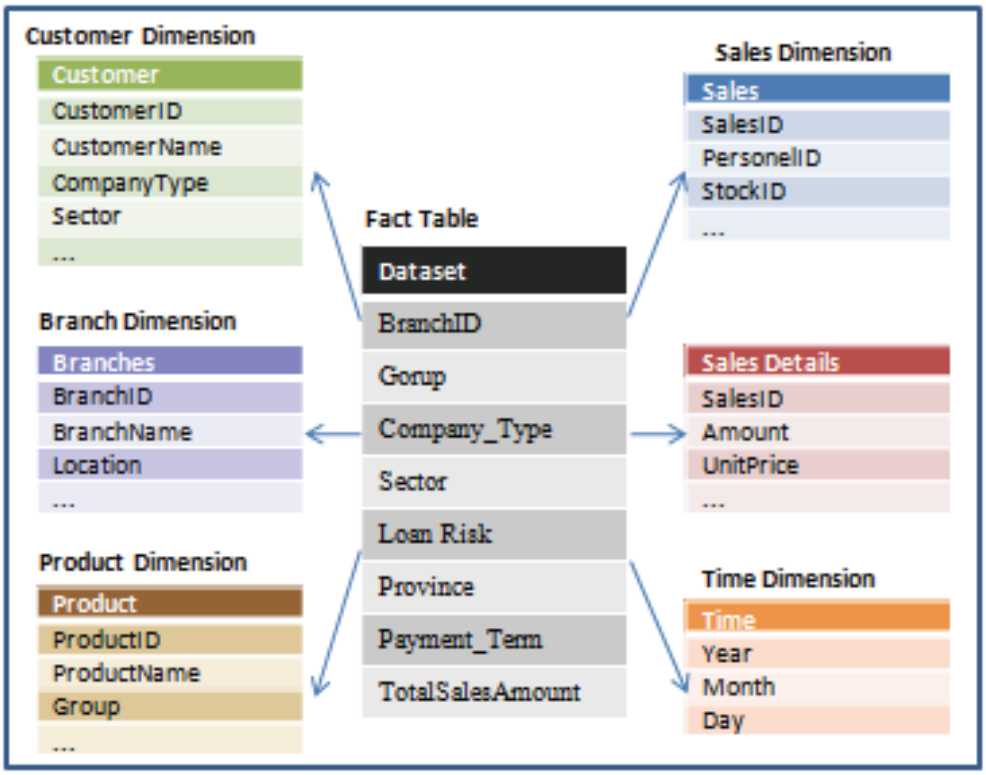


Fig. 4. Star schema of the data warehouse used in the case study

In the case study, once clustering task was used to find customer segments with similar profiles, and then association rule mining was carried out to the each customer segment for product recommendation. The main advantage of this application is to be able to adopt different product recommendations for different customer segments. Based on our service-based data mining architecture, Fig. 5 shows the knowledge discovery workflow constructed in this case study, which represents pre-processing steps, potentially repeatable sequences of data mining tasks and post-preprocessing steps. So, we defined a data mining application as an executable software program that performs two data mining tasks and some complementary tasks.

In the scenario, first, (1) the client sends a business request and then (2) this request is sent to application server for invoking data preparation service. After data-preprocessing, (3) data warehouse is generated, (4) clustering service is invoked to segment customers, and then (5) clustering results are evaluated to ensure the quality of clusters. After this step, (6) more than one ARM web services are executed in parallel for discovering association rules for different customer segments. (7) After the evaluation of ARM results by using Lift and Loevinger thresholds, (8) the results are filtered according to user-defined parameters to highlight particular subsets of patterns mined. For example, users may want to observe only association rules with k-itemsets, where k is number of items provided by user. Finally, (9) visualization service is invoked to plot a graph for displaying results.

Fig. 5. An example knowledge discovery workflow

Given the design and implementation benefits discussed in section 3.3, another key aspect in evaluating the system is related to its performance in supporting data mining services execution. In order to evaluate the performance of the system, we performed some experiments to measure execution times of the different steps. The data mining application described above has been tested on deployments composed from 4 association rule mining (ARM) web services; in other words, customers are firstly divided into 4 groups (customer segments), and then 4 ARM web services are executed in parallel for different customer segments (clusters). Each node was a 2.4 GHz Centrino with 4 GB main memory and network connection speed was 100.0 Mbps. We performed all experiments with a minimum support value of 0.2 percent. In the experiments, we used different datasets with sizes ranging from 5Mbytes to 20Mbytes.

While in the clustering experiments we used the customer and their transactions (sales) data available at the data warehouse, in the ARM, we used products and transaction details (sales details) data. Expectation-Maximization (EM) algorithm for clustering task and Apriori algorithm for ARM were implemented as two separate web services. The execution times have been shown in Table 1. It reports the times needed to complete the different phases: file transfer, data preparation, task submission (invoking the services), data mining (clustering and ARM), and results notification (result evaluation and visualization).

Values reported in the Table 1 refer to the execution times obtained for different dataset sizes. The table shows that the data mining phase takes averagely 81.1% of the total execution time, while the file transfer phase fluctuate around 12.8%. The overhead due to the other operations – data preparation, task submission, result evaluation and visualization – is very low with respect to the overall execution time, decreasing from 6.5% to 5.4% with the growth of the dataset size. The results also show that we achieved efficiencies greater than 73 percent, when we execute 4 web services in parallel, instead of one web service.

Dataset Size	File Transfer	Data Prepar.	Task Submission	Data Mining			Results Notification
				EM	Apriori	Total	
5 MB	3,640	1,820	212	7,110	29,156	36,266	691
10 MB	5,437	1,995	253	13,251	24,031	37,282	720
15 MB	8,287	2,064	248	18,343	23,477	41,820	862
20 MB	11,071	2,218	264	34,528	23,233	57,761	1,485

Table 1. Execution times (in milliseconds) needed to complete the different phases

The *file transfer* and *data mining* execution times changed because of the different dataset sizes and algorithm complexity. In particular, the *file transfer* execution time ranged from 3,640 ms for the dataset of 5MB to 11,071 ms for the dataset of 20MB, while the *data mining* execution time ranged from 36,266 ms for the dataset 5 MB to 57,761 ms for 20MB.
In general, it can be observed that the overhead introduced by the SOA model is not critical with respect to the duration of the service-specific operations. This is particularly true in typical KDD applications, in which data mining algorithms working on large datasets are expected to take a long processing time. On the basis of our experimental results, we conclude that SOA model can be effectively used to develop services for KDD applications.

4.2 Discussion and evaluation

The case study has been useful for evaluating the overall system under different aspects, including its performance. Given these basic results, we can conclude that SOMiner is suitable to be exploited for developing services and knowledge discovery applications in SOA.
In order to improve the performance moreover, the following proposals should be considered:

1. To avoid delays due to data transfers during computation, every mining server should have an associated local data server, in which data is kept before the mining task executes.
2. To reduce computational costs, data mining algorithms should be implemented in more than one web services which are located over different nodes. This allows the execution of the data mining components in the knowledge flow on different web services.
3. To reduce computational costs, the same web services should be located over more than one node. In this way, the overall execution time can be significantly reduced because different parts of the computation are executed in parallel on different nodes, taking advantage at the same time of data distribution.
4. To get results faster, if the server is busy with another task, it should send the user an identifier to use in any further communication regarding that task. A number of idle workstations should be used to execute data mining web services, the availability of scalable algorithms is key to effectively using the resources.

Overall we believe the collection of features of SOMiner make it a unique and competitive contender for developing new data mining applications on service-oriented computing environments.

5. Conclusion

Data mining services in SOA are key elements for practitioners who need to develop knowledge discovery applications that use large and remotely dispersed datasets and/or computers to get results in reasonable times and improve their competitiveness. In this chapter, we address the definition and composition of services for implementing knowledge discovery applications on SOA model. We propose a new system, SOMiner that supports knowledge discovery on SOA model by providing mechanisms and higher level services for composing existing data mining services as structured, compound services and interface to allow users to design, store, share, and re-execute their applications, as well as manage their output results.

SOMiner allows miners to create and manage complex knowledge discovery applications composed as workflows that integrate data sets and mining tools provided as services in SOA. Critical features of the system include flexibility, extensibility, scalability, conceptual simplicity and ease of use. One of the goals with SOMiner was to create a data mining system that doesn't require users to know details about the algorithms and their related concepts. To achieve that, we designed an interface and toolkit, handling most of the technical details transparently, so that results would be shown in a simple way. Furthermore, this is the first time that a service-oriented data mining architecture proposes a solution with semantic web services. In experimental studies, the system has been evaluated on the basis of a case study related to marketing. According to the experimental results, we conclude that SOA model can be effectively used to develop services for knowledge discovery applications.

Some further works can be added to make the system perform better. First, security problems (authorization, authentication, etc.) related to the adoption of web services can be solved. Second, a tool can be developed to automatically transfer the current traditional data mining applications to the service-oriented data mining framework.

6. References

Ali, A.S.; Rana, O. & Taylor, I. (2005). Web services composition for distributed data mining, *Proceedings of the 2005 IEEE International Conference on Parallel Processing Workshops, ICPPW'05*, pp. 11-18, ISBN: 0-7695-2381-1, Oslo, Norway, June 2005, IEEE Computer Society, Washington, DC, USA.

Ari, I.; Li, J.; Kozlov, A. & Dekhil, M. (2008). Data mining model management to support real-time business intelligence in service-oriented architectures, *HP Software University Association Workshop*, White papers, Morocco, June 2008, Hewlett-Packard.

Brezany, P.; Janciak, I. & Tjoa, A.M. (2005). GridMiner: A fundamental infrastructure for building intelligent grid systems, *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, pp. 150-156, ISBN: 0-7695-2415-x, Compiegne, France, September 2005, IEEE Computer Society.

Chen, N.; Marques, N.C. & Bolloju, N. (2003). A Web Service-based approach for data mining in distributed environments, *Proceedings of the 1st Workshop on Web Services: Modeling, Architecture and Infrastructure* (WSMAI-2003), pp. 74-81, ISBN 972-98816-4-2, Angers, France, April 2003, ICEIS Press 2003.

Chen, P.; Wang, B.; Xu, L.; Wu, B. & Zhou, G. (2006). The design of data mining metadata web service architecture based on JDM in grid environment, *Proceedings of First International Symposium on Pervasive Computing and Applications*, pp. 684-689, ISBN: 1-4244-0326-x, Urumqi, China, August 2006, IEEE.

Cheung, W.K.; Zhang, X-F.; Wong, H-F.; Liu, J.; Luo, Z-W. & Tong, F.C.H., (2006). Service-oriented distributed data mining, *IEEE Internet Computing,* Vol. 10, No. 4, (July/August 2006) pp. 44-54, ISSN:1089-7801.

Congiusta, A.; Talia, D. & Trunfio, P. (2007). Distributed data mining services leveraging WSRF, *Future Generation Computer Systems*, Vol. 23, No. 1, (January 2007) 34–41, ISSN: 0167-739X.

Congiusta, A.; Talia, D. & Trunfio, P. (2008). Service-oriented middleware for distributed data mining on the grid, *Journal of Parallel and Distributed Computing, Vol.* 68, No. 1, (January 2008) 3-15, ISSN: 0743-7315.

Du, H.; Zhang, B. & Chen, D. (2008). Design and actualization of SOA-based data mining system, *Proceedings of 9th International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID/CD),* pp. 338–342, ISBN: 978-1-4244-3290-5, Kunming, November 2008.

Guedes, D.; Meira, W.J. & Ferreira, R. (2006). Anteater: A service-oriented architecture for high-performance data mining, *IEEE Internet Computing*, Vol. 10, No. 4, (July/August 2006) 36–43, ISSN: 1089-7801.

Jackson, T.; Jessop, M.; Fletcher, M. & Austin, J. (2007). A virtual organisation deployed on a service orientated architecture for distributed data mining applications, *Grid-Based Problem Solving Environments*, Vol. 239, Gaffney, P.W.; Pool, J.C.T. (Eds.), pp. 155-170, Springer Boston, ISSN: 1571-5736.

Olejnik, R.; Fortiş, T.-F. & Toursel, B. (2009) Web services oriented data mining in knowledge architecture, *Future Generation Computer Systems,* Vol. 25, No. 4, (April 2009) 436-443, ISSN: 0167-739X.

Perez, M.; Sanchez, A.; Robles, V.; Herrero, P. & Pena, J.M. (2007). Design and implementation of a data mining grid-aware architecture, *Future Generation Computer Systems,* Vol. 23, No. 1, (January 2007) 42–47, ISSN: 0167-739X.

Sairafi, S.A.; Emmanouil, F.S.; Ghanem, M.; Giannadakis, N.; Guo, Y.; Kalaitzopolous, D.; Osmond, M.; Rowe, A.; Syed, J. & Wendel, P. (2003). The design of discovery net: Towards open grid services for knowledge discovery, *International Journal of High Performance Computing Applications*, Vol. 17, No. 3, (August 2003) 297–315, ISSN: 1094-3420.

Stankovski, V.; Swain, M.; Kravtsov, V.; Niessen, T.; Wegener, D.; Kindermann, J. & Dubitzky, W. (2008). Grid-enabling data mining applications with DataMiningGrid: An architectural perspective, *Future Generation Computer Systems,* Vol. 24, No. 4, (April 2008) 259–279, ISSN: 0167-739X.

Swain, M.; Silva, C.G.; Loureiro-Ferreira, N.; Ostropytskyy, V.; Brito, J.; Riche, O.; Stahl, F.; Dubitzky, W. & Brito, R.M.M. (2009). P-found: Grid-enabling distributed repositories of protein folding and unfolding simulations for data mining, *Future Generation Computer Systems,* Vol. 26, No. 3, (March 2010) 424-433, ISSN: 0167-739X.

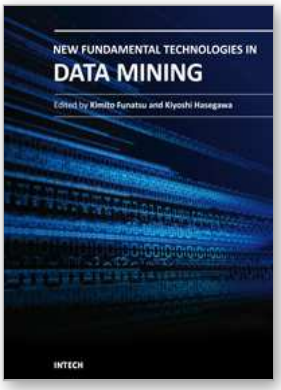
Talia, D. (2009). Distributed data mining tasks and patterns as services, *Euro-Par 2008 Workshops - Parallel Processing, Lecture Notes in Computer Science,* pp. 415-422, Springer Berlin / Heidelberg, ISSN: 0302-9743.

Talia D. & Trunfio, P. (2007). How distributed data mining tasks can thrive as services on grids, *National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation (NGDM'07),* Baltimore, USA, October 2007.

Tsai, C.-Y. & Tsai, M.-H. (2005). A dynamic web service based data mining process system, *Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05),* pp. 1033-1039, IEEE Computer Society, Washington, DC, USA.

Wu, S.; Wang, W.; Xiong, M. & Jin, H. (2009). Data management services in ChinaGrid for data mining applications, *Emerging Technologies in Knowledge Discovery and Data Mining,* pp. 421-432, Springer Berlin / Heidelberg, ISSN: 0302-9743.

Yamany, H.F.; Capretz, M. & Alliso, D.S. (2010). Intelligent security and access control framework for service-oriented architecture, *Information and Software Technology*, Vol. 52, No. 2, (February 2010) 220-236, ISSN: 0950-5849.



New Fundamental Technologies in Data Mining

Edited by Prof. Kimito Funatsu

ISBN 978-953-307-547-1
Hard cover, 584 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

The progress of data mining technology and large public popularity establish a need for a comprehensive text on the subject. The series of books entitled by "Data Mining" address the need by presenting in-depth description of novel mining algorithms and many useful applications. In addition to understanding each section deeply, the two books present useful hints and strategies to solving problems in the following chapters. The contributing authors have highlighted many future research directions that will foster multi-disciplinary collaborations and hence will lead to significant development in the field of data mining.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821