TOWARDS DATA MINING SERVICES ON THE INTERNET WITH A MULTIPLE SERVICE PROVIDER MODEL: AN XML BASED APPROACH

Shonali Krishnaswamy, Arkady Zaslavsky
School of Computer Science & Software Engineering
Monash University, 900 Dandenong Road, Caulfield, VIC 3145, Australia
{shonali.krishnaswamy,arkady.zaslavsky}@csse.monash.edu.au

Seng Wai Loke
School of Computer Science and Information Technology
RMIT University, GPO Box 2476V, Melbourne, VIC 3001, Australia
swloke@cs.rmit.edu.au

ABSTRACT

The emergence of Application Service Providers hosting Internet-based data mining services is being seen as a viable alternative for organisations that value their knowledge resources but are constrained by the high cost of data mining software. In this paper, we present a new multiple service provider model of operation for the Internet delivery of data mining services. This model has several advantages over the currently predominant approach for delivering data mining, services such as providing clients with a wider variety of options, choice of service providers and the benefits of a more competitive marketplace. We have developed XML DTD's to support the interaction protocols for the multiple service provider model including: specification of the task preferences of clients, specification of the functionality of data mining service providers and the exchange of information to access data and computational resources to perform the task. We have also developed a matching algorithm to map the task preferences of the client preferences to the functionality of the data mining service providers.

1 Introduction

Organizations are increasingly placing emphasis on knowledge as the key to providing them with a competitive edge and supporting the strategic decision making process. Data mining is an important technology for organizational "business intelligence". The recent trend of the Application Service Provider (ASP) paradigm is leading to the emergence of Internet-based data mining service providers such as digimine™ (http://www.digimine.com/) and Information Discovery™(http://datamine.aa.psiweb.com/). This option of Internet-delivery of data mining services is emerging as an attractive option for small to medium range organisations, which are the most constrained by the high cost of data mining software, and consequently, stand to benefit by paying for software usage without having to incur the costs associated with buying, setting-up and training.

Application Service Providers underline the commercial viability of "renting" software out [Clark-Dickson 1999]. Thus, instead of buying an expensive software package and installing it, organizations logon to an application service provider (either through the Internet or dedicated communication channels), use the packages provided by the ASP and pay for this usage. Data mining has several characteristics, which allow it to fit intuitively into the ASP model. The features that lend themselves suitable for hosting data mining services are as follows:

Diverse requirements: Business intelligence needs within organisations can be diverse and vary from customer profiling and fraud detection to market-basket analysis. Such diversity requires data mining systems that can support a wide variety of algorithms and techniques. Data mining systems have evolved from stand-alone systems characterised by single algorithms with little support for the knowledge discovery process to integrated systems incorporating several mining algorithms, multiple users, various data formats and distributed data sources. This growth and evolution notwithstanding, the current state of the art in data mining systems makes it unlikely for any one system to be able to support all the business intelligence needs of an organisation. Application Service Providers can alleviate this problem by hosting a variety of data mining systems that can meet the diverse needs of users.

Increased demand for business intelligence. Technologies like e-commerce provide an opportunity for small and medium range companies to compete in global markets, which were previously the domain of large organisations and multi-nationals. These companies are hitherto looking towards business intelligence tools to provide them with a competitive edge, by maximising the gain obtained from their information resources and supporting their strategic

decision-making process. The high cost of data mining software can be prohibitive for small to medium range organisations. In such cases, the application service providers are a viable and intuitive solution.

Need for immediate benefits. The benefits gained by implementing data mining infrastructure within an organisation tend to be in the long term. One of the reasons for this is the significant learning curve associated with the usage of data mining software. Organisations requiring immediate benefits can use ASP's, which have all the infrastructure and expertise in place.

Specialized Tasks. Organisations may sometimes require a specialised, once-off data mining task to be performed (e.g. mining data that is in a special format or is of a complex type). In such a scenario, an ASP that hosts a data mining system that can perform the required task can provide a simple, cost-efficient solution.

There are two aspects to delivering web-based data mining services. The first approach focuses on the provision of data mining facilities and access to this infrastructure as a web-based service [Krishnaswany et al. 2000b; Krishnaswany et al. 2001]. The second concept involves providing data mining *models* as services on the Internet [Sarawagi & Nagaralu 2001]. The currently predominant modus-operandi is for organisations that require data mining services to send their data to the service provider (with whom the company typically has a long term service level agreement (SLA)) and access the results through a web interface. However, there are situations where this model is not adequate. Consider the following scenarios and the questions they raise.

Case 1. An organisation requires a specialised data -mining task to be processed, for example, it has acquired a data set that is of a complex data type and needs this to be mined for patterns. The company's data mining service provider does not have the ability to process this task.

1.What are the mechanisms for locating the appropriate service providers?

2.How is the most cost-effective and efficient service provider determined? What are the performance metrics that are available?

3.What languages for task description such as DMQL (Data Mining Query Language) [Han et al. 1996] and the Microsoft OLE DB standard for data mining [Microsoft 2000] does the service provider support?

4.In the context of ASP hosted services, what are the parameters required to specify a data mining task in addition to the traditional specifications such as the type of output required, the data set, the background knowledge, qualitative and quantitative measures [Han et al. 1996] For instance, in the above case, how is it specified that the requirement is mining spatial data?

Case 2. An organisation is unwilling to ship its sensitive data across to the service provider.

1.Does the service provider have the infrastructure to provide on-site mining? What are the mechanisms that can make this possible?

Case 3. An organisation is particular about having an increased level of control over the data mining process.

1.How would they be able to specify requirements such as the mining algorithm to be used or the need for comparative results from different mining techniques?

It is evident that the concept of providing Internet-based data mining services is still in its early stages and there are several open issues – some of which are generic to ASP's - such as:

- Lack of performance metrics for the quality of service.
- Lack of well-defined models for costing and billing of data mining services.
- Standards that still in their early stages (in the case of data mining, standards such as Microsoft's OLE DB [Microsoft 2000] initiative are still emerging and PMML (Predictive Model Markup Language) [Grossman etal. 1999; Ramu 1998] focuses on describing predictive models generated by the data mining process, rather than data mining services)
- Need for mechanisms to describe data mining task requests and services.

In this paper, we consider two models of operation for data mining service providers – the current approach involving a single service provider and a new model involving multiple service providers - and discuss the information exchange between clients and service providers in both these models. We use the interaction protocols between clients and the service providers to motivate the need for specification of data mining task requests that adequately represent the requirements and constraints of the clients and also illustrate the importance of description mechanisms for data mining systems and services in order to support Internet delivery of such services. We present XML document type definitions for describing data mining task requests and the functionality and services of data mining service providers. The paper is organized as follows. Section 2 surveys the landscape of commercial data mining service providers and emerging infrastructures to support e-services. Section 3 presents the interaction protocols for the currently predominant single service provider model and introduces a multiple service provider model for Internet delivery of data mining services. It discusses the comparative differences in operation and interaction protocols between the two models. Section 4 presents the information exchange process for the different models of operation. Section 5 presents the document type definitions for the different XML messages that are

exchanged between clients and data mining service providers to support interactions in a multiple service provider model. Section 6 presents our matching algorithm for showing how client preferences can be mapped to the functionality supported by service providers. Section 7 compares our work with related research and section 8 concludes the paper.

2 Related Work

The context for the work presented in this paper is the current modus operandi of Internet-based data mining service providers and the emerging infrastructures to support market places of e-services. We present a multiple service provider model for delivering web-based data mining services as an alternative to the currently predominant approach. This model is suited for operation in e-services environments, which allow registration and discovery of service providers. In this section we present a survey and comparative analysis of commercial ASP's who provide data mining services to motivate the benefits of our multiple service provider model. We then discuss emerging e-services platforms and architectures that facilitate a multiple service provider model for data mining services.

2.1 Case Studies - Review of Commercial Data Mining Service Providers

In this section we discuss the operation of commercial data mining service providers. The potential benefits and the intuitive soundness of the concept of hosting data mining services is leading to the emergence of a host of business intelligence application service providers. We first present a brief description of the following ASP's: digiMine (http://www.digimine.com), Information Discovery (http://datamine.aa.psiweb.com/), iFusion (http://www.kineticnetworks.com/), ListAnalyst.com (http://www.listanalyst.com/), WhiteCross Systems [http://www.whitecross.com/] and WebMiner (http://www.webminer.com/). We have directed our analysis towards those vendors who primarily focus of hosting data mining systems as an ASP service. We then present a comparative evaluation of these service providers in terms of their focus, services and functionality, charging models and mode of operation.

2.1.1 digiMine™

digiMine™ (http://www.digimine.com/) is a business intelligence service provider that offers the following services:

- Warehousing Service – extracts data from multiple data sources and builds a scalable data warehouse
- Analytic Services – performs different forms of data analysis including web site usage, shopping cart analysis and customer profiling
- Data Mining Services – applies data mining algorithms (developed in-house by digiMine) to the clients data
- Data Enhancement Services – cleans data by removing duplication and reducing noise and error

The model of operation involves installation of a digiMine tool called DataSlurper at the client site, which encrypts and compresses the data for transfer. DataSlurper can extract logs from web servers and data from commercial DBMS's. The clients also have the option of sending their data through FTP to the ASP. On completion of the transfer, the data is cleaned and loaded into a warehouse and mining is performed. The results are accessed as reports via a secure web interface.

2.1.2 Information Discovery™

Information Discovery™ (http://datamine.aa.psiweb.com/) is primarily a data mining ASP whose focus is on *pattern management*, which is the storage, querying and retrieval of patterns discovered by data mining. They have a query language called *Pattern Query Language (PQL)*, which facilitates the querying of patterns from a pattern base. The mode of operation is for the data to be transferred to the ASP site where data mining is performed. The discovered patterns are loaded in to a pattern-base, which resides at the client site and can then be accessed by the client locally using PQL. Explanation documents are automatically generated for the patterns, to aid in their understanding by the client. The patterns are combined and updated on a monthly basis. The data mining system that is hosted is one that was developed by Information Discovery and focuses on mining relational data.

2.1.3 iFusion™

iFusion™ from Kinetic Networks (http://www.kineticnetworks.com/) also operates by transferring data from the clients into their local server. They offer data integration services, which extract the data from multiple sources and translate it into the format required by their data mining system. The data analysis service performs the mining and delivers standardized reports. It also supports customisable ad-hoc reports and real-time querying. The results are accessed via a secure web interface. Unlike in Information Discovery, a consultant provides the explanation for the results.

2.1.4 ListAnalyst.com™

ListAnalyst.com™ (http://www.listanalyst.com/) focuses on customer profiling. Clients are required to upload lists of customer profiles (and any other related information) through a web interface. The client can also choose one of the four services offered:

- List Profiler – which finds the "best" market segment for the client
- List Compare and Clone – which analyses differences between customers and prospective customers
- Response Multiplier – which identifies customers who are most likely to respond to a campaign
- Sales/Profit Multiplier – which identifies customers that account for most profits

The analysis is performed using their product MarketMinerä and the results are sent to the clients.

2.1.5 WhiteCross Systems™

The focus of WhiteCross™ (http://www.whitecross.com/) is on mining web logs for the client. It operates by extracting information from clients web logs and processes it by combining it with any other relevant data on a fast server at the ASP site. This type of data mining is directed towards e-businesses who might require analysis of visitors to their web site. The model of operation involves WhiteCross consultants determining client needs such as frequency of reports and type of access. The client then has to ensure that the log files and other data sources can be accessible by the ASP. The ASP delivers predefined reports via a secure web interface. The reports can be in any required format, including HTML, text or PDF. The clients have full access to the database containing the results of the analysis. This database can be accessed through a web interface for further drilling and viewing customized reports (i.e. the ASP supports a thin client).

2.1.6 WebMiner™

WebMiner™ (http://www.webminer.com/) is data mining service for e-commerce web sites. The principal data mining technique used is predictive modelling. It uses predictive modelling techniques to drive marketing strategies for e-commerce web sites (i.e. convert browsers into buyers). Interestingly, WebMiner is the only ASP who has published their charging model. The charging model includes three components:

- Set-up Fee – this is a one time cost for performing data analysis and implementing the ASP infrastructure
- Maintenance Usage Fee – this is a monthly fee for on-going monitoring and maintenance. It includes web reports, IT support and analysis updates (the frequency of which depends on the clients activity levels)
- Performance Commission – this is a percentage of the total sales made due to WebMiner's focused offers to clients (who accept the offer and make a purchase).

It is evident that WebMiner operates on the principal of managing the entire profiling and targeting of customers for a given e-business client. Thus they provide IT support, reports to the clients and drive the marketing strategy based on their predictive modelling techniques. They offer more than merely mining the data and consequently charge on commission basis.

It can be seen that the currently predominant modus operandi for data mining ASP's is the single-service provider model. It is evident that all of today's data mining ASP's operate using a client-server model, which requires the data to be transferred to the ASP servers. There is no data mining service provider who uses an alternative approach (e.g. mobile agents) to deploy the data mining process at the client's site. However, the development of research prototypes of distributed data mining (DDM) systems such as JAM [Tewari & Maes 2000], Papyrus [Ramu 1998], Bodhi [Kargupta et al. 1998] and DAME [Krishnaswany et al. 2000a; Krishnaswany et al. 2000b] show that this technology is a viable alternative for distributed data mining. The use of a secure web interface is the most common approach for delivering results (e.g. digiMine, WhiteCross and iFusion), though Information Discovery sends the results to a "pattern-base" (or a knowledge-base) located at the client site. Another interesting aspect is that most service providers host data mining tools that they have developed (e.g. digiMine, Information Discovery, ListAnalyst.com, and WhiteCross). This is possibly because the developers of data mining tools are seeing the ASP paradigm as a natural extension to their market. This trend might also be due to the know-how that data mining tool vendors have about the operation of their systems. A comparative evaluation of the six data mining service providers is summarized in table 1.

2.2 E-Services Infrastructures

It is obvious that the mode of operation for ASP's hosting data mining services does not reflect a marketplace environment of e-services where clients can make ad-hoc requests and service providers compete for tasks. However, the emergence of e-services platforms such as e-speak (http://www.e-speak.hp.com/) and enabling infrastructures such as UDDI (http://www.uddi.org/) facilitate the registration and discovery of service providers. This has the potential to bring about a transformation to the current model of operation. In this section, we present an overview of the following emerging standards and systems that support services on the Internet:

- E-Speak is an e-services infrastructure developed by Hewlett Packard (http://www.e-speak.hp.com/). It provides the underlying technology to support registration of services by service providers and location of services by clients. A service provider registers a service by providing a description using a "vocabulary".

Table 1. Comparison of Data Mining Service Providers

	digiMine	Information Discovery	IFusion	ListAnalyst.com	WhiteCross	WebMiner
Data Mining Tasks	Customer profiling, Web site usage, Shopping Cart Analysis, Associations	Pattern discovery and analysis	General data analysis – not specified	Customer profiling	Mining Web Logs	Predictive modelling of customers of e-business web sites
Data Mining Tools	In-house algorithms – not specified	Data Mining Suite from Information Discovery	In-house algorithms – not specified	MarketMiner from ListAnalyst.com	Not specified	In-house algorithms – not specified
Operational Model	Transfer data from client, perform mining at ASP site	Transfer data from client, perform mining at ASP site	Transfer data from client, perform mining at ASP site	Transfer data from client, perform mining at ASP site	Transfer data from client, perform mining at ASP site	Mange the entire IT infrastructure and marketing strategy
Data Transfer	FTP, automated data extraction, encryption and transfer	Not specified	Transfer mode not specified, offers data integration services	Clients upload through a web interface	Client has to give access to web logs, which are accessed by the system	ASP manages the web logs and marketing strategies
Services	Data warehousing, analysis, mining and cleaning	Data mining and monthly updates of the patterns, automated explanation of the patterns	Data integration, analysis and mining, support from business consultants	Customer profiles and market segmentation, comparison of customer profiles, identify most like customers who will respond and those that contribute most to profits	Mining of web logs, clients have full access to database of results	Mining of web logs, proactive marketing
Result Delivery Model	Secure web interface	Results are loaded onto a pattern-base at the client site	Secure web interface	Sent to the client	Clients query a database of results through a web interface	Not specified
Result Format	Reports	Patterns	Reports – standardized and customisable, real–time querying	Reports	Reports (in any required presentation format – HTML, PDF etc), further analysis / drilling from database of results	Not specified
Charging Model	Not specified	Not specified	Not specified	Not specified	Not specified	Set up and maintenance fees, performance commission

•Web Services Description Language (WSDL) (http://msdn.microsoft.com/xml/general/wsdl.asp) is an XML format developed by Microsoft to support the communication and network details to support exchanging messages over the web. Elements in a WSDL document include information such as the port, the port type, the network bindings and the actual message that is being exchanged. It facilitates web services at the lower level of data communication and is not intended for describing the semantics of services.
•Universal Description, Discovery and Integration (UDDI), is a standard developed and supported by Ariba, IBM and Microsoft (http://www.uddi.org/). UDDI is a universal registry that allows global registration and description of web services to facilitate the location of services by clients and the interaction to enable usage of the service.
•E-Business-XML (eb-XML) is an initiative supported by United Nations and OASIS (http://www.ebxml.org). The stated objective of eb-XML is to support open and interoperable e-business. The eb-XML architecture provides mechanisms to define business processes and their associated messages and the registration and discovery of business process sequences with the related message exchanges.

In summary, it can be seen that there are several emerging standards and infrastructures to support Internet delivered services. These technologies provide an opportunity for an Internet market place of data mining services. In the following sections of the paper, we present such a model and evaluate its benefits and disadvantages.

3 Models of Operation for Data Mining Service Providers

This section presents alternative models of operation for data mining service providers. These models illustrate the context of interaction and communication between "clients" and data mining service providers. We discuss the existing single service provider model – where one ASP host several data mining systems - and present our alternative multiple service provider model – where several ASP's host one or more data mining systems.

3.1 Single Service Provider Model

This model, as illustrated in figure 1, has simpler operational semantics of the two and is the currently predominant approach to providing Internet-based data mining services. A client organization has a single service provider who meets all the data mining needs of the client. The client is well aware of the capabilities of the service provider and there are predefined agreements regarding quality of service, cost and protocols for requesting services. The service provider hosts one or more distributed data mining systems, which support a specified number of mining algorithms. The service provider is aware of the architectural model, specializations, features and required computational resources for the operation of the distributed data mining system.

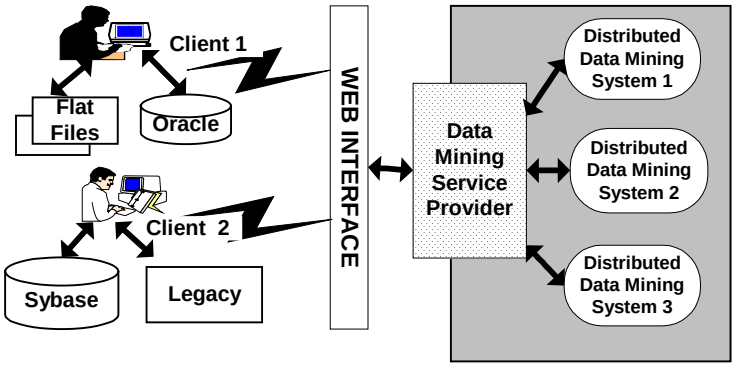

Figure 1 Single Service Provider Model

The interaction protocol for this model is as follows:
1. Client requests a service using a well-defined instruction set from the service provider.
2. The service provider maps the request to the functionality of the different DDM systems that are hosted to determine the most appropriate one.
3. The "suitable" DDM system processes the task and the results are given to the client in a previously arranged format.

The implication of having to map task requests to an appropriate DDM system is that it necessitates a structured means of specifying the functionality of each individual DDM system in terms of the algorithms and data types it supports and its architecture (i.e. whether it performs distributed mining in client-server mode, uses mobile agents or

a hybrid approach). Additional information that is required might be performance metrics such as the time taken to perform a given task by a system, the computational resources provided for the system and details such as the format in which a data-mining task is specified and specifying the output produced. The key to automated mapping of task requests to appropriate DDM systems is a specification language for describing data mining task requests and the services, architectures and functionality of data mining systems that are hosted by ASP's. While such a language is not paramount for the functioning of this model, it is definitely advantageous. The primary characteristics of the single service provider approach for hosting data mining services are:

- It satisfies the basic motivations for providing data mining services and allows organisations to avail the benefits of business intelligence without having to incur the costs associated with buying software, maintenance and training.
- The cost for the service, metrics for performance and quality of service are negotiated on a long-term basis as opposed to a task-by-task basis. For example, the number of tasks requested per month by the client and their urgency may form the basis for monthly payments to the service provider.

However, the model has limitations such as the inability to meet the needs of the scenarios outlined in section 1. It implicitly lacks the notions of competition and that of an "open market place" which gives clients the highest benefit in terms of diversity of service at the best price. In summary, this model falls short of allowing the Internet to be an electronic market place of "services" (as it is becoming for goods).

3.2 Multiple Service Provider Model

This model, as illustrated in figure 2, is characterized by clients being able to request data mining services from several service providers who host one or more DDM systems.

This approach provides a higher level of flexibility for the client and represents the establishment of an open, virtual market place of data mining service providers. The multiple service provider model operates in the form of a "federation" [Krishnaswany et al. 2001]. The "federation manager" is a coordinating component in the system that manages the interactions between the client and the data mining service providers. The interaction protocol for this model is as follows:

1. The client requests a service by providing a task specification to the federation manager. It must be noted that the parameters for specifying the task must be well defined and must facilitate the requests to be made at the level of granularity that the client deems appropriate.

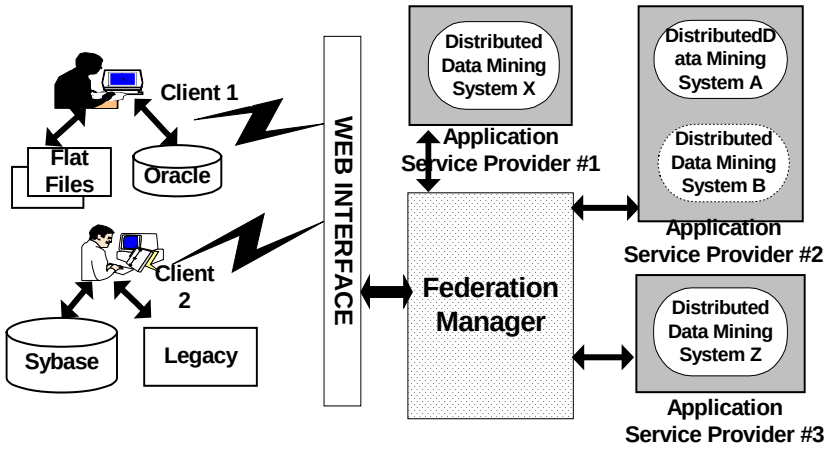

Figure 2 Multiple Service Provider Model

2. The federation manager broadcasts the client's requests to the data mining service providers that are registered with it. The federation manager maintains information about each data mining service provider such as the name, address, contact information, DDM systems hosted, algorithms, architectures and functionality supported by those systems and the computational resources that the service provider has.

3. The data mining service providers evaluate the requested task against the capabilities and functionality of the DDM systems that they host.

4. If they can meet the needs of the requested task, the data mining service providers respond by presenting an estimate of the cost, possible time frame for completion of the task and liabilities for not meeting the targeted response time. This information is presented to the federation manager in a specific, structured format.

5. The federation manager can either present the responses it receives along with the information that it already maintains about the respective service providers to the client or in more sophisticated environment can perform matching of client preferences with the capabilities of the service providers and rank the service providers on that basis and present this to the client. There is also scope for automated negotiation to be incorporated into this stage of the interaction protocol. It must be noted that service-provider ranking, preference matching and automated negotiation are emerging research issues in the e-services domain. We present a brief discussion on these aspects in section 6.3, but it is not the core focus of this paper.
6. The client decides which service provider it deems most appropriate and informs the federation manager and the chosen service provider.
7. The service provider gives the client a legal document/contract, which makes the commitment to maintain the confidentiality of the data that is mined and the consequent knowledge that is produced.
8. The client is then required to provide a security deposit in the form of a credit card number to the federation manager. The actual payment is made on completion of the task and provision of results to the client.
9. The client and the data mining service provider exchange information regarding the transfer of data, passwords to access systems and the mode of transfer of results.
10. The data mining service provider processes the task, provides the results to the client (in the agreed format and method) and informs the federation manager of task completion.
11. The client acknowledges the completion of the task to the federation manager and the payment is made to the service provider.

This model overcomes the limitations and restrictions imposed by the previous approach in meeting the needs of the requirements outlined in section 1. The operation of this model centres around two concepts. Firstly, there is a need for a well-defined and structured mode for exchange of information between the clients and the service providers. The issues that arise in this context are:

- What are the parameters that need to be specified at the various stages of the interaction protocol?
- What is an appropriate format and medium for the information exchange?

Secondly, there is a need for a coordinating entity such as the "federation manager". The principal role of the federation manager is to maintain a registry of data mining service providers and their basic capabilities. It also acts as an intermediary in the interactions between the clients and data mining service providers. The federation manager must provide the infrastructure for the following basic operations:

- Allowing data mining service providers to register and de-register themselves with the federation.
- Maintaining information that reflects the current capabilities of the data mining service providers that are registered.
- Coordinating the interactions between the clients and service providers.

As opposed to the previous approach, the multiple service-provider model involves a short-term contractual arrangement between the client and the service provider. This necessitates the determination of service costs and quality of service levels (such as response time) on a task-by-task basis. This also requires contracts of confidentiality for the data and the knowledge that is produced to be drawn for every service that is required. In summary, this model provides a wider choice of data mining services for clients and caters for selection of the most appropriate and cost-efficient service provider by the implicit competition in the approach. The Internet provides the infrastructure to facilitate an on-line market place for electronic goods; the multiple service provider model presented above, is a step towards establishing an on-line market place for data mining services.

4 Information Exchange Process

The previous section illustrated the variation in the interaction protocols and the information exchanges between the clients and the data mining service providers based on the different models of service provider organization. At a high level of abstraction the information exchanged can be classified into five categories as shown in table 2.

There is a significant difference in the manner in which the information exchange takes place between the models involving single and multiple service providers. The principal difference, as illustrated in figure 3, is that information such as access information for the client and access information for the service provider need only be exchanged only once in the single service provider model. This does not hold for the multiple service provider model, where the contractual agreement between the service provider and the client is relatively short-term (and typically limited to a single task). This in turn necessitates exchanging all the information on a per-task basis. Further, in the multiple service provider model, the data mining task description, client access information and service provider access information (indicated in figure 3 as stages (3) and (4)) take place only after the appropriate service provider is selected by the client on the basis of the descriptions provided (as indicated by stage (2)). After

selecting the service provider the information exchange occurs directly between the client and the service provider by passing the intermediary.

Table 2 Summary of Information Exchanged

Information	Description	Provider
Data Mining Task Request	General description of the task and the preferences of the client	Client
Service Provider Description	General description of the service provider capabilities	Service Provider
Data Mining Task Description	Detailed specification of the mining task (e.g. condition, decision attributes etc.)	Client
Client Access	Specification of how to access the client for either transferring data or to mine the data locally	Client
Service Provider Access	Specification of how to access the results on completion of the task	Service Provider

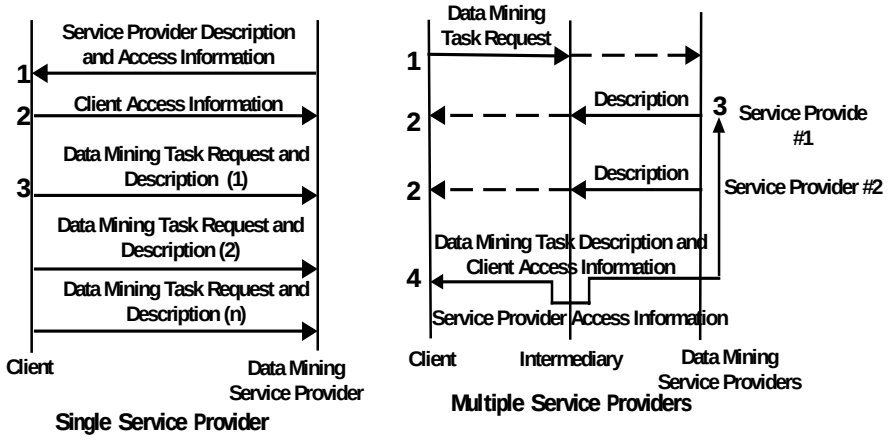

Figure 3. Comparison of the Information Exchange Process

Based on the exchanges presented in figure 3, we now specify the information content that is passed between clients and data mining service providers at each stage of the interaction. We use XML documents to manage the exchange of information. XML is widely used in metadata standards for data content representation and is suitable for standardized information interchange in many domain specific contexts [Goldfarb & Prescod 1998]. Moreover, it is now the defacto approach for exchanging information in e-services environments. We believe that XML provides a suitable basis because of the following reasons:

- XML query languages can be used to query a collection of XML documents (http://www-db,research.bell-labs.com/user/simeon/xquery.html). Thus, the federation manager can query a collection of service provider descriptions to locate an appropriate service.
- XML is humanly readable.
- The Document Object Model (DOM) allows access to XML documents from with programming languages.
- XSLT (Extended Style Sheet Language Transformation) (http://www.w3.org/TR/xslt) allows XML documents to be converted to other languages such as HTML, WML.

The following section presents the XML Document Type Definitions (DTD) for the documents/messages that are exchanged.

5 Document Type Definitions

As discussed in section 3, there are five types of documents that are exchanged in the interaction between clients and data mining service providers. They are:

- Data mining task request
- Service provider description
- Data mining task description
- Client access information
- Service provider access information

In this section, we present the structure and content of these documents along with fragments of XML code as illustrative examples. The complete document type definitions are presented in the Appendix.

5.1 Data Mining Task Request

A data mining task request is initiated by the client and is a general description of the type of service required and the client's preferences and constraints for the task. It does not contain a detailed specification of the data mining task. This separation of the client's task request and the actual task specification is to facilitate the short-term interaction model where the client might not wish to broadcast sensitive/confidential information about their data. The specification of the task description has been developed to accommodate clients who might wish to include detailed requirements such as format of the results and the type of algorithms to be used, and those who might not have such fine grained knowledge about the data mining process and might wish to describe their requirements at a higher, more general level of abstraction. The information contained in this DTD is as follows:

Task Descriptor. The task descriptor indicates the basic type of data mining task that is required such as market basket analysis, customer profiling, and fraud detection. A client requiring customer-profiling and fraud-detection would specify this as follows:

```
<task-descriptor task-type = "customer-profiling">
    <misc> </misc>
</task-descriptor>
<task-descriptor task-type = "fraud-detection" >
    <misc> </misc>
</task-descriptor>
```

Algorithm Preference. This indicates the preference or requirement for a specified data mining algorithm to be used. This is optional and can be left unspecified when the client has no preference. All constraints specified in this DTD allow specification of whether the constraint is preferred or required. A client specifying a preference for the algorithm ID3 would include the following:

```
<algorithm-preference>
     <algorithm-details name ="ID3"> </algorithm-
details>
    <pref-constraint choice="preferred"> </pref-
constraint>
</algorithm-preference>
```

Output Format. This specifies the preferred or required output format for the results (e.g. association rules, classification rules, visual/graphical output). The clients can also optionally specify the knowledge integration technique they require or prefer (for cases where the data is distributed and there is a need for integrated results). The following code fragment specifies the preference for results in the form of decision trees and the use of PMML (Predictive Model Markup Language) as the knowledge integration technique.

```
<result-preference>
     <result-details result-format="decision-trees">
     </result-details>
     <knowledge-integration><KI-technique
type="PMML">
     </KI-technique></knowledge-integration>
    <pref-constraint choice="preferred"> </pref-
constraint>
</result-preference>
```

Comparative Results. This allows the client to specify that they require the dataset to be mined using two or more mining techniques and a comparative analysis of results to be presented. For example, the following code fragment specifies a preference for a comparative analysis of results (presented as decision trees) using the algorithms ID3 and C4.5.

```
<comparative-results>
     <algorithm-details name ="ID3">
        <pref-constraint choice="preferred">
        </pref-constraint>
     </algorithm-details>
     <algorithm-details name ="C4.5">
        <pref-constraint> choice="preferred">
        </pref-constraint>
     </algorithm-details>
    <result-details result-format = "decision-trees">
    </result-details>
    </KI-technique></knowledge-integration>
    <pref-constraint choice="preferred"> </pref-
constraint>
</comparative-results>
```

Data Type. This indicates the type of data that has to be mined (e.g. text, ascii, temporal, relational etc.).
Data Format. Additionally, if the data is in a standardised format it can be specified.
Dataset Size. This information is provided so as to allow the service provider to form an accurate estimate of the response time. The following example illustrates a specification of the data type as text and the data set size as 30 gigabytes.

```
<data>
      <data-types data-type="text"/>
     <size unit="GB" measure="30"/>
</data>
```

Location. This indicates the preferred or required location for mining to occur. Thus, the client can specify this option as either “local” (to indicate that the data cannot be shipped and must be mined on site), “remote” (to indicate that the data can be mined remotely) and “anywhere” (to indicate no preference). In case “local” or “anywhere” is specified, the computational resources that are available for the data mining process must also be declared. This allows service providers to estimate the response time for the task (in the context of the local resources). The following example specifies the location preference as “local” and states that the client has one dedicated server running Windows NT with 128 MB of memory and 233 MHz processor.

```
<location location-preference="local"/>
<comp-resources>
   <server name="muruga.csse.monash.edu.au"
dedicated="YES">
      <type>
         <stand-alone>
            <configuration OS="Windows-NT">
               <config-total CPU="233 MHz"
               memory="128 MB"/>
            </configuration>
         </stand-alone>
      </type>
   </server>
</comp-resources>
</location>
```

Response Time. This indicates the required response time for completion of tasks and can be specified as a range with an upper and lower limit. The unit can be specified as either hours or days. The following example specifies that the client has an upper limit of 20 hours for the completion of the task.

```
<response time>
   <time measure="hours">
      <range-constraint max="20" min="0"/>
    </time>
</response-time>
```

Cost. This can be used to optionally indicate the cost that the client is willing to bear for the service. The cost is also specified as a range. In the following example, the client specifies that they are willing to pay a maximum of USD 5000 for the task.

```
<cost measure="USD">
    <range-constraint max="5000"/>
</cost>
```

Miscellaneous Services. This part of the DTD allows clients to specify other miscellaneous services they might require such as pre-processing, support for mobile-users, parallel processing. The dm-task-request DTD is attached as Appendix A.

5.2 Service Provider Descriptor

The service provider descriptor contains information about the service provider and the data mining systems hosted by the service provider. In [Krishnaswany et al. 2001] we presented Distributed Data Mining Systems – Markup Language [DDMS-ML], an XML DTD that allows distributed data mining systems to specify their architectures and functionality. The service provider descriptor principally contains a modified version of DDMS-ML documents to represent and describe each data mining system that is hosted. The information contained in this document is as follows:

Service Provider Details. This includes information such as the name of the service provider, address, business registration details and contact details (phone, fax, email and web address). A sample fragment of the XML document is illustrated as follows:

```
<spd>
<sp-details name="ABC"
    bus-reg="X1X10X1X1"
    address="900 Dandenong Rd, Caulfield East, VIC - 3145,
Australia"
    email="system@abc.com" phone="61 3 9903 2000"
    fax ="61 3 9903 1077" url="http://www.abc.com"
    date="01/01/2001/>
 <spd>
```

Capability Descriptor. This provides a high level description of the type of data mining tasks that the service provider supports such as: fraud-detection, market-basket-analysis, customer-profiling, dependency-analysis, clustering, segmentation and predictive-modelling. Further, we have incorporated the facility to specify any other provider specific description. For example, to specify that the DAME system supports dependency analysis:

```
<capability-descriptor>
    task-type="dependency-analysis"
</capability-descriptor>
```

Computational Resources. This component includes information about the computational resources that the service provider supports. A service provider can have several servers. A server is either a stand-alone system or a parallel server or a cluster. A server may or may not be dedicated for distributed data mining (depending on whether the service provider hosts a variety of applications). A server's physical configuration such as the operating system, the CPU, the memory and the number of nodes (if the server is a cluster) is recorded. This information allows the client to understand the infrastructure of the service provider. Consider the following example, where the service provider has two dedicated servers –one having Linux as its operating system and one having Solaris:

```
<comp-resources>
   <server name="nemesis.csse.monash.edu.au"
dedicated="YES">
      <type>
         <stand-alone>
            <configuration OS="Solaris">
               <config-total CPU="" memory="128 MB"/>
            </configuration>
         </stand-alone>
      </type>
   </server>
   <server name="milkyway.sd.monash.edu.au"
dedicated="YES">
      <type>
         <stand-alone>
            <configuration OS="Linux">
               <config-total CPU="" memory="128 MB"/>
            </configuration>
         </stand-alone>
      </type>
   </server>
</comp-resources>
```

Task Specific Details. This includes details that are specific to a particular task. Depending on the nature and longevity of the relationship between the clients and the service provider, this information may or may not be exchanged. Further, in the multiple service provider this information is provided after the task request has been made. The information specified in this component are details such as the estimated response time for the task, the cost of the task (depending on where it is deployed and what architectural model is requested by the client) and the liabilities in terms of constraints and the respective cost-reductions (e.g. in case of an inability to meet the estimated response time, the cost of the task reduces by *x* dollars per hour of delay). In the following example, the service provider undertakes to complete a task in 10-14 hours for a cost of $4000. The service provider also agrees to reduce $200 for every half hour after the 14 hour period has lapsed.

```
<task-specific-details>
 <task-metrics>
   <response time>
    <time measure="hours">
         <range-constraint max="20" min="0"/>
    </time>
   <liability-constraint="Complete Task in Specified
Time"
     cost-reduction="$200 per additional half hour"
   </response-time>
   <arch-model architecture="client-server"/>
  </task--metrics>
  <cost amt ="$4000">
    <arch-model architecture="client-server"/>
   </cost>
</task-specific-details>
```

Data Mining Systems. This includes details about the different data mining systems that are hosted by the service provider. In this section we present a description of the Distributed Agent Based Mining Environment (DAME) [Krishnaswany et al. 2000a; Krishnaswany et al. 2000b], which is our hybrid distributed data mining system, as the example XML document that conforms to the service-provider-descriptor DTD. A data mining system is described in terms of its functionality and architecture as follows:

•*Meta Information.* This part of the document contains information about the DM system such as its name, version, date of development, organisation and developer. We impose the constraint that within each service provider descriptor DM system must have a unique name.

```
<dm-system name="DAME" version="1.0"
    organisation="Monash University"
    developer="Shonali Krishnaswamy"
    date="March 2000"
</dm-system>
```

•*Architectural Model.* This component states whether the DM system uses the client-server approach, the mobile agent paradigm or a hybrid model (integration of both client server and mobile agents) for distributed data mining. This information is important in situations where a client requires a particular architectural model. For instance, a situation where the data to be mined is sensitive and the client does not want the data to be transported to the service provider will warrant that a DM system that uses mobile agents be used. It is also possible then that the mobile agent performs its task and is destroyed and not allowed to leave the site to provide further protection to the client. Further, if the system supports the mobile agent model and/or a hybrid approach, the agent environment/toolkit required needs to be specified, to enable the client to have the requisite agent server in place for the data to be mined at their site. The DAME system supports the hybrid architectural model.

```
<arch-mode architecture="hybrid"/> </arch-model>
```

•*Specialisations and Features.* This section of the document allows DM systems to describe their distinguishing functions and special services. Similar to the data types, the DTD has some pre-specified options such as support for parallel algorithms, optimisation, cost-efficiency, pre-processing, mobile users and visualisation. However, it also allows a DM system to present any other special features that it may possess. This component also includes information about support for “knowledge integration” which is the process of integrating results obtained from

distributed data sets. The DAME system supports cost-efficiency, optimisation and mobile users. It does not support knowledge integration.

```
<features>
    <services type="cost-efficiency"/>
    <services type="optimisation"/>
    <services type="mobile-users"/>
    <knowledge-integration>
        <KI-technique type="None"/>
   </knowledge-integration>
</features>
```

•*Algorithms.* This component specifies the mining algorithms that are supported by the DM system. The specification includes details such as the algorithm's name and optionally the version. This is followed by details regarding the structure of the input file for the algorithm and the type of output model produced. The current version only allows the specification of a text input file. Specifying the structure of complex data files is a non-trivial task and is not part of our current focus. For such input data, the DTD only allows specifying the data type and the respective file extension required. This component of the document allows the presentation of details about algorithm usage to clients who might wish to use a particular system. The algorithms supported by the DAME system are described as follows:

```
<algorithms>
 <algorithm algorithm-name="ID3" algorithm-
version=""
   algorithm-author="" algorithm-date=""
   algorithm-misc="WEKA package"
 <ip-file file_extension=".arff">
    <data-type type="ascii"/>
    <structure>
          <text-file-descriptor
          header="@relation relation-name"
          comments="%"
          attribute-descriptor="@attribute attribute-
name type"
          attribute-types="Nominal,  string" data-
descriptor="
          @data value1, value2,...,value n" field-
delimiter=","/>
   </structure>
 </ip-file>
 <op-model>Decision Tree</op-model>
</algorithm>
</algorithms>
```

•*Data Types.* This part of the document states the data types that can be mined using a given DM system. The following options have been specified: text, relational, spatial, temporal, image, video, multimedia, object-oriented and hypertext. However to cater for flexibility and extensibility, the DTD allows specifying other data formats apart from the ones listed above. The algorithms in the DAME system support only text data.

```
<data-types>
     <data-types data-type="ascii"/>
</data-types>
```

The service-provider-descriptor DTD is provided attached as Appendix B.

5.3 Client Access Information

This client-access DTD (refer Appendix C) contains information that the client provides to the service provider so that the data can be accessed or transferred (depending on whether the client wishes the mine to be performed on-site or at the location of the service provider). This information is released after the contractual agreement of maintaining the confidentiality of the data is finalized. The information contained in this document is as follows:

Local Data Access . This indicates the access mode when the mining has to be performed locally at the client's site. The service provider dispatching mobile agents to the client's site typically achieves this. In this case, the client needs to specify the location where the mobile agent server is running (in terms of the host and port) and where the data is located (more than one location can be specified for distributed datasets). In some instances the client might require the agent toolkit/environment to be provided by the service provider to enable local deployment. This requirement should also be specified in this document/message. The client also needs to set the security permission in the mobile agent server to allow access by the service provider's agents. The following example specifies that the client does not require the mobile-agent server and gives the host and port where the server is running and the location of the dataset (directory and file name).

```
<local-data-access>
 <ma-server ma-server-requirement="NO">
  <ma-server-location
       host-name="muruga.csse.monash.edu.au"
       port="50">
  </ma-server-location>
  <data-location
       directory="c:\dmsp"
       file-name="dataset1.txt">
  </data-location>
 </ma-server>
</local-data-access>
```

Data Transfer Mode. This indicates how the data is transferred for it to be mined in a remote location. Obviously, this indicates that the client specified the "remote" or "anywhere" option in dm-task-request document. The options include:

- Client transfers data to a specified location of the service provider.
- Service provider transfers data from specified locations of the client. In this case, access to the data must be specified in terms of hosts, ports, user names, passwords and directories where the data is located.

In the following example, the client chooses the option of having the service provider transfer the data from two of its servers. The client specifies the directories and the data files from which the data has to be transferred. It also contains the user names/passwords that the client has created for the service provider to access its file servers.

```
<data-transfer-mode>
 <sp-from-client>
  <ma-server-location
        <host host-name="muruga.csse.monash.edu.au"
              host-port="20"
              host-username="dmsp"
              host-password="dmsp1"
              host-directory="c:\dmsp\dataset1.txt">
        </host>
        <host host-name="krishna.csse.monash.edu.au"
              host-port="20"
              host-username="dmsp"
              host-password="dmsp1"
              host-directory="c:\dmsp\dataset2.txt">
        </host>
 </sp-from-client>
</data-transfer-mode>
```

The client-access DTD is attached as Appendix C.

5.4 Data Mining Task Descriptor

There are on-going initiatives and standardization efforts to specify the parameters that are needed for a data mining process such as Microsoft's OLE DB for data mining [Microsoft 2000] and data mining query languages such as DMQL [Han et al. 1996]. We do not wish to propose a parallel scheme for describing/ specifying a data mining task. Typical parameters that need to be specified include attributes to be considered in the mining process, the condition and decision attributes, data types and distribution of the values of the attributes. Given such a specification, the service provider should feed this in to the mining algorithms/systems that are hosted to process the task.

5.5 Service Provider Access Information

The service-provider-access DTD specifies how the client can access the service provider to deposit the data if they choose to and how to access task status information and the results. The information contained in this document is as follows:

Access. This specifies the host, port, user name, password and directory for the client to submit the data to be mined and is similar to the specification of the *data transfer mode* in the client-access DTD.

Security. This specifies the encryption to be performed on the data prior to transfer.

Task Status. This specifies the URL of the site from where clients can obtain status reports of their tasks. It also specifies the user name and password to access this site as shown in the following example.

```
<task-status>
     <url url-name="http://www.abc.com/client1/ts-
status.html
          url-username="client1"
          url-password="client1-password"/>
</task-status>
```

Results. This specifies the URL of the site from which the results can be downloaded/viewed along with the user name and password to access this site and is identical in syntax and structure to the *task status* specification shown previously. The service-provider-access DTD is attached as Appendix D.

In this section, we have presented XML DTD's to support the interactions between clients and Internet-based data mining service providers through a structured exchange of information. The interactions are driven by the need to first identify service provider(s) that can best meet the preferences and requirements of the clients and then support the Internet-based processing of the data mining task by exchanging information at a finer level of granularity. Thus, the interactions occur at two levels:

1.Identification of service providers, which is supported by allowing the specification of the client's task requirements (in the dm-task-request DTD) and the capabilities/functionality of the service providers (in the service-provider-descriptor DTD).

2.Upon successful identification of a suitable service provider, the interactions are directed towards to exchanging detailed information such as access to computational resources, data, data transfer mechanisms and details of the data mining task itself. This stage is supported by the service-provider-access DTD, the client-access DTD and the data-mining-task-descriptor DTD.

The following section presents a discussion on the aspects and issues of service provider identification.

6 Matching Client Preferences With Service Provider Capabilities

As discussed in the previous section the dm-task-request-descriptor (which captures the task constraints of the client) and the dm-service-provider-descriptor (which allows specification of the functionality of the service provider) facilitates the identification of service providers that satisfy client requirements (as maximally as possible). However, it is important to establish that there is a semantic correspondence between the DTD's that we have proposed, which allows the explicit mapping of the preferences and needs of clients to the functionality provided by different service providers. This is important as it establishes that the design/structure of the DTD's is consistent with their stated objectives of facilitating the identification of appropriate service providers. Thus, we show that for all constraints specified in a task request, there is a corresponding mapping in the description of service providers whereby it can be established whether a service provider can satisfy the required constraint or not. This is significant in the context of the operation of e-service environments/platforms, where a fundamental requirement is the ability to locate service providers, identify one or more service providers that can satisfy the requirements of the clients (to varying degrees) and rank the service providers in the order of how best they satisfy those requirements. Additionally, systems like MARI [Tewari & Maes 2000] may be able to support some forms negotiation. E-services platforms such as e-speak (http://www.e-speak.hp.com/), eb-XML (http://www.ebxml.org) allow definition of messages, attributes and business processes that facilitate the identification of services. By showing that our DTD's facilitate the matching of constraints (of clients) and functionality (service providers), we establish that these specifications can easily be incorporated into an e-services platform.

6.1 XML Path Language (XPath)

We use XML Path Language (XPath) to illustrate the correspondence between the dm-task-request DTD and the service-provider-descriptor DTD. XPath is a World Wide Web Consortium (W3C) (http://www.w3c.org/TR/xpath) standard for addressing parts of an XML document. In addition, XPath also provides an intuitive means for matching and testing for patterns in an XML document. XPath models a document as a tree, where the nodes represent elements, attributes and namespaces. The basic syntactic unit of XPath is the expression, which is evaluated to result in an object of the following possible types: a set of nodes, a string, a Boolean or a number. An XPath expression is specified as an *absolute path* (referred using the syntactic notation '/') from the root of the document tree or as a *relative path* from a specific location (*context node)*. A path expression consists of a sequence of one or more *location steps* separated by '/'. A location step consists of three parts:

1.*Axis* specifies the relationship between the nodes that the XPath expression refers to and the context node. For example, in the expression child::a (which selects the a element children of the context node), child is the axis. The relationships that can be specified in the axis include: child, parent, descendent, following-sibling, preceding-sibling, following, preceding, self, self-or-descendent, self-or-ancestor, attribute (which contains the attributes of the context node) and namespace (which contains the namespace of the context node).

2.*Node Test* specifies the node type of the selected nodes. In the expression child::a, the node type is a.

3.*Predicates* – specify constraints that refine the node selection process. For example, the previous expression child::a, can be further refined by adding a predicate such as child::a[position()=1], which selects the *first* 'a' element child of the context node.

A detailed treatment of the XPath syntactic conventions is outside the scope of this paper and readers are referred to([http://www.w3c.org/TR/xpath).

6.2 XPath for Illustrating Document Correspondence

In this section, we present our scheme for matching client preferences with the functionality supported by service providers. The matching is done as illustrated in figure 4.

Begin
Let C be the set of constraints specified by the client
Let S be the set of services provided by the service provider
Let $xp_i(a)$ be an XPath expression for document i relating to a, where $a \in S$ or $a \in C$
Let $\oplus$ be the operation which determines the basis for comparing $xp_i(a)$ and $xp_i(b)$
For each $c \in C$
Begin
Let $s \in S$
Check $xp_{dm\text{-}task\text{-}request}(c) \oplus xp_{dm\text{-}service\text{-}provider\text{-}descriptor}(s)$ evaluates to true
End for
End

Figure 4 Matching Process

Thus, the matching process involves evaluating an XPath expression for each constraint specified in the dm-task-request document and comparing the result obtained with the result of evaluating a corresponding XPath expression in the service-provider-descriptor document. The comparison operation will vary depending on the semantics of the document and includes equality, inclusion and other logical operators. The XPath expressions that have to be evaluated and "matched" using the above process are as follows:

1.Task Description Matching. These XPath expressions match the description of the task provided by the client with the description of the tasks supported by the service provider. Either (i) or (ii) must evaluate to yield a positive result.

(i) //task-descriptor@task-type $\subseteq$ (//capability-descriptor@task-tpe or //capability-descriptor/misc)
(ii) //task-descriptor/misc $\subseteq$ (//capability-descriptor@task-tpe or //capability-descriptor/misc)

Note that in the above expression (i), xp(c) is //task-descriptor@task-type, $\oplus$ is $\subseteq$ and xp(s) is (//capability-descriptor@task-tpe or //capability-descriptor/misc).

2.**Algorithm Preference Matching.** These expressions match the algorithm preferences of the client with algorithms supported by the DDM systems hosted by the service provider. If this constraint is specified as a "required" constraint then both (i) and (ii) must be satisfied and if it is an "optional" constraint then if (i) is satisfied and (ii) is not, it would still be considered as a positive evaluation.

(i) (//algorithm-preference/algorithm-details@name $\subseteq$ //dm-system/algorithms@algorithm-name)
(ii) (//algorithm-preference/algorithm-details@version $\subseteq$ //dm-system/algorithms@algorithm-version)

Further, the dm-system under consideration for (i) and (ii) must be the same (in instances where the service provider hosts multiple data mining systems and the document includes descriptions for each system). Thus in the service-provider-descriptor,

(//dm-system@name[algorithms@algorithm-name] = //dm-system@name[algorithms@algorithm-version])

3.**Result Preference Matching.** These expressions match the result preferences of the client with output models supported by the DDM systems hosted by the service provider. The result preferences include the format in which the results are wanted and the knowledge integration technique specified. Rule (i) specifies the matching for the format of the result and rule (ii) specifies the matching for the knowledge integration aspect.

(i) (//result-preferences/result-details@result-format $\subseteq$ ((//dm-system/op-model@result-format) or (//dm-system/op-model/misc))

or

(//result-preferences/result-details/misc $\subseteq$ ((//dm-system/op-model@result-format) or (//dm-system/op-model/misc)))

(ii) (//knowledge-integration/KI-Technique $\subseteq$ //knowledge-integration/KI-Technique)

As before, the dm-system under consideration for (i) and (ii) must be the same (in instances where the service provider hosts multiple data mining systems and the document includes descriptions for each system). Further the dm-system under consideration for the algorithm and result preferences must be the same. Thus in the service-provider-descriptor,

(//dm-system@name[op-model@result-format] = //dm-system@name[knowledge-integration/KI-Technique])

4.**Data Preference Matching.** This is a required constraint (i.e. there are no optional preferences allowed for this constraint).

(i) (//data/data-types@data-type ⊆ (//dm-system/data-types/data-type or //dm-system/data-types/data-type/misc-data-types))

The dm-system under consideration for the algorithm and result preferences must be the same as the dm-system that matches the data type requirements of the task. This ensures that there is no scope for inconsistencies such as one dm-system meeting the data-type constraint and another satisfying the other constraints.

5.**Comparative Result Preference Matching.** These expressions evaluate whether or not a service provider can provide comparative requests required by the client. The comparative results are specified by the client as a set of algorithms and a result-format.

(i) (//comparative-results/algorithm-details@name ⊆ //dm-system/algorithms@algorithm-name)
and
(//comparative-results/algorithm-details@version ⊆ //dm-system/algorithms@algorithm-version)
and
(//comparative-results/result-details@result-format ⊆ ((//dm-system/op-model@result-format) or (//dm-system/op-model/misc)))
or
(//result-preferences/result-details/misc ⊆ ((//dm-system/op-model@result-format) or (//dm-system/op-model/misc)))

Unlike the algorithm, result and data-type preferences, the comparative-result constraint need not be satisfied by one dm-system. A service provider who is able to provide the comparison required by using algorithms supported by different systems (which are hosted) can therefore stand to gain by being able to satisfy such a constraint, which may not be satisfied by any one particular system.

6.**Location Preference Matching.** These expressions evaluate whether or not a service provider can provide meet the mining location constraints of the client. Either (i) or (ii) must evaluate to a positive result.

(i) (//location/location-preference = local) and (//dm-system/arch-model/ = mobile-agent or //dm-system/arch-model = hybrid)

(ii) (//location/location-preference = remote) and (//dm-system/arch-model/ = client-server or //dm-system/arch-model = hybrid)

This constraint too must be satisfied by the same dm-system that satisfies the algorithm, result and data-type preferences.

7.**Response Time Preference Matching.** This expression evaluates whether the response time specified by the service provider is within the range specified by the client or not. Rule (i) and (ii) must be satisfied.

(i) (//response-time/time/range-constraint@min >= //task-metrics/response-time/time/range/constraint@min)

(ii) (//response-time/time/range-constraint@max <= //task-metrics/response-time/time/range/constraint@max)

8.**Cost Preference Matching.** This expression evaluates whether the cost specified by the service provider is within the range specified by the client or not. Rule (i) and (ii) must be satisfied.

(i) (//cost/range-constraint@min >= //task-metrics/cost/amt)

(ii) (//cost/range-constraint@max <= //task-metrics/cost/amt)

9. **Miscellaneous Preferences Matching.** This expression evaluates whether the miscellaneous preferences specified by the client are satisfied by the dm-system of the service provider that meets the algorithm, result, data and location preferences. Rule (i) or (ii) must evaluate to a positive result.

(i) (//misc-services/services ⊆ //dm-system/features/services)

(ii) (//misc-services/misc ⊆ //dm-system/features/services)

The preceding rules specify the XPath expressions for mapping the constraints specified by the client with the services provided by the data mining service providers. As it can be seen, for every constraint specified in the client's dm-task-request document, there is a corresponding XPath expression which points to the appropriate sections of the service-provider-descriptor document. These matching rules form the basis for determining the extent to which a service provider can meet the requirements specified by a client with respect to a data mining task. Thus an e-service environment can use these matching rules to identify and rank service providers for a given task request. We briefly discuss the concepts of matching and ranking in e-service environments in section 6.3.

6.3 Matching and Ranking Service Providers in E-Services Environments

The ability to find service providers that best satisfy the constraints of clients and to negotiate on the terms of an e-service transaction are important components in the multiple service provider model (and in any environment that supports and facilitates e-services). It is often possible that a single service provider will not be able to meet all the requirements specified by the client and that there will be several service providers who may be able to satisfy different combination of preferences. The current focus of the research in this area is directed towards service discovery and matching is performed by simple attribute/key word matches (e.g. in e-speak) (http://www.e-speak.hp.com/). Further this matching occurs at a high level of abstraction – for instance a service provider who registers as a "data mining service provider" will be presented to clients who request for "data mining services". Multi Attribute Resource Intermediary (MARI) [Tewari & Maes 2000] is a research prototype that is being developed at MIT Media Labs to support negotiation in e-service environments. This system requires clients to constrain their preferences within a scale, whereby the less the client is willing to compromise on an attribute, the more constrained it is. Each attribute has a utility function associated with it, which forms the basis for ranking of service providers. There can be several techniques/measures that can be used to rank service providers. We are currently investigating this emerging issue in e-services research. Automated negotiation in e-services is also an important issue in this context, however this is not the current focus of our work.

7 Comparison with Related Work

We now present a brief overview of related research. To the best of our knowledge there is no other work that focuses on describing the interactions and exchange of information to support the Internet delivery of data mining services in a multiple service provider domain. However, there are two data mining/business intelligence related XML initiatives – namely Predictive Model Markup Language (PMML) [Grossman etal. 1999] and OLE DB for Data Mining [Microsoft 2000]. PMML is an XML-based approach for describing the predictive models that are generated as the output of data mining processes. PMML is primarily used for knowledge integration of results obtained by mining distributed data sets. Microsoft's OLE DB for data mining is a description of a data mining task in terms of the data sets that are being mined and allows specification of the attributes to be mined, the attributes to be predicted and the type and format of the attributes. It incorporates a version of PMML and is primarily intended to allow "plug and play" of data mining algorithms. As discussed in the previous section, this specification (as it evolves into a standard) can easily be used in conjunction with our work, which focuses on specifying user preferences for data mining tasks that are processed by application service providers.

In section 2, we outlined emerging standards and systems that support e-services. We now discuss how our model can be used in conjunction with those systems/standards.

- *E-Speak*. E-Speak allows description of services through a user-defined "vocabulary". The service provider descriptor presented in this paper is a possible vocabulary for data mining service providers using e-speak. However, whether e-speak allows for the type of interaction required by data mining tasks (given the implicitly confidential nature of the process) needs to be examined.
- *UDDI*. The XML DTD's presented in this paper provide the basis for data mining service providers using UDDI to describe themselves.
- *eb-XML*. This can be used to implement the multiple service provider model for hosting data mining services. The information exchange process that we have developed for data mining services and the XML messages that we have specified can be easily incorporated for use in an eb-XML environment.

While the XML DTD's presented in this paper provide the basis for information exchanges in such a virtual market-place of data mining services, they can also be used in the current model of operation to automate the following stages of the interaction:

- The DTD's allow specification of data transfer preferences. For systems such as digiMine, which allow different options for data transfer, the XML DTD's can facilitate capturing client preferences.
- The systems require information about access to client computational resources. The client-access DTD can be incorporated in this stage of the interaction.
- Systems such as WhiteCross and iFusion, which support data integration from multiple data sources can capture data location details using the client-access DTD.
- The most common basis for delivering results is through a secure web interface. The service-provider-access DTD allows defining this information.
- Further, preferences such as the format for delivering results and the type of reports can be specified using the dm-task-request DTD.

In conclusion, it can be seen that the DTD's allow specification of more detailed information about task preferences, service provider capabilities and access information than is required by the current model of interaction between clients and data mining service providers. The DTD's have been developed with the flexibility to support different parts of the interactions in the single-service provider model. Further, while the dm-task-request DTD and the service-provider-descriptor DTD are primarily useful in a multiple service provider, the client-access DTD and the service-provider DTD are suitable for use in either model of operation. The distinct contribution of our research is the identification of the processes required to support the delivery of data mining services and specify the information content that needs to be exchanged between organisations requiring data mining services and data mining service providers.

8 Conclusions and Future Work

The emergence of Internet-based data mining service providers is proving to be a viable means for satisfying the business intelligence needs of knowledge-centric organizations. In this paper we have presented a multiple service provider model as an alternative operational model to the currently predominant approach among data mining service providers. We have shown the advantages of the new model with respect to providing clients with a wider variety of options, choice of service providers and the benefits of a more competitive marketplace. We have presented the structure and contents of XML documents/messages to support the interactions in the multiple service provider model of operation for data mining service providers. The XML DTD's support the different stages of interaction including: specification of the task preferences of clients, the specification of the functionality of data mining service providers and the exchange of information to access data and computational resources to perform the task. Thus, they facilitate identification of suitable service providers (that can satisfy the preferences and requirements of the client) and provide support for performing a data mining task through the Internet. The DTD's have been designed to cater for the incorporation of organizational constraints and requirements that arise when data mining services are outsourced using the Internet. We have presented a matching algorithm that uses XPath expressions to map client preferences to the functionality supported by service providers. We have also presented a comprehensive survey of related work in this domain. In summary, we have presented a novel model for delivering Internet-based data mining services, defined the interaction protocols for the model and developed an XML based information exchange mechanism to support this model. The current focus of our work is the ranking schemes for service providers. We believe that ranking, along with negotiation techniques for e-services are important research questions in this area. In summary, we see that the potential benefits of the multiple service provider model and the emergence of platforms and standards to support e-services are significant factors that impact positively on the realization of our work. This research takes the creation of a virtual market place for data mining services one step forward.

REFERENCES

Clark-Dickson,P., (1999), "Flag-fall for Application Rental", Systems, (August), pp.23-31.

Goldfarb,C,F., and Prescod,P., (1998), "The XML Handbook", Prentice-Hall PTR, New Jersey, USA.

Grossman,R,L., Bailey,S., Ramu,A., Malhi,B., Hallstrom,P., Pulleyn,I., and Oin,X., (1999), "The Management and Mining of Multiple Predictive Models Using the Predictive Modelling Markup Language (PMML)", *Information and Software Technology*, Volume 41, pp. 589-595.

Han,J., Fu,Y., Wang,W., Koperski,K., and Zaiane,O., (1996), "DMQL: A Data Mining Query Language for Relational Databases", Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, June.

Kargupta,H., Park,B., Johnson,E., Riva Sanseverino,E., Di Silvestre,L., and Hershberger,D., (1998),"*Collective Data Mining From Distributed Vertically Partitioned Feature Space*", in KDD-98 Workshop on Distributed Data Mining, New York, USA, AAAI Press.

Krishnaswamy,S., Zaslavsky,A., and Loke,S,W., (2000a), "An Architecture to Support Distributed Data Mining Services in E-Commerce Environments", Proceedings of the Second International Workshop on Advanced Issues in E-Commerce and Web-Based Information Systems, San Jose, Californinia, June 8-9, pp.238-246.

Krishnaswamy,S., Loke,S,W., and Zaslavsky,A., (2000b), *"Cost Models for Heterogeneous Distributed Data Mining"*, Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering, Chicago, Illinois, July 6-8, pp.31-38.

Krishnaswamy,S., Zaslavsky,A., and Loke,S,W., (2001), "Federated Data Mining Services and a Supporting XML Markup Language", Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34), Hawaii, USA, January, In the "e-Services: Models and Methods for Design, Implementation and Delivery" mini-track of the "Decision Technologies for Management" track.

Microsoft OLE DB for Data Mining, URL: http://www.microsoft.com/data/oledb/dm.htm, March, 2000.

Morency,J., (1999), "Application Service Providers and E-Business", Network World Fusion Newsletter, URL: http://www.nwfusion.com/newsletters/nsm/0705nm.html

Ramu,A,T., (1998), "Incorporating Transportable Software Agents into a Wide Area High Performance Distributed Data Mining Systems", Masters Thesis, University of Illinois, Chicago, USA.

Sarawagi,S., and Nagaralu,S,H., (2000), "Data Mining Models as Services on the Internet", SIGKDD Explorations, vol. 2, issue. 1, http://www.acm.org/sigkdd/explorations/ , accessed 01 April, 2001.

Stolfo,S,J., Prodromidis,A,L., Tselepis, L., Lee,W., Fan,D., and Chan,P,K., (1997), "JAM: Java Agents for Meta-Learning over Distributed Databases", in *Proceedings of the Third International Conference on Data Mining and Knowledge Discovery (KDD-97)*, Newport Beach, California, (eds) David Heckerman, Heikki Mannila, Daryl Pregibon, and Ramasamy Uthurusamy, AAAI Press, pp. 74-81.

Tewari,G., and Maes,P., (2000), "A Generalized Platform for the Specification, Valuation, and Brokering of Heterogeneous Resources in Electronic Markets", in *Lecture Notes in Artificial Intelligence* (LNAI) 2033, Springer-Verlag, pp.7-24.

Universal Description, Discovery and Integration (UDDI) - URL:http://www.uddi.org/

Note: All DTD's have been parsed for validity and well-formedness.

APPENDIX A (DM-TASK-REQUEST DTD)

```
<?xml version="1.0" encoding="UTF-8"?
<!-- edited with XML Spy v3.0.7 NT (http://www.xmlspy.com) by Shonali Krishnaswamy (Monash
University) -->
<!--                    Data Mining Task Request                    -->
<!ELEMENT dm-task-request (task-descriptor+, algorithm-preference*, result-preference*, comparative-
results?, data, location, response-time, cost)>
<!--                    Task Descriptor                    -->
<!ELEMENT task-descriptor (misc)>
<!ATTLIST task-descriptor
     task-type (fraud-detection | market-basket-analysis | customer-profiling | dependency-analysis |
clustering | segmentation | predictive-modelling) #IMPLIED
> <!ELEMENT misc (#PCDATA)>
<!--                    Algorithm Preferences                    -->
<!ELEMENT algorithm-preference (algorithm-details, pref-constraint?)> <!ELEMENT algorithm-details
EMPTY>
<!ATTLIST algorithm-details name CDATA #IMPLIED version CDATA #IMPLIED >
<!ELEMENT pref-constraint EMPTY>
<!ATTLIST pref-constraint choice (preferred | required | unconstrained) #REQUIRED
>
<!--                    Result Preferences                    -->
<!ELEMENT result-preference (result-details+, knowledge-integration, pref-constraint?)>
<!ELEMENT result-details (misc?)>
<!ATTLIST result-details
     result-format (rules | visual | decision-trees | association-rules | classification-rules | decision-rules |
predictive-models) #IMPLIED
>
<!ELEMENT knowledge-integration (KI-technique)*>
<!ELEMENT KI-technique EMPTY>
<!ATTLIST KI-technique type (PMML | BODHI | meta-learning | None | Unspecified) #REQUIRED >
<!--                    Comparative Results                    -->
<!ELEMENT comparative-results (algorithm-details, algorithm-details+, result-details?)>
<!--                    Dataset Meta Information                    -->
<!ELEMENT data (data-types, data-format?, size)>
<!ELEMENT data-types (misc?)>
<!ATTLIST data-types
     data-type (spatial | temporal | multimedia | image | video | text | relational | ascii | hypertext)
#REQUIRED
```

```
>
<!ELEMENT data-format (#PCDATA)>
<!ELEMENT size (#PCDATA)>
<!ATTLIST size
    unit (KB | MB | GB) #REQUIRED
    measure CDATA #REQUIRED
>
<!--                    Location Preferences                    -->
<!ELEMENT location (comp-resources)*>
<!ATTLIST location
    location-preference (local | remote | anywhere) #REQUIRED>
<!ELEMENT comp-resources (server+)>
<!ELEMENT server (type)>
<!ATTLIST server
    name CDATA #IMPLIED
    dedicated (YES | NO) #REQUIRED
>
<!ELEMENT type (stand-alone | parallel | cluster)>
<!ELEMENT stand-alone (configuration)>
<!ELEMENT parallel (configuration)>
<!ELEMENT cluster (config-total, configuration+)>
<!ATTLIST cluster
    no-of-nodes CDATA #REQUIRED
>
<!ELEMENT config-total EMPTY>
<!ATTLIST config-total
    CPU CDATA #REQUIRED
    memory CDATA #REQUIRED
>
<!ELEMENT configuration (config-total)>
<!ATTLIST configuration
    OS (Windows-NT | Solaris | Linux | Unix | Mac) #REQUIRED
>
<!--                    Response Time Preferences                    -->
<!ELEMENT response-time (time, pref-constraint?)>
<!ELEMENT time (range-constraint)>
<!ATTLIST time
    measure (hours | days) #REQUIRED
>
<!ELEMENT range-constraint EMPTY>
<!ATTLIST range-constraint
    min CDATA #IMPLIED
    max CDATA #IMPLIED
>
<!--                    Cost Preferences                    -->
<!ELEMENT cost (range-constraint, pref-constraint?)>
<!ATTLIST cost
    measure CDATA #REQUIRED
>
<!--                    Misc. Preferences                    -->
<!ELEMENT misc-services (services | misc)*>
<!ELEMENT services EMPTY>
<!ATTLIST services
    type (mobile-users | parallel-algorithms | pre-processing) #REQUIRED
>
```

APPENDIX B (SERVICE-PROVIDER-DESCRIPTOR DTD)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0.7 NT (http://www.xmlspy.com) by Shonali Krishnaswamy (Monash University)  -->
<!--                                    Service Provider Descriptor                -->
<!ELEMENT spd (sp-details, comp-resources, dm-system+, task-specific-details)>
<!ELEMENT sp-details (capability-descriptor)*>
<!ATTLIST sp-details
      name CDATA #REQUIRED
      bus-reg CDATA #REQUIRED
      address CDATA #REQUIRED
      email CDATA #REQUIRED
      phone CDATA #REQUIRED
      fax CDATA #REQUIRED
      url CDATA #REQUIRED
      date CDATA #REQUIRED
>
<!ELEMENT capability-descriptor (misc)>
<!ATTLIST capability-descriptor
      task-type (fraud-detection | market-basket-analysis | customer-profiling | dependency-analysis |
clustering | segmentation | predictive-modelling) #IMPLIED >
<!ELEMENT misc (#PCDATA)>
<!--                                    Computational Resources                -->
<!ELEMENT comp-resources (server+)>
<!ELEMENT server (type)>
<!ATTLIST server
    name CDATA #IMPLIED
    dedicated (YES | NO) #REQUIRED
>
<!ELEMENT type (stand-alone | parallel | cluster)>
<!ELEMENT stand-alone (configuration)>
<!ELEMENT parallel (configuration)>
<!ELEMENT cluster (config-total, configuration+)>
<!ATTLIST cluster
      no-of-nodes CDATA #REQUIRED
>
<!ELEMENT config-total EMPTY>
<!ATTLIST config-total
    CPU CDATA #REQUIRED
    memory CDATA #REQUIRED
>
<!ELEMENT configuration (config-total)>
<!ATTLIST configuration
      OS (Windows-NT | Solaris | Linux | Unix | Mac) #REQUIRED
>
<!--                                    Distributed Data Mining System                -->
<!ELEMENT dm-system (sys-info, arch-model, data-types, features, algorithms)>
<!--                                    System Information                -->
<!ELEMENT sys-info EMPTY>
<!ATTLIST sys-info
      name ID #REQUIRED
      version CDATA #REQUIRED
      organisation CDATA #REQUIRED
      developer CDATA #REQUIRED
      date CDATA #REQUIRED
>
```

```
<!--                    Architectural Model                    -->
<!ELEMENT arch-model (ma-toolkit)*>
<!ATTLIST arch-model
    architecture (mobile-agent | client-server | hybrid) #REQUIRED
>
<!ELEMENT ma-toolkit (misc)*>
<!ATTLIST ma-toolkit
    ma-toolkit-name (aglet | gypsy | voyager | grasshopper | jseal | mole | moa | dAgents | concordia |
tacoma | telescript | oddysey) #IMPLIED
>
<!--                    Data Types Supported                    -->
<!ELEMENT data-types (data-type)+>
<!ELEMENT data-type (misc-datatypes)*>
<!ATTLIST data-type
    type (spatial | temporal | multimedia | image | video | text | relational | ascii | hypertext) #REQUIRED
>
<!ELEMENT misc-datatypes ANY>
<!--                    Specialisations and Features                    -->
<!ELEMENT features ((services)*, (knowledge-integration)*, (misc-features)*)>
<!ELEMENT services EMPTY>
<!ATTLIST services
    type (mobile-users | optimisation | cost-efficiency | parallel-algorithms | pre-processing | visualisation)
#REQUIRED
>
<!ELEMENT knowledge-integration (KI-technique)*>
<!ELEMENT KI-technique EMPTY>
<!ATTLIST KI-technique
    type (PMML | BODHI | meta-learning | None | Unspecified) #REQUIRED
>
<!ELEMENT misc-features ANY>
<!--                    Algorithms                    -->
<!ELEMENT algorithms (algorithm+)>
<!ELEMENT algorithm ((ip-file), (parameter)*, (op-model))>
<!ATTLIST algorithm
    algorithm-name CDATA #REQUIRED
    algorithm-version CDATA #IMPLIED
    algorithm-author CDATA #IMPLIED
    algorithm-date CDATA #IMPLIED
    algorithm-misc CDATA #IMPLIED
>
<!ELEMENT ip-file (data-type, structure?)>
<!ATTLIST ip-file
    file_extension CDATA #REQUIRED >
<!ELEMENT structure (text-file-descriptor)>
<!ELEMENT text-file-descriptor EMPTY>
<!ATTLIST text-file-descriptor
    header CDATA #REQUIRED
    comments CDATA #REQUIRED
    attribute-descriptor CDATA #REQUIRED
    attribute-types CDATA #REQUIRED
    data-descriptor CDATA #REQUIRED
    field-delimiter CDATA #REQUIRED
>
<!ELEMENT parameter EMPTY>
<!ATTLIST parameter
    parameter-name CDATA #IMPLIED
```

```
    description CDATA #IMPLIED
>
<!ELEMENT op-model (misc?)>
<!ATTLIST op-model
    result-format (rules | visual | decision-trees | association-rules | classification-rules | decision-rules |
predictive-models) #IMPLIED
>
<!--                    Task Specific Information                    -->
<!ELEMENT task-specific-details (task-metrics+, cost+)>
<!ELEMENT task-metrics (response-time, arch-model)>
<!ELEMENT response-time (time, liability+)>
<!ELEMENT time (range-constraint)>
<!ATTLIST time
    measure (hours | days) #REQUIRED
>
<!ELEMENT range-constraint EMPTY>
<!ATTLIST range-constraint
    min CDATA #IMPLIED
    max CDATA #IMPLIED >
<!ELEMENT liability EMPTY>
<!ATTLIST liability
    constraint CDATA #REQUIRED
    cost-reduction CDATA #REQUIRED >
<!ELEMENT cost (arch-model)>
<!ATTLIST cost
    measure CDATA #REQUIRED
>
```

APPENDIX C (CLIENT-ACCESS DTD)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0.7 NT (http://www.xmlspy.com) by Shonali Krishnaswamy (Monash
University)  -->
<!--                    Client Access                    -->
<!ELEMENT client-access (data-transfer-mode | local-data-access)>
<!ELEMENT data-transfer-mode (client-to-sp | sp-from-client)>
<!--                    Client- to -Provider                    -->
<!ELEMENT client-to-sp (#PCDATA)>
<!--                    Provider-from-Client                    -->
<!ELEMENT sp-from-client (host)+>
<!ELEMENT host EMPTY>
<!ATTLIST host
    host-name CDATA #REQUIRED
    host-port CDATA #REQUIRED
    host-username CDATA #REQUIRED
    host-password CDATA #REQUIRED
    host-directory CDATA #REQUIRED
>
<!--                    Local Data Access                    -->
<!ELEMENT local-data-access (ma-server)+>
<!--                    Mobile Agent Server                    -->
<!ELEMENT ma-server (ma-server-location, data-location+)>
<!ATTLIST ma-server
    ma-server-requirement (YES | NO) #REQUIRED
>
<!ELEMENT ma-server-location EMPTY>
<!ATTLIST ma-server-location
```

```
    host-name CDATA #REQUIRED
    port CDATA #REQUIRED
>
<!--                    Data Locations                    -->
<!ELEMENT data-location EMPTY>
<!ATTLIST data-location
    directory CDATA #REQUIRED
    file-name CDATA #REQUIRED
>
```

APPENDIX D (SERVICE-PROVIDER-ACCESS DTD)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.0.7 NT (http://www.xmlspy.com) by Shonali Krishnaswamy (Monash University)  -->
<!--                    Service Provider Access                    -->
<!ELEMENT sp-access (data-submission+, encryption, task-status, results)>
<!--                    Data Submission                    -->
<!ELEMENT data-submission (host)+>
<!ELEMENT host EMPTY>
<!ATTLIST host
    host-name CDATA #REQUIRED
    host-port CDATA #REQUIRED
    host-username CDATA #REQUIRED
    host-password CDATA #REQUIRED<
    host-directory CDATA #REQUIRED
>
<!--                    Data Encryption                    -->
<!ELEMENT encryption (#PCDATA)>
<!--                    Task StatusAcess                    -->
<!ELEMENT task-status (url)>
<!ELEMENT url EMPTY>
<!ATTLIST url
    url-name CDATA #REQUIRED
    url-username CDATA #REQUIRED
    url-password CDATA #REQUIRED
>
<!--                    Results Access                    -->
<!ELEMENT results (url)>
```