Management of Data Mining Model Lifecycle to Support Intelligent Business Services

Ismail Ari, Jun Li, Jhilmil Jain, Alex Kozlov
HP Laboratories, Palo Alto
HPL-2008-37
April 24, 2008*

data mining models, model lifecycle, SOA, BSM, BI, BPM

Information Technology (IT) management is going through its third phase of evolution. In the first phase, it was about managing silos of devices (servers, storage) and applications. In the next phase, best practices frameworks such as IT Infrastructure Library (ITIL) prescribed a service-oriented approach to IT management. Now, we are seeing a shift of focus from the bottom-up IT-driven approaches to top-down business-driven strategies. This final phase also called Business Service Management (BSM) will enforce business accountability on people as well as IT by automating and managing business processes and leveraging IT Service Management (ITSM) practices.

While the IT industry is moving forward with these modern concepts, they have left behind critical processes and soft IT assets unmanaged, especially at the intersection of business processes with Business Intelligence (BI). In this paper, we describe the design and implementation of a system that manages data mining model assets of an organization that can support business processes in making real-time decisions and forecasts. We address technical challenges related to model aging, scalability, model lifecycle, and metadata management. We also identify a number of grand challenges related to organizational ethnography (such as analysis of semantic gaps between business analysts and statisticians), visualization support, and longitudinal studies needed to identify evaluation metrics.

Internal Accession Date Only Approved for External Publication

Submitted to ACM Computer Human Interaction for Management of IT (CHIMIT'08), Nov, 2008

Management of Data Mining Model Lifecycle to Support Intelligent Business Services

Ismail Ari, Jun Li, Jhilmil Jain, Alex Kozlov [1]
Hewlett-Packard Laboratories
1501 Page Mill Rd,
Palo Alto, CA, USA, 94541
{ismail.ari,jun.li,jhilmil.jain}@hp.com, alex.kozlov@turn.com

ABSTRACT

Information Technology (IT) management is going through its third phase of evolution. In the first phase, it was about managing silos of devices (servers, storage) and applications. In the next phase, best practices frameworks such as IT Infrastructure Library (ITIL) prescribed a service-oriented approach to IT management. Now, we are seeing a shift of focus from the bottom-up IT-driven approaches to top-down business-driven strategies. This final phase also called Business Service Management (BSM) will enforce business accountability on people as well as IT by automating and managing business processes and leveraging IT Service Management (ITSM) practices.

While the IT industry is moving forward with these modern concepts, they have left behind critical processes and soft IT assets unmanaged, especially at the intersection of business processes with Business Intelligence (BI). In this paper, we describe the design and implementation of a system that manages data mining model assets of an organization that can support business processes in making real-time decisions and forecasts. We address technical challenges related to model aging, scalability, model lifecycle, and metadata management. We also identify a number of grand challenges related to organizational ethnography (such as analysis of semantic gaps between business analysts and statisticians), visualization support, and longitudinal studies needed to identify evaluation metrics.

Categories and Subject Descriptors

H.1.2 **[Models and Principles]** User/Machine Systems- *Human factors* H.2.8 [**Database Management**] Database Applications- *Data mining,* H.5.3 **[Information Interfaces and Presentation]** Group and Organization Interfaces- *Asynchronous Interaction.*

General Terms

Management, Design, Human Factors.

Keywords

Data mining models, model lifecycle, SOA, BSM, BI, BPM.

CHIMIT'08, Nov 14–15, 2008, San Diego, CA, USA.

1. INTRODUCTION

The synchronization of business and IT is happening at a rapid pace due to the proliferation of web services, Service-Oriented Architecture (SOA) principles [8], ITSM best practices such as ITILv3 [13], and Business Process Management (BPM) systems. The benefits observed in several industries such as increased business agility is in turn causing a push for further automation of processes, integration of disparate systems, and real-time use of Business Intelligence (BI) gained from data stored in data warehouses and continuously streamed from operational systems.

Today, enterprises collect huge amounts of data on their customers, partners, and employees as well as their operational and financial systems. They hire statisticians (either locally or outsourced) to create data mining models that analyze collected data to help business analysts create reports and identify trends, so that they can optimize their channel operations, improve service quality, track customer profiles; ultimately reducing costs and increasing revenue. Unfortunately, steps that start with data acquisition and lead to business outcomes are not *operationalized* today. The processes are not modeled and automated and the insights gained are buried in silos of systems and applications. As a result, the business-level responses are almost always after the fact (*e.g.* a critical process has failed or a sales opportunity has been lost). BSM practices plan to enforce real-time accountability on people as well as IT to change this picture by defining business metrics in terms of Key Performance Indicators (KPIs) and Service Level Agreements (SLAs). BSM depends on emerging integration technologies, Business Activity Monitoring (BAM) tools, and concurrent adoption of ITSM practices to be successful.

However, this ideal picture is not easy to achieve. For example, data mining models that help deliver BI require deep understanding of intricate statistics and algorithms as well as in-depth domain knowledge. It takes complex and sometimes composite models to predict the next purchase of customers who buy diapers, or browse digital cameras or finance offers. Furthermore, models have a high up-front development cost both time-wise and monetary, which discourages frontline businesses from applying even matured mining techniques. However, as business processes and BI systems are drawn closer together due to increased system integration, enterprises are beginning to realize the value of managing "soft" IT assets including data mining models. The major contribution of this paper is describing design and implementation of a model management system that

[1] Alex Kozlov was previously at HP. He now works for Turn.com; Contact him at alex.kozlov@turn.com

increases model utility and value across the enterprise closing the remaining gaps and helping deliver "BI to the masses".

However, there are serious technical as well as ethnographic challenges regarding building, updating and sharing complex data mining models across the enterprise: (1) All models inevitably age over time and their predictive performance changes as the products, customers, and business environments change (especially in the Electronics domain that offers different generations rapidly). For example, the concept of a "high-end" digital camera shifts from 3 to 5 to 10 Megapixels within a few years. One has to continuously feed new data into a model, monitor its performance and constantly tune its parameters to retain good prediction results. (2) Mining algorithms can now be easily obtained from off-the-shelf BI suites [19][20][23] and it is common to find practical BI deployments that incorporate hundreds of data mining models in banks, retailers, insurance companies, telcos and even casinos [18][24]. However, there is lack of support for effectively managing and utilizing these large collections. Manual management is impractical, faulty, and unresponsive to quick change. (3) Then, there is the issue of timely-communication between the business analysts and the statisticians or model developers. Large-scale and dynamic nature of the models and the organizations makes it impractical to timely inform multiple parties about model-related events. Due to the lack of continuous monitoring and alerts, models that deteriorate and don't catch emerging patterns or forecast accurately are detected and updated only after serious business consequences. Today, stakeholders email each other and developers overwrite model parameters causing the valuable interactions to be lost or buried in emails and databases. (4) Finally, there is a semantic gap between statisticians who talk about regressions, accuracy, and ROC vs. business analysts who talk about customer retention strategies, addressable markets, *etc*.

Overall, current data mining systems have not effectively addressed the challenges mentioned above. Specifically, we observed these problems during the development of HP Labs' Retail Store Assistant (RSA) platform [1], which aims to provide real-time personalized offers to customers through multiple retail channels including kiosks, web, and mobile devices using data mining models as well as business rules. The personalized coupons are presented to users by combining information from a user's current shopping list, user's past profile, business rules set in the frontline campaign process, and BI gained from this and other users' overall purchase behaviors (*e.g.* via clustering). For example, the system uses data mining models to predict customer's preferences and best matching products. Each component is published as a web service and the overall design uses a SOA.

Figure 1 summarizes the business and BI ecosystem described above and will guide us through the rest of our paper. It shows statisticians, business analysts and customers as the entities or stakeholders involved in the BSM process, *i.e.* the unified and "intelligent" (referring to BI support) business service management. Statisticians or model developers, who can belong to different organizations, create and train data mining models over enterprise data. They are tasked with maintaining the performance of model predictions. Business people are responsible for successful operation of frontline systems as well as customer satisfaction. They analyze reports and create written

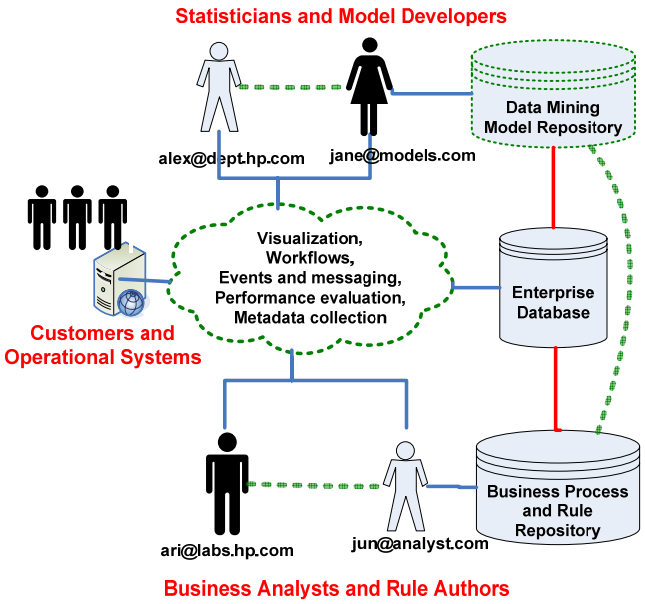

Figure 1: Enterprise entities involved in and affected from management of data mining models.

rules or policies (today mostly manually) to drive the operational systems. In the future, BSM practices will require these people to get involved in designing and managing automating business processes and authoring actionable business rules. Our contributions are highlighted with dashed lines. Through the design of data mining model management system, we facilitate collaboration among entities in the same or different groups and reduce delays in maintaining models and integrating gained knowledge into operational system. We use workflows, messaging systems, automated performance evaluation, rich metadata collection, and (currently basic) visualization to put it all together. We also track inter-model and model-rule or model-process dependencies and allow cross-pollination of gained knowledge to enable continuous process improvement.

2. Background on Concepts Used

This section describes novel concepts that emerged in the last decade, which affected the design of our solution. It gives an overview of SOA, BSM, BPM, and core services including BI.

2.1 SOA and BSM

We have already briefly mentioned the SOA-based retail platform as our driving business application. Today, the IT industry is considering two delivery strategies for SOA: top-down and bottom-up. The majority of organizations that are adding web services to their existing application environments are practically adopting the bottom-up approach. However, according to Thomas Erl [8] the so called "bottom-up strategy" is in reality a misnomer and not a SOA strategy, since the existing architectures remain unchanged and SOA principles are ignored. In the top-down strategy, the business model is analyzed and the business processes are also made service-oriented. Since both the processes and the people are involved, top-down SOA delivery is the right approach and constitutes a big part of the BSM practice. During the design of our service-oriented retail platform we met with retailers to understand their business models and identify critical

processes to be able to help them as well as their customer, thus following a top-down strategy.

However, we learned that even simple business processes can become quite complex due to the involvement of multiple people/roles, organizations, and IT systems across multiple channels. When BSM practices will add the pressure of handling events and exceptions in real-time and require compliance with business metrics and other regulations, processes will have to turn automated and use BI tools to make critical decisions. In addition, when IT and business transformations are completed business processes will be described as a combination of artifacts such as workflows, models, business rules and a set of web services (wrapping new functional units or legacy systems). Each of these components will iteratively depend on data, views, reports and data mining models stored in data warehouses and other BI tools.

2.2 Business Process Management (BPM)

Today, a BPM system lies at the core of many SOA offerings providing the language and tools to express business logic in an executable form (*e.g.* BPEL [29]). BPM tools also enable message routing, message mapping, and choreography among published services and applications. BPM can help process instances acquire analytical support from the BI services or contact rule engines to decide which action to take based on the business conditions.

Figure 2 illustrates a conceptual view of a SOA environment where all the well-known and emerging engines are presented as core services. On the top there is a list of applications such as Email, Customer Relationship Management (CRM), Enterprise Resource Planning (ERP) and Supply Chain Management (SCM). The applications are used by the Human Resources (HR), finance, sales, marketing, and other strategic departments of an enterprise to operate business on a daily basis. Next, the figure shows the operational systems, *aka* channels, including the web, mobile, email, application servers and the sector-specific kiosks including the Point of Sale (POS) and Automatic Teller Machines (ATM). These channels are the sources of data streams that carry the clicks, transactions, sensors readings, *etc*. associated with simple and complex (correlated) business events. All entities in this picture are logical; each service could run in a standalone physical server or get virtualized and distributed over clusters of servers and storage (as shown with the dashed boxes in Fig.2).

The Enterprise Service Bus (ESB) is similar to a computer bus or a network router, fixing problems associated with point-to-point service connectivity. It has highly-optimized XML/SOAP message queues at its core. Usually, a BPM system is layered on top of the message queues to handle the service orchestration and schema mapping. We show BPM as an external entity to ESB to denote potential access to its services via other data and protocol adapters (HTTP, SMTP, TCP/IP, *etc.*). A UDDI registry service can also be a part of or complementary to the ESB, providing service lookup and access to web service definitions (WSDL).

2.3 Core SOA Services and BI

We've seen proliferation of specially-purposed computing *engines* and *appliances* over the last decades, which include the database engine, ETL engine, analytics engine, pub/sub engine, rule engine, search engine, and more recently stream and event processing engines. We believe that these engines will be presented as independently-scalable core services and collaborate with other services in the SOA environment. Decoupling core

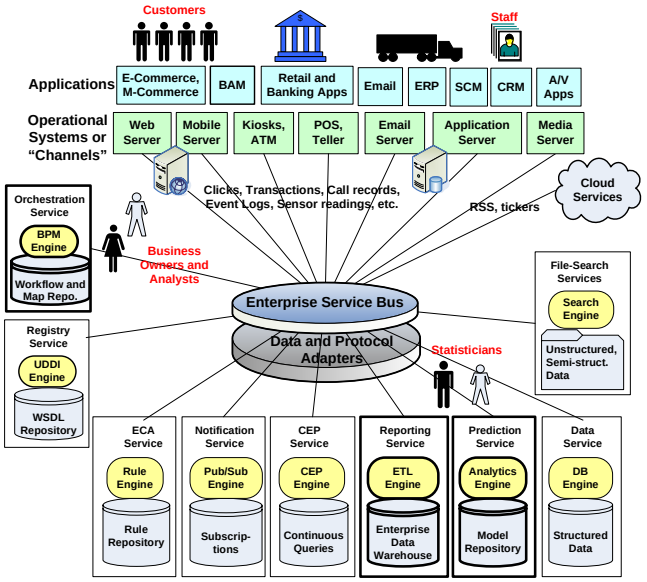

Figure 2: SOA environment where processing engines are presented as services and applications access services through multiple channels. Application-specific services are not shown.

components from each other improves reusability of each service and the flexibility of the overall architecture. Trying to unify all these existing and emerging services into DBMS is not a viable strategy.

Figure 2 highlights reporting (*aka* OLAP) and prediction services as BI services. As more real-time data stream and complex event processing applications emerge we will include Complex Event Processing (CEP) service into this category to detect temporal, spatial or logical patterns from event streams. ETL engine is responsible for *extracting* the historical data from data services, *loading* them into the data warehouse and *transforming* them for fast access to summarized reports. Analytics engine goes further and does algorithmic processing on historical data using data mining models, to extract patterns and make forecasts. Finally, data and file services provide access to structured and unstructured data, respectively, thus supporting BI and other applications and application-specific services. The rule engine is presented as an Event-Condition Action (ECA) service and the Publish-Subscribe engine is presented as a Notification service. ECA service can be contacted by other services to retrieve enterprise policies or trigger rule-based decisions.

For instance, in our retail platform both frontline coupon offer process and the model management workflows would be registered with the BPM engine/service. Retail managers would continuously decide what promotional rates to offer on certain conditions and register their decisions as IF-THEN rules with the ECA service. During a specific customer interaction, the offer process would consult reporting service to learn which segment this customer belongs to and ask analytics engine on *e.g.* the probability of this customer switching to other retailers [5]. Then, it would fetch product price, availability and profit margin information. The process would compile and send this information to the specific ruleset or policy in the ECA service to finalize the decision for the percentage of personalized discount that will be applied to this customer for this product at this time period, in this store, at this region, *etc.*

3. Our Solution

The issues with model complexity due to business complexity cannot be simplified magically (see "No Magic" in [18]). Tools can help sift through large data and help store models, but it takes coordination among people in different roles, different departments, and even different organizations (due to partnership or outsourcing) to discover actionable knowledge. Our system is designed to increase access of multiple stakeholders to model development process, automate tedious/repetitive tasks, reduce delays, and finally capture rich information to track model provenance and help with future inquiries and management.

Our system can track a data mining model lifecycle from creation, to business inclusion, to performance deterioration ("aging" or "decay"), to maintenance, and finally to expiration and archival. It also tracks rule-model and model-to-model dependencies, so that if a model needs to be evicted from the repository due to poor performance, then related entities that depend on this model are informed promptly. We track dependencies over multiple repositories using a BPM system (Microsoft Biztalk) with web service capabilities described below. We also define rich and extensible model-related metadata to track model provenance and capture the collective intelligence that is today left in experts' minds and in silos of proprietary tools. Rich metadata helps close the semantic gap between business analysts and model developers leading to actionable BI. For example, the campaign process in our retail platform uses two data mining models, *customer behavior change detection* and *fuzzy product matching*, to present personalized coupons. These models will be maintained by the model management system described here.

3.1 Model Lifecycle

We implemented a special workflow in Microsoft Biztalk BPM to orchestrate data mining model management, specifically to handle model lifecycle-related events. Any other BPM system allowing visual or programmatic description of workflow logic could have been used. The orchestration publishes a receive port as a web service to receive model-related event calls and queries. Note that this service is for management purposes and not for applications to query a specific model's prediction. Internally, the workflow calls model metadata repository to store or query model metadata and to advance models' status. Details of repository API for adding-removing models, getting model tags/authors, adding dependencies, etc. are straightforward and therefore skipped for brevity. Model repository is currently implemented as a .NET library and the data is stored in Microsoft SQL Server. Figure 3 shows a partial diagram of the orchestration that synchronizes the communication of business analysts, statisticians and the automated model performance evaluation routines over model-related events. The orchestration first checks the model event type in the event payload and switches to the branch associated with that type of event that represents different model states.

These states are shown in Figure 4: model created, referenced, unreferenced, deteriorated, to be expired, and expired. Model is assumed to be in the creation state until it is deployed. A MODEL CREATED event (zoomed in Fig 3. as red box) informs the orchestration that statisticians have deployed a new model into the model repository. Statisticians can raise this event by calling the web service published by the orchestration. The event carries the model identifier and other basic model attributes shown. The new model goes to an Active state and starts to get managed by our

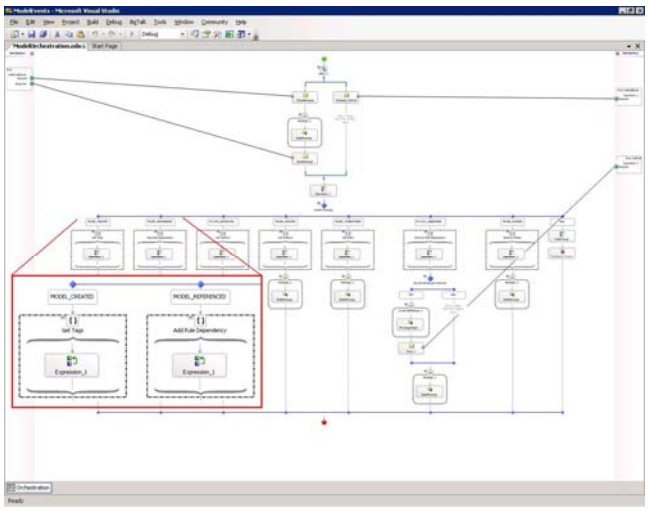

Figure 3: Data mining model management workflow implemented in Microsoft Biztalk.

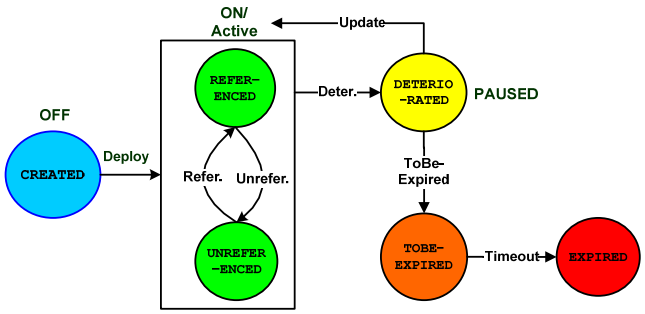

Figure 4: Model lifecycle tracked by a hierarchical state machine.

system. Note that Figure 4 is a hierarchical state machine where Active state embeds REFERENCED and UNREFERENCED sub-states. Models that enter Active or ON state are initialized to Unreferenced sub-state. The Active state also keeps the history of the last state (for each model), so that it can return to this state when models exit and reenter Active state.

The related business analysts and other users will be informed when a new model is created and deployed. The orchestration does this by compiling a list of subscribers based on subscription keywords and notifying them. If analysts choose to integrate this model's prediction in their business rules or processes, then they reference the model's endpoint (*e.g.* Web Service URI recorded in metadata) and raise the MODEL REFERENCED event to inform and be informed about future model status updates. Processes can reference model services directly or via a rule-model binding. The Reference event payload includes a unique business rule id and the model id and the orchestration adds the rule-model dependency information to the given model's metadata in the model repository. When model developers receive MODEL DETERIORATED events as a result of the orchestration's periodic performance scans (described later) they determine if they can fix the model by updating internal parameters and if so they raise a MODEL UPDATED event. The orchestration handles update notifications similar to deteriorations by forwarding them to related people such as authors of this model and dependent business rules. If the model cannot be fixed (*e.g.* due to products

becoming obsolete), then developers can choose to set a model expiration time and raise a MODEL TOBEEXPIRED event. The orchestration will find the business rules dependent on this model using the model repository API and notify the associated rule authors. It also sets a timer, so that a MODEL EXPIRED event is raised at the expiration time. When the rule authors receive expiration notification, they revise their business rules based on other active models in the repository, remove references to expiring models, and raise the MODEL UNREFERENCED event to the orchestration. When a model finally expires, business rules won't be able to access it to get predictions and the orchestration removes the business rule ids from the model metadata. If no pending references exist, the orchestration directly raises a MODEL EXPIRED event to finish the process. This will remove the model from the system and no further notifications related to this model will be raised.

Note that data mining model lifecycle management and metadata collection (described next) is representative of the BSM functionalities such as configuration, measurement, fault and trouble ticketing, and inventory management [4].

3.2 Metadata for Data Mining Models

By retaining collective intelligence gained over time, rich metadata helps track model provenance and close remaining gaps between the business analysts and the model developers. Figure 5 shows model metadata at the top level also showing details of basic attributes. Basic attributes include the globally unique identifier (GUID), model name that provides a quick verbal reference, and the textual description that explains what the data mining model is about. We also track model creation, last access and expiration times. Model authors are the developers that need to be informed about model lifecycle events. Data mining algorithm specifies particular algorithms (decision tree, logistic regression, *etc.*) used in construction of this model. Other metadata fields include model schema for describing I/O attributes, model assumptions, tags/keywords, training dataset information, performance evaluation methods, event triggers or thresholds, rules that depend on this model, and finally inter-model dependencies.

An input attribute can be selected simply from a column in the database (*e.g.* CustomerId) or it can be an aggregated attribute (*e.g.* The total purchases over the last 3 months). The output attribute represents the result/attribute that the data mining model is trying to calculate or predict (*e.g.* The top 10 coupons to offer, churn rate, retention rate, or customer response probability). In addition, a data mining model works best (or only works) under certain conditions. The model developer can document these model assumptions (*e.g.* data ranges, input data quality, *etc.*) during the model construction and update them with gained knowledge over time. This rich information, beyond simple versioning and dependency tracking, gets transferred to the peer model builders or business analysts who rely on the correct operation of models when making business decisions or creating business rules. Imagine a practical business intelligence system containing hundreds to thousands of different models; it would be a daunting task to identify these buried assumptions, even if they could eventually be discovered by inspecting the entire data mining model. As model complexity increases and inter-model and model-rule dependencies proliferate, a model management system such as ours becomes a necessity. An analyst or manager cannot sift through raw database tables, try to understand SQL queries, or even locate the models during a business chaos such as all "personalized" coupons coming out the same, customers refusing to pay, or out-of-stocks occurring.

Models can be tagged by their authors to allow indexing and textual search. By querying a tag, we retrieve models that share the same or similar tags, thus finding models that are semantically linked or related to each other. This helps with model selection process before model composition. More interestingly, social tagging can be applied creating a perfect application for *Enterprise 2.0* (*i.e.* Web 2.0 for the enterprise). Business owners, IT and statisticians can gradually open models that reach a certain level of maturity for use by a broader community including customers doing their personal projects. We have not experimented with social tagging yet, but our system enables such scenarios.

Business rules are also represented by GUIDs in their respective rule repository (shown in Fig.1). When a model-change event happens (*e.g.* model deterioration) the author of a dependent business rule gets notified by the orchestration. Different models can also depend on each other through versioning or other taxonomies. For example, a model can be the result of back-fitting of another model, thus having an "improved model" (parent-child) relationship. Models can also share the same data source, but might use different data mining algorithms, thus having a "Models

Figure 5: Model metadata and basic attributes.

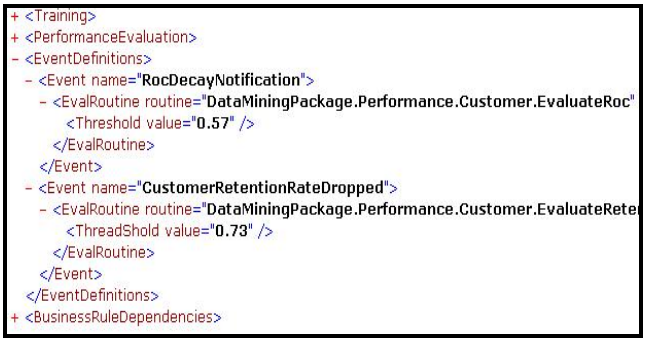

Figure 6: Model performance evaluation and event triggering.

Sharing Training Dataset X" (peer or siblings) relationship. Similarly models can be composed using machine learning to create a supermodel and this would create both parent-child and sibling relationships. Our system, through the use of rich metadata, allows tracking and managing different types of complex model relations.

Developers can use their favorite BI tools to build data mining models. Next, they can export their model schema (*e.g.* using the standard Predictive Model Markup Language –PMML specification [21] format) and enrich that with metadata such as our basic model attributes, model assumptions while also exposing their performance evaluation routines through public API (SOAP, HTTP, message queues, *etc.*) to be called by our orchestration periodically or per-event-based. Associated with the performance evaluation routines, they define event trigger predicates shown in Figure 6. Simple examples include: "ROC < 0.57" or "customer retention rate < 0.73". The orchestration scans through the associated performance evaluation results and raises the related event (*e.g.* deteriorated) if the criteria for "EvaluateRoc" routine evaluates as TRUE.

3.3 A Preliminary Testbed

We built the orchestration in Figure 3 and the model metadata repository. To test the functionality of model orchestration and other system components, we tracked lifecycles of a few exemplary models related to retailing as shown in Figure 7. The figure displays the model events for our *CouponPrediction* model. This model was created, referenced, deteriorated and updated in a day time and then referenced again the next day. Figure 8 displays the ROC-based model performance results for two CouponPrediction data mining models. It highlights models with deteriorated performances in red color.

The goal of this testbed and the web front-end is to demonstrate that the workflow engine is working, model events are being captured and metadata is tracked. When deployed in the enterprise setting, this system will support a community of statisticians and business entities. Therefore, we need to run field studies to understand the needs of this environment to identify requirements of the visualization support. We present a preliminary analysis of this emerging ethnography and list a number of grand challenges for the CHIMIT community in Section 6.

We do not report or claim any performance results in this paper. However, to get a sense of execution times of different algorithms, we compared Decision-Tree, Naïve-Bayesian, and Clustering over the same training dataset with ~18K rows on a 3.20 GHz server with 3.5GB memory. Decision tree was the fastest and finished in 2 seconds, whereas Naïve-Bayesian took 8 seconds and Clustering 48 seconds. The results would depend on resource (CPU, memory, I/O) capacities, model and algorithmic complexity, training data size and several other factors related to system configuration. In addition, the quality of predictions can only be judged by statisticians and business analysts under certain circumstances.

4. Related Work

We address challenges that start after models are built. Our goal is to let developers use their favorite tools (MSAS, SAS, FairIsaac, Oracle) to build data mining models and then register them into our model management system. In other words, this work does

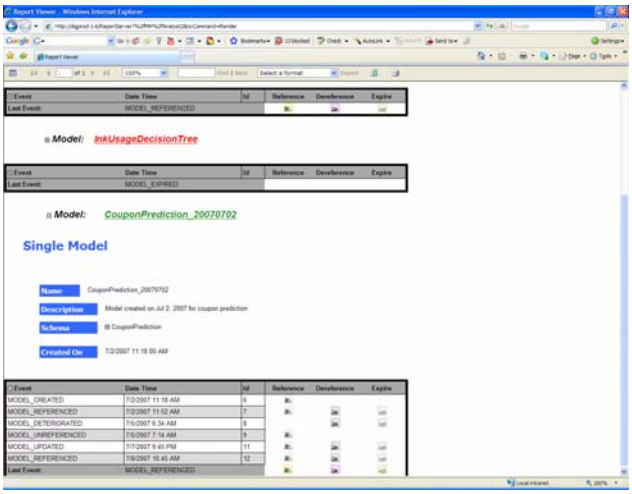

Figure 7: Web front-end to model management system showing a few models in our repository and their status.

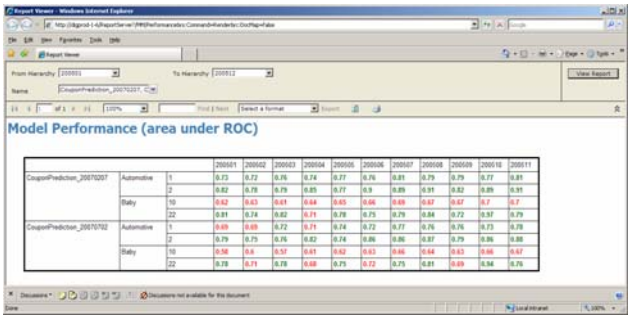

Figure 8: Model performance comparison where deteriorated models are marked with red color.

not focus on the details of algorithmic issues and attribute/parameter selection in model construction, which differentiates our work from most existing tools and systems [19][20][23][9]. No prior art addresses integration of data mining model collections with business processes to automatically provide business insights, while also addressing the model aging, scalability, timely-communication and semantic gap challenges. Microsoft Analysis Services (MSAS) provides Analysis Management Objects (AMO) library to create, modify and delete data mining objects such as cubes and mining models. Their "collections" can contain only mining models built on the same data whereas our system can cover and relate all mining models and other business objects in an enterprise. We use model-dependency tracking to relate models to rules and other models. Dependency tracking and decay monitoring capabilities of SAS Enterprise Miner, FairIsaac Model Builder, and Oracle Data Mining & Analytics are limited compared to our system. Data mining models are more complex constructs then web service descriptions (WSDL) and application programming interface (API) definitions. Therefore, service and application lifecycle management systems [12] focusing primarily on versioning do not solve problems addressed here.

Recent work by Chen, *et al.* [22] compares data mining models based on business applicability, but they do not address management issues that we discuss in this paper. Other recent work shows the importance of making models first-class citizens

of database and calls these special repositories modelbases [15]. Yet, there hasn't been any work providing integration of data mining modelbases with business processes to automatically provide business insights, while also discussing the model aging, scalability, timely-communication and semantic gap challenges mentioned before. There also has been increased interest in real-time ETL [25] and data stream processing recently. These topics are complementary and orthogonal to our current system in providing real-time BI.

A list of grand challenges for the data mining field are identified in KDD 2007 [26], which confirms some of our findings and suggestions by stating that the heavy dependence of online marketing applications on large datasets makes "data mining and statistical data analysis, modeling, and reporting an essential mission-critical part of running the on-line business."

5. Technical Challenges

There are two dimensions to model prediction in our data mining model management system. One is model evaluation or training with new data, which can take significant amount of processing time and is better suited as an offline or asynchronous operation. The other dimension is model querying and access to model's prediction results, which can be completed as a synchronous call or query (although analytical queries usually involve multiple joins and are therefore more complex than simple select or aggregation queries). As we've seen in Section 3.3 complex models over large datasets take minutes or even hours to evaluate. A single server cannot handle evaluation of more than a few models simultaneously due to intensive CPU, memory, and I/O needs. Therefore, if our system is deployed as a multi-tenant or hosted service providing model evaluation and query support simultaneously, then we'll also need to address scheduling and load balancing issues over a centralized server farm or distributed systems.

Portability issues follow performance and scale issues. One needs to be able to export/import models among different platforms from *e.g.* Oracle, Microsoft, IBM, SAS, or HP. Although web services provide loose coupling between processes and models, due to variety of algorithms (Decision Tree, Naïve Bayesian, Linear Regression, Logistic Regression, Association Rules, Neural Nets, and Clustering) supported in different platforms one may not be instantiate a model on another platform. Our model metadata practice and standards such as PMML alleviate some of the problems, but not all.

Finally, we will have to quantify performance of the management process to satisfy requirements of ITSM and BSM practices. We can evaluate success of our management system in terms of model and business process up-down times and its affects on BSM (or business value) by the prediction accuracy for when the models are up and active. We can also quantify success in terms *utilization* or of how many business-level queries were answered by the models in the management system. These discussions on business impact lead to human aspects of our system and will be discussed in the next section.

6. Human Factors Challenges

For our system to be truly usable and scalable, and have business impact, we need to complete the ethnographic studies, select the best visualization options and do the evaluation through longitudinal studies. This emerging field at the intersection of business processes and BI creates challenges not addressed by the Computer-Human Interaction (CHI) community before. We list some of the grand challenges specific to the field and overview related future work in this section.

6.1 Field Studies

Though extensive ethnographic field studies focusing on system administrators have been conducted [2][7][11][14][16] to the best of our knowledge, there is no published work that focuses on studying the data mining model developers or statisticians and business analysts that build, maintain and integrate scalable data mining models.

The open questions in this space are: what tasks are structured and which ones are unstructured as defined in [2]? Which category of tasks is more predominant and is there an overlap between structured and unstructured tasks? To address these research agendas, there is a need to understand the following: How do statisticians maintain models; Does the individual maintenance process differ from collaborative maintenance; How do model developers communicate with each other to handle error conditions and exceptions; How do they synchronize with the business analysts for handing off deliverables such as reports; How do they overcome semantic gaps? Who provides the ontological mapping? Does further synchronization between the two user groups ensue when the analysts use query and analysis tools to examine the information? How does business-to-model feedback mechanism work? Which metadata is most useful? How do business analysts handle errors/exceptions when models fail?

We need to conduct multiple field studies to systematically analyze the work practices, favored tools, decision making approaches, and mental models of statisticians and business analysts to create user models. User modeling will provide us a theoretical framework to identify technical, social, cognitive, and business challenges faced by these key stakeholders. This will be followed by applying task analysis techniques (*e.g.* hierarchical task analysis, goals-operator-methods-selection, cognitive task analysis, and activity theory) and work analysis techniques (such as flow, sequence, cultural, artifact, physical) to understand the perception, comprehension and projection activities undertaken by these users.

6.2 Visualization as Dashboards

If critical aspects of the business can be identified, then dashboard design would be simplified. In general, the dashboard should show only the key business and system needs. The exact metrics and widgets will vary from business to business. If we follow an analogy from cars to explain the variation, the dashboards of a sedan car and a stick-shift sports car will be quite similar in functionality displaying speed, temperature, and gas, except that the sports car will also have the tachometer to measure RPM for maximum performance. Similarly, dashboards used by several industry sectors may include total revenue and expense reports from accounting, employee counts and open requisitions from HR, and other values from across the enterprise.

Based on the field study findings, we will identify design guidelines for dashboard visualization. It is important to create design guidelines to address cognitive complexity, spatial organization, information coding, state transitions, orientation & help, navigation, querying, data set reduction, and other types of

interactions. Past literature and anecdotal observations [10][17] suggest that dashboard visualization may be of interest to this particular user group as dashboards are able to quickly provide a visual snapshot or overview of the enterprise goals, metrics, benchmarks, and the system status, results and alerts.

We have identified four dimensions that can be considered for designing dashboards:

1) Introspection: Understanding the knowledge requirements of the specific domain with respect to monitoring, analysis, and management stages is critical. Monitoring typically comprises of average task time, number of queries running and queued, number of models in each stage of the lifecycle (*i.e.* created, referenced, etc.). Analysis typically comprises of interactions to support further drill down in the data to visualize the correlation between the models and the lifecycle and the details of the model itself. Management of models comprises of information such as model creator(s), model users, and associated metadata. Also, since information is typically pulled from various data sources, the granularity (per model, classification of models) of the data to be displayed, and the frequency of update (hourly, daily, weekly) is currently unclear.

We see the following metrics taking precedence in model management. How many total models? How many models in each state? When events caused model transitions? When did they occur? Who triggered the event? Who is or how many processes are using this model? Will it be useful to visualize the state machine and to whom? How long does it take to process? What is the arrival rate of new training data? How frequently is it being updated? What are the most problematic models in which areas?

2) Customization: Different users will have different views of the process, so we need to identify the appropriate data to be visualized for the various user groups such as consumers, executives, project managers, analysts, statisticians and other technical staff. Field studies will also enable us to understand how the access/privilege aspects should be managed (*e.g.* Who can create individuals or collaborative models? Who can modify dashboard components by adding functions, modifying thresholds, and alerts?)

Various vertical sectors and different organizational roles will also require insights into different metrics and have different drill-down (or roll-up) capabilities, including down to the level of specific business or IT events. Ultimately, using customized dashboards the real-time visibility of the organizational operation is increased attracting more eyes on specific processes and emerging issues.

We observe that since model development is a long and tedious process requiring domain expertise, each developer will only have a few or few tens of models under his management. On the other hand, business analysts manage thousands of rules and policies and after BI integration they will also have to manage the dependencies of business assets to mining models. This imbalance in management scale adds another dimension to dashboard visualization complexity.

3) Presentation: After the content has been customized for each user type, we need to address the design, layout and navigation issues such as: Where should the thresholds of the rules and alerts be defined; Should the main dashboard include the execution and management of exceptions (i.e. rules that trigger it, actions to be taken, and recipients to be informed); How should we tie the visual representation of the overall lifecycle of the models with "querying" each model (i.e., switch between overview and detail and back) ?

4) Adaptation: of both visualization and evaluation to support continuous improvement as required by ITIL v3 best practices. Appropriate visualization support is required as the system evolves from low to high utilization and as complex dependencies increase over time.

6.3 Longitudinal Evaluation

In the final phase, we will develop appropriate metrics and methodologies to evaluate dashboard visualization. Many techniques for visualization of large scale data have been proposed; we would like to learn the aspects of the visualization that are most effective for our particular application. Are the proposed visualizations such as speedometers/gauges, radar graphs, scatter plots, pie charts, bar charts or histograms, tabular displays, area or 3D graphs truly usable, effective and useful? What are the strengths and weaknesses? Are the visualization techniques and methods also under continuous improvement? How can we carefully integrate these tools into solutions that address real life problems?

If the global and departmental KPIs are not realistic or the business value is not clear, then all these tools and portals carry the risk of becoming yet another system to manage, an organizational overhead and a cost center. Therefore, Project and Portfolio (PPM) tools should be integrated into the picture and used effectively to define objectives and requirements, allocate and optimize resources, and monitor progress down to task-level and time spent on each task [our system can track time spent on each model lifecycle stage, since everything is clearly defined].

We conducted informal usability testing of various visualization techniques, and observed that that while usability studies can help identify design issues, longitudinal research is essential for analyzing behavior over time especially for knowledge management tools where usage of monitoring, analysis, and management changes with time. Models can take days and even years to be created and they evolve in iterative cycles. As mentioned before organizations go through a feedback cycle of model creation, analysis, planning rules and alert, reviewing alerts. Thus, we find that the strengths and weaknesses of the visualization can only be effectively measured and evaluated over time.

We assume that the data is being collected on critical processes. In reality, businesses that are trying to cover gaps in processes may find that they're not collecting any data related to that process. Without this step handled, BI tools would be useless to those specific processes. Therefore, "BSM, PPM, and BI products will likely converge in the near future, with products that have their roots in BI emerging as the winner for many organizations." [elusive]. We believe that longitudinal research will enable us to uncover the impact of the environment on the visualization by studying the context of usage and how learning and using the visualization changes over time.

7. Conclusions

Serious challenges on managing data mining models and integrating them with online services in service-oriented

architectures (SOA) have been addressed in this paper. These challenges include model aging, management scalability, timely-communication among parties on model changes, semantic gap on interpreting models, and business process integration. By addresses these challenges we hope to provide sustainable and real-time Business Intelligence (BI) to business services and operational systems.

At this stage, it is hard to put an exact or even expected business value on our model management system and our claim is supported by field experts, e.g. “Projects and systems are so complex that few CIOs can predict direct impact on business” [4]. For now, our goal is to simplify BI tasks, enable collaboration and make analytics more tangible and relevant to business users. It suffices to say that the BI and BPM/SOA markets are both measured in billions.

8. Acknowledgements

We’ve like to thank our managers Mohamed Dekhil and Henry Sang for sponsoring this project.

9. REFERENCES

[1] Ari I., Li J., Ghosh R., Dekhil M., Providing Session Management As a Core Business Service, WWW 2007.

[2] Bailey J., Kandogan E., Haber E., Maglio P.P., Activity-based Management of IT Service Delivery, In CHIMIT 2007.

[3] Barrett, R., Kandogan, E., Maglio, P. P., Haber, E. M., Takayama, L. A., Prabaker, M., Field Studies of Computer System Administrators: Analysis of System Management Tools and Practices. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work.

[4] Biddick Micheal, Hunting the elusive CIO dashboard, InformationWeek, 2008.

[5] M.-C. Chen, A.-L. Chiu and H.-H. Chang, Mining changes in customer behavior in retail marketing, Expert Systems with Applications 28 (4), pp. 773–781. 2005

[6] Chieh-Yuan Tsai and Min-Hong Tsai, A Dynamic Web Service based Data Mining Process System, 5th IEEE International Conference on Computer and Information Technology (CIT), 2005, pp 1033-1039.

[7] Dijker, B., A Day in the Life of System Administrators, SAGE, http://sageweb.sage.org

[8] Erl. T, Service-Oriented Architecture, Concepts, Technology and Design, Prentice-Hall, 2005.

[9] FairIsaac Enterprise Decision Management (EDM) http://www.fairisaac.com

[10] Few, S. Information Dashboard Design: The Effective Visual Communication of Data”, O'Reilly Media, Inc., January 24, 2006, ISBN-10: 0596100167.

[11] Haber, E. and Kandogan, E., Security Administrators in the Wild: Ethnographic Studies of Security Administrators. SIG CHI 2007 Workshop on Security User Studies: Methodologies and Best Practices.

[12] HP SOA Systinet Registry/Repository, http://www.hp.com/go/soa

[13] IT Infrastructure Library (ITIL) v3, http://itil.org

[14] Kandogan, E., and Haber, E. Security Administration Tools and Practices. Book: Security and Usability: Designing Secure Systems that People Can Use. O'Reilly Media, Inc., 2005, pp374-394.

[15] Liu B., Tuzhilin A., Managing large collections of data mining models, Comm. of the ACM, Vol 51. No:2, Feb 2008

[16] Maglio, P.P., Kandogan, E. and Haber, E. "Distributed Cognition Analysis of Attention and Trust in Collaborative Problem Solving." Proc. Cognitive Science 2003.

[17] Malik, S. Enterprise dashboards : design and best practices for, Wiley, August 12, 2005, ISBN-10: 0471738069.

[18] Michael Meltzer, Using data mining on the road to successful BI, Parts II-III, DMReview.com, Sep-Oct 2004.

[19] Microsoft SQL Server 2005 Analysis Services, http://www.microsoft.com/sql/technologies/analysis/

[20] Oracle Data Mining, http://www.oracle.com/technology /products/bi/odm

[21] PMML, Predictive Model Markup Language, Version 3.2 http://www.dmg.org/pmml-v3-2.html

[22] Ron Kohavi, Neal J. Rothleder, And Evangelos Simoudis, Emerging trends in business analytics, ACM Communications 2006.

[23] SAS Institute Model Manager, http://www.sas.com/ technologies/analytics/modelmanager/

[24] Sumathi S. and Sivanandam S. N., Data mining in customer value and customer relationship management, Studies in Computational Intelligence (SCI) 29, 321-386, 2006. Springer-Verlag.

[25] Thomsen C, Pedersen T., Lehner W., RiTE: Providing On-Demand Data for Right-Time Data Warehousing, ICDE 2008

[26] Usama Fayyad, From Mining the Web to Inventing the New Sciences Underlying the Internet, KDD 2007 Keynote.

[27] Wei M, Ari I., Li J., Dekhil M., ReCEPtor: Sensing complex events in data streams for SOA, HPL-2007-176, 2007.

[28] Witten I H., Frank E., Data Mining, Practical machine learning tools and techniques, Morgan Kaufmann, 2000.

[29] Web Services Business Process Execution Language (WSBPEL), OASIS, http://www.oasis-open.org