Market Intelligence Portal: An entity-based system for managing market intelligence

by Z. Su
J. Jiang
T. Liu
G. T. Xie
Y. Pan

Gathering market intelligence (MI) can be critical for the success of an enterprise, and in-depth analysis of this information is helpful in understanding products, business trends, and information concerning competitors. Because of the huge volume of this information and the high rate of its growth, there is a great demand from enterprises for automated MI management systems. In this paper, an information portal for MI management systems is presented. It is based on a semantic web approach and provides the user a single access point for all relevant information about a specific MI topic. In contrast with traditional knowledge portal methods, our work is based on entity-level computing technologies rather than document-level technologies. We present a knowledge network called EntityNet, which is generated by an information synthesis process and enables the network to store entity-level knowledge and provide semantic services for the knowledge worker. From the systems perspective, EntityNet's most significant features are its customized document-processing flow and personalized category view. The framework can integrate various text collections, apply data-mining and dissemination functions on the collections with a defined process flow, and present a personalized browsing and searching interface.

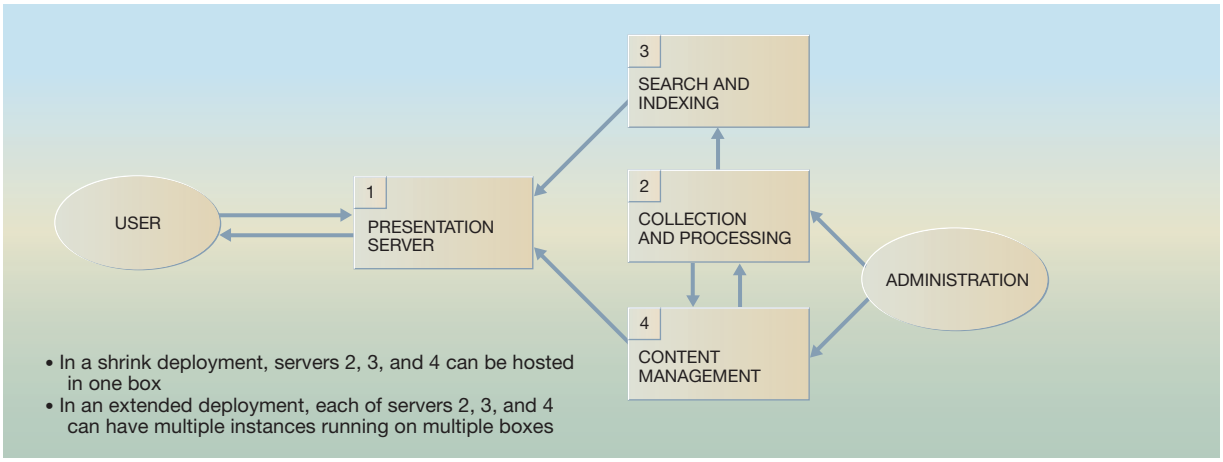
Enterprise executives often need to integrate market intelligence information from various sources, but its current volume impedes their manual efforts. Moreover, with the appearance of efficient information digitalization methods and rapid expansion of the Internet, the quantity of the digital data available has dramatically increased. For these reasons, automated tools are needed to assist in the entire process of retrieving appropriate documents, extracting useful information from within the documents, and analyzing the results. These automated tools are defined as market intelligence (MI) systems and are critical in the conduct of business by enterprise customers.

During the past decade, many efforts have been made in this field. Generally, MI research and system development efforts have focused on storage and data-mining technologies. Data warehousing and on-line analytical processing (OLAP) have typically been used to solve data extraction, transformation, data cleaning, storage, and mining issues.[1] Previous efforts have used document-based technologies and supported document-level functions such as full text search, document classification, and so on.

In this paper, a system for market intelligence management, the Market Intelligence Portal (MIP), is presented. It is an information management system which can automatically collect daily market information from various sources, such as Web sites, file systems, mail servers, and so on. Once the data have

0018-8670/04/$5.00 © 2004 IBM

Figure 1 MIP system architecture



been collected, useful information is automatically extracted from the documents and then organized according to users' requirements. The system also provides a Web portal to publish the processed information, where a user can customize his or her own category design and visualization method.

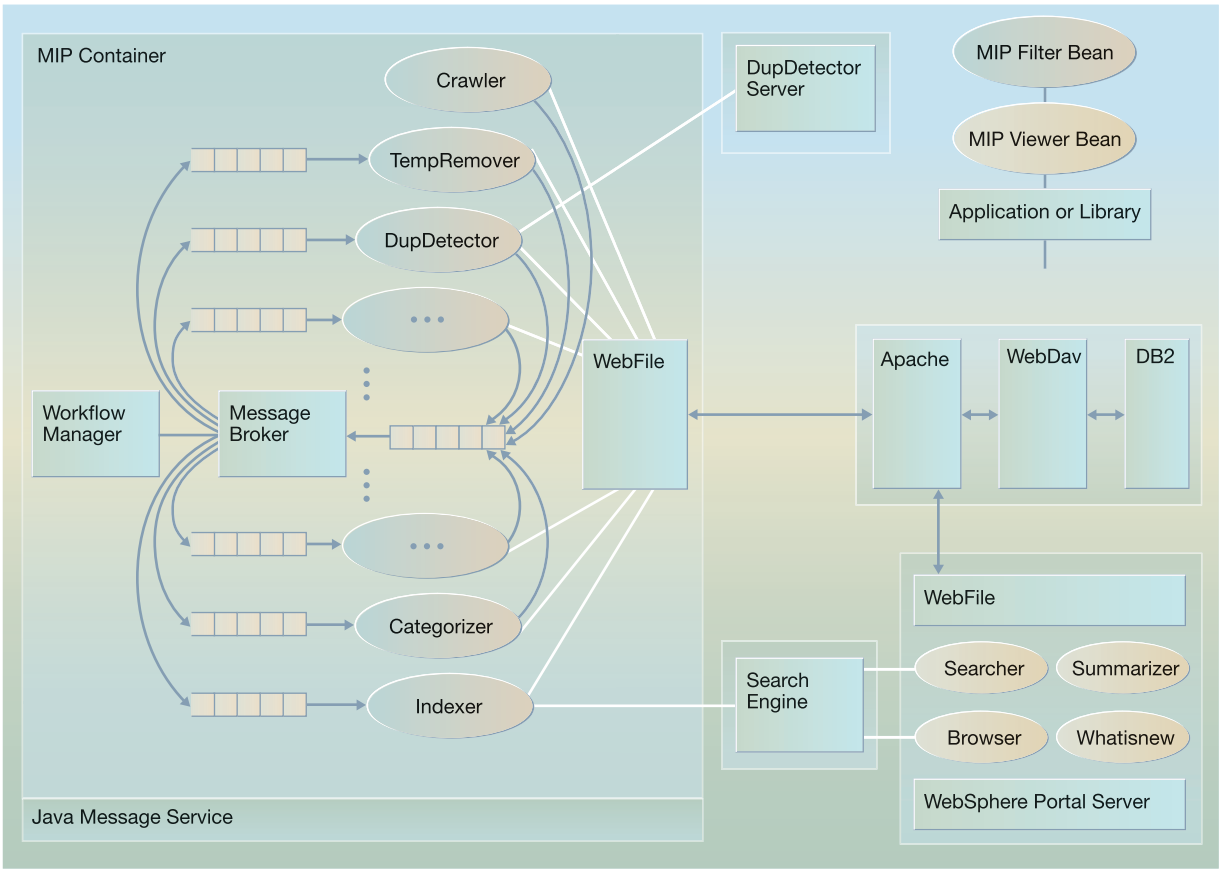
Figure 1 shows the system architecture of MIP. As shown, there are four subsystems. At the client side, a presentation server generates the Web pages according to the user's defined category structure. A user can define his or her own category organization and thus perform browsing and searching operations on the document repository, based on this organization. The technology behind this personalized service is a flat classification scheme, which regardless of the user's personalized category design, requires only the system classification model to be trained and updated. All personalized classifications can be generated based on this classification model at very low cost. Because only one classification model needs to be trained, this framework has a big advantage in reducing computing and storage costs. A user can perform searching and browsing functions on his or her document directory corresponding to the defined category tree. The search and indexing subsystem provides a keyword search service on categories through the presentation server.

A salient feature of MIP is that it supports customized document-processing flow. This is achieved by two subsystems on the server side: the collection and processing module and the content management module. Customized document-processing flow allows any system administrator to specify any data source he is interested in and ask the MIP system to crawl the documents from those sources according to a user-defined schedule. At the same time, the system administrator can also define the steps used in filtering the crawled documents by applying different kinds of "miners" in a given sequence and configuration. The mining sequence and configuration data are stored in the workflow manager of the collection and processing module. The original crawled documents and the meta-data that is the output of the miners is stored in the content management module.

A "message broker" assists in the communication between different MIP miners. All MIP miners are located in the collection and processing module. Several miners have been implemented and are ready for use, including a crawler, a template remover, and engines for duplicate and similarity detection, summarization, and categorization. These miners use the WebFile application programming interface (API), an interface we defined based on the WebDAV[2] protocol, to communicate with the content management module. Figure 2 shows the software architecture of the server-side subsystems.

Generally, three steps are involved in this framework. First, all relevant data related to the specific MI topic are collected (or "crawled") from different places, such as Web sites, databases, or file systems. Sec-

Figure 2 Software architecture of MIP server-side subsystem



ondly, the crawled pieces of raw data are imported into a single data repository. By applying data-mining and information-extraction technologies, useful information from the raw data is generated and mapped onto a predefined taxonomy. Finally searching and browsing services on the results are provided to end users.

In the following sections, we introduce related work and the methodology for internal knowledge representation in MIP. The individual components of the framework are then discussed in detail, followed by system implementation descriptions and conclusions.

Background

The Unstructured Information Management Architecture[3] (UIMA) is an architecture developed at the IBM Watson Research Center and provides a common framework for the integration of unstructured information management technologies. It specifies an architecture and framework for developing, describing, and composing text analysis engines (TAEs) and advanced search capabilities. MIP uses UIMA as the platform for unstructured document processing engines. As mentioned in the previous section, we designed a customized document-processing flow in MIP, and this workflow engine can integrate various information-processing algorithms and engines including engines packaged as UIMA TAEs.[4] We also developed a set of Chinese and English processing engines in MIP which are packaged as TAEs to satisfy common use cases. In the section "Information synthesis," we describe those engines in detail.

WebFountain*[5] is another related project that combines advanced text analytics technologies, vast data sources, and key applications to offer customers the ability to quickly discover business insights and transform their processes, strategies, and decision mak-

ing based on the extracted insight information. Through the analysis of billions of Web pages, bulletin board postings, Web logs, and other related open or licensed feeds and proprietary data, it discovers emerging trends, unexpected relationships, and complex patterns that are costly to acquire by manual methods. In contrast with the service-oriented business model for WebFountain, MIP is a solution that can be installed on a customer site and that supports user customization. It provides fewer analytic tools than WebFountain, focusing instead on providing rich user-interface features to assist in browsing and searching a large document repository.

The IBM WebSphere** Portal[6] (WSP) includes a tool for building an enterprise taxonomy with the goal of high accuracy and high coverage for very large document collections with highly diverse styles and quality of writing. It is represented as an element in a portal and document management system for corporate customers and has been adopted by IBM for internal enterprise document management. MIP focuses on the whole process from collections to browsing, with the taxonomy subsystem as only one element of the system. Also, the MIP taxonomy subsystem provides a personalized categorization framework for end users, which is not the focus in the categorization feature of the WSP.

From a data-mining perspective, related work includes that of TEMIS,[7] an IBM partner, which develops text-mining technologies for the products Online Miner and ID-Insight Discoverer** and provides solutions for customers. The Online Analyst project[8] also offers an intelligent agent to assist end users to read and quickly analyze huge volumes of documents retrieved online. Both projects have significantly advanced the field of mining for information integration. In contrast with their approaches, MIP is focused on a rich user interface for browsing and the process of knowledge collection.

The unique features of MIP are its use of highly detailed text analysis (such as entity and relationship extraction) in real applications, its customized workflow for the processing of unstructured data in documents, and our patented flat classification scheme to support high-efficiency personalized categorization.

In the following section, we introduce the knowledge representation scheme used in MIP. Following this, details of the components of the MIP framework are presented.

Knowledge representation in MIP

As we have mentioned, the functionalities of the MIP system are entity-based. Aside from the document-level operations, MIP combines entities and entity-relation-extraction technology with semantic web activities, and generates a semantic network structure to store knowledge. In this section, we introduce the knowledge extraction process and its representation scheme in MIP.

Information synthesis. Our work uses the semantic web architecture and constitutes an automated process for retrieving appropriate documents, extracting useful data about market intelligence, and finally making the processed data available at the portal server. We refer to this process as information synthesis (IS), and it is illustrated in Figure 3.

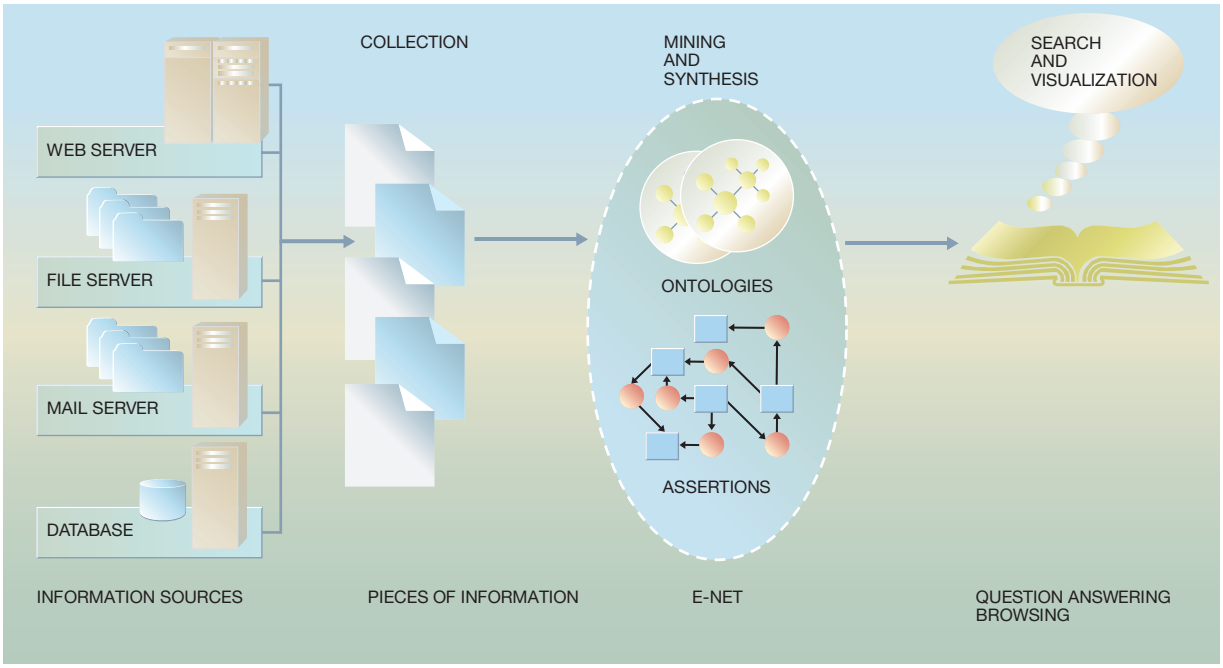
According to our definition, an IS process can be divided into three steps:

1) Data crawling from different sources
2) Analysis and integration of the extracted data into a specific ontology schema
3) A unified information access interface and services for end users based on this ontology

The first step is one of data collection, in which documents are collected from different sources. The information source can be an e-mail server or a Web site. Later, the collected documents will be cleaned and filtered by many engines in a user-defined sequence. The detailed process of data collection and filtering is introduced in the section, "System implementation."

Among the document-processing engines, there is an entity and entity-relation-extraction engine to extract entities and relationships between those entities from documents. Entity and entity-relation extraction is a natural language processing (NLP) technique that extracts two concepts and their relationship to each other from a sentence. Its task is the same as extracting a subject-verb-object from a sentence, labeling the type of verb in order to determine the kind of relationship the subject and object nouns have to each other, and then storing it in a database for question answering or text mining. Because this is a powerful analysis tool, it has been used in intelligence applications as well as in CRM (customer relationship management), where it is being applied today in some advanced question-answering systems. Representative systems and products on the market are

Figure 3 Data flow for information synthesis



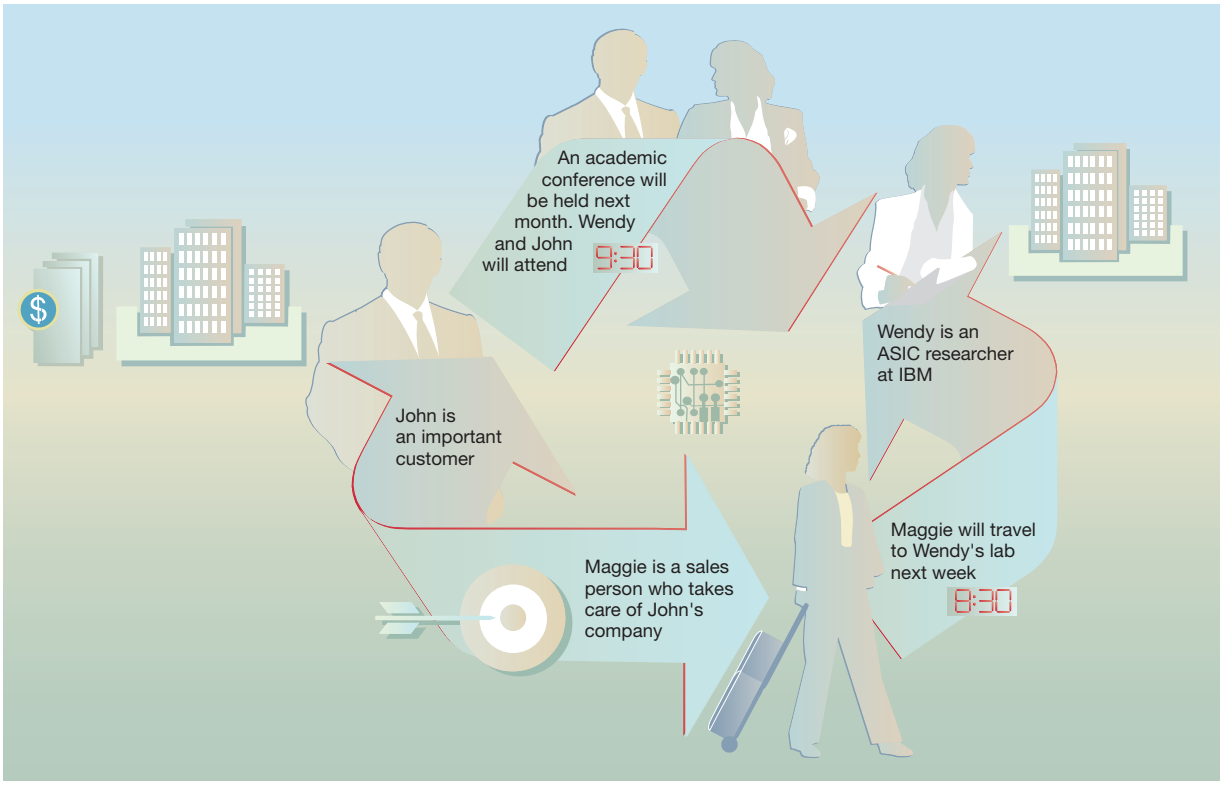
eQuery[9] and Languistics.[10] In MIP, we have used a statistical learning method to perform this task,[11] and research into this function is ongoing.

EntityNet. Once the entities and entity relationships have been extracted, they will be mapped onto a predefined domain-specific ontology by domain experts. This ontology may be different in different domains. As shown in Figure 3, we integrate the extracted data (knowledge) into an ontology-supported semantic network. In our framework, this semantic network is called EntityNet or E-Net. By using entity and entity-relationship extraction and other NLP methods, we can extract entities from unstructured information sources and find the relationships between those entities. They may also be extracted directly from structured information sources such as databases. Thus, all elements stored in E-Net are entities, and this enables it to support inference for question-answering systems and data schema for semantic search.

As a low-level data representation scheme, we propose the use of the Resource Description Framework[12] (RDF). The objective of RDF is to support the interoperability of meta-data. RDF allows descriptions of Web resources to be made available in machine-understandable form. A Web resource is any object with a Uniform Resource Identifier (URI) as its address. This enables the semantics of objects to be expressible and exploitable. Once it is widely deployed, RDF will enable services to develop processing rules for automated decision-making about Web resources. As summarized in *The New Review of Information Networking,* "RDF is based on a concrete formal model utilizing directed graphs that allude to the semantics of resource description. The basic concept is that a Resource is described through a collection of Properties called an RDF Description. Each of these Properties has a Property Type and Value."[12] Any resource can be described with RDF as long as the resource is identifiable with a URI. We chose RDF as the information-synthesis data model to store the E-Net and the association operations based on it.

At the client side, MIP provides a portal for end users where search and browsing services are provided. Besides the traditional services such as full-text searching, classification, and summarizing, due to the storage of knowledge in E-net, MIP can provide more services, including semantic searching and browsing.

Figure 4 Example of an E-net



To visualize the data, in addition to the traditional directory structure view, we are working on using Scalable Vector Graphics[13] (SVG) to display the entities and their relationships in two-dimensional space. This graphic user interface will be very helpful for users, assisting in browsing the dataset and making it easily understandable.

Figure 4 shows an example demonstrating how the concept of E-net supports semantic searching and browsing. In this example, John is an important customer of Maggie, an IBM salesperson. Maggie is investigating ways to impact John's area, ASIC (application specific integrated circuit) design. With E-Net, Maggie discovers that Wendy of IBM Research is a recognized scientist in ASIC design. Most importantly, she finds that Wendy will attend an academic conference next month in which John will also participate. Maggie then learns that when she travels to a company near Wendy's lab next week, she can meet with Wendy and ask her to have a personal talk with John during the conference. As we can see from the figure, the entities are linked with each other, and these links can be very helpful for semantic searching (inference) and data visualization.

Components of MIP

In this section, the component technologies of MIP are presented in detail. From the perspective of system architecture, MIP is a classical client-server system. As mentioned in the introduction, it has two novel features: its personalized category service at the client side, and its customized workflow for document process at the server side.

Customized workflow. MIP provides the system administrator with the option to customize workflows for document processing. The workflow is a sequence of operations on documents. Usually, the workflow begins with a crawler. It could be a Web page crawler or another kind of crawler such as a Domino crawler or file-system crawler. This crawler will "crawl" documents from locations specified in the configuration

designed by the administrator. Subsequently, many document process engines will analyze those crawled documents according to the specified sequence and configuration. For example, as a first step, a Web page will usually be filtered by a template remover engine so that irrelevant data such as advertisement links in a news page will be discarded. Then the filtered page will be checked by a duplicate detection engine, and so on. The following are a set of engines we have developed for MIP.

Duplicate detection engine. This is used for determining if a document's content is duplicated in other existing documents at a database. One scenario leading to duplication results from the "mirrors" that many Web sites have. The contents of the original site are duplicated on the mirror sites. Another scenario occurs when a document is modified slightly and published in another place. It is desirable to remove such duplicated information during browsing or searching. In MIP, we use a histogram-based measure of similarity of word distribution in documents to determine the similarity of two documents and remove redundant information. A complex index method has also been applied to improve the speed of the duplicate detection process for large collections of documents.

Template remover engine. The task of this engine is to remove unrelated contents from Web pages, such as advertisement links and menu icons in news pages. After this engine removes this content, the text information of a document can be extracted.

Format conversion engine. The task of this engine is to convert different document formats into a unified format. In MIP, all document formats are converted to plain text.

Entity extraction and mapping engine. This engine is used to extract named entities from the documents. The extracted entities are mapped onto a user-defined ontology for building E-Net.

Classification engine. This is a Support Vector Machine[14] (SVM) engine, which is a binary classification engine widely used in language processing. Based on this engine and the flat classification method introduced in the subsection "Flat classification scheme," MIP provides a personalized classification service for the user.

Clustering engine. This engine can perform unsupervised clustering analysis on documents.[15] Basically, the clustering engine can automatically classify the documents into a number of clusters based on features automatically extracted from those documents.

As mentioned before, the workflow module can manage and configure these engines. The workflow module has three components: the workflow manager, the engine deployment descriptor, and the message broker.

Workflow manager. The workflow manager can be considered as a container for the various functional engines described in the previous section. The workflow manager provides the system administrator with an interface to define his personal workflow by combining and configuring a set of functional engines. MIP is designed to be deployed in an environment with various data resources (Internet or Intranet) and formats (HTML [Hypertext Markup Language], Word** documents, PowerPoint** documents, and PDF [Portable Document Format]). Users can use the workflow manager to define many concurrent tasks and accommodate different workflows. The defined tasks can be viewed as one or more directed workflow graphs.

In addition to defining various workflow topologies, users can also define the appropriate inputs and outputs of each miner according to its deployment descriptor. This design follows the "pipeline-filter" model[16] in software architecture, which provides considerable flexibility and scalability. Once a workflow is activated, the workflow manager will validate it and "translate" this graph into a complicated data structure in memory for later usage.

Engine deployment descriptor. Before a functional engine is deployed in MIP, its deployment descriptor, which contains information about how to manage it at runtime, must be provided by the engine provider. The deployment descriptor is an XML (eXtensible Markup Language) document which contains structural information about the engine, such as the functionality of the engine and its functional interfaces and Java** classes, as well as its external resource dependencies. The external resources could be *InFields*, *OutFields*, message queues, and database schema. The *InFields* are a collection of URIs, including possible data resources to be received and processed by this engine. The *OutFields* are also a collection of URIs, including possible outputs to be created by the engine. Message queues are used for asynchronous communications between the engine and the message broker. Database schema define the

schema of database tables the engine needs. The following is an example of a deployment descriptor:

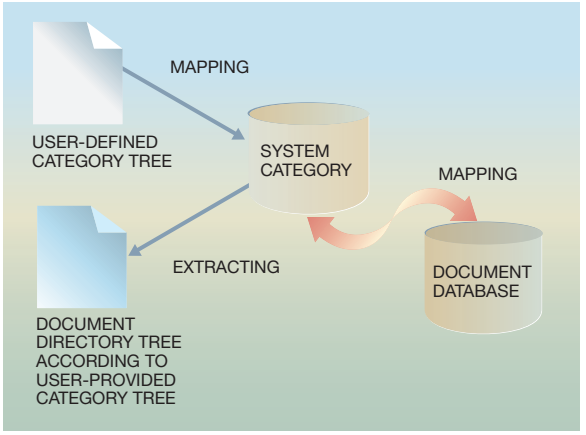
```
<miner id = "Indexer">
    <miner-name> Indexer </miner-name>
    <function-description> build index </function-
      description>
    <infields>
       <field-1>. . .</field-1>
    </infields>
    <outfields>
       <field-1>. . .</field-1>
              . . .
    </outfields>
    <message-queue> IndexerQueue </message-
       queue>
    <table-url>. . .</table-url>
. . .
</miner>
```

Message broker. The message broker is a key component in the message-driven architecture of the workflow management module. In addition to forwarding messages among engines, the message broker is responsible for dynamic routing. It acts like a mail server to determine the message receiver for each message generated by the engines.

Each engine produces messages and sends them to the message broker. The message contains the event type and associated parameters. It is the responsibility of the message broker to look up the defined workflow graph and decide which engine should receive the message. The message broker uses a finite-state-machine model to control the message flow. All data generated in a workflow are stored in a database which contains raw data crawled from various resources, temporary data, and final results.

We allocate two repositories for different purposes. Repository 1 contains raw files and temporary data, accessed by the engines. In the database design level, repository 1 may be a database in a DBMS (Database Management System). It contains several tables and indexes corresponding to the associated engines. The URIs of these tables can be specified in the InFields and OutFields of a message. Repository 2 contains the processed data, known as viewers, which will be accessed by end users. It is the responsibility of the workflow manager to integrate the useful information scattered over several tables in repository 1 into repository 2. This process is just the routine information synthesis discussed previously. Viewers can access these refined data without knowledge about the actual processing of the raw data.

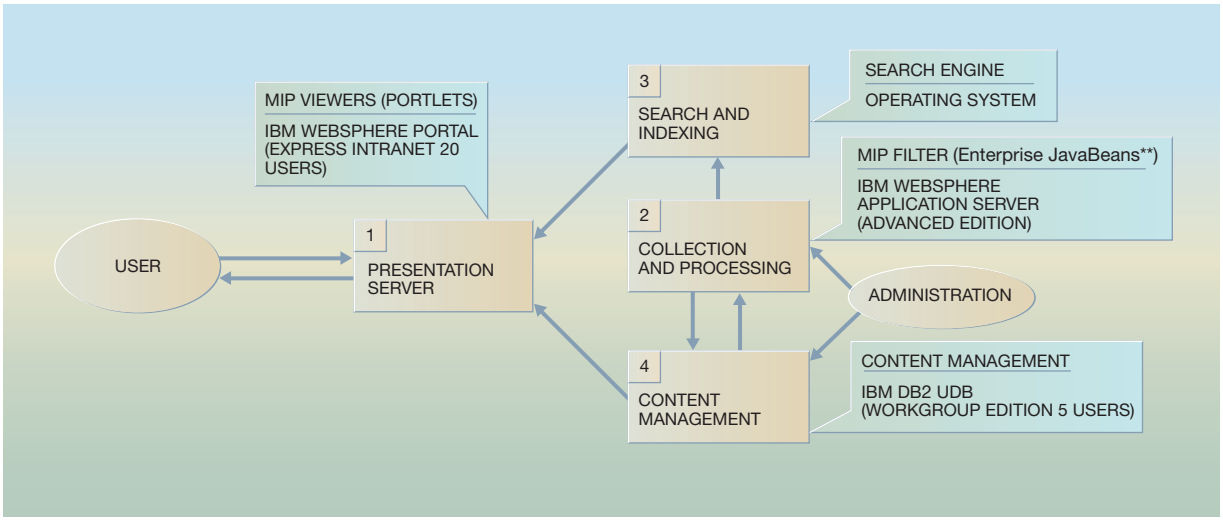
Figure 5 Block diagram showing the flat classification method



Flat classification scheme. The MIP system provides a portal for end users to access market intelligence information. In this context, a user-friendly interface is quite important. With the development of computing technology, users need personalized information classification services, allowing them to define their own category structure for information browsing. The biggest problem in providing these services is heavy computation and storage costs because traditional methods need to build and update a classification model for each user. In MIP, a general classification model for personalized service is used. Using this framework, no matter how different users' personalized category designs may be, only the system classification model needs to be trained and updated. All personalized classifications could be generated based on this classification model with very low costs. Because only one classification model needs to be trained, this framework has a big advantage in saving huge computing and storage costs.

Figure 5 shows the general system architecture of the flat classification. This classification framework is based on binary classifiers for personalized classification introduced in the earlier section, "Customized workflow." To create a new personalized category design, all that is necessary is to apply our algorithm to the existing system category repository. No personalized classifiers need to be trained, and all personalized document classifications are gener-

Figure 6 System architecture design with software platform selections



ated from a unified classification model. Thus, this method is very efficient and practical for implementing personalized classification services.

System implementation

In this section, we present a market intelligence system which was implemented for an Asian market-research company based on our MIP design. Although not all components have been integrated into this real application, this example will make the MIP framework more easily understandable.[17]

Architectural overview of the system. Figure 6 shows the software platform architecture for the MIP system that was implemented. The first component is the MIP viewer, located in the presentation server, which provides a Web portal for users accessing content, checks the user's authorization, and manages user profiles and access logs. The second component is for collection and processing. It contains the crawler, workflow controller, and several annotators for language processing, including Chinese segmentation, the duplication detector, the categorizer, and the summarizer. The file format converter is also contained in this component. The third component is the search engine. The last component is the content manager, which manages the DB2 UDB (Universal Database*) and intelligent content analysis.

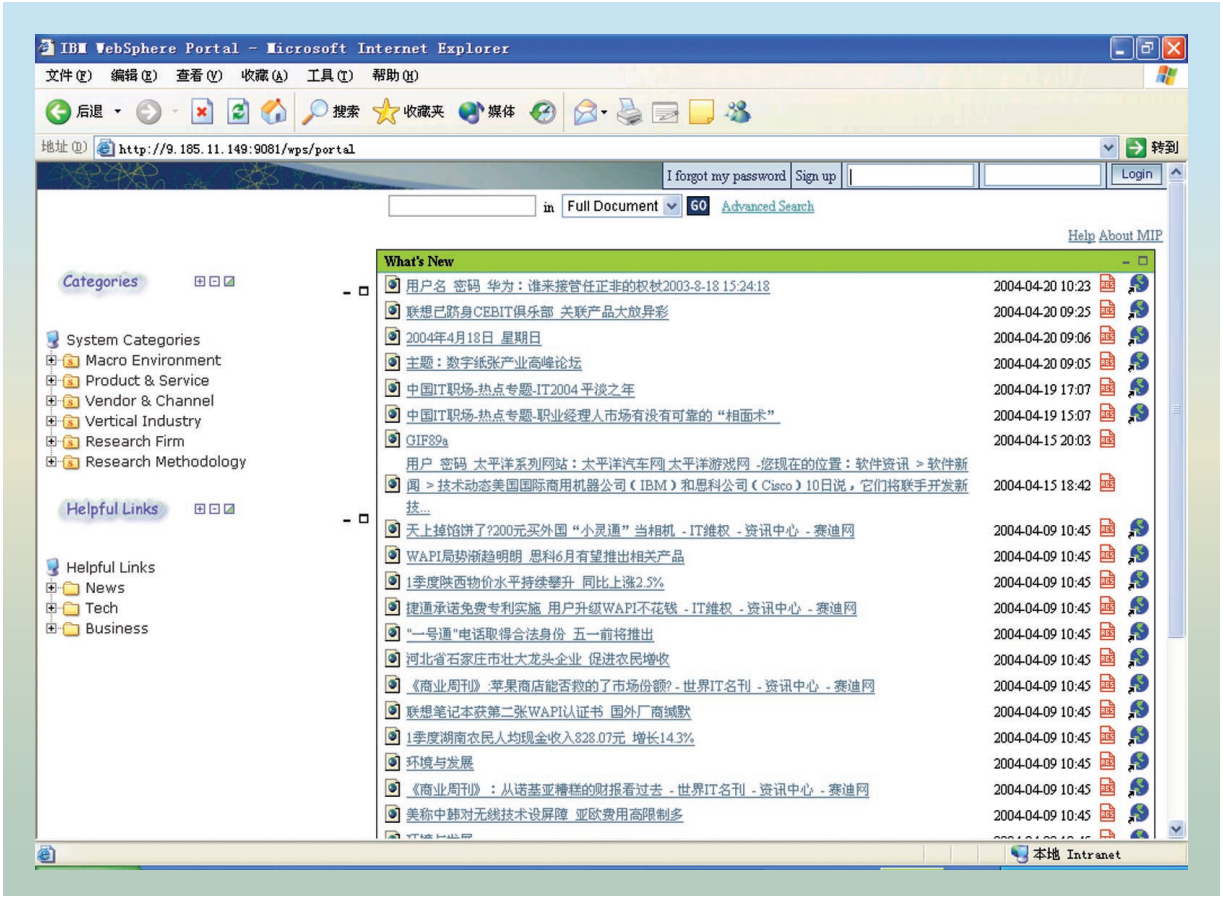
The system can monitor several Web sites (in Chinese or English), by using a "start" URI and a value for depth (in Web pages). It can also process the folders in machines located on a LAN. Through the EIP (Enterprise Information Portal) Domino crawler,[18] MIP can also access Domino databases. The crawling tasks can be scheduled by the system administrator.

The system can detect duplicated content in documents with a similarity threshold set by the system administrator. The administrator can define categories by providing sample documents for training associated with corresponding categories. After training, the model for the categorization can be generated and updated based on future inputs. Based on this model, the system can automatically put downloaded documents into categories. The system also provides tools for the system administrator to set the categories of the documents manually. Users can generate their category designs based on a predefined taxonomy and achieve personalized classification.

For efficiency in full-text search, the index table can be generated dynamically while documents are being collected. For Chinese documents, the index contains Chinese word segmentation information. The index table also contains information on the document title, author, and so on.

The entity and relationship extraction engine was implemented for the system, supporting English and

Figure 7 Entrance page of system



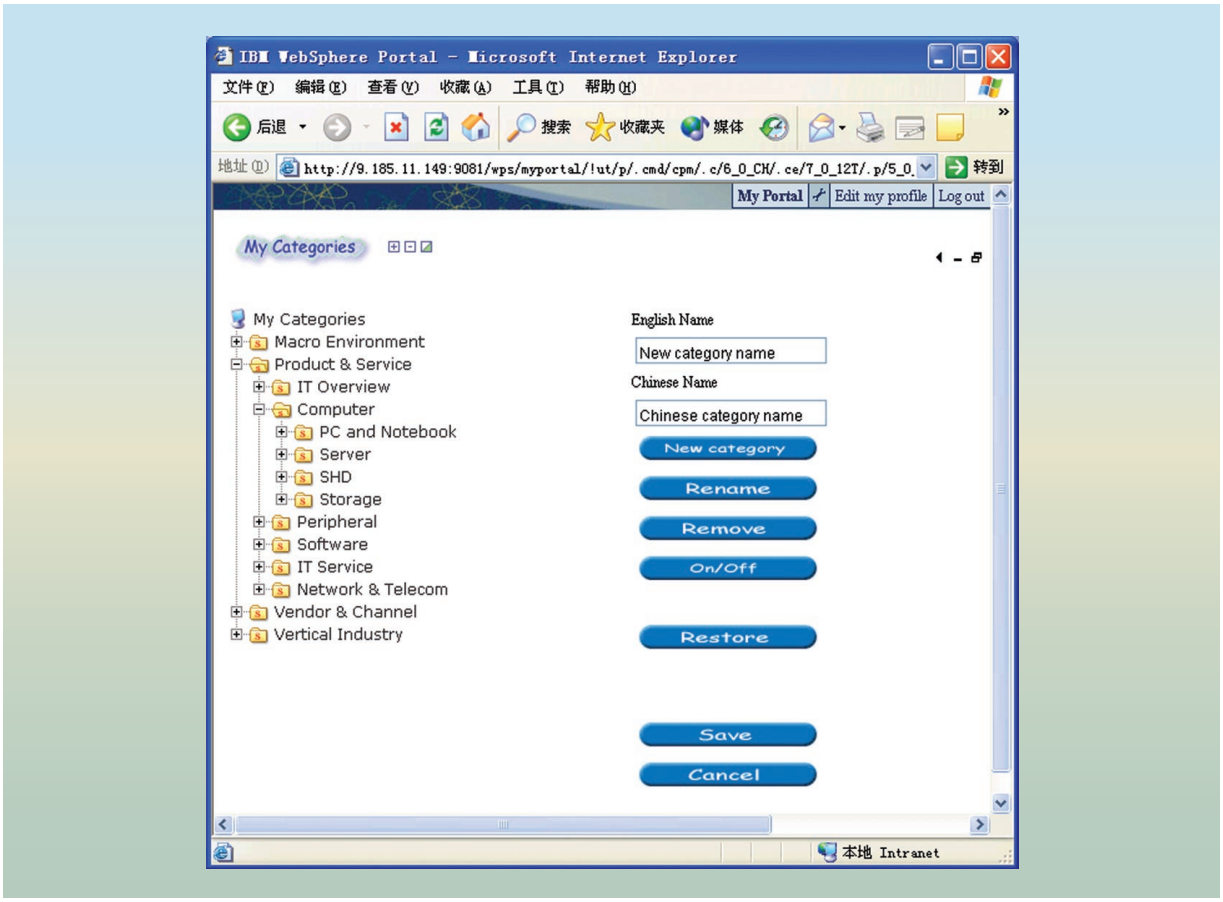
Chinese. Based on this engine, simple question answering can be performed, such as identifying elements with a relationship to a given person or company. In the near future, we plan to add more relationship types to the extraction engine, to support a wider range of questions.

Portal interfaces. As shown in Figure 7, the default home page for user access contains five areas: a navigation bar, search bar, copyright bar, login panel, and main viewer. The navigation bar contains the system category tree and default bookmarks. All client and administrative interfaces are packaged as WebSphere Portal Server portlets.

Once users are logged into this portal, they can define their own categories, based on the system category structure. The detailed method of personalized category creation has been introduced in a previous section. Figure 8 shows the interface for the user's category design.

For navigation and searching, users can navigate freely among system categories or personal categories. In addition, the system can recommend related documents based on the current one. If users think that a document is important, they can add the document to their personal "bookcase." Searching can be executed within the documents' full text or their titles, and users may search by category, author, or date, or based on Boolean operations among them. For full-text search, the keywords in the title are given higher priority. Intelligent document analysis yields a good ranking result. The system analyzes document content, including important named entities, topics, and frequency of user access to a document.

Figure 8 Interface for customized category design



Conclusions and future work

The MIP framework can integrate various text collections, apply data-mining and dissemination functions on the collections with a defined process flow, present a personalized browsing and searching interface, and help users in market intelligence information management. MIP has been used for rapid application development for some important customers in China, including a market-research company and an Asian news agency, with very positive feedback.

Our future work will focus on two areas. One area involves technologies such as visualization, relationship extraction, and ontology mapping. The other area involves the extension of this framework from MI to other areas in knowledge management.

* Trademark or registered trademark of International Business Machines Corporation.

** Trademark or registered trademark of Temis France Corporation, Sun Microsystems, Inc., or Microsoft Corporation.

Cited references and notes

1. W. F. Cody, J. T. Kreulen, V. Krishna, and W. S. Spangler, "The integration of business intelligence and knowledge management," *IBM Systems Journal* **41,** No. 4, 697–713 (2002).
2. WebDAV Resources, http://www.webdav.org/.
3. UIMA, The Unstructured Information Management Architecture Project, http://www.ibm.com/research/uima.
4. The Chinese entity spotter engine developed for MIP was made available as one of the TAEs for IBMs UIMA demo.
5. WebFountain, http://www.almaden.ibm.com/webfountain/.
6. WebSphere Portal for Multiplatforms, http://www-306.ibm.com/software/genservers/portal/.

7. Global Solutions Directory, Text Mining Server, http://www.developer.ibm.com/solutions/isv/igssg.nsf/list/bycompanyname/86256B7B0003EBBF86256DF600491B20?OpenDocument.
8 A. Zanasi, "Web Mining through the Online Analyst," in *Data Mining II*, N. F. F. Ebecken and C. Brebbia, Editors, WIT press.com electronic library (2000), pp. 3–14, http://library.witpress.com/listchapters.asp?q_bid=107&q_subject=Computing%20_%20Information%20Management.
9. Syracuse University School of Information Studies, Center for Natural Language Processing, http://cnlp.org/tech/equery.asp.
10. Languistics, http://www.languistics.com/.
11. V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc, New York (1995).
12. R. Iannella, "An Idiot's Guide to the Resource Description Framework," *The New Review of Information Networking* **4** (1998), http://www.dstc.edu.au/cgi-bin/redirect/rd.cgi?http://archive.dstc.edu.au/RDU/reports/RDF-Idiot/.
13. J. Ferraiolo, J. Fujisawa, and D. Jackson, *Scalable Vector Graphics 1.1 Specification*, W3C (January 2003), http://www.w3.org/TR/SVG/.
14. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, New York (2000), http://www.support-vector.net/.
15. Z. Su, L. Zhang, and Y. Pan, "Document Clustering Based on Vector Quantization and Growing-Cell Structure," *Proceedings of the IEA/AIE 2003, Developments in Applied Artificial Intelligence, 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Laughborough, UK; Lecture Notes in Computer Science 2718 Springer (2003), pp. 326–336.
16. S. Rongviriyapanish and N. Lévy, "Variations sur le Style Architectural Pipe et Filter," *Proceedings of the AFADL 2000, Actes du 3eme Colloque sur les Approches Formelles dans l'Assistance au Développement de Logiciels*, Grenoble, France (2000), pp. 81–95, http://www-lsr.imag.fr/afadl/Programme/Articles/levy.ps.
17. The visualization method described in this paper was not fully implemented; instead, a limited machine-generated graph was presented to users. The question-answering system was implemented, but only for very simple queries. RDF storage and MIP are not connected in the current system. There is some meta-data support, but it is currently based on a relational database (DB2).
18. *WebSphere Portal Content Management*, http://design.torolab.ibm.com/software/webservers/portal/library/extend/InfoCenter/wpf/fea_cm.html.

Accepted for publication April 19, 2004.

Zhong Su *IBM Research Division, China Research Lab, 2/F HaoHai building, No. 7, 5th Street, ShangDi, Beijing (suzhong@cn.ibm.com).* Dr. Su is a research staff member at the IBM China Research lab. He received his Ph.D. degree in computer science at Tsinghua University in 2002. He has worked on a wide range of software systems. Currently, he leads a project on business intelligence and related solutions in the area of banking and telecommunications.

Jianmin Jiang *IBM Research Division, China Research Lab, 2/F HaoHai building, No. 7, 5th Street, ShangDi, Beijing (jiangjm@cn.ibm.com).* Dr. Jiang is a research staff member at the IBM China Research Lab. He received his Ph.D. degree in computational mathematics at Tsinghua University in 1997. He has worked on natural language processing and developed an English-Chinese translation engine. His current research interests are text mining and information retrieval.

Tao Liu *IBM Research Division, China Research Lab, 2/F HaoHai building, No. 7, 5th Street, ShangDi, Beijing (liutao@cn.ibm.com).* Mr. Liu is a research and development engineer at the IBM China Research Lab. He received his master's degree in computer science and technology at Tsinghua University in 2003. He is currently working on design and development for the next generation of market intelligence portals.

Guo Tong Xie *IBM Research Division, China Research Lab, No. 7, 5th Street, ShangDi, Beijing (xieguot@cn.ibm.com).* Mr. Xie is a researcher in the information and knowledge team at the IBM China Research Lab. He obtained his B.Sc. degree in 2000 and a M.Sc. degree in 2003, both in computer science, from the Xi'an Jiao Tong University. Before joining IBM Research in 2003, he worked on distributed computing and e-business. His main research interests are in the application of semantic web technologies to information integration, business integration, and knowledge management. He is currently leading a group on business semantics research.

Yue Pan *IBM Research Division, China Research Lab, 2/F HaoHai building, No. 5th Street, ShangDi, Beijing (panyue@cn.ibm.com).* Dr. Pan is a research staff member at the IBM China Research lab and manages the information and knowledge team. He received his Ph.D. degree in automatic control at the Chinese Academy of Science in 1996. He has worked on a platform for Internet content selection, translingual search, text mining, and other software systems. Dr. Pan and his team are currently developing technologies to capture business semantics, and integrate, analyze, and visualize structured and unstructured information for business intelligence.