Data Mining and Knowledge Discovery, 12, 127–150, 2006
© 2005 Springer Science + Business Media, Inc. Manufactured in the United States.
DOI: 10.1007/s10618-005-0018-2

Organizing Multiple Data Sources for Developing Intelligent e-Business Portals

JIA HU hujia@kis-lab.com
NING ZHONG zhong@maebashi-it.ac.jp
Department of Information Engineering, Maebashi Institute of Technology, 460-1 Kamisadori, Maebashi, Gunma, Japan 371-0816

Received April 3, 2005; Accepted August 31, 2005

Published online: 8 April 2006

Abstract. Enterprise applications usually involve huge, complex, and persistent data to work on, together with business rules and processes. In order to represent, integrate, and use the information coming from the huge, distributed, multiple sources, we present a conceptual model with dynamic multi-level workflows corresponding to a mining-grid centric multi-layer grid architecture, for multi-aspect analysis in building an e-business portal on the Wisdom Web. We show that this integrated model will help to dynamically organize status-based business processes that govern enterprise application integration.

We also present two case studies to demonstrate the effectiveness of the proposed model in the real world. The first case study is about how to organize and mine multiple data sources for behavior-based online customer segmentation, which is the first crucial step of personalization and one-to-one marketing. The second case study is about how to evaluate and monitor data quality, which in return can optimize the knowledge discovery process for intelligent decision making. The proposed methodology attempts to orchestrate various mining agents on the mining-grid for integrating data and knowledge in a unified portal developed by a service-oriented architecture.

Keywords: intelligent e-business portals, the Wisdom Web, multi-layer grid, dynamic multi-level workflows, multi-database mining

1. Introduction

The Web has far-reaching impacts on both academic research and business operations. Developing intelligent e-business portals is one of the most sophisticated applications of Web Intelligence (WI), which needs to be supported by various WI technologies (Zhong, 2004; Zhong and Liu, 2004; Zhong et al., 2003a). An e-business portal enables an enterprise or a company to create a *virtual organization* (or a *virtual community*) on the Web where key production/information steps are outsourced to partners and suppliers. In other words, an e-business portal is a single gateway to personalized information needed to enable informed business decisions, in which all of the contents related to the virtual organization can be accessed, although such kind of organization information is geographically distributed over multi-site, multi-data repositories, and multi-institution.

As the huge and multiple data sources are coupled with geographic distribution of data, users, systems, services, and resources in the specific types of enterprises, the Grid platform is needed as a powerful middleware for developing e-business portals (Foster and Kesselman, 2003b). Furthermore, workflow management systems address enterprise process automation problems, which refer to a formal, executable description of a business process (Alonso et al., 2004).

In this paper, we present a conceptual model with three levels of dynamic workflows, namely *data-flow*, *mining-flow*, and *knowledge-flow*, corresponding to the grid with three layers called *data-grid*, *mining-grid*, and *knowledge-grid*, respectively, for transforming data to active knowledge in a unified portal. Furthermore, such a multi-layer grid is a mining-grid centric one for multi-aspect analysis, and for severing status-based business processes (Hu and Zhong, 2004, 2005).

The rest of the paper is organized as follows. We discuss the background and related work in Section 2. In Section 3, we describe the architecture of our e-business portal and its main features. Sections 4 and 5 discuss how to model business processes with workflows and how to collect multiple data sources with Web farming, respectively. In order to demonstrate how to effectively use the proposed model in real world applications, we present two case studies. The first case study described in Section 6 is about how to organize and mine multiple data sources for behavior-based online customer segmentation, which is the first crucial step of personalization and one-to-one marketing. The second case study described in Section 7 is about how to evaluate and monitor data quality, which in return can optimize the knowledge discovery process for intelligent decision making. Finally, Section 8 gives conclusions and our future work.

2. Background and related work

2.1. Enterprise applications and portals

Enterprise applications are about the display, manipulation, and storage of large amounts of complex, persistent data and the support or automation of business processes with that data (Fowler, 2003). Data is the most important asset of a company, which is persistent because it needs to be around between multiple runs of applications and processes. It will often outlast the hardware that originally created and saved it, and outlast operating systems, and applications. The new information could be merged with old data seamlessly. Business models live in a world dominated by data and knowledge. As far as business models are concerned, this is a world of business information (BI) and knowledge management (KM) (Pyle, 2003). Although some investigators have stressed the importance of knowledge management for developing advanced information systems (Detlor, 2004; Maier, 2004), they did not touch the point of how to automatically transfer the data to knowledge and how to use all kinds of knowledge with wisdom. As Charles H. Spurgeon[1] said "Wisdom is the right use of knowledge; to know how to use knowledge is to have wisdom."

Increasingly, as the Internet is driving enterprises to run their business with agile, efficiency, and wisdom, the enterprises have to integrate the existing sources, services, and applications. Hence, portals are a good choice of enterprises in the long term, as portals are largely based on existing Web application technology and integrate diverse interaction channels at a central point with an aggregated view across all information (Wege, 2002).

2.2. Design pattern and evolution

Layering is one of the most common techniques that software designers use to break apart a complicated software system (Buschmann et al., 1996; Fowler, 2003). In a layered system, the higher layer uses various services defined by the lower layer, but the lower layer is unaware of the higher layer. Furthermore, each layer usually hides its lower layer from the layers above. The hardest part of a layered architecture is deciding what layers to have and what the responsibility of each layer should be.

The notion of layers becomes more apparent in the 90's with the rise of client/server systems. Client/server applications remove dependencies on hardware platform, but are still dependent on underling application infrastructures such as .Net, J2EE, or CORBA, all of which have varying degrees of scalability and reliability problems (Xu, 2003).

As the business logic is more and more complex, it is awkward and tedious by embedding the logic directly in the client side. The object-oriented community proposes a three-layer system (Fowler, 2003). In this approach, there is a presentation layer for user interface, a domain layer for business logic, and a data layer for data access and management. Hence, the intricate domain logic can be separated from the user interface as a self-governed layer. With the rise of the Internet, suddenly enterprises want to deploy their own applications with a Web browser. However, if business logic is buried in a rich client in the client/server environment, then all the business logic is necessary to be redone to have a Web interface. A well-designed three-layer system could just add a new presentation layer and be done with it.

2.3. Advanced architecture

Grid computing attempts to provide a powerful middleware for coordinated and controlled resources sharing and problem solving in dynamic, multi-institutional *virtual organizations* (Foster and Kesselman, 1999, 2003a; Foster et al., 2001). The main challenge of Grid computing is the complete integration of heterogeneous computing systems and data resources with the aim of providing a global computing space through the use of standard protocols. An important function of Grid middleware components is to discover resources and information about these resources in order to optimize their use (Stork, 2002).

Although most of Grid projects have focused on resource sharing in the distributed environment, researchers begin to touch about how to employ knowledge processing on the Grid (Cannataro et al., 2004; Churcin et al., 2002). Congiusta et al., 2003 have been developing an environment for geographically distributed high-performance knowledge discovery applications. However, how to combine data mining and knowledge discovery with reasoning and how to use multiple information sources need to be investigated at a unified way in depth.

Increasingly, it is becoming necessary to develop any higher level services that can automate the process and provide an adequate level of performance and reliability. Xu et al. proposed a three-layer conceptual model for service visualization on the Grid (Xu, 2003). The model aimed to implement the service visualization and decouple applications from the hardware, removing scalability issues and other hardware limitations, while providing flexibility to meet new business opportunities. However, the model just mapped the Web application to the Grid environment, and how to implement the goals was not discussed.

While Grids have become almost commonplace, the use of Grid resource management is far from ubiquitous because of many open issues of the field, including multiple layers of schedulers, the lack of control over resources, the fact that resources are shared, and that users and administrators have conflicting performance goals. Nabrzyski et al. defined Grid resource management as the process of identifying requirements, matching resources to applications, allocating those resources, and scheduling and monitoring Grid resources over time in order to run Grid applications as efficiently as possible (Nabrzyski et al., 2004). They were also trying to use multiple layers for schedulers in the Grid environment.

Meanwhile, Deelman et al., 2003 discussed issues associated with workflow management in the Grid in general and provided a description of how to generate executable workflows on the Grid accordingly. Furthermore, Gil et al., 2003 used artificial intelligence planning techniques to compose valid end-to-end workflows on the Grid (Gil et al., 2004). So far, most of the workflow systems on the Grid maintain a static specification with a single layer model, in this way it will be hard for them to tackle problems in the real world. This paper attempts to present a dynamic three-level workflow management system with respect to a three-layer grid for enterprise application integration.

2.4. Multi-database mining

Multi-database mining is an important research area because (1) there is an urgent need for analyzing data in different sources, (2) there are essential differences between mono- and multi-database mining, and (3) there are limitations in existing multi-database mining efforts (Zhang et al., 2003). Wu and Zhang simplifies each enterprise as a two-level organization with a central company and multiple branches. They firstly proposed an approach named local pattern analysis for identifying patterns in multi-database of such enterprise by weighting (Wu, 2003).

3. Grid-based e-business portal architecture

The architecture described here aims at modeling the next generation of intelligent applications for enterprises. As stated in Han and Chang, (2002), data mining holds the key to uncovering and cataloging the authoritative links, traversal patterns, and semantic structures that will bring intelligence and direction to our Web interactions. Hence, it is a real challenge regarding the integration of multiple data sources and the transformation from such data to useful knowledge via mining.

Figure 1 shows the architecture of an e-business portal that has been developing by us. In this e-business portal, there are mainly four kinds of data sources deployed on the data-grid, namely customer, product, transaction, and Web usage datasets. Various data mining methods are employed as agents on the mining-grid for various service-oriented, multi-aspect analysis (Zhong et al., 2005). Furthermore, the rules and models mined from multiple data sources are stored on the knowledge-grid, so that they will be refined into active knowledge by reasoning and inferring with the existing/background knowledge. The active knowledge is employed to provide personalized services for customers, portals, and enterprise marketers through the three-layer grid.

The status-based business processes in the e-business portal are dynamically organized by using the workflow management system. The workflows are divided into three levels, namely data-flow, mining-flow, and knowledge-flow, corresponding to the three-layer grid, respectively. They are generated dynamically, based on the conditions

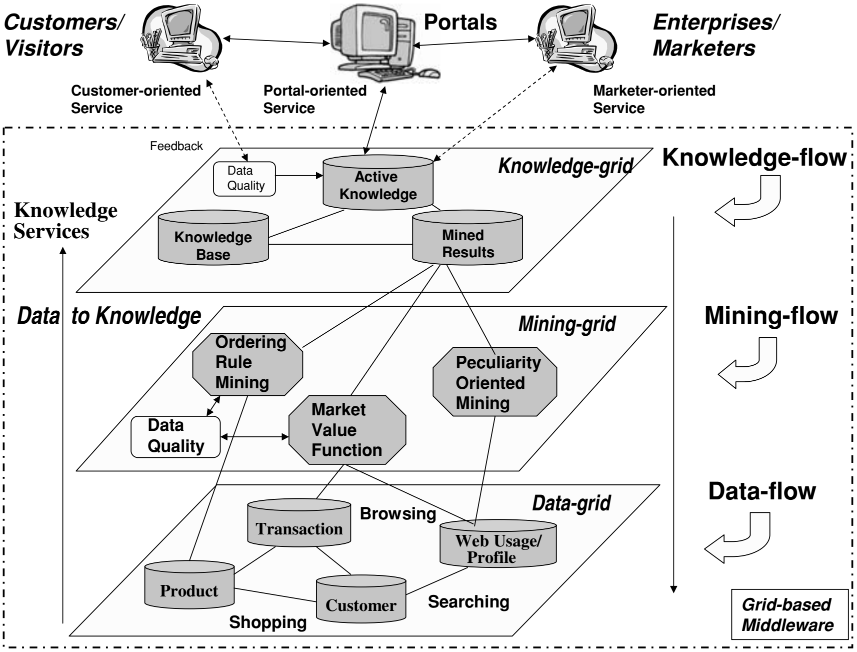

Figure 1. The architecture of an e-business portal.

(situations), data quality analysis, mining process, and available knowledge sources. In this model, lower level applications and services provide middleware support for higher level applications and services, thereby opening the door to developing more complex, flexible, and effective systems.

The architecture is deployed on Grid middleware and services, i.e., it uses basic Grid services to build specific knowledge services. Following the integrated Grid architecture, these services can be developed in different ways using the available Grid toolkits and services. The current implementation is based on the Globus toolkit (Foster and Kesselman, 1997), so that the Grid-enabled applications can be accessed by any end users via a standard Web browser.

4. Modeling business process with workflow

4.1. Knowledge-flow vs. knowledge-grid

The knowledge-flow is a collection of dynamically generated workflow processes related to generating, reasoning, refining, evaluating, and employing knowledge on the knowledge-grid for various knowledge services.

From the top-down perspective, the knowledge-grid is supported by both the mining-grid and the data-grid for serving the customers, portals, and enterprise marketers. From the bottom-up perspective, the data-grid supplies data services for the mining-grid, and then the mining-grid produces new rules and models for the knowledge-grid to generate active knowledge.

In general, several kinds of rules and models can be mined from different data sources by multi-aspect analysis. The results cannot be utilized for knowledge services until they are combined and refined into more general ones to form *active knowledge*, by meta-learning and reasoning. Distributed Web inference engines on the knowledge-grid will employ such active knowledge with various related knowledge sources together to implement knowledge services and business intelligence activities (Tomita et al., 2004; Zhong, 2004).

From the viewpoint of applications, there are mainly three kinds of knowledge-flows, namely customer-oriented flow, portal-oriented flow, and marketer-oriented flow on the knowledge-grid, respectively, as shown in Figure 2.

- Customer-oriented flow: A customer-oriented flow is invoked by a series of customer activities, for example, visiting portal, making registration, searching, browsing, shopping, and paying. It is also utilized to make a dynamic recommendation to a Web user (visitor and potential customer) based on the user profile and the usage behavior.
  In the traditional workflow, the whole process is defined in advance. In our applications, however, several data mining methods will be involved as agents in a mining process, in order to learn user behavior well. Hence, the workflow will never be completely static any more. Our workflows consist of static components defined in advance and dynamic components generated by learning in various situations. In order to represent the *dynamic* part in our workflows, we defined our own symbol to denote a dynamic task (see the symbol to denote *Mining Services* in Figure 2).

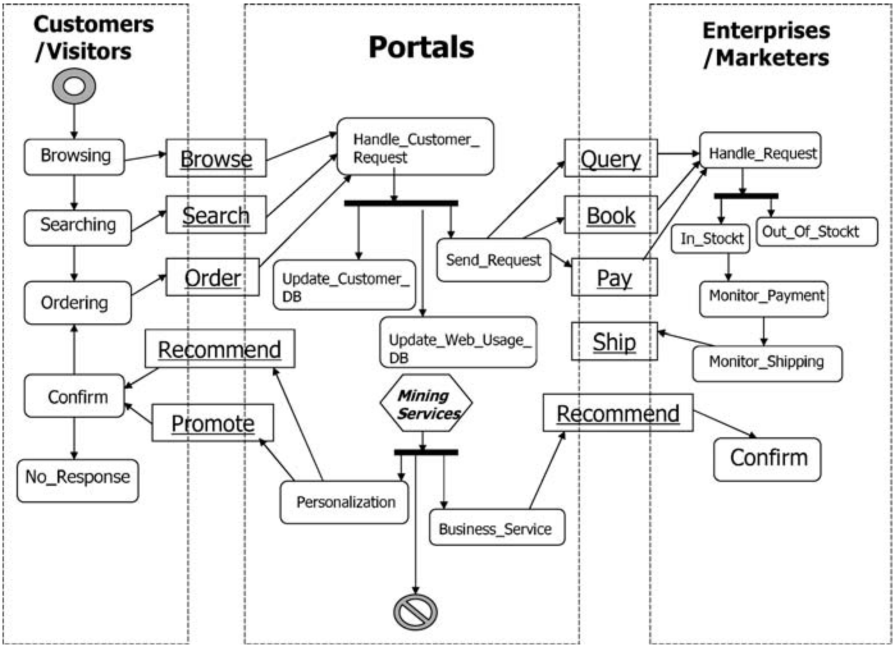

Figure 2. An activity diagram of knowledge-flows.

- Portal-oriented flow: A portal is an interface for customers, and various activities such as clicks and browsing sequences will be recorded for surfing behavior analysis. The main process of a portal-oriented flow aims at supplying personalized services for different customers by dynamically modifying the website's contents, links and website's structure.
- Marketer-oriented flow: Marketers (managers of enterprises) are another class of users of a portal. The marketer-oriented flow aims at supporting marketers to do market and sales analysis and helping them to make intelligent business decision. It combines Web usage data with marketing data to give information about how visitors used the portal for marketers.

4.2. Mining-flow vs. mining-grid

The mining-flow is a collection of processes related to planning, organizing, controlling, and managing a data mining process dynamically for different mining tasks on the mining-grid. From the top-down perspective, different data mining methods are deployed on the mining-grid as agents for mining services. The three kinds of knowledge-flows, as mentioned on Section 4.1, will evoke the corresponding mining services with standard interfaces when needed. However, on the mining-grid, different mining methods work just like agents, that is to say, they are working in an autonomic, distributed-cooperative mode.

One of the main reasons for developing multiple data mining agents on the mining-grid is that we cannot expect to develop a single data mining method to solve all problems since the complexity of real world applications. Hence, various data mining agents need to be cooperatively used in the multi-step data mining process for performing multi-aspect analysis as well as multi-level conceptual abstraction and learning.

The other reason for developing multiple data mining agents on the mining-grid is that when performing multi-aspect analysis for complex problems, a data mining task needs to be decomposed into sub-tasks. Thus these sub-tasks can be solved by using one or more data mining agents that are distributed over different computers and multi-data repositories on the Grid. Thus the decomposition problem leads us to the problem of distributed cooperative system design.

One of the key issues on the mining-grid is how to dynamically plan the mining-flow on it. That is to say, how to find, locate, integrate, and use the distributed resources for a mining process. The data mining process planning can be used to form such mining-flows (Zhong et al., 2001).

Figure 3 is an example of the mining-flow, i.e., the MVF (market value function) mining-flow. The MVF agent involves the identification of customers having a potential market value by studying the customers' characteristics and needs, and selects certain potential customers to promote (Zhong et al., 2004). Hence, when the MVF mining-flow is started, it will employ a data-flow first to get the appropriate data sources to analyze, and when the mining process is finished, the mining result, i.e., the mined market values, will be stored into the corresponding dataset on the data-grid. Whether the mined results are good enough or not for use will be left for evaluation on the knowledge-grid.

The other key issue on the mining-grid is to determine when and how to start a mining-flow. We use data quality analysis to solve such a problem. Hence, a mining-flow will start in two situations, the first is that it is invoked by a knowledge-flow from the

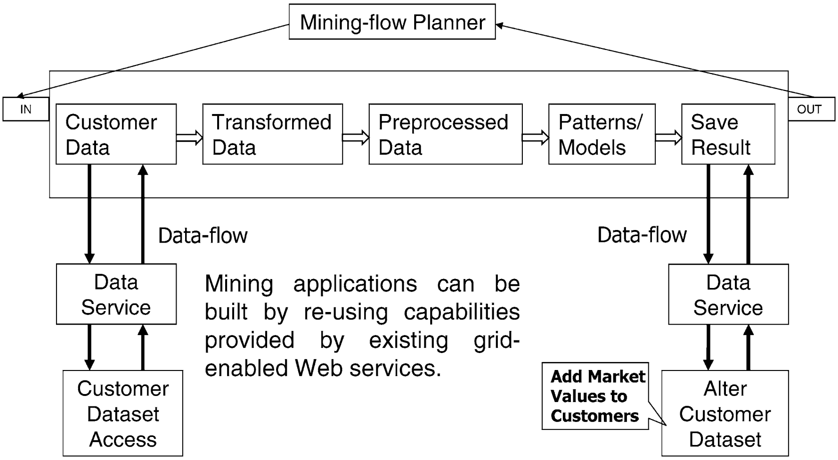

Figure 3. An example of the MVF mining-flow.

knowledge-grid; the other is that the data quality is good enough. Data quality analyzing services are deployed on the mining-grid, which can keep watching the data sources automatically. The mining-flow will not be invoked until the data quality is satisfied.

4.3. Data-flow vs. data-grid

The data-flow is a collection of descriptions for the dynamic relationship among multiple data sources on the data-grid. From the beginning, we only have the product dataset. The customers and Web usage information are accumulated during the business interaction between customers and a portal. Typically, there are mainly four kinds of relations between the data sources, namely solid relation, strong relation, weak relation, and uncertain relation, respectively.

If a customer has bought some kind of product, a solid relation is built between the customer and the product. A MVF mining-flow may build a strong relation between the customer and the product, which indicates some customer has strong possibility to buy some product. A Web usage mining-flow will build a weak relation between the data sources, because Web usage mining usually gives the orientation of a group of users. A uncertain relation indicates so far there is insufficient data to build the relation between the data sources. All the relationships are stored in the transaction database. Data storing and retrieving are deployed on the Grid platform, such as Globus, as a standard Grid service. OGSA-DAI is used to build database access applications (OGSA-DAI Project).

5. Collecting multiple data sources with Web farming

In order to log high-level movements and actions of customers effectively, not only Web mining, but also Web farming should be employed. Web farming focuses on how to design the website and collect the information such as Web usage data effectively for further mining process. Web farming is the systematic refining (or cultivating) of

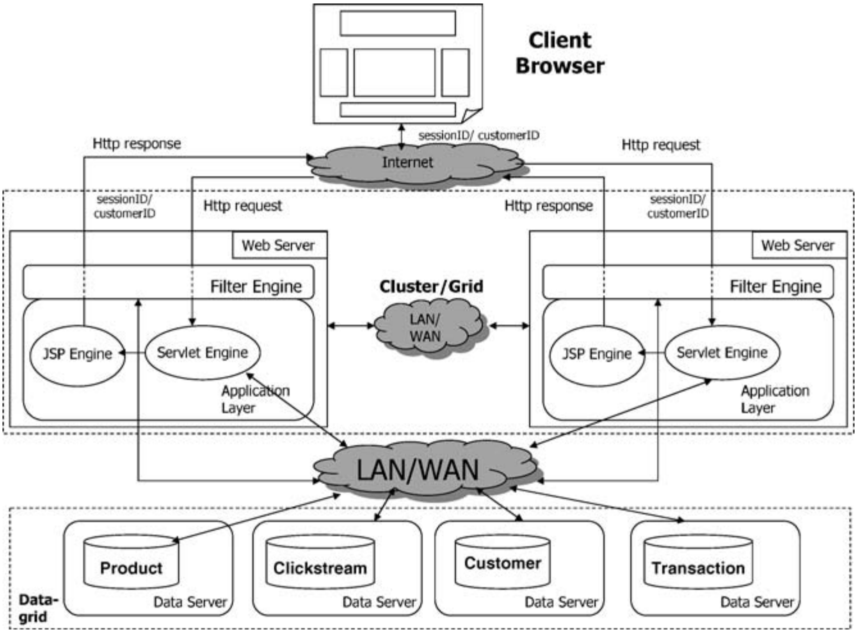

Figure 4. An application of the proposed model in Figure 1.

information resources on the Web to increase the productivity of providing business-relevant information that is valuable to an enterprise (Hackathorn, 1998).

Extending Web mining to Web farming is treated more like a large agricultural business including planting and harvest. Web farming extends Web mining into an evolving breed of information analysis in a whole process of Web-based information management including seeding, breeding, gathering, harvesting, refining, and so on (Zhong, 2004).

5.1. Architecture

An application of the proposed model (in Figure 1) is shown in Figure 4. A big change to enterprise applications in the last few years has been the rise of Web-browser-based user interfaces and Web-enabled services. They bring with them a lot of advantages: no client software to install, a common UI approach, and easy universal access (Fowler, 2003).

Hence, the Web hosting service becomes increasingly complex and important, which is usually developed as a portal. In Figure 4, two Web servers are organized together in LAN or WAN as a cluster for handling the request from clients. The Web server cluster is a popular architecture adopted by enterprises as a way to create scalable and highly available solutions.

Web server's job is to interpret the URL of a request and hand over control to a Web server application. In a Web server cluster, a user's request is automatically transferred between the nodes, and the user is completely unaware of this shift. In this case, any part of Web server log in each server is incomplete. Also, the Web applications deployed

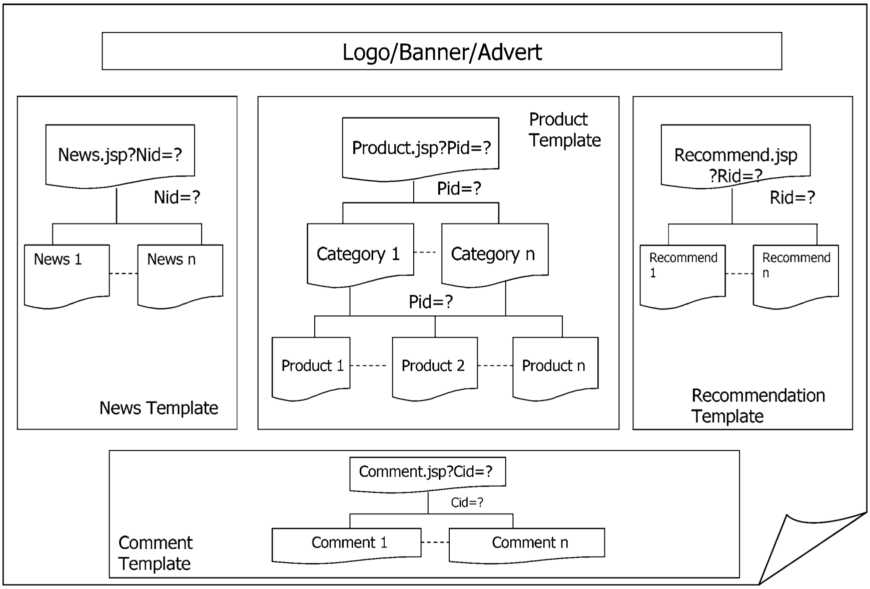

Figure 5. A unified format of a dynamic server page.

on the Web server will recall the data server to perform the business logic. The data server can be distributed in LAN or WAN. In complex portals, Web application servers are independent with Web servers, and can also be organized as an application server cluster for load balancing, failover, and QoS. In Figure 4, enterprise applications are compacted as an application layer deployed in the Web server. Since the application server has to maintain the context of a user's session and related data, the application server is the logical choice for logging user interactions.

A unified format of a dynamic server page is shown in Figure 5. The page is dynamically composed by five parts. A banner or logo is usually placed on the top of a page. In the middle, the product's detail information is displayed. On the left side, the news channel will keep publishing and updating the public information. On the right side, the personal recommendations with respect to a specific product or customer are listed. On the bottom, other customers' reviews and comments with respect to the product being viewed are displayed.

In this case, we do not need to build one page for each product, on the contrary, we only need to maintain four templates, namely *product*, *comment*, *news*, and *recommend*, respectively, for organizing and displaying the related information dynamically for different users in one page. The *product* template is responsible for displaying product information for users, which takes the parameter *Pid* to distinguish different products. On the same time, the *news* template takes *Nid* and the *comment* template takes *Cid* to distinguish the different news and reviews, respectively. The *recommend* template shows the personalized recommendations for different customers.

The *product* template is connected with *comment*, *news*, and *recommend* templates by *Pid*. The presentation is driven by different contents requested. The Web usage, content, and structure data should be considered in a unified way. For example, the following URL is designed to pass the request of *productID = 34* to the servers:

http://server:port/webApp/product.jsp?Pid=34

Although the portal can hide the customers' request in the URL, we will describe that "highlighting" the customers' choices in the URL will help to detach the clickstream module from the business logic and build a general clickstream log module with *filter*.

5.2. Modeling clickstream log

Generally speaking, the content and even the structure are dynamically composed with requests from different users. The website is not page-centric any more, but content-centric. In this situation, Web content, structure, and usage data should be considered in a unified way. Also, as each page is generated dynamically, the proxy or cache server will not work that means we can record the compete user request.

Hence, we can add the request parameter, such as Pid, and customer information, such as customerID or sessionID to the clickstream log:

remotehost rfc931 authuser (*date*) "*request*" *status bytes customer-id click-type click-id*

For example, when user *helen* requested the following URL,

http://www.maebashi-it.org:8088/protal/product.jsp?Pid=99

We got the following clickstream log: *60.43.46.224 - - 08/Dec/2005:15:02:10 -0800]* "*GET /portal/Product.jsp HTTP/1.0*" *200 14912 helen Pid 99*

The format of the clickstream is similar to the common server log format. If the application layer can get the customerID, it will be saved, such as helen; if not, a sessionID will be saved. Also, the action will be saved at the same time, not just the page link. Hence, we can connect the specific information on the Web with the users.

With the *click-type* and *click-id* added, we can log the customers' action on the portal. With the *customer-id* added, we can associate the action with a specific user. Also, we can use this *customer-id* to track the actions across multiple websites.

The most significant part of API 2.3 is the addition of filters — objects that can transform a request or modify a response (Hall, 2001). A filter is a program that runs on the server before any associated servlet or JSP page. Filters are preprocessors of the request before it reaches a servlet, and/or postprocessors of the response leaving a servlet.

5.3. Session tracking

HTTP is a stateless protocol: it provides no way for a server to recognize whether a sequence of requests is all from the same client. The traditional session tracking techniques used by developers are, user authorization, hidden form fields, URL rewriting, and persistent cookies (Hall and Brown, 2001; Hunter, 2001).

We could combine the existing techniques for session tracking and log the user activity as best as we could. First, when a client requests a page from the server, a unique sessionID is generated automatically by the server. This sessionID will be logged with the requested page in the clickstream log, also the sessionID is returned back and saved as a persistent cookie. If the client browser did not receive the cookies, it will automatically transfer to URL rewriting. If the user registered with an ID, the customerID will be used to identify different customers. And the customerID will be logged in clickstream log to replace the sessionID. On the next time, when a registered customer comes back, the server will automatically get his/her information from cookies and associate his/her click with the specific customer.

The disadvantage is that a customer cannot simultaneously maintain more than one session at the same website. If the customer logs in from another location in a session, he/she will automatically logout from the previous site so that one customerID can only login from only one location simultaneously.

The clickstream log acquisition mechanism is shown in Algorithm 1. We can see that steps 5 through 8 of this procedure are repeated as long as the customer has not left the portal, i.e. the customer session is not terminated.

Algorithm 1: Clickstream log acquisition process

```
1. The client sends a request for the first page to the Web server;
2. The filter saves the request in the clickstream log;
3. The Web server randomly assigns a unique sessionID for the client and sends the
   sessionID back with the response;
4. The sessionID is saved as a server cookie for the client automatically;
5. The client sends the request for the next page with the unique sessionID;
6. The filter saves the request and the sessionID in the clickstream log;
7. The request from the client is passed to the Web server;
8. The next page with the contents is uploaded to the client machine.
```

6. Case study 1: Behavior-based online customer segmentation

In this section, we present a case study on behavior-based targeted marketing for illustrating the effectiveness of the proposed architecture stated above. Behavior-based targeted marketing is a Web-based one that is possible for a vendor to personalize his product messages for individual customers at a massive scale on the enterprise portal. The ability to track users' surfing behavior down to individual mouse clicks has brought the vendor and end-customers closer than ever before. Modeling customer groups and performing multi-level promoting are an efficient strategy for personalized recommendation and one-to-one marketing. Broadly speaking, online customers can be divided into the following five categories (Liu et al., 2004):

- Random customers have no strong intention to purchase something, and just wander among pages;
- Rational customers are new to the website, but know clearly what they want and select a direction based on the information of hyperlinks;
- Recurrent customers are familiar with the Web structure, and can find the useful information right away;
- Indirect customers usually buy products for others;
- Key customers are the most valuable customers to a company.

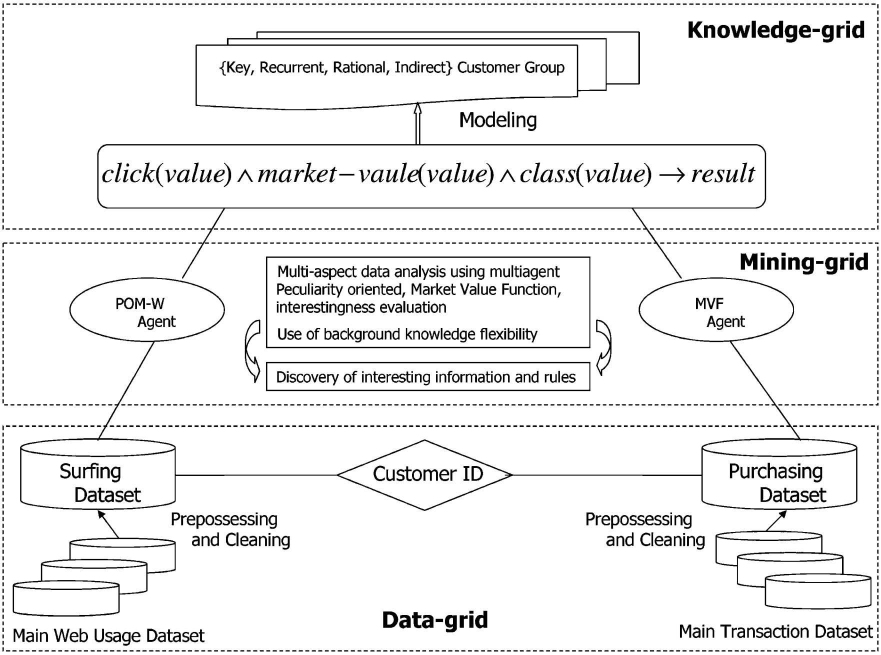
Knowledge-grid
{Key, Recurrent, Rational, Indirect} Customer Group
Modeling
click(value) ∧ market−vaule(value) ∧ class(value) → result
Mining-grid
POM-W Agent
Multi-aspect data analysis using multiagent
Peculiarity oriented, Market Value Function,
interestingness evaluation
Use of background knowledge flexibility
Discovery of interesting information and rules
MVF Agent
Surfing Dataset
Customer ID
Purchasing Dataset
Prepossessing and Cleaning
Prepossessing and Cleaning
Main Web Usage Dataset
Data-grid
Main Transaction Dataset

Figure 6. A framework of the behavior-based targeted marketing data mining, corresponding to the three-layer grid architecture shown in Figure 1.

Our objective is to find the relationship between customers' surfing and purchasing behaviors and divide them into different groups based on different behaviors.

Figure 6 shows a framework of the behavior-based targeted marketing data mining, corresponding to the three-layer grid architecture shown in Figure 1, for performing online customer segmentation and multi-level targeted marketing efficiently and effectively.

First, two kinds of datasets, *surfing and purchasing*, are deployed on the data-grid, and their relationship is connected by the data-flow. The surfing dataset is generated by carefully cleaning the Web usage dataset and only the click number of a customer to a product is picked out. On the other hand, the purchasing dataset is generated from the transaction dataset, which indicates the purchase number of a customer to a product in a time period.

Second, two kinds of mining agents, called POM-W (peculiarity oriented mining in the Web usage) and MVF (targeted market value function), deployed on the mining-grid, are employed to mine the surfing and purchasing datasets, respectively, under the mining-flow management.

Finally, the mined results by the POM-W and MVF agents are combined and refined into more general ones (i.e. combining customer's surfing and purchasing behaviors) to form *active knowledge*, by meta-learning and reasoning with various related knowledge sources on the knowledge-grid.

6.1. *POM-W agent*

As stated in Giudici (2001), an online visitor typically visits only a few different pages of all pages available. This leads to a rather sparse data table when the data are arranged according to all the categories and products corresponding to customers. Previous studies also show that each customer will spend most of his/her time to visit his/her favorite products or categories, which are only a small part of the whole categories listed in the website. Hence, in the surfing dataset, to each product, most of customers will have click numbers in an average level. There is no doubt that the customer who has remarkable click numbers (in other words, peculiar click numbers) to a product usually reveals his/her strong interest on this product. As the data represents a relatively small number of objects and, furthermore, those objects are very different from other objects in a dataset, the POM-W (peculiarity oriented mining in the Web usage) agent is employed (Zhong et al., 2003b).

The basis process is as follows, in the surfing dataset, let $A_1, A_2, \ldots, A_m$ represent different products shown as columns. Let x_{ij} represent the click number of customer i to product j, and n is the number of tuples. The peculiarity of x_{ij} can be evaluated by the *Peculiarity Factor*, $PF(x_{ij})$,

$$PF(x_{ij}) = \sum_{k=1}^{n} N(x_{ij}, x_{kj})^{\alpha} \tag{1}$$

where N denotes the conceptual distance, α is a parameter which can be adjusted by a user, and $\alpha = 0.5$ is used as default.

The peculiarity factor is calculated by the conceptual distances, $N(x_{ij}, x_{kj})$, with the following equation,

$$N(x_{ij}, x_{kj}) = |x_{ij} - x_{kj}| \tag{2}$$

The selection of peculiar data after the evaluation for peculiarity factors is based on a threshold value in Eq. (3),

$$\textit{threshold} \;=\; \textit{mean of } \; PF(x_{ij}) + \beta \times \textit{standard deviation of } PF(x_{ij}) \tag{3}$$

where β can be adjusted by users, and $\beta = 1$ as default. If $PF(x_{ij})$ is over the threshold value, x_{ij} is a peculiar data. The details about peculiarity oriented mining refer to Zhong et al. (2003c).

6.2. *MVF-agent*

Although the POM-W agent stated above is able to help us to find the peculiar customers based on his/her click numbers, we still need to know why this customer is peculiar. Hence, another kind of mining agent called MVF (targeted market value function) is employed for further analysis (Zhong et al., 2004).

Targeted marketing involves the identification of customers having potential market value by studying the customers' characteristics and needs, and selects certain customers to promote. Underlying assumption is that similar type of customers tend to make similar decisions and to choose similar services or products. Formally, an information table is a quadruple:

$$S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$$

where U is a finite nonempty set of objects, At is a finite nonempty set of attributes, V_a is a nonempty set of values for $a \in At$, $I_a : U \rightarrow V_a$ is an information function for $a \in At$. Each information function I_a is a total function that maps an object of U of exactly one value in V_a. An information table represents all available information and knowledge. Objects are only perceived, observed, or measured by using a finite number of properties.

A market value function is a real-valued function from the universe to the set of real numbers, $r : U \rightarrow \Re$. For the targeted marketing problem, a market value function ranks objects according to their potential market values.

A linear market value function is of the form:

$$r(x) = \sum_{a \in At} w_a u_a(I_a(x)) \tag{4}$$

where w_a is the weight of attribute a, and $u_a : V_a \rightarrow \Re$ is a utility function defined on V_a for an attribute $a \in At$. x is the one of the elements in U.

Consider an attribute taking its value from V_a. For $v \in V_a$,

$$u_a(v) = \frac{\Pr(v \mid P)}{\Pr(v)} \tag{5}$$

where $\Pr(v \mid P)$ denotes the probability distribution of attribute value v in P. $\Pr(v)$ denotes the probability distribution of attribute value v in U.

Consider an attribute taking its value from V_a. For $v \in V_a$,

$$\omega_a = \sum_{v} \Pr(v \mid P) \log \frac{\Pr(v \mid P)}{\Pr(v)} \tag{6}$$

For each customer, the purchasing history can be recorded in the transaction dataset and then is transformed to generate the purchasing dataset. Using Eqs. (4)–(6), we can calculate each customer's market value based on each specific product and category. As stated in (Zhong et al., 2004), the MVF agent is effective to sort customers based on some attributes, such as possibility to buy some product. The MVF agent can use not only demographic information, but also the past purchase information of the customers.

6.3. *Learning active knowledge on the knowledge-grid*

The mined results from two different mining agents are stored on the knowledge-grid, respectively, and they are combined and refined into more general ones to form *active knowledge* by meta-learning and reasoning.

For the case study, the rules generated by meta-learning have the following common expression:

$$rule_{no.}\text{: } click(value) \wedge market{-}value(value) \wedge class(value) \rightarrow result.$$

where

$$\begin{aligned}
&click(value) = \{high, low\}\\
&market{-}value(value) = \{high, low\}\\
&class(value) = \{positive, unknown\}\\
&result = \{modeling(value), action(value)\}, \textit{and}\\
&\ modeling(value) = \{key, rational, recurrent, indirect\}\\
&\ action(value) = \{promoting, none\}.
\end{aligned}$$

In the rule expression, *class* parameter indicates whether or not the customer has bought the product or accept the service; for detail information (see, Zhong et al., 2004). For example, if a customer always visits some category and product, and has a higher market value to this category or product, also he/she is in the positive set (i.e., bought this product), we can divide this customer into the key customer group, with the following rule:

$$\begin{aligned}
rule_1\text{: } & click(high) \wedge market{-}value(high) \wedge class(positive)\\
& \rightarrow modeling(key)
\end{aligned}$$

If we have the following rule:

$$\begin{aligned}
rule_2\ \text{:} & click(high) \wedge market - value(high) \wedge class(unknown)\\
& \rightarrow action(promoting)
\end{aligned}$$

which means the customer always visits some product, and also has a higher market value to this product, and he/she has not bought this product yet, we can make a promotion to him/her right now. Furthermore, if various related background knowledge sources are available, more useful *active* rules can be discovered from the primary ones through an explanation-based inductive reasoning process.

Once we obtain behavioral customer segmentation, any existing recommendation algorithm for cross-sell and up-sell can be employed for the targeted groups. Our segmentation is based on the propensity to consume. Hence, the products that the customer already owned should be filtered out to avoid seemingly trivial recommendations. In this case, association rules could be employed to find other related products.

6.4. Experimental evaluation

6.4.1. Dataset description. In our experiments, two related datasets are used, one is a customers' surfing dataset about 13 products visited by 7548 visitors, the other is

Table 1. Part of results from POM-W agent.

ID	p_1	PF(x)	p_2	PF(x)	p_3	PF(x)	...
c_1	1	2019.6	1	2672.9	1	2768.6	...
c_2	1	2019.6	1	2672.9	0	285.1	...
c_3	2	3263.3	1	2672.9	0	285.1	...
c_4	0	1824.4	0	672.2	0	285.1	...
c_5	0	1824.4	0	672.2	0	285.1	...
...	...	...	...		...	...	...
T		2915.5		2179.5		1304.4	...

a transaction dataset. We randomly select 3000 customers from both surfing dataset and transaction dataset for experiments, each customer is described by 22 attributes. The surfing dataset has 3000 rows which indicate the 3000 different customers and 13 columns which indicate the each product available in the shopping website.

Hence, x_{ij} in Eq. (1) indicates the click number of some customer (c_i) to some product (p_j) accumulated in a certain period, not just in a session. To some specific products, most customers have average click numbers. Very interesting customers have higher click numbers. Also, most customers will only visit part of all available products.

6.4.2. Result of peculiarity oriented mining. At first, we find the peculiar data in all columns respectively by using the method stated in Section 6.1. Table 1 shows the result of peculiarity values calculated by Eq. (1). Note that in Table 1 the last tuple T is the threshold values calculated in Eq. (3) and β is 1. Usually, the customers who have higher click numbers to some products are picked up as the peculiar ones.

6.4.3. Results of market value function. The market value can be computed by Eq. (4). We use the following threshold value, which is similar to Eq. (3):

$$\mathit{threshold} = \mathit{mean\ of}\ r(x_{ij}) + \beta \times \mathit{standard\ deviation\ of}\ r(x_{ij}) \tag{7}$$

to determine if the market value is high or low. In Eq. (7), β can be adjusted by a user and $\beta = 1$ is used as default.

Part of results are displayed in Table 2. In Table 2, only the market value of first product (p_1_mv) and the difference between a market value and the corresponding threshold ($p_1\_diff\_mv$) are displayed.

6.4.4. Results of customer segmentation Based on the results shown in Tables 1 and 2, we can get rules for customer segmentation as shown in Table 3.

We can see that 6.1% of 3000 customers is selected as a key customer group for product $p1$. As they have already bought this product, we can recommend the related products to this customer group. As this group is the most valuable group to a company, the detail behavior of each customer should be investigated carefully, such as visiting frequency, attachment time to each product, so that we can tailor the personalized service to each customer. As shown in Table 3, 10.7% of all customers are rational customers and 3.0% are recurrent one.

Table 2. Part of results from MVF agent.

ID	p_1*_mv*	p_1*_diff_mv*	...
c_1	0.0981002	−0.0018998	...
c_2	0.0980301	−0.0019699	...
c_3	0.0980301	−0.0019699	...
c_4	0.0758767	−0.0241233	...
c_5	0.0753944	−0.0246056	...
...	...	...	...
T	0.1		...

Table 3. Discovered rules for customer segmentation.

ID	*rule*	No.	Ratio
r_1	*click*(*high*) ∧ *market-value*(*high*) ∧ *class*(*positive*) → *modeling*(*key*)	183	6.1 %
r_2	*click*(*high*) ∧ *market-value*(*high*) ∧ *class*(*unkown*) → *action*(*promoting*)	0	0
r_3	*click*(*high*)∧*market-value*(*low*) ∧ *class*(*positive*) → *modeling*(*rational*)	320	10.7 %
r_4	*click*(*high*) ∧ *market-value*(*low*) ∧ *class*(*unkown*) → *action*(*promoting*)	0	0
r_5	*click*(*low*) ∧ *market-value*(*high*) ∧ *class*(*positive*) → *modeling*(*recurrent*)	91	3.0 %
r_6	*click*(*low*) ∧ *market-value*(*high*) ∧ *class*(*unkown*) → *action*(*promoting*)	38	1.3 %
r_7	*click*(*low*) ∧ *market-value*(*low*) ∧ *class*(*positive*) → *modeling*(*indirect*)	964	32.1 %
r_8	*click*(*low*) ∧ *market-value*(*low*) ∧ *class*(*unkown*) → *action*(*none*)	1404	46.8 %

7. Case Study 2: Data quality analysis

The other key question in the mining-grid is to determine when and how to start the mining processes. We use data quality analysis to solve this kind of problem. Hence, a mining-flow will start in two situations, the first is that it is invoked by a knowledge-flow from the knowledge-grid; the other is that the data quality is good enough.

Multiple data sources are prepared for a mining-flow. Some data mining services will be invoked, after that, the generated knowledge will be stored in *Mined Results* database for further use (see Figure 1). Different methods of data quality analysis are needed for different mining methods. Data quality analyzing services are deployed on the mining-grid, which can keep watching the data automatically. The mining-flow will not be invoked until the mining status is satisfied.

Figure 7 shows a data quality analysis flow for the MVF agent. The MVF mining-flow will not be started until we have enough data in positive and negative datasets, respectively. If we want to know who is interested in our portal. A data quality analyzing service will keep watching the customer-related datasets, until the following conditions is satisfied.

- We have enough members in the customer dataset U,
- We have enough registered customers P,
- We have enough data in the negative dataset D.

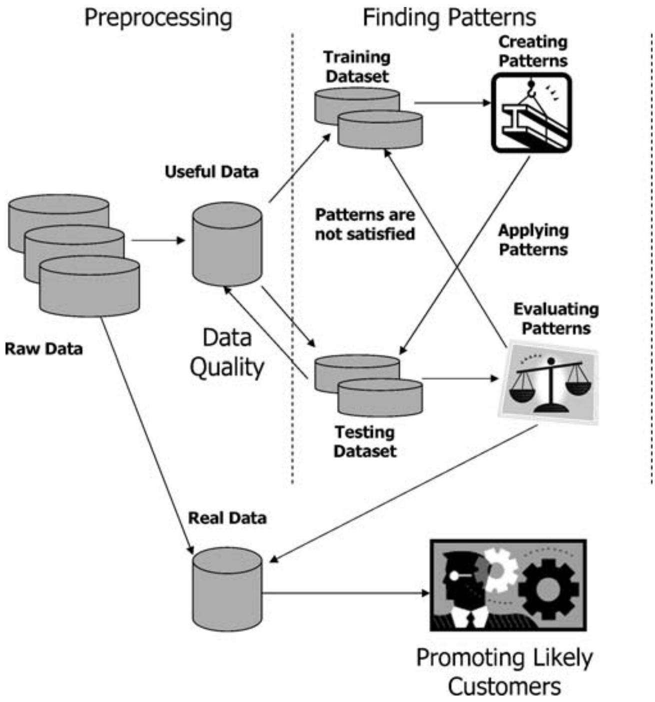

Figure 7. The data quality analysis flow for the MVF agent.

7.1. The lift value

The lift index is used as the evaluation criterion (Ling and Li, 1998), since it has a better intuitive meaning for the targeted marketing problem and it has a nice property that it is independent to the number of the responders. After all testing examples are ranked using the MVF agent; we divide the ranked list into 10 equal deciles, and see how the original responders distribute in the 10 deciles. If regularities are found, we will see more responders in the top deciles than the bottom deciles.

Let S_i denote how many positive examples are in the ith deciles. The lift index is defined as follows,

$$S_{lift} = (1.0 \times S_1 + 0.9 \times S_2 + \cdots + 0.1 \times S_{10}) \Big/ \sum_{i=1}^{10} S_i$$

If the distribution of the buyers in the 10 deciles is random (no regularity is found), then the S_{lift} would be around 50%. In the best situation when $S_1 = \sum_i S_i < 10\%$, $S_{lift} = 100\%$. In the worse case when $S_{10} = \sum_i S_i$ (and rest $S_i = 0$), $S_{lift} = 10\%$.

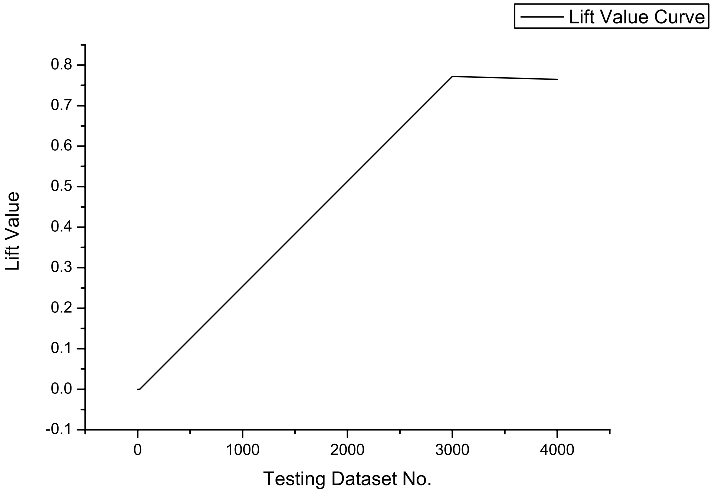

Figure 8. Lift value influenced by testing dataset no.

7.2. Selection of testing dataset

Figure 8 shows the relationship between the lift value and the number of instances selected for testing dataset.

From the ever beginning, the lift value will increase very fast when the number of testing dataset increases. After that, the increasing will begin to go smoothly until the point the lift value will decrease, even the number of the testing dataset keeps growing. The data quality analysis flow will monitor the slope of this curve, such as

$$Slope(Lift) = (Lift(n) - Lift(n-1))/(No.(n) - No.(n-1)) \tag{8}$$

When the slope of the curve reach zero, we can use the pattern for building the model of the dataset. In other words, at this time, the instances are enough to describe the pattern of the raw dataset. When the pattern of the raw dataset changes, the curve will change also and the data quality analysis flow will start again to calculate the new pattern.

7.3. Attribute selection

Besides the number of the testing dataset, the selected attributes for calculating market value and the distribution of attribute values will also influence the model. In the above situation, the mining process is almost the same, the difference is the data quality items. With the size of databases growing rapidly, data dimensionality reduction becomes an important factor in building these models, in other words, if these models can be estimated by part of attributes, labor and communication costs for data collection could be reduced dramatically. For different mining methods, corresponding data quality services should be developed.

We will apply attribute reduction algorithms based on rough set theory and the entropy in information theory to the market value function model in order to select attributes during estimating attribute weights. For the attribute weights in Eq. (6), it can be drawn from information-theoretic measures such as the Information Gain:

$$\omega_a = H(\{a\}) - H(\{a\}|D) \tag{9}$$

where $H(\{a\})$ denotes the entropy value of the attribute a in U, and $H(\{a\}|D)$ denotes the conditional entropy value of the attribute a given by D.

Hence, it is reasonable to apply attribute reduction algorithms based on rough set theory and information theory to reducing attributes, while estimating weights of attributes, according to CEBARKNC, Eq. (9) can be modified to the following equation:

$$\omega_a = \begin{cases} H(\{a\}) - H(\{a\}|D), & \text{if } a \in B \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where B is the relative reduction. It is obvious that weight of an attribute in B calculated by Eq. (10) equals to Eq. (9). Eq. (10) removes unimportant attributes.

In boosting, weights are assigned to each training example. A series of hypothesis is learned. After a hypothesis is learned, the weights are updated to allow the subsequent hypothesis to pay more attention to the examples misclassified by the previous hypothesis. There are thousands of data in a database. The CEBARKNC needs to spend much time to compute the conditional entropy. Hence, sampling data is necessary. We can assign weights to each training example through the boosting method and then select examples with higher weights to build market value function model (Huang et al., 2003).

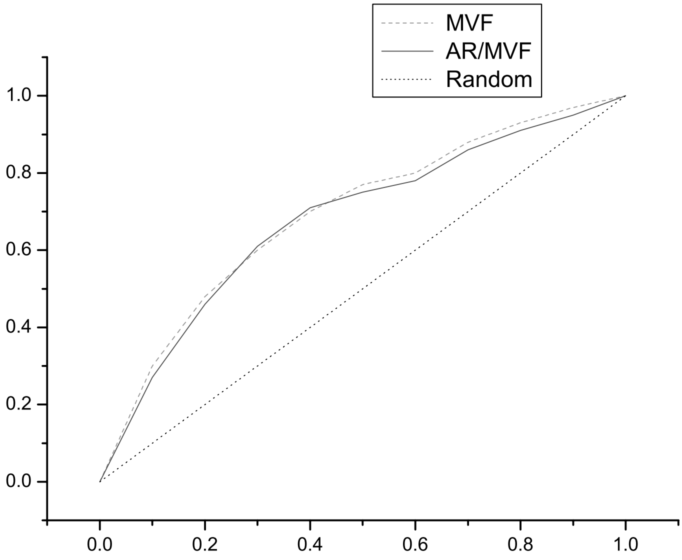

Figure 9. Cumulative lift curve.

The result of attribute reduction (AR) and the normal market value function model is shown in Figure 9. We can see that the model can dramatically reduce the attribute number. For different mining methods, data quality analysis will be totally different. Hence, developing appropriate and effective data quality methods is also a challenge for the mining-grid.

8. Concluding remarks

The paper presented a conceptual model with dynamic multi-level workflows corresponding to a multi-layer grid architecture, for multi-aspect analysis in distributed, multiple data sources, and for dynamically organizing status-based business processes. We illustrate how to use the multi-layer grid conceptual model for building an e-business portal.

As a case study, we presented the first step of multi-database mining based on the proposed model. In this approach, we tried to find the relationship between the customer's surfing and purchasing behaviors, and performing multi-level marketing strategies for targeted customer groups. We showed that collecting the customer-related data carefully and building a holistic customer profile will help to learn the customer behavior effectively. By following this way, the enterprise will perform more personal recommendation and promotion.

In comparison with the related work (Berman et al., 2003; Foster and Kesselman, 2003a; Nabrzyski et al., 2004; Preuner and Schrefl, 2000, 2003; Priebe and Pernul, 2003) the most novel features of our approach are such that,

- It is based on the paradigm of the Wisdom Web based computing in which Artificial Intelligence (AI) (e.g., knowledge discovery and data mining, intelligent agents, knowledge representation, planning, and social intelligence) and advanced Information Technology (IT) (e.g., data/knowledge grids, ubiquitous computing, and workflow) are incorporated to make a reality of e-business portals.
- It is one of the first attempts to build a domain-independent multi-layer grid architecture in which various data mining agents are organized on the mining-grid to connect the data-grid and the knowledge-grid for business intelligence.

Here we would like to emphasize that how to manage, analyze, and use the information intelligently from different data sources is a problem not only exists in the e-business field, but also exists in e-science, e-learning, e-government, and all intelligent Web information systems (Zhong, 2004). The development of an enterprise portal on the Wisdom Web and Knowledge Grids is a good example for trying to solve this problem (Liu, 2003).

Note

1. Charles H. Spurgeon, English preacher of 19th century 1834–1892.

References

Alonso, G., Casati, F., Kuno, H., and Machiraju, V. 2004. Enterprise Application Integration, Web Services — Concepts, Architectures and Applications, Springer, pp. 67–92.

Berman, F., Fox, G., and Hey, A.J.G. (Eds.) 2003. Grid Computing: Making the Global Infrastructure a Reality. John Wiley & Sons.

Buschmann, F. et al. 1996. Pattern-Oriented Software Architecture: A System of Patterns. Wiley.

Cannataro, M., Congiusta, A., Mastroianni, C., Pugliese, A., Talia, D., and Trunfio, P. 2004. Grid-based data mining and knowledge discovery. In N. Zhong and J. Liu (Eds.), Intelligent Technologies for Information Analysis. Springer-Verlag, pp. 19–45.

Congiusta, A., Pugliese, A., Talia, D., and Trunfio, P. 2003. Designing grid services for distributed knowledge discovery. Web Intelligence and Agent Systems: An International Journal, 1:91–104.

Curcin, V., Ghanem, M., Guo, Y., Köhler, M., Rowe, A., Syed, J., and Wendel, P. 2002. Discovery net: towards a grid of knowledge discovery. Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 658–663.

Deelman, E., Blythe, J., Gil, Y., and Kesselman, C. 2003. Workflow management in GriPhyN. In J. Nabrzyski et al. (Eds.) Grid Resource Management. Kluwer Academic Publishers, pp. 99–116.

Detlor, B. 2004. Towards Knowledge Portals: From Human Issues to Intelligent Agents, Information Science and Knowledge Management. Kluwer Academic Publishers.

Foster, I. and Kesselman, C. 1997. Globus: A metacomputing infrastructure toolkit. The International Journal of Supercomputer Applications and High Performance Computing, 11(2):115–128.

Foster, I. and Kesselman, C. 1999. The Grid: Blueprint for a Future Computing Infrastructure, 1st edition. Morgan Kaufmann.

Foster, I. and Kesselman, C. 2003a. The Grid: Blueprint for a Future Computing Infrastructure, 2nd edition. Morgan Kaufmann.

Foster, I. and Kesselman, C. 2003b. Concepts and architecture. In I. Foster and C. Kesselman (Eds.) The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, pp. 37–64.

Foster, I., Kesselman, C., and Tuecke, S. 2001. The anatomy of the grid: Enabling scalable virtual organization. International Journal of High Performance Computing Application, 15(3):200–222.

Fowler, M. 2003. Patterns of Enterprise Application Architecture. Addison Wesley Professional.

Gil, Y., Deelman, E., Blythe, J., Kesselman, C., and Tangmunarunkit, H. 2004. Artificial intelligence and grids: Workflow planning and beyond. IEEE Intelligent Systems, Special Issue on e-Science, 19(1):26–33.

Giudici, P. 2001. Association models for web mining. Data Mining and Knowledge Discovery, 5:183–196.

Hackathorn, R.D. 1998. Web Farming for the Data Warehouse. Morgan Kaufmann.

Hall, M. 2001. More Servlets and JavaServer Pages. Sun Microsystems Press.

Hall, M. and Brown, L. 2001. Core Web Programming, 2nd edition Sun Microsystems Press.

Han, J.W. and Chang, K. Ch. 2002. Data mining for web intelligence. IEEE Computer, 35(11):64–70, 2002.

Hu, J. and Zhong, N. 2004. Organizing dynamic multi-level workflows on multi-layer grids for developing e-business portals. Proc. 2004 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 777–778.

Hu, J. and Zhong, N. 2005. Developing e-business portals with dynamic multi-level workflows on the multi-layer grid. Proc. 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 196–201.

Huang, J.J., Liu, Ch. N., Ou, Ch. X., Zhong, N., and Yao, Y. Y. 2003. Attribute reduction of rough sets in mining market value functions. Proc. 2003 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 470–473.

Hunter, J. 2001. Java Servlet Programming, 2nd edition O'Reilly.

Ling, Ch. X. and Li, Ch. H. 1998. Data mining for direct marketing: Problems and solutions. Proc. 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 73–79.

Liu, J. 2003. Web intelligence (WI): What makes wisdom web?. Proc. 18th International Joint Conference on Artificial Intelligence, pp. 1596–1601.

Liu, J., Zhang, Sh. W., and Yang, J. 2004. Characterizing web usage regularities with information foraging agents. IEEE Transactions on Knowledge and Data Engineering, 16(5):566–584.

Maier, R. 2004. Knowledge Management Systems: Information and Communication Technologies for Knowledge Management, 2nd edition. Springer-Verlag.

Nabrzyski, J., Schopf, J.M., and Weglarz, J. 2004. Grid Resource Management: State of the Art and Future Trends. Kluwer Publishing.

Preuner, G. and Schrefl, M. 2000. A three-level schema architecture for the conceptual design of web-based information systems. *World Wide Web*, 3(2):125–138.

Preuner, G. and Schrefl, M. 2003. Integration of web services into workflows through a multi-level schema architecture. Proc. 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, pp. 51–60.

Priebe, T. and Pernul, G. 2003. Towards integrative enterprise knowledge portals. Proc. 12th International Conference on Information and Knowledge Management, pp. 216–223.

Pyle, D. 2003. Business Modeling and Data Mining. Morgan Kaufmann.

Stork, H.G. 2002. Webs, grids and knowledge spaces: Programmes, projects and prospects. Journal of Universal Computer Science, 8(9):848–867.

The OGSA-DAI project: http://www.ogsadai.org.uk/.

Tomita, K., Zhong, N., and Yamauchi, H. 2004. Coupling global semantic web with local information sources for problem solving. Proc. 1st International Workshop on Semantic Web Mining and Reasoning, pp. 66–74.

Wege, C. 2002. Portal server technology. IEEE Internet Computing, 6(3):73–77.

Wu, X. and Zhang, S. 2003. Synthesizing high-frequency rules from different data sources. IEEE Transactions on Knowledge and Data Engineering, 15(2):353–367.

Xu, M., Hu, Zh. H., Long, W.H., and Liu, W. 2003. Service virtualization: Infrastructure and applications. I. Foster and C. Kesselman (Eds.) The Grid: Blueprint for a Future Computing Infrastructure, 2nd. Morgan Kaufmann, pp. 179–189.

Zhang S., Wu, X., and Zhang, C. 2003. Multi-database mining. IEEE Computational Intelligence Bulletin, 2(1):5–13.

Zhong, N., Liu, Ch. N., and Ohsuga, S. 2001. Dynamically organizing KDD processes. International Journal of Pattern Recognition and Artificial Intelligence, *World Scientific*, 15(3):451–473.

Zhong, N. 2004. Developing intelligent portals by using WI technologies. J.P. Li et al. (Eds.) Wavelet Analysis and Its Applications, and Active Media Technology. World Scientific, 2, pp. 555–567.

Zhong, N., Hu, J., and Motomura, S. 2005. Building a data mining grid for multiple human brain data analysis. Computational Intelligence, An International Journal, 21(2):177–196.

Zhong, N. and Liu, J. 2004. The alchemy of intelligent IT (iIT): Blueprint for future of information technology. N. Zhong and J. Liu (Eds.) Intelligent Technologies for Information Analysis, Springer Monograph, pp. 1–16.

Zhong, N., Ohara, H., Iwasaki, T., and Yao, Y.Y. 2003a. Using WI technology to develop intelligent enterprise portals. Proc. International Workshop on Applications, Products and Services of Web-based Support Systems, pp. 83–90.

Zhong, N., Yao, Y.Y., Liu, Ch. N., Ou, Ch. X., and Huang, J.J. 2004. Data mining for targeted marketing. N. Zhong and J. Liu (Eds.) Intelligent Technologies for Information Analysis, Springer-Verlag, pp. 109–131.

Zhong, N., Yao, Y.Y., and Ohshima M. 2003b. Peculiarity oriented multidatabase mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):952–960.