Leverage the power of information management for Service-Oriented Architecture (SOA)-based modeling, architecture, design, and implementation. See the various services that information management offers arranged into a stack view and get a detailed description of each. The authors start with metadata management and the importance of metadata integration, move into an examination of services that information management offers, and then present a SOA case study. Finally, the authors list some tools for the services discussed.

Introduction

In this article you'll learn about specific services, such as the following:

- Metadata management
- Extract Transformation Load (ETL)
- Federation
- Data placement (such as replication and caching)
- Data modeling
- Search
- Analytics

We then present a case study that uses SOA to validate data quality, and end with a list of tools for various services. After reading this paper, you should be better able to unleash the power of information management to help build a robust and balanced SOA, enabling information and business integration and avoiding common mistakes such as isolated data silos, data inconsistencies, and untapped information assets.

SOA is more than Web services

Figure 1 shows a logical view categorizing services that information management offers based on their following value propositions:

- Security
• Collaboration
• Availability
• Manageability
• Information consumption

While no single product offers all of these services, taken together these services create a complete information management framework under SOA. Notably, while some articles might place metadata management at the bottom of the information management stack, we depict it in a way that shows that metadata management is pervasive and intertwined with the rest of other services. In fact, SOA is a metadata-driven architecture (see "Metadata Evolution Management in Your SOA" in the Resources section). Therefore, we begin with metadata management in the second half of this paper.

**Figure 1: Information management in SOA**

**Metadata management**

**Metadata, metamodel, and meta-metamodels**

The most common definition of metadata is data about data -- which doesn't really say much. Depending on the discipline, metadata can mean different things. In essence, metadata is information about data's structure (syntax) and meaning (semantics). Examples of structural approaches to metadata are Relational Database Management Systems (RDBMS) catalogs, Java library catalogs, and XML DTDs and schemas. Each of these defines how data looks and how it is used. From the semantic point of view, metadata provides meaning for data. Examples include descriptions in data dictionaries, annotations, or ontologies.

Furthermore, there are instance and class metadata in the content management arena. Instance metadata simply is data stored in a content management metadata repository and refers to objects stored somewhere else, such as documents, Web pages, audio, and video files. Entries in taxonomy and indexes are also considered to be instance metadata. Class metadata is, in some respects, equivalent to RDBMS catalogs and XML schemas, which describe the structure of instance metadata.
Metamodels (also known as meta-metadata) define the structure and semantics of metadata. Examples of standardized metamodels include Unified Modeling Language (UML) and Common Warehouse Meta-model (CWM). The meta-metamodel layer is comprised of the description of the structure and semantics of meta-metadata. It is an attempt to provide a common language that describes all other models of information. Meta Object Facility (MOF) is a standard for meta-metamodels (see Resources).

**Figure 2: MOF metadata architecture**

![MOF metadata architecture diagram]

It is vital for metadata producers to adhere to the standards in metamodels, metadata interfaces, meta-metamodels, and query languages to achieve maximum interoperability and reach for wider metadata consumers, such as data warehouses, analytics, and modeling tools. SOA relies on such cohesive standards in order to dynamically match service producers and consumers, monitor BEPL flow, and improve the traceability of IT resources and business processes.

**Considerations for metadata management**

When we reengineer metadata management, XML obviously is a default data format for metadata because of its ubiquity. Within a single vendor or an organization, centralized approaches are often preferred in order to encourage metadata asset reuse and to reduce development effort and confusion. Also, standardization is the preferred approach. For instance, IBM® uses the open source Eclipse Modeling Framework (EMF) as a common metadata integration technology. EMF provides metadata integration for tools and run-time, so that all the software developed on top of EMF shares a common understanding of other applications. In an ideal situation (though it might be difficult in the short-term), one metadata repository stores all metadata artifacts. Services offered by information management, such as SSO, ETL, federation, quality, search, versioning, and workflow can be invoked for data, content, and metadata management when they are needed.

Regarding the XML repository, there are two popular storage mechanisms for storing XML metadata. They are RDBMS and native XML repositories. Each has its advantages and disadvantages. Some of the determining factors are performance, flexibility, bandwidth, interoperability, support of user-defined data types, and data quality assurance.

Across vendors, enterprises or industries level, the federated approach is a more practical method for metadata management. A virtual metadata repository allows applications to access and
aggregate heterogeneous metadata sources through a single API. Physical metadata artifacts can be stored either in their original locations or using a ETL/replication/cache method to improve performance and metadata placement. Automatic discovery, mapping, and transformation among diverse metadata sources are critical to improve metadata manageability.

**Relationships among data, content, and metadata management**

On one hand, metadata provides the glue that enables programs to talk to each other (in fact, one vendor calls its metadata repository *SuperGlue*). On the other hand, requirements for metadata management are very similar to data and content management. Metadata management needs to offer the same types of services on security, collaboration, QoS, and manageability as on data and content management. Metadata management also needs to incorporate SSO, ETL, federation, quality, search, versioning, workflow and storage persistence. The automation and orchestration requirements for metadata management tend to be even greater than for data and content management, because the audience of metadata is really mostly computer programs.

Nevertheless, the good news is that asset reuse and service orchestration can be achieved by building metadata management on top of well-architected, SOA-based information management. This illustrates the importance of reengineering information management into SOA-based and reusable components.

**Challenges of metadata integration**

As we stated earlier, integrating metadata is more challenging than integrating data and content. Many factors contribute to the difficulty of metadata integration. They include the following:

- Metadata is pervasive and, in many cases, invisible to users.
- Metadata and metamodels, in many products, have their own proprietary format. This is especially true for content management.
- In content management, adding metadata to content is typically facilitated by manual workflows. A great deal of content lacks good metadata to enable integration and search.
- Metadata integration requires higher levels of automation and orchestration than data and content integration. This, in turn, requires higher levels of automated discovery, transformation, mapping, and semantic understanding.
- Vendors might choose to keep their proprietary metadata format for fear of losing current customers.
- It takes time and effort to transform to metadata standards such as MOF.

**Business value of metadata integration**

SOA is largely a metadata-driven architecture. To understand the high-level business value of metadata integration, let’s begin by taking a bird’s eye view. Figure 3 illustrates the importance of metadata integration within the context of On Demand Business. Based on information standards, metadata enables seamless information exchange. Given well-integrated metadata, information can freely flow from one place to another across boundaries imposed by operating systems, programming languages, locations, and data formats. Thus metadata can be thought of as the "brain" in information integration. Furthermore, information integration enables business...
integration, either across departments within an enterprise or across enterprise boundaries. It provides the following:

- It provides a single and complete view of customers, partners, products, and business through data warehouses or federation.
- It facilitates business performance management using analytical services.
- It enhances business applications with broad information access.
- It enables business process transformation with continuous information services.

Lastly, business integration is one of the cornerstones of an on demand business. Business integration differentiates itself from previous Enterprise Application Integration (EAI) by using IT technology to serve business objectives, rather than the reverse. Therefore, it is not an overstatement to say that metadata integration is the brain of an On Demand Business.

**Figure 3: Metadata integration is the brain of On Demand Business**

![Metadata integration is the brain of On Demand Business](image)

Examples of high-level metadata integration values include:

- Facilitating data/content integration from heterogeneous sources.
- Improving time to market for new applications and allowing faster application integration.
- Smoothing the process of inter-/intra-enterprise business integration.
- Providing new insight by enabling analysis of fully integrated information.
- Enabling impact analysis through change management and predictive analysis.

**Data and content federation: A decentralized approach**

Federation is the concept that a collection of resources can be viewed and manipulated as if they were a single resource, while retaining their autonomy (with little or no impact to existing applications or systems) and integrity (not corrupting data or content in existing applications or systems). Needless to say, autonomy and integrity are two important prerequisites for federation.

Since the late 1990s, data federation has emerged as a distinct approach from the centralized approach that data marts and warehouses had been using. Data federation strives to leave data in its original location and to create what can be thought of as a virtual database. Similarly, content federation has emerged in recent years to enable access to and aggregation of heterogeneous content sources. These decentralized approaches reduces data and content redundancies, bandwidth, storage, on-going synchronization, and additional administrative cost associated with
a centralized approach. Real-time access to distributed information sources also brings new capabilities to business intelligence, one example being compliance with legal and regulatory requirements. For developers, data federation reduces the need to write and maintain custom APIs for various data sources and to acquire highly specialized skills.

The top concern with data federation is performance. To improve performance, federation frequently uses caching, materialized query tables (MQTs), and distributed query optimization and execution. Caching and MQTs create and manage tables at the federated server, which can be a full or subset of rows from target federated data sources. As a cutting-edge tool, IBM WebSphere® Information Integrator takes into account the following:

- Standard statistics from source data (such as cardinality or indexes)
- Data server capability (such as join features or built-in functions)
- Data server capacity
- I/O capacity
- Network speed (please refer to the IBM Redbook, "DB2II: Performance Monitoring, Tuning and Capacity Planning Guide" in the Resources section)

ETL: A centralized approach

Extract-transform-load (ETL) is one of the oldest technologies for data integration and is closely allied with data warehousing and business intelligence. It enables data consolidation, migration, and propagation. ETL tools extract, transform, and load data from one or more data sources to one or more targets. ETL was, for some time, the backbone of information integration and still is very popular today. Unlike straightforward extract and load operations, transformation is the most complicated piece, as there is a need to understand, convert, aggregate, and calculate data. The benefits of ETL and data warehousing can be diminished by high costs, slow turn-around time, and incomplete sets of information in data sources.

Centralized and de-centralized approaches compliment each other, and there are major benefits when both approaches are combined.

The centralized approach involves some of these elements:

- Access performance or availability requirements demand centralized data.
- Currency requirements demand point-in-time consistency, such as close of business.
- Complex transformation is required to achieve semantically consistent data.
- The centralized approach is typically used for production applications, data warehouses and operational data stores.
- The centralized approach is typically managed by ETL or replication technologies.

The decentralized approach involves the following considerations:

- Accesses performance and load on source systems that can be traded for overall lower cost implementation.
- Currency requirements demand a fresh copy of the data.
- Data security, licensing restrictions, or industry regulations restrict data movement.
• The decentralized approach can combine mixed format data, such as customer ODS with related contract documents or images.
• Query requires real-time data, such as stock quote, on-hand inventory.

Data replication and event publishing

Data replication moves copies of data from one location to another location. The target location could be either a centralized location, such as data warehouse, or another distributed place on the network. In a grid environment, replication and cache services are used to create the Placement Management Service to meet Quality of Service (QoS) goals. Depending on the access patterns and location of the consuming applications, a Placement Management Service can improve response time and information availability by creating caches or replicas (See "Towards an information infrastructure for the grid" in Resources). In a Web application environment, data and content replication are often used to move data or content from the staging server (usually only for administrators) to the production server when data or content are ready to be published for public consumption. The staged data governance gives organizations greater control over the flow and life cycle of information. For example, a Web site supports multiple national languages. When a piece of data or content element needs to be translated before it can be published on the site, it is populated to the staging server first. Only after it gets translated and optionally approved by administrators is it replicated to the production server and subsequently made available to the public.

Replication can be used in conjunction with either centralized or decentralized approaches. The major differences between ETL and data replication are that ETL usually moves data to a centralized location after applying vigorous data cleansing and transformation rules, takes much longer, and moves larger amounts of data. This is in contrast to data replication, which moves a much smaller set of data, as it is going to central or distributed locations in a more automatic fashion. Data replication can access data in real-time or near-real-time. The primary goal of ETL is to analyze or monitor data and produce business intelligence, whereas the goals of data replication are mostly related to performance, data governance, and data availability. Lastly, ETL and data replication can complement each other nicely, in other words, one can use the data replication function to move data faster to data marts and warehouses, and the data transformation function in ETL can deliver greater flexibility and higher data quality in the data replication arena. In order to reuse the logic in different tools, easily callable and loosely-coupled information services need to be in place.

Unlike ETL and data replication, event publishing does not know where the data is going and how it will be used. Changes from source tables are published in XML format or other data formats to a message queue. It is the responsibility of the applications to retrieve published events and take proper actions, such as triggering a business process or transforming data before applying it to a target data source. The loosely coupled architecture separates service providers and consumers, and allows data events to be independent from applications.

Logical data and semantic information modeling

Logical data modeling is one of software development's best practices and also one of the most easily neglected areas when a development organization is under time and budget pressure. While
logical data modeling is often skipped during in-house development, organizations frequently buy or acquire Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), or other sorts of packages. The result is that there are many versions of data models referring to the same thing within an organization, and each data source has its own data model and meta-model. For example, it is not unusual to have different terms referring to customers -- CRM calls it a *customer*, the accounting system calls it as *client*, and sales system calls it as *buyer*. The textbooks and theorists tend to begin with the logical enterprise data model, then move onto a physical data model (such as Entity Relationship Diagram), code generation, and development, but the order is often reversed in reality.

In practice, organizations often build, buy, or acquire databases in pieces, and data remains in isolated islands. On occasion these organizations recognize a need to integrate the data. What do they do next? Often they dive into piles of documents, millions of lines of code, and terabytes of data to discover what types of information they produce and consume, not to mention that they need to discover and document after the fact the relationships among various data models and business processes. On the bright side, certain automatic data discovering and profiling tools can speed up processes and relieve the pain of performing these tasks. Many organizations might eventually derive a logical enterprise data model so that individual systems can be mapped to the common logical model. Transformation is required in certain cases, such as transforming one currency to another. In the end, physical data models are mapped to an Enterprise Data Model -- a common logical data model shared by an enterprise. An Enterprise Data Model provides maximum benefits if it is designed at the beginning as a part of Model Driven Architecture. Nevertheless, it is still invaluable as a result of the above reverse engineering steps. The main benefits of an Enterprise Data Model are:

- Provides an overview of enterprise information assets.
- Reinforces the practice of using IT technologies to support business processes.
- Reduces the cost and risks of Enterprise Information Integration (EII), Enterprise Application Integration (EAI), and data warehousing.
- Enables asset-based reuse of data, metadata, and meta-models.
- Improves data and metadata quality.
- Facilitates communication among business analysts, data modelers, developers, and database administrators.

Semantic information modeling (ontology) moves beyond structural, logical data modeling in the sense that it models semantics (meaning) and relationships of data. It unifies vocabularies (terms and concepts) across multiple knowledge domains. There are a number of problems ontology solves particularly well, such as problems with the following (see also "Semantics FAQs" in the Resources section):

- Information integration
- Model transformation
- Translation
- Data cleansing
- Search
- Navigation

Semantic information modeling (ontology) moves beyond structural, logical data modeling in the sense that it models semantics (meaning) and relationships of data. It unifies vocabularies (terms and concepts) across multiple knowledge domains. There are a number of problems ontology solves particularly well, such as problems with the following (see also "Semantics FAQs" in the Resources section):

- Information integration
- Model transformation
- Translation
- Data cleansing
- Search
- Navigation
• Text understanding
• Document preparation
• Speech understanding
• Question-and-answer issues

Data profiling

Data profiling is a process to discover the following:

• Data formats
• Patterns
• Characteristics
• Rules
• Hidden relationships

Data profiling also provides numerous benefits, including the following:

• Improves organizations' understanding of their data.
• Helps Electronic Data Management (EDM).
• Facilitates data mapping and transformation.
• Improves data quality.
• Builds baselines for performance tuning.
• Assists semantic modeling.

Data profiling aims to understand information better and create additional metadata about objects.

Data, content, and metadata quality

Data quality can make or break an enterprise information management strategy, which in turn determines the success of its business integration strategy. Data quality issues are reported to be one of the main reasons data warehousing projects miss deadlines. Poor data quality can cause misinformed decisions, ineffective operations, missed opportunities, and on occasion punishment by the organization or marketplace. Data quality no longer sits on the shelf as a luxury, nice-to-have item; instead, it has become a key operational element for businesses.

Examples of data quality problems are:

• Missing data for required fields
• Inconsistent data entries
• Incorrect or inaccurate data entries

Due to the inherent complexity of data quality work, some organizations opt to out-source such work to third-party service providers. We will take a look at a case study later in this paper.

Content quality is often neglected partially because evaluating content quality is a much harder task than evaluating data quality. Content, after all, is unstructured, and quality standards are thought of to be more subjective or arbitrary. Content quality is typically not in the scope of
technology projects. It is not well-regarded from an organizational perspective. However, in a SOA environment, content quality becomes more important due to SOA's fluid nature. If data errors or poor quality content are not being caught early on, they get propagated everywhere. Content quality criteria differ by the types of contents, but there are some common criteria to evaluate content quality, such as the following:

- Relevancy
- Timeliness
- Expiration
- Content validation
- Rating
- Duplication
- Link checking

Metadata quality has received increased attention lately due to the increasing demand for metadata management capabilities. Techniques that are used to improve data quality, such as standardization, profiling, inspection, cleansing, transformation, and validation also apply to metadata quality improvement.

Strong data typing is the key to ensuring consistent interpretation of XML data values across diverse programming languages and hardware. However, current XML technology only allows the schema validation for a single document, but an effective way to validate data types (including user-defined data types) and enforce semantic strong-typing across different schemas or data sources (such as between relational databases and OO data type facility) is missing. Standardizing on XML Document Type Definitions (DTDs) or schemas, which many industries are attempting to do as a solution to this problem, is insufficient, as issues on XML DTDs or schemas validation, semantic consistence, and compatibility still exist when you need to integrate data across multiple industries, which is a basic requirement for On Demand Business.

**Search and query**

Within enterprise search, there are many different types of searches: keyword, Boolean, range, faceted metadata, semantic, natural languages and parameterized. No matter which type of search, the purpose is to provide a consolidated, correlated, and ranked result set that enables quick and easy access to information. To facilitate search, indexing (not to be confused with indexes in relational databases) is used to index key words, concepts and instance metadata of unstructured content, such as Web pages, e-mail database, or file systems, so they can be searched and retrieved. Relational databases can also be indexed for faster and more flexible search.

Although many organizations realize the importance of integrating structured and unstructured information, today's search results are still unrelated to each other. What users get is a list of links that point to *potentially related information*. Users have to crawl through the search results to find information they need and to correlate it with the original intent of the query. This is largely a manual process. We think there is a strong need to research on using search and query to achieve *one query, one result set* across data and content.
Databases generally have their own search functions. The most generic search function is through query language, such as SQL and XQuery. Database search is great to retrieve structured and exactly matched data, but it requires highly specialized knowledge on query construction and data model familiarity. The users of database search are typically developers or database administrators. Besides, database search is not designed for relevance ranking, fuzzy search or multiple keywords. Therefore, database search is limited in scope. To achieve high performance, flexibility, relevance ranking, and so on, some search engines connect to databases directly, extract data, and generate indexes from databases. One example is IBM WebSphere OmniFind.

Analytics

As we illustrated in the previous ETL section, data warehouses consolidate data into a central location to enable better decision-making, cross-departmental reporting, and data mining. The traditional analytics include reporting, data mining, dashboards, scorecards, and business performance management. As competition increases, operations become more complex, and regulations become more restrictive over time. Organizations want to access heterogeneous data sources in real-time in order to make the following improvements:

- Employ integrated information to predict market trends.
- Understand customers better.
- Increase operation efficiency.
- Ensure compliance to regulations.
- Derive new knowledge.

All of these trends drive the increased demand for analytical capabilities in information management. Analytics have moved from the back office to the front line. For example, if a salesperson knows an existing client's contract, service experience, its industry trends, its competitors and customers, he or she will be in a much better position to form a customized sales proposal specific to that client. Lastly, analytics frequently necessitates information integration across heterogeneous information sources. For instance, to evaluate quality, a car manufacturer needs to correlate accident reports (stored in a document management system), dealers' repair records (stored in a relational database), drivers' risk factors, and environmental factors (stored in knowledge management system). The future of analytics will build increased intelligence to access and correlate information from heterogeneous information sources in order to allow new insights and business decisions.

Related services

The following services are described as related not because they are not important to information management, but because they are common to business processes and application integration as well.

SSO, access control, and audit

Single-Sign-On (SSO) to heterogeneous information sources, access control, and auditing the viewing and modification of information all build a foundation for a secure environment for information management. SSO asks users who are you, access control asks what can you do,
and audit keeps track of what have you done. The benefits of SSO are many; it reduces user frustration, lowers developing effort, and increases productivity. Access control ensures that only people with correct rights can access data and content. Some businesses require highly sophisticated access rights management such as Digital Rights Management. Audit service adds additional security to data and content. Viewing, inserting, modifying, and deleting information can all be audited and easily reported. With increasing demand on security and regulatory compliance, the combination of SSO, access control, and audit service builds a solid foundation for enterprise information management.

**Workflow and version control**

Both workflow and version control are designed to foster collaboration in a team environment. Data, content, and metadata management, application code development, and processes all need workflow to allow people to collaborate while establishing consistent points through version control so they can refer back to it later. Workflow links people, processes, and information into one ecosystem. Each part of the system -- people, processes and information -- is very interactive, and the interactions among them are even more dynamic. For example, a company sets up a program that every employee can submit their ideas on any topic. Depending on the domains of ideas (information), they will be routed, reviewed, and worked by different people (processes, people). Thus, a highly robust and adaptive workflow is needed to be able to handle unanticipated situations. Once you develop such a workflow service, it can be called by different applications, such as document management, HR system, or knowledge management.

**Portal**

Industry analysts predict that enterprise portals combined with Web services will take off within the next twelve months. Portals integrate applications and information and present them to the end users as one unified view. Since EII provides an abstraction layer, developers are able to access and aggregate various information sources, maintain the codes, and achieve performance and security requirements without writing customized adopters. As the result, application development can reduce time, cost and skill demands, and the portal users can access the wide variety of information effortlessly. Most importantly, end-to-end business processes can be integrated easily and quickly.

**Case study: An example of data quality service**

Services such as enterprise search, data quality and validation, and analytics in the information management stack are often good candidates for outsourcing. The framework of information management under SOA opens up a new and increasingly popular business model. Let's take a look at a case study of offering data validation services, a subset of data quality services, through SOA.

Many e-commerce companies need to verify addresses, telephone numbers, and social security numbers, as well as other identifying information in real-time in order to prevent mistakes and fraud or to comply with laws and regulations, such as Sarbanes-Oxley. Because of the complexity of data quality validation, some companies subscribe to data validation services from third-party
providers instead of developing in-house solutions. Some companies offer data validation and quality services and provide real-time address and telephone number validation over the Internet. Typically, after the customers fill out e-commerce applications and submit them online, e-commerce companies wrap customers' information into XML documents and send it to data validation companies through Web services, Simple Object Access Protocol (SOAP), and Web Services Description Language (WSDL). The receiving companies verify the data in real-time within the same customer transaction. For the customers, they get instant feedback and are able to correct or cancel the transaction.

In the past, if any data errors occur during the process, e-commerce got undeliverable addresses or e-mails days or even months later; meanwhile, customers wondered what happened to their account. As a result of data validation services through SOA, e-commerce companies are relieved from the burden of maintaining and updating gigabytes of database information that contains millions of people's names, phone numbers, and deliverable addresses, including information from other countries and territories.

Conclusion

The authors examined each of the services that information management offers and gave special attention to metadata management and integration. Although there are many types of services, and these might initially seem overwhelming, you can see the main point of information management if you remember the following value proposition:

- Security
- Collaboration
- Quality of Service
- Manageability
- Consumption

Hopefully, this paper makes you aware of the great importance and broad scope of information management. Armed with knowledge of the individual pieces and their interactions, you are able to unleash the power of information management and build a robust and balanced SOA.

Acknowledgement

The authors would like to thank Susan Malaika and Norbert Bieberstein for their excellent feedback and Robert D. Johnson for his support.

IBM information management products

The following table shows you the information management services and the IBM products available to implement these services.

**Table 1. IBM information management products**

<table>
<thead>
<tr>
<th>Information management services</th>
<th>IBM products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explore the different approaches to information management in SOA</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td>DB2® Data Warehouse Edition; DB2 Cube Views; DB2 Alphablox; DB2 Entity Analytics</td>
</tr>
<tr>
<td><strong>Content federation</strong></td>
<td>WebSphere® Information Integrator, Content Edition</td>
</tr>
<tr>
<td><strong>Data federation</strong></td>
<td>WebSphere Information Integrator</td>
</tr>
<tr>
<td><strong>Data modeling</strong></td>
<td>Rational® XDE; alphaWorks Data Architect for DB2 Document Management</td>
</tr>
<tr>
<td><strong>Data profiling</strong></td>
<td>WebSphere ProfileStage</td>
</tr>
<tr>
<td><strong>Data quality</strong></td>
<td>WebSphere QualityStage</td>
</tr>
<tr>
<td><strong>ETL</strong></td>
<td>WebSphere DataStage; DB2 Warehouse Manager</td>
</tr>
<tr>
<td><strong>Logical and semantic information modeling</strong></td>
<td>IBM Research Ontology management system (Snobase)</td>
</tr>
<tr>
<td><strong>Metadata repository</strong></td>
<td>WebSphere MetaStage; alphaWorks XML Registry</td>
</tr>
<tr>
<td><strong>Search</strong></td>
<td>WebSphere Information Integrator OmniFind Edition</td>
</tr>
</tbody>
</table>