Abrizah Abdullah, et al. (Eds.): ICOLIS 2007, Kuala Lumpur: LISU, FCSIT, 2007: pp 31-40

Towards a Highly Configurable Data Mining Middleware through Metadata Approach for Access to Unlimited Data Sources

Lai Ee Hen and Sai Peck Lee
Department of Software Engineering
Faculty of Computer Science & Information Technology
University of Malaya, 50603 Kuala Lumpur, Malaysia
E-mail: laiee@perdana.um.edu.my; saipeck@um.edu.my

Abstract

Information stored in a consumer database has become a valuable asset for an organization in today's information age. It houses vital, hidden information that can be extracted using data mining tools to solve real-world problems in engineering, science, and business. However, due to the rapidly increasing data, data mining tools face substantial challenges in extracting information from data sources such as XML, relational databases, flat files, spreadsheets and so forth. Worst still, these data sources might reside on different locations across different boundaries. Therefore, in order to support new data sources, data mining tools need to provide the flexibility to access the wide range of data sources through straightforward configurations. In this paper, we provide an overview of a proposed data mining middleware known as Java-based Data Mining Middleware which extensively makes use of such configurations through metadata and XML to access unlimited data sources that reside on different locations. Additionally, the proposed middleware also supports a wide range of data mining techniques through trouble-free configurations. Additional data sources and data mining techniques can simply be plugged into the middleware programmatically and non-programmatically.

Keywords: Metadata; XML; Data Mining Middleware; Trouble-free configuration

1. Introduction

In today's market place, information in a consumer database, which is the most valuable asset of an organization, consists of strategic important hidden information. Using data mining, patterns and relationships in the data can be extracted to uncover the hidden messages, which are important to the business to increase revenues and reduce costs. According to Sanjiv Purba (2006), data mining is the process of deriving useful information from a data source through the use of creative queries. It includes the identification of undetected relationships without the needs of applying specialised approaches. Data mining is alternatively known as Knowledge Discovery from Data. However Data Mining is actually an essential step among the processes of knowledge discovery. These processes involved in sequence are Data Cleaning, Data Integration, Data Selection, Data Transformation, Data Mining, Pattern Evaluation and lastly Knowledge Presentation (Jiawei and Micheline 2006).

Data mining has received an enormous attention by many sectors ranging from manufacturing, marketing, investment and so forth due to the substantial contributions provided. Many organizations use data mining to help them to manage the phases of the customer life cycle such as obtaining new customers and retaining existing customers. For instance, with the help of data mining, the characteristics of customers can be uncovered so that companies can target prospects with similar characteristics.

Realising the ability of data mining in helping organizations to understand the complete value of their corporate data, this research aims to propose an architecture of a data mining middleware known as Java-Based Data Mining Middleware (JDMM). JDMM is a

highly configurable platform-, data source-, and data mining technique-independent middleware, which is accessible from front-, back- and web-office environments. In short, users are able to use JDMM in any platform to mine any data sources without geographical boundaries using a wide range of data mining techniques to produce comprehensive and meaningful results in any types of report formats to help organizations in their business's decision-making.

Over the past decades, the Internet has become widely adopted. With the web evolution, the amount and speed of data interchanged have increased a great deal and so as the data sources. This has led to the development and the quick acceptance of Extensible Markup Language (XML) (Giovanna, Marco and Daniele, 2005). XML is a technology for creating markup languages to describe data of any type in a structured manner (Harvey, et al. 2000). Undeniable, the future of XML will be the foundation for all data manipulation and data transmission. In addition, the use of databases has grown in all enterprises and almost all organizations need a form of data storage. These databases range from flat file, relational database, object-oriented database, XML and so forth. With XML as a versatile, readable meta-language that can be understood easily due to its' capable of self-describing the information content from various data sources which include semi-structured and structured documents (Lee and Lee, 2005), the proposed JDMM will support both structured and unstructured data in different data sources through the use of XML in data transmission and manipulation.

The proposed JDMM supports a variety of data mining functionalities. One of the features is the ability to mine data using unlimited number of data mining techniques. This is a necessity as different users are interested in exploring different kinds of knowledge.

In order for JDMM to mine any data source using any data mining technique to produce any proprietary reporting format, JDMM needs to be highly flexible and configurable. This is possible through the use of XML as Metadata. Metadata is defined as data about data (Euzenat, 2002). These metadata functioned as the "content descriptors" of the different data mining techniques, data sources and reporting formats (Yoshinobu et al., 2005) Through the configuration of metadata and XML, users are able to plug in any data mining techniques, data sources, and reporting formats.

In the remaining of the paper, we focus on highlighting the challenges in data mining tools. Based on the challenges, we propose the architecture of a highly configurable data mining middleware that will gear towards these challenges by adopting XML and Metadata technologies.

2. Data mining challenges

Choosing an appropriate data mining tool is not an easy task. According to Nisbet (2004), a good data mining tool is a tool that will be able to achieve ease of use, provide an acceptable accuracy and the ability of the system to perform all data mining common task.

Many data mining technologies and tools are available in the market, however many issues need to be considered in order to design a good data mining tool. One of the challenges in data mining is to mine diverse knowledge in databases (Jiawei and Micheline, 2006). Different users might have different interest of knowledge. A data mining system needs to cater for a spectrum of data mining techniques to help in uncovering and extracting hidden patterns to stay competitive in their fields of business. However, in this information age, user needs and requests are always changing and are very dynamic in nature. Providing predefined sets of data mining techniques might not be sufficient in this dynamic environment.

Presentation and visualization of data mining results are also one of the cofactors in the designing a data mining system. It serves no purpose if a data mining system indicates it is able to uncover some new knowledge, however users find it difficult to understand. Knowledge discovered needs to be comprehensive such as providing high-level languages and visual representations. These can be achieved through the adoption of expressive knowledge representation techniques such as graph, trees, tables and so forth (Jiawei and Micheline, 2006).

In addition, a data mining system might face the challenges of limited supports such as inability to provide platform independence and data source independence (Sanjiv, 2006). For example, if a new database management system is released which is not supported in the current version of data mining system used, the user would have to be forced to choose either to buy a new system that includes the support for the data source or wait for an updated release of their existing data mining system which is rather very inflexible.

Performance attribute is a crucial consideration of JDMM to mine large data sources. To ensure performance is not a bottleneck, implementation of JDMM's data mining techniques need to be efficient. Furthermore, the caching framework of JDMM needs to implement in-memory caching data structures efficiently in order to handle large unmined and mined data. All in-memory caching data structures need to be thread safe to avoid deadlock within JDMM. Other than that, JDMM needs to implement proper garbage collection to ensure that all unused objects are released efficiently.

3. Related work

Throughout the research of this paper, we have selected a set of data mining tools to further study their functionalities and services provided. These tools are IBM Intelligent Miner, SPSS Clementine, SAS Institute Enterprise Miner, Oracle Data Miner, and Microsoft Business Intelligence Development Studio. Our study mainly focuses on addressing challenges and issues listed at the previous section.

At the time of our study, a few tools like Microsoft Business Intelligence Development Studio, Oracle Data Miner, and SAS Institute Enterprise Miner only support a predefined set of data sources. In order to implement new data sources, it requires thorough understanding of the data source API specification. For instance, Oracle Data Miner only supports JDBC compliant driver and Microsoft Business Intelligence Development Studio only supports more ODBC, OLEDB and other types of predefined data sources.

Data mining tools such as Microsoft Business Intelligence Development Studio is platform dependent. The tool depends on .NET Framework which currently only supports the Windows platform. Theoretically, Mono allows .NET based applications such as Microsoft Business Intelligence Development Studio to run on operating systems such as UNIX and Linux. Unfortunately, implementing Mono (Mono, 2007) in platforms such as BSD (OpenBSD, FreeBSD and NetBSD) and Mac OS X are not straightforward. SPSS Clementine, on the other hand, releases different binaries on different platforms. Lastly, Oracle Data Miner uses the same binaries on different platforms, and as such, is platform independent.

Oracle Data Miner and Microsoft Business Intelligence Development Studio use a caching strategy in performing data mining. Unfortunately, these tools only cache a certain percentage of both unmined and mined data in the application tier. The caching strategy intends to offload computing cycles from the backend systems (for example, SQL Server Analysis Services in the case of Microsoft Business Intelligence Development

Studio, and Oracle Data Mining in the case of Oracle Data Miner (2006). However, we might have scenarios whereby two users might be mining the same data set and this causes redundancy in terms of work performed (i.e., computed data is computed again) since both the unmined and mined data are not fully persisted or cached in the backend systems.

4. Proposed Data Mining Middleware Architecture

A conceptual high-level architecture of JDMM is depicted in Figure 1. Users are connected to JDMM through two different web applications that reside on the Tomcat Servlet Container. The web applications are known as JDMM Web Configurator and Web JDMM. Both of these web applications support different user roles. In business's point of view, users from the front-end web environment (Web JDMM) are allowed to analyze dynamic data sources which may vary such as MySQL, MSSQL, text files, Excel files, XML files or other data sources to solve business's decision problems. In the technical user's point of view, users with technical knowledge are able to perform detailed configuration on the internal components of JDMM through the JDMM Web Configurator. They are able to plug in all kinds of adapter into JDMM for connecting to different data sources, data mining techniques and reporting formats. All the configurations are done using XML as metadata to describe the adapters. In addition, these configurations are done through point of click which shields the complexity from the users. The Enterprise Java Bean (EJB) server acts as a retrieval engine and consists of different adapters to interconnect different data sources with JDMM. After data retrieval, these data are automatically organized by JDMM to create a data mining model using a specific data mining technique through the adapter framework. At this stage, the result can be stored into a data mining repository or directly to a persistent data store such as a relational database (MySQL). At a specific point of time interval, the result from the repository will be transferred to the persistent data store. The primary objective of the repository is to cache results so that computed results are not computed again. The result is a XML file that will then be published and delivered to the user as either PDF file, XLS file or any proprietary format that are incorporated into JDMM.

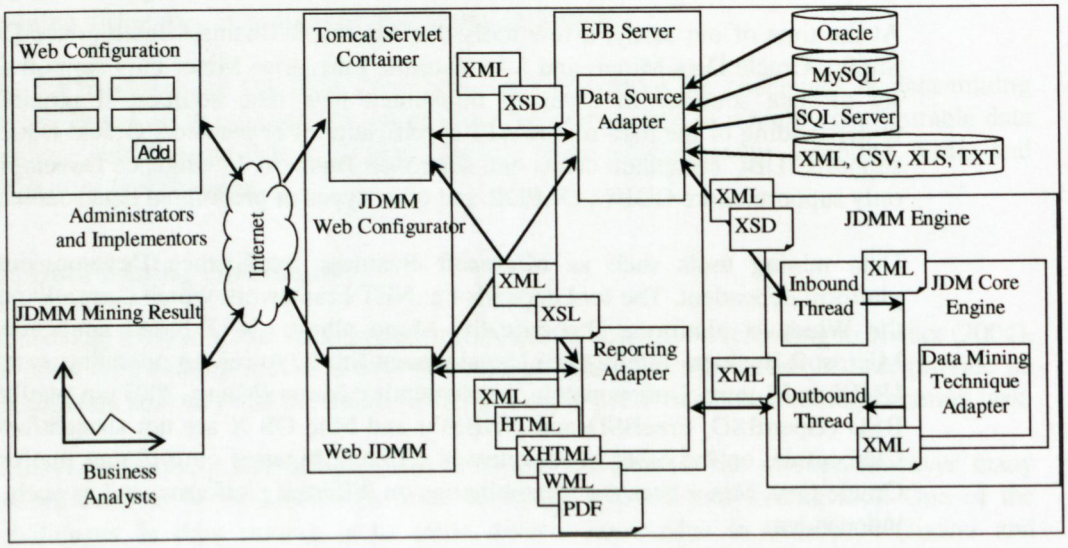

Figure 1: JDMM Architecture Information Flow

(a) JDMM User Roles

JDMM is accessible by three possible roles of users namely JDMM Implementor, JDMM Administrator, and JDMM Business Analyst. JDMM are accessible through JDMM Web Configurator and Web JDMM.

JDMM Web Configurator is a web application that provides a user-friendly interface, allowing the implementers to specify different data sources, mining techniques and reporting results. JDMM Implementors are the technical users who hold the responsibility of plugging in any adapters through JDMM Web Configurator. These adapters enable users to connect to any data sources using any data mining techniques to generate any type of report. Through simple configuration using JDMM Web Configurator, any new data mining techniques, data sources or report formats created will also be supported by JDMM through the flexibility of adapter framework.

On the other hand, JDMM Administrator will administer all JDMM instances and be responsible for the uptime of JDMM through JDMM Web Configurator. Lastly, JDMM Business Analysts represents non-technical users which are responsible for business decision-making such as business decision-making through JDMM to solve real-world business problems such as planning a marketing campaign, forecasting product growth and other types of business decisions. They will access JDMM through Web JDMM.

(b) JDMM Information Interchange

JDMM is a data mining middleware that is accessible from front-, back- and web-office environments. In order to ensure a common standard of data communication and data store, XML is the preferred language for information interchange between JDMM components. This is due to the numerous advantages XML can offer.

- XML can be straightforwardly usable over the World Wide Web.
- XML can support a wide variety of applications.
- XML is compatible with SGML.
- XML documents are human-legible and reasonably concise.
- Any shortness in XML markup is of minimal importance.

XML is a plain-text, Unicode-based meta-language (Dare, 2003). It is a language for defining markup languages. It is not tied to any programming languages, operating systems, or software vendors. Hence, XML will help us in ensuring JDMM to be a platform- and language- independent data mining middleware. In addition, XML is the preferred choice of data format as compared to data formats such as comma separated value files and RTF because XML can easily represent both tabular data (such as relational data from a database or spreadsheets) and semi-structured data (such as a Web page or business document). In addition, XML is in a text-based format which supports internationalization by being fully Unicode compliant.

(c) Highly Configurable JDMM Adapter Framework

Based on Figure 1, JDMM provides an extensible JDMM Adapter Framework namely Data Source Adapters, Reporting Adapters, and Data Mining Techniques Adapters. As mentioned, one of the issues a data mining system need to take notes is the ability to mine diverse knowledge in databases. Hence, the proposed JDMM will accept wide a range of data sources such as relational databases, object-oriented databases, flat file and so forth from heterogeneous environment through the used Data Source Adapters. Each adapter consists of sets of classes helping JDMM to connect to a specific data source. Any new data source available will only need to be plugged into JDMM Adapter through minimal

configuration in an XML file. An XML file will serve as a generic configuration file for JDMM implementors to specify properties pertaining to a specific data source. For example, for a typical relational database data source, JDMM implementors are able to specify the user and its associated password along with the database driver and URL (Uniform Resource Locator) to access a specific relational database. On the other hand, JDMM implementors are also able to specify properties such as data source path pertaining to data sources such as CSV, XML and TXT files.

Similar to Data Source Adapter, the Data Mining Technique Adapter also allows different data mining techniques to be plugged and configured into JDMM, similarly through simple XML-based configuration. Each technique is governed by adapters which are pluggable rule adapters specified in an XML configuration file. As such, JDMM is able to use other components to apply a specific data mining technique. These proposed adapters can implement many different data mining algorithms such as hypothesis testing, time series, normal distribution, binomial distribution and many other types of algorithms.

Lastly, the proposed JDMM also provides another type of adapter called Reporting Adapter through the JDMM Adapter Framework. Due to the diverse knowledge of interest among users, their demand on the result may vary. Yet again this is made possible through the Reporting Adapter which is used to describe different mined data into any formats such as pdf file, Excel spreadsheet, csv file, text file, xml file, html/htm file and other types of format. In this Reporting Adapter, besides using XML as the medium to describe the mined data, Extensible Stylesheet Language (XSL) is also incorporated into the adapter. XSL is a flexible and powerful language, official recommendation of the World Wide Web Consortium (W3C), for transforming XML documents into other document such ash HTML, WML, PDF and so forth (O'Reilly, 2001). It comprises three technologies namely XPath, XSL Transformation and XSL Formatting Objects (Ray, 2003). With these technologies rolled into one XSL, it enables JDMM to navigate through the XML, transform and format the XML document into any format that is recognizable by the target source for decision making.

(d) Application of JDMM Metadata on Interpreted Data

The Reporting Adapter validates all interpreted data in the form of XML files, which was originally stored in the caching data structures within a repository in JDMM Engine, against an XSD file (JDMM metadata) to verify that the interpreted data conforms to JDMM. The functionalities of Report Adapter are shown in Figure 2.

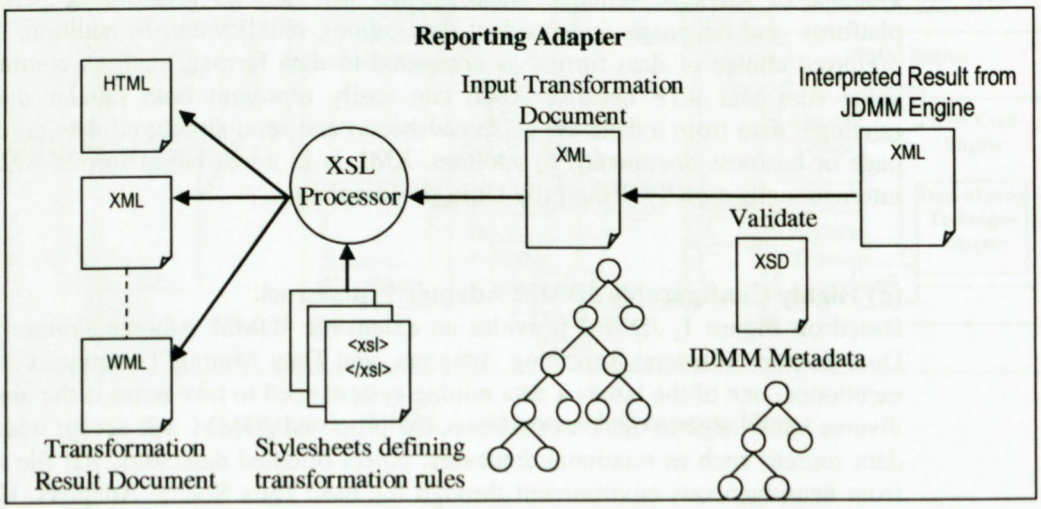

Figure 2: Functionalities of Reporting Adapter

Most of the caching data structures in JDMM are key-value pairs type of data structures which are thread safe as JDMM is a multi-threaded server-based application. The JDMM metadata serves as a standard to verify that all XML instances of mined data are conformed to the JDMM. By making use of XML-based parsers conforming to W3C standards, JDMM transforms the XML interpreted data to a tree-based data structure (also known as DOM tree or document). All compliant XML instances are then processed by a XSL processor which serves to generate target outputs such as HTML, XHTML, WML, XML and other types of outputs.

5. JDMM Detailed Architecture

In addressing the challenge of diverse platforms, JDMM needs to be implemented as a platform independent middleware in order to run on different operating systems such as Windows and UNIX. Java, as the language that supports "Write Once Run Anywhere Concept", is chosen to be the language used in developing JDMM. With the managed runtime of Java Runtime Environment, development can focus solely on the business logic of JDMM.

Data retrieval process done in the EJB Server as shown in Figure 1 is channeled to JDMM Engine. The detailed architecture of JDMM is depicted in Figure 2. The architecture of the JDMM Engine is divided into two threads namely Inbound Threads and Outbound Threads. An Inbound Thread manages all incoming uninterpreted operational data (raw data), whereas an Outbound Thread manages all outgoing interpreted operational data (mined data).

Both the Send Adapter and Receive Adapter are part of the Adapter Framework. Each adapter is configurable through the JDMM Web Configurator and each configurable parameter is stored in a XML file. JDMM uses a set of XSD schema files (also known as JDMM metadata) to formally describe both the uninterpreted data (unmined data) and interpreted data (mined data). JDMM schema is an abstract representation of an object's characteristics and relationships to other objects. In this case, the objects refer to both uninterpreted data and interpreted data. A JDMM XML schema typically represents the interrelationships between the attributes and elements of an XML object (for example, a document or a portion of a document in JDMM). Through the use of metadata, JDMM is able to formalize uninterpreted data to ensure that the data conforms to JDMM.

Both the Send and Receive Pools use a common thread pool component to ensure that all data mining requests are processed concurrently as opposed to the sequential approach which is slower. As such, data mining requests can be handled simultaneously as a separate entity. Hence, if deadlock occurs, the deadlock only happens on a particular thread and the processing does not affect other threads.

JDMM relies heavily on XML as a mode of communication with other JDMM components. As such, JDMM needs to verify that each XML containing uninterpreted data conforms to JDMM. Besides ensuring JDMM to be a highly configurable data mining middleware, JDMM also needs to ensure that all interpreted data can be easily ported to other data mining products or data mining standards. In this case, JDMM will use XSD files as a mode to ensure that all interpreted data conforms to a specific product or data mining standard.

Java-Based Data Miner (JDM) is a pure Java API for developing data mining applications. The idea is to have a common API for data mining that can be used by clients without users being aware or affected by the actual vendor implementations for

data mining. The JDM architecture consists of three logical components, the API, the Data Mining Engine (DME), and the metadata repository (MR). The API is an exposed programming interface that provides access to the services provided by the DME. The API shields the data mining user from the actual implementation in the DME and any associated sub-components used by the DME. The DME is the engine providing the services that can be used by JDMM users through the API defined above. It can be implemented as a server known as Data Mining Server (DMS). The third component is the MR which is used to persist data mining objects. These persisted data mining objects are again used by the DME for data mining operations. The metadata repository can exist as a flat file system or a relational database. The three logical components are grouped into one physical system or they can exist independently as separate components. Apart from these, JDMM implementors can also implement additional components and tools to enhance the vendor implementation of the JDM.

JDM Extension is an extension to JDM that includes additional data mining models, data scoring and data transformations. JDM Extension is based on a highly-generalized, object-oriented, data mining conceptual model using Data Mining Group's Predictive Model Markup Language (PMML) data mining standards. PMML is a XML markup language to describe statistical and data mining models ("Predictive Model Markup Language," 2005). This JDM Extension also expose the data mining techniques used through Data Mining Techniques Adapter allowing different mining techniques to be adopted during data mining process. Finally, the interpreted results in XML format are sent to different data sources as output through send adapters.

JDMM will be designed as an information-intensive software, and as such, a proper caching is needed to allow maximum performance. Data Mining Repository (DMR) stores serialized objects captured from data sources at EJB Server. JDMM allows caching policies such as "no caching", "least recently used", "most recently used" and "caching duration" to be configured in an XML file. These policies are intended to tune the amount of cached data. All these policies such as caching algorithm, caching interval and other types of properties are specified in an XML-based configuration file residing on the where JDMM resides. The sole objective of the memory repository is to reduce any I/O during the process of mining the data sources. Majority of the data mining processes within JDMM are performed using the cached data from DMR. DMR is further divided into Caching Repository and Non-Caching Repository as shown in Figure 3. The Caching Repository stores both interpreted and uninterpreted data into JDMM Directory Service. The JDMM Directory Service catalogs each transformation and descriptor that occurs during clusterings, associations and others into a volatile directory. Implemented as part of JDMM is a snap-in which allows JDMM implementors to configure the JDMM Directory Service. All configurations within JDMM Directory Service are stored in multiple configurable XML files.

JDMM is cross platform and has no data access restrictions. Connecting to other data sources is made possible by plugging in an adapter component of the data source into the JDMM. The open architecture of JDMM intends to match with today's rapidly changing business requirements. JDMM transforms data and deploys advanced analytical and visualization techniques.

6. Conclusion

JDMM uses several XML schemas (also known as metadata) to describe the structure of both unmined and mined data. The schemas contain attributes and elements which are considered important by the authors as guidelines to formally explain both the unmined and mined data. JDMM uses XML as a mode of communication between each component

within JDMM, for example, the JDMM repository. The JDMM serves as a central place where both unmined and mined data is stored, cached and maintained. Both the unmined and mined data residing in the repository are distributed over one or more networks to share the heavy workloads during data mining. JDMM also uses the memory to perform data mining as the memory is many times faster than disk. Such an architecture intends to decrease the amount of time required to perform data mining parallely in a heavily cached distributed computing environment.

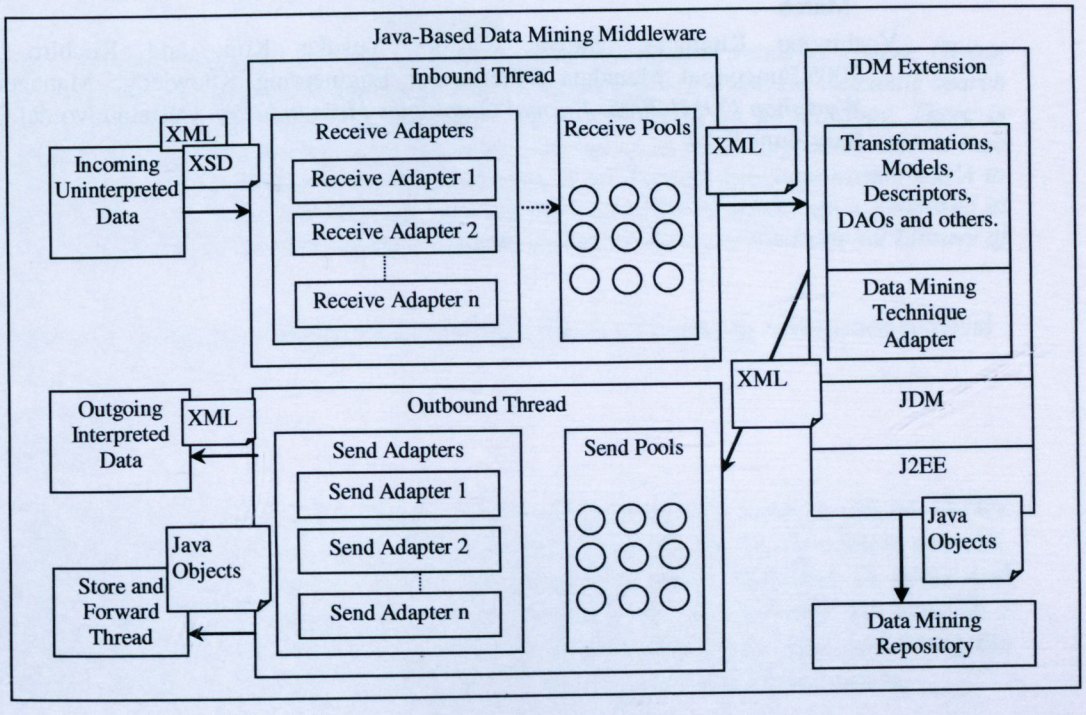

Figure 3: Detailed Software Architecture of JDMM

References:

B2 Intelligent Miner Library. 2002. *Using the Intelligent Miner for Data. IBM.* Version 8 Release 1.

Dare, Obasanjo. 2003. *Understanding XML.* Available at: http://msdn.microsoft.com/library/default.asp?url=/library/enus/dnxml/html/UnderstXML.asp. Microsoft Corporation. July.

Euzenat, J. Eight. 2002. Questions about Semantic Web Annotations. *IEEE Intelligent Systems.* March/April.

Giovanna Guirrini, Marco Mesiti and Daniele Rossi. 2005. Impact of XML Schema Evolution on Valid Documents. WIDM'05. *Proceeding the Seventh ACM International Workshop on Web Information and Data Management.* Bremen, Germany. November 5.

Jiawei Han and Micheline Kamber. 2006. *Data Mining: Concepts and Techniques.* Second Edition. Elsevier.

Lee ,Ching Kum and Lee Sai Peck. 2005. Integrating Xml With Relational Databases Using Middleware Approach, *Malaysian Journal of Computer Science.* Vol. 18, no. 2: 1-10

Nisbet, Robert A.. 2004. *How to choose a data mining suite.* Available at: http://www.dmreview.com/editorial/newsletter_article.cfm?nl=bireport&articleId=1000465&iss ue=20003.

Oracle Data Miner. Oracle Corporation. 2006. Available at http://www.oracle.com/technology/products/bi/odm/odminer.html.

Ray, Erik T. 2003. *Learning XML.* 2nd Edition. O'Reilly.

Sanjiv Purba. 2006. *Handbook of Data Management.* Viva Books Private Limited.

SQL Server 2005 Documentation. 2005. Microsoft Corporation.

O'Reilly, Doug Tidwell. 2001. *XSLT: Mastering XML Transformation.*

Mono. 2007. *What is Mono?* Available at: http://www.mono-project.com/Main_Page. March

Yoshinobu Kitamura, Naoya Washio, Yusuke Koji, and Riichiro Mizoguchi. 2005.Functional Metadata Schema for Engineering Knowledge Management. *First Workshop FOMI 2005: Formal Ontologies Meet Industry.* Castelnuovo del Garda (VR). Italy. June 9-10