Information Integration Software Solutions
March 2003

DB2 Information Management Software

Information integration—Extending the data warehouse

By Dr. Barry Devlin
IBM Software

Table of Contents

Data warehousing is changing, changing to address new business needs. Of course, many of the original requirements remain—like the need to deliver business value, the requirement for data cleanliness and consistency and the ability to dice, slice and dig into the information at will. The fundamental new requirements concern timeliness and extensibility—to give business users the ability to see current data, remote data or unstructured data when they need it. All this must be seamlessly integrated with the historical information they have typically delivered through data warehouses and marts.

These requirements have emerged over the past few years. Typically, IT organizations address them by creating operational data stores (ODS) or simply by trying to load the data warehouses more rapidly and regularly with more and more data. However, supplying data warehouses with fresh real-time data is very expensive, and not cost-effective for the majority of enterprises. Furthermore, some data cannot or need not be kept in the data warehouse even though it may be of critical value, because its usage, size or format is inappropriate for the data warehouse or user queries.

To successfully address these needs, businesses require additional methods of integrating and delivering information without requiring all data to be stored in the data warehouse first. This is the goal of IBM's vision for information integration. The aim is to make data location and format transparent to the user or application, enabling both centralized, local access, as found in traditional data warehouses, as well as distributed access to remote data from within the same infrastructure.

So do not fear! We are not proposing that you throw away all of the tooling and investment you have made in data warehousing. In fact, in many ways, information integration is a natural and logical extension of what you have already done to create and maintain your existing data warehouses.

This white paper first briefly outlines the historical evolution of data warehousing over the last 20 years. It shows how the integration of diverse data was the foundation of data warehousing, and thus, how the concept of information integration is a logical corollary. It also describes how the increasing need for near real-time data as well as a certain level of read/write operations in the warehouse have forced compromises to the original data warehouse architecture.

Second, the paper focuses on the distributed access aspect of information integration, as delivered in IBM DB2® Information Integrator. It illustrates how the technology can address these new business needs by providing direct access to data in its original location under certain defined and controlled circumstances. In particular, it discusses where this federation functionality is useful and where it should not be used.

Finally, this paper gives a number of examples of how DB2 Information Integrator can extend existing data warehouses to address emerging business needs.

Data warehousing—20 years growing

The original business rationale for data warehousing is well known. Simply put, the purpose of a data warehouse was to provide usable and understandable business information to end users. Although some of this information already existed in the business IT systems, it was clear that immense quantities of raw data were also available, and could be converted into useful information.

To address these business needs, IBM and others proposed a three-layer data architecture in the mid- to late 1980s, which is now widely accepted. This is shown in Figure 1. But, why so many layers? There are two basic reasons. First is performance. If complex end-user queries are allowed to run against operational systems that have been designed and optimized for other purposes, they are likely to significantly impact the performance of these base systems. In addition, the response times for end user queries is likely to be poor. These considerations lead directly to a minimum of two layers in the data architecture, one operational in nature and the other informational.

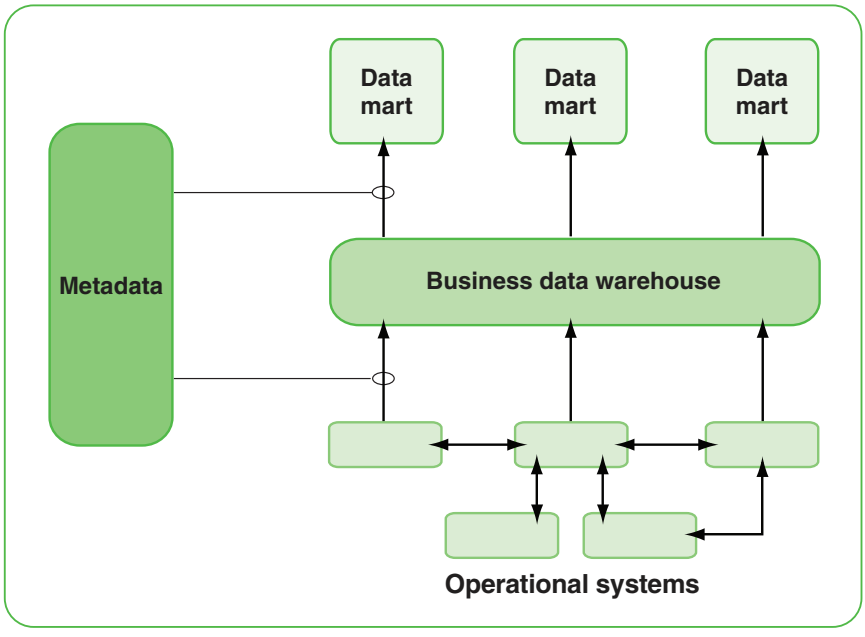

Figure 1: Data warehouse—three layer architecture.

The second reason for the three-layer architecture was to enable multiple business views built on a consistent information base. This requires a little explanation. First, it is recognized that the various operational systems have different views of the world, defined at different points in time and for varying purposes. The definition of "customer" in one system, for example, may be different from that in another. The record sets may overlap and the details may be inconsistent. To provide a consistent, overall view of the business, the first step is to reconcile the basic operational systems data. This reconciled data and its history is stored in the business data warehouse (BDW) in a largely normalized form. Although this data is consistent, it is still neither in the form required by the business nor can it be queried in a performant manner. The third layer in the architecture, the data marts, addresses these problems. Here, the reconciled data is further transformed into information sets that support end-user needs for different views of the business and can be easily and speedily queried.

One of the obvious trade-offs in the three-layer architecture is the introduction of a significant level of latency between data arriving in the operational systems and its appearance in the data marts. In the past, this was of little importance to most companies. In fact, many would have been happy to achieve the one-day latency that the architecture can easily provide, as opposed to the multi-week reconciliation timeframes they often faced. However, the emergence of e-business, customer relationship management (CRM), call centers and other initiatives in the 1990s demanded even shorter latency periods, down to sub-minute levels in some cases.

As shown in Figure 2, IT organizations responded by introducing the ODS and operational data marts into the data warehouse architecture. In contrast to the BDW and traditional data marts, which are read-only, these new components were characterized by the fact that end users could update as well as read data in these components. Architecturally, the ODS can be envisaged either as an additional layer between the operational systems and the BDW, implying that all data should pass through the ODS, or as a form of bypass to the BDW, as shown in the figure. The ODS is responsible for the transmission back and forth of near real-time data between the operational systems and the data marts. This necessitates a parallel process to subsequently reconcile data at the different levels of the architecture. However, introducing bidirectional and even circular data flows into the architecture can cause problems with data consistency throughout the environment.

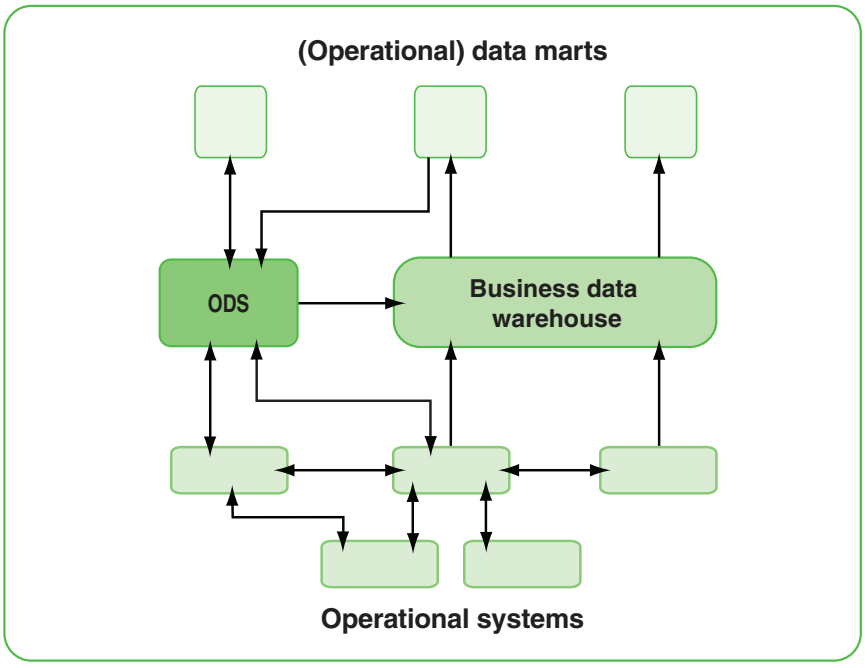

Figure 2: Adding the operational data store.

The creation of operational data stores in both warehousing and in non-warehousing projects has accelerated over the last few years. As a result, the complexity of these projects has increased enormously, as designers struggle to ensure data integrity in a highly duplicated environment while trying to reduce the latency of data movement between the layers. While many enterprises have successfully done this with IBM DB2 Universal Database™, a federated approach would be easier and more cost effective to use in many applications.

The increasing need for near real-time data access and read/write access to what was formerly seen as a purely informational environment are not the only problems that IT organizations face. In addition, there is a growing interest in combining traditional structured data with the unstructured variety.

Unstructured data, or content as it is also known, had been sitting quietly outside the borders of data warehousing for many years. Typically residing in proprietary content stores or flat files, such data had been largely ignored by the users and builders of data warehouses, although some analysts estimate it may comprise 85 percent of all digitized data. While there have been some attempts to link these different types of data through indexing systems, such projects have been overshadowed by the amount and complexity of work involved in creating data warehouses solely for structured data.

However, this situation has changed in the last few years. The Internet has spawned vast stores of largely unstructured content. As CRM has evolved, companies have begun to see substantial value in relating customers' transactions (structured data) to other interactions, such as phone calls, faxes, e-mails and so on, that consist largely of unstructured content. Textual descriptions, and even photos and videos, are also potential data sources, as they provide the context of the traditional transactional data. Slowly, but surely, the data warehouse has expanded beyond its traditional user base to the rest of the enterprise.

To date, data warehousing vendors have attempted to address these new requirements mainly through the expansion and improvement of traditional tool sets. Support for unstructured content has been built into relational databases. ETL vendors have added near real-time support, such as individual record handling, replication and support for message queuing, to their tool suites.

However, given the substantial revolution in both business and technical requirements previously described, it may be surmised that a gradual evolution in product functionality within the bounds of the existing architecture may be insufficient. The architecture itself needs to be revisited and extended. And so, it is time to step forward into the world of information integration.

Information integration—extending the data warehouse architecture

The primary underlying principle of information integration is that users should be able to see all of the data they need, as if it resided in a single source. In effect, information integration technology shields the requester from all of the complexities associated with retrieving data from diverse locations, with differing semantics, formats and access methods. Users, or applications acting on their behalf, are able to address the data through a standard language, such as structured query language (SQL), extensible markup language (XML) or a standard Web services or content API. As a result, they see information transparently, without regard for its physical implementation.

This result can be achieved in one of two ways, or indeed, by a combination of them. The two primary methods of information integration are: (1) providing distributed access to data through data federation, and (2) moving the data to a location that is more efficient or consistent for the application, an approach called data consolidation or placement. Distributed access corresponds to the Enterprise Information Integration (EII) category of technology, while data placement can be mapped to ETL and replication. Together, these sets of function form the heart of what is required to integrate information. In the simplest terms, federation takes a query in one location and distributes the appropriate parts of it to act upon the data wherever and in whatever form it resides. Data consolidation, on the other hand, combines data from a variety of locations into one place, in advance, so that a user query does not need to be distributed. Both approaches require extensive and largely common supporting functionality.

Both distributed access and data placement require underlying mapping, transformation and caching functionality. Furthermore, since the same data may, depending on the business requirement, need to be consolidated in some cases and federated in others, a common set of transformation and mapping functionality is required to support both approaches to maintain data consistency across the business. Mapping provides the ability to understand the relationships between different pieces of data. Transformation combines the data related through mapping, by providing the ability to transform data between different representations. Caching provides a temporary data store that can improve the performance of federation by transparently storing a local copy of a result set.

These functions depend on a detailed description of the environment in which they operate. This description includes business meaning, relationships, location, technical format and so on. In short, metadata. Such metadata must be both comprehensive and consistent, and be useful from the discovery and definition phases of an integration project, through to the operation of a federated query. A comprehensive and logically consistent set of metadata, whether materialized in a single physical store or distributed across multiple stores, is the fundamental basis for integrating information.

The relationship between information integration and data warehousing

Today's layered data warehouse architecture is founded on the premise that all of the data required for a particular end-user inquiry or report should be brought together in a single data mart or at least in a single data warehouse environment using ETL functionality. This is done to provide stability, consistency of data and guaranteed access to data.

But how can the new requirements for lower data latency, reduced storage of rarely used data and access to remote and varied data sources be supported? The answer clearly lies in the distributed query approach. Federation functionality provides the possibility of maintaining the logical appearance of a single warehouse or mart, without the prior physical movement of all of the data.

Does this mean the traditional approach to warehousing should be abandoned? Absolutely not! Federation neither could nor should replace the whole data warehouse approach. A fully federated or virtual data warehouse is not suggested, for all the well-known reasons of performance, inconsistency and autonomy. Rather, federation should be used to extend and enhance existing data warehouses to address specific business needs and in certain well-understood and limited circumstances. The information integration approach to data warehousing therefore embraces both data placement and federation.

Real-time data access

Federation can play an important role when businesses require access to specific items of real-time data in combination with traditional, historical or analytical data already resident in the data warehouse. This is shown in Figure 3. An end-user query or report that is largely based on historical, consolidated data available in the data mart also requires some up-to-the-minute information. In a traditional data warehouse architecture, this real-time data must be fed continuously into the data mart, usually through an ODS. Not only must significant quantities of such data be stored in the mart, but the ETL environment must be capable of sustained near real-time throughput.

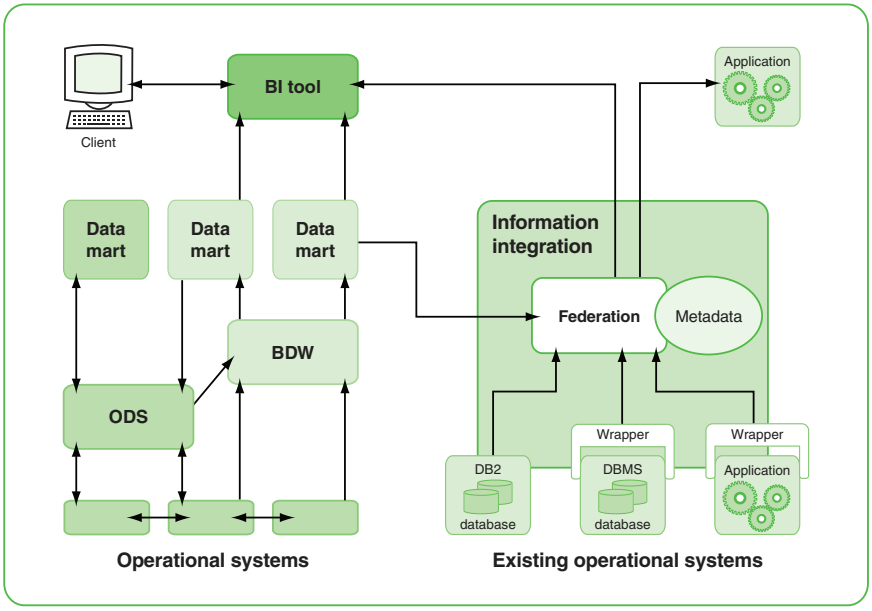

Figure 3: Federated access to real-time data.

Federation provides a simpler and more elegant solution in many situations. When the end-user query is run, a simple request for a specific piece of information can be sent to the operational system, and the result returned and joined with the information retrieved from the data mart. Operational systems that can be accessed in this way include those based on DB2 Universal Database, a variety of relational and non-relational databases, as well as IBM WebSphere® MQ queues and Web services. This functionality eliminates the need to store near real-time data in the mart or to enable the ETL environment to handle such data.

Note an important qualification. The query that is sent to an operational system should be simple, of a type that the operational system was designed to handle efficiently, and return a few specific pieces of information. This limits any performance impact on the operational system and network.

Federated query uses standard SQL, which allows for transparent use with existing business intelligence (BI) analytical tools. In this way, existing BI tools can access local and remote relational and non-relational data. This protects the business investment in existing tools and leverages IT developers' skill and expertise in working with these tools and their SQL-based paradigm. Federation is not limited to accessing real-time data. Any data can be accessed in this way, removing the need to store it in the warehouse or mart. It is well known that much of the data in warehouses is there in case it might be needed. However, in many deployments, a lot of this data—20 to 50 percent—is rarely accessed at all. Where data is used infrequently and already exists elsewhere, federated query allows access to such data in its original locations. When the data is historical, it may be necessary to keep it in the data warehouse since the only other copy is on magnetic backup tape. But when the required data remains in the operational system database, federation can allow its elimination from the warehouse while still supporting end users' needs.

And there is an additional benefit. With such a federated infrastructure in place, organizations can also enable operational applications to easily access data in the warehouse and join it with existing operational data from distributed sources, as shown at the top right of Figure 3. While this is not a warehouse application in the strict sense of the word, it clearly provides additional possibilities to reuse warehouse data.

Access to unstructured content

Figure 4 shows an additional way in which federation can extend the data warehouse. In this case, the business requirement is to incorporate some unstructured data or content within reports generated in the data warehouse environment. In the traditional data warehouse architecture, the approach would be to load the required content from the source into the warehouse and then query all of it in the normal way. However, such data is usually voluminous. Even when organizations are willing to have such large quantities of data in the warehouse, other problems arise. Such content may be volatile or outside the organizational span of control, on the Internet or in a partner's data stores, for example. In this instance, it may be difficult to know when the data has changed, and thus when to load a new version.

Again, the benefit of federation is that it allows access to the content when and as required. When the report is run, a sub-query is dispatched to the original content source and returns only the information required in its most up-to-date form.

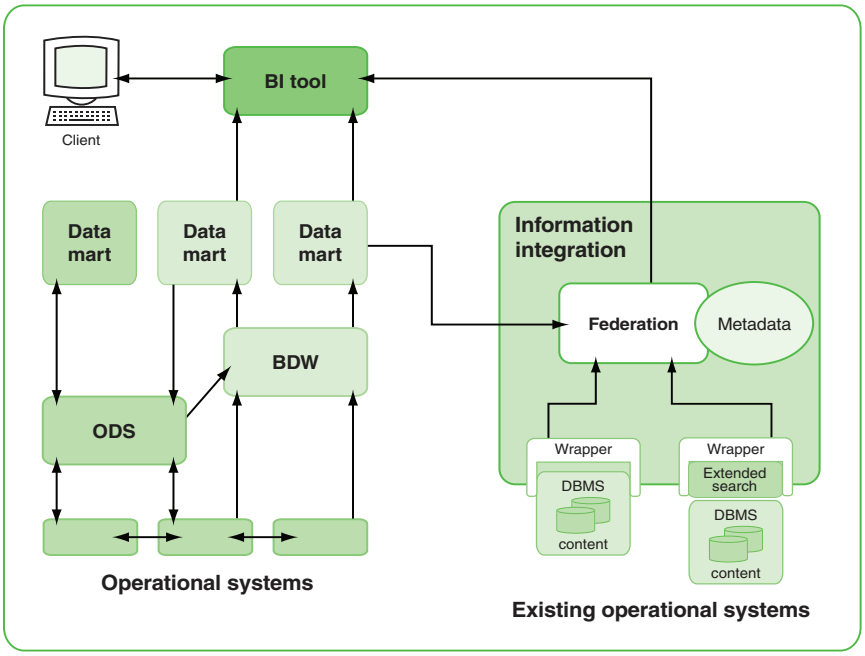

Figure 4: Federated access to unstructured content.

Federating data marts and business data warehouses

It bears repeating that IBM does not recommend eliminating data warehouses and marts by shifting user query and reporting to a pure federated infrastructure layer. Virtual data warehouses have been tried many times and most have failed to deliver the value end users require. Federation does not replace data warehouses. Federation extends the existing data warehouse concept.

A third possible extension to the data warehouse is shown in Figure 5, which addresses a situation that is widespread throughout the industry today. This is the much-maligned, but ubiquitous, situation of having more than one data warehouse within a company. This situation can arise from mergers, acquisitions or simply as a result of independent or uncoordinated investments in different divisions. As these separate warehouses appear or grow, it is usually not long before management wants to compare or join the information residing in multiple warehouses. This is a difficult requirement to satisfy through the traditional data warehouse architecture. In this model, the approach would be either to try to load the contents of the second warehouse into the first one, or to create an overview warehouse containing data from both original warehouses. Quite apart from the volumes of data involved, there is the substantial problem of trying to create a unified data model that covers both sources and allows the data from one warehouse to be loaded into another.

In this case, a federated approach again provides a much simpler solution. A federated query addresses only the subset of the data needed to answer the management request. There is no need to load all of the data from one warehouse into another, avoiding the need to create at least one extra copy of the data. While the difference between the two models does not go away, the federation approach allows incremental focus on only one subset of the model at any one time—the part that supports the federated query.

As is also shown in Figure 5, one or more of the BDWs can be included in the federated query. This allows the inclusion in the result set of detailed-level data, which was not previously included in the data mart in the ETL population step. Clearly this use of federation is not limited to cases where there are multiple warehouses. It is equally applicable in a single-warehouse environment, allowing occasional access to data at the BDW level for users of specific data marts.

It should be noted that this is an approach that can be grown over time, increasing the scope of the federation at each step until eventually all the data in the marts is available together. In this way, the almost inevitable inconsistencies in meaning or content that have arisen among different warehouses as they grew, can be gradually discovered and addressed incrementally. Finally, the business can decide whether or not to physically combine the original marts. With the analysis now complete, the design of the combined mart is significantly simplified. In this manner, the federated approach delivers incremental value with staged effort throughout the process—a very desirable attribute of the approach.

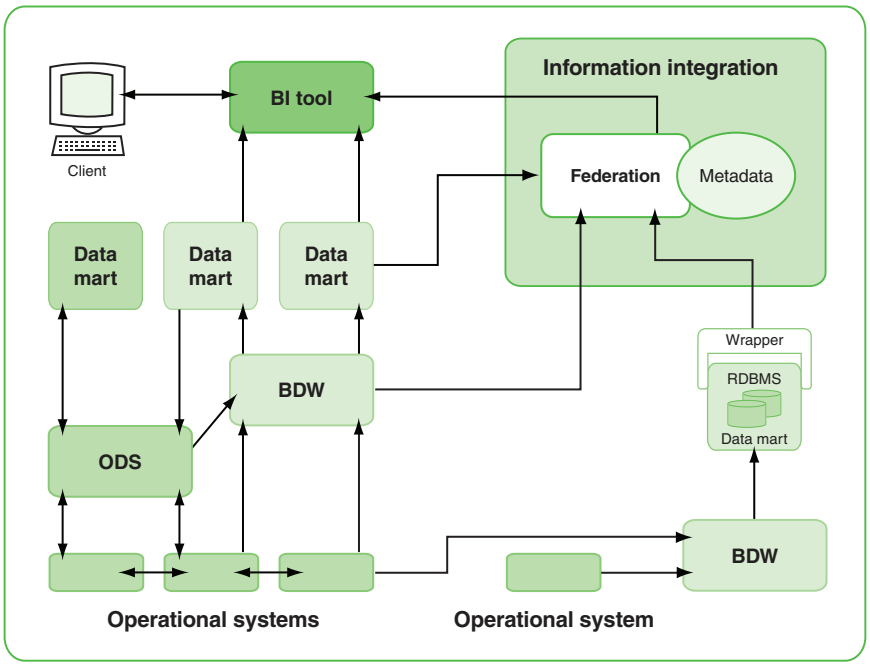

Figure 5: Federated access to data marts.

To federate or not to federate

But, what are the trade-offs of using this approach? One fact to consider is that the federated queries act against remote sources, including operational systems. While one might argue that this could impact the operational application's performance, this can be mitigated by sending only simple and specific queries to the operational system. This is an entirely different matter from sending complete, complex end-user queries into the same environment. In the federated case, the implications for performance can thus be predicted and managed.

Another potential problem is how to logically and correctly link data in the warehouse environment with that in the remote systems. This is the same problem that must be addressed when designing the ETL population processes of a warehouse. The same detailed analysis and understanding of the sources and their relationships to the targets is required. Proper data modeling is still necessary. On occasion, it will be clear that the relationship is too complex, or the source data quality too poor to allow federated access. In some cases, if one understands the design of the ETL processes that populate the data warehouse, then one can re-use that design when building the federated query. In general, federation does not, in any way, reduce the need for detailed analysis or modeling. In fact, it may necessitate more rigor in the process, because of the real-time, inline nature of any required transformations.

These considerations govern the circumstances under which federation can be used to extend the data warehouse. Federation is a powerful approach when one needs access to real-time data, to content that is not easily stored in the warehouse or to data that is seldom used. It is more appropriate for occasional queries than for ones that are regularly and predictably repeated and can benefit from pre-processing of the source data. It is also useful for queries that must access non-relational data.

However, where complex transformations or cleansing of the source data is required, federation cannot easily cope, and loading the data into the warehouse is recommended. If a complex query is predictable and regularly repeated, it may be more reasonable to load the data once into the warehouse and access it locally.

Clearly, federation is not a panacea for all data access problems; but it is quite capable of addressing some well-known requirements. One can also envisage that as the federation tooling is improved and the enterprise data environment is better integrated, the opportunities for using federation will expand. Specifically, we anticipate Web services to be capable of providing more sophisticated data transformations and cleansing in real-time, extending the ways federated queries can be used.

Upon accepting the possibility that federation allows data to be read from the operational systems as part of a data warehouse query, the next logical question is if federation should be used to write data back into the operation system. The technology provides for this, but in order to minimize the potential impact on data integrity and security within the operational environment, this should be done through the applications responsible for creating and maintaining the data in these operational systems.

Using IBM DB2 Information Integrator in your data warehouse

IBM DB2 Information Integrator, along with its predecessor products—IBM DB2 DataJoiner® and IBM DB2 Relational Connect—support IBM's vision of information integration. DB2 Information Integrator provides EII functionality, allowing access and query across an integrated view of diverse and distributed data. Typically, the data resides in a variety of databases and in many formats, including:

- DB2, Informix, Oracle, Sybase, SQL Server and Teradata databases
- XML, ODBC, OLE DB and Microsoft Excel file formats
- Web services, message queues, flat files and IBM Lotus® Extended Search data sources.

The following examples illustrate how the three architectural patterns can be incorporated in typical data warehousing situations, using DB2 Information Integrator.

Getting to up-to-date account information through the warehouse

Consider a call center in a bank or other financial institution. Agents have access to a variety of information about customers through the data warehousing infrastructure. Such information may include details of transactions over some period of time as well as summarized or derived information showing behavioral trends, marketing opportunities and so on. In this scenario, the most recent information available in the warehouse is as of close-of-business yesterday. And, most of the trending and marketing information is only updated monthly, so it may be a number of weeks out of date.

It is readily apparent with this environment that agents will be unable to easily support customers who make inquiries about today's transactions. Furthermore, the agents find it difficult to use customer calls as potential sales opportunities because of the lack of real-time information on the calling customer who moments before performed an online banking transaction. If the bank were to use only the traditional data warehouse architecture, solving the first of these issues would involve running the ETL processes in near real-time mode, while the second issue would lead to storing huge amounts of additional data in the warehouse, of which only a small portion would ever be used when specific customers call.

DB2 Information Integrator can solve these problems, using the approach shown in Figure 3. DB2 Information Integrator provides the federated query function to allow direct access to up-to-the-minute data on customers residing in DB2 and other relational and non-relational databases.

When a customer calls with a problem relating to today's transactions, the agent can run a simple SQL query (simple from the agent's point of view, that is) that joins consolidated customer information in the warehouse with the day's relevant transactions across the branch system, ATM and Internet banking channels. Within this join, DB2 Information Integrator maps from the consolidated customer number in the warehouse to the potentially various customer numbers in the underlying operational systems, submits relatively simple queries to each of those systems and finally joins the result sets for the agent. This is likely to be a simpler and less costly approach than upgrading the ETL system to near real-time.

DB2 Information Integrator can also help the agent to make more informed choices when considering marketing to a customer who has called in for another reason. Here, trend or opportunity data in the warehouse is joined with real-time indicators such as account balance or recent major transactions from the operational systems. The business benefit here is achieved without the need to store substantial amounts of data that may never be used in the warehouse.

Supply chain optimization and business activity monitoring

A classic stress test for business intelligence applications is managing inventory levels in hundreds of locations, from hundreds of suppliers. To accomplish this, many retail distribution companies have turned to digital dashboards to monitor stock levels and prevent the dreaded "stock out" situation when a customer wants to buy an item but the shelf is empty. This is actually a real-time information integration problem. The dashboard must be based on exception alerts, because there are anywhere from 5,000 to 30,000 items being tracked daily in, say, 250 stores. Some stock items can be sold out in one day when unanticipated events occur. The production system can generate a simple alert to the dashboard saying, "We crossed the threshold level—order more." But it is now up to the knowledge worker to take action. Ideally, the knowledge worker would like to push a few buttons and have a variety of information gathered and delivered quickly. The information that is needed to make a decision includes:

- Stock level history on the alerted item for the last three weeks compared to a year ago for the same items (from a DB2 data warehouse)
- The current invoices and shipment orders for those items to that store (from the ERP procurement system)
- A summary of stores within fast shipping distance who may have some excess stock to offer (from a "shipping" data mart not based on DB2)
- A quick check of promotions in that store to see what may be driving the stock level down (from a content management system holding coupons and advertisements)
- The Top 10 news items in that store region for the last few days (Web services news feeds)

In order to deliver to the knowledge worker all these facts as the basis for hundreds of "on the spot" decisions, the choice is between building an enormously complex application program and using DB2 Information Integrator. The federated system can gather this information in a single request to a variety of heterogeneous databases as well as unstructured data. Within seconds, the knowledge worker can determine that no shipments are likely to arrive soon, find the reason for the stock out, locate the most likely sources of replenishment, and route inventory from the stores that have excess to those more in need. Then, on to the next alert.

Executive information systems

Today's executive information systems focus largely, if not exclusively, on providing structured data to their users. However, it is widely recognized that a substantial amount of the input to executive decision-making is in the form of unstructured content. Such content is excluded from current executive information systems because of difficulties in accessing it, or joining it with structured data or the volumes it would represent if stored in the data warehouse. Furthermore, such content is often sourced externally and may be rather volatile.

DB2 Information Integrator addresses all of these problems using the approach illustrated in Figure 4. Together with Lotus Extended Search, DB2 Information Integrator provides tools and access to a wide variety of content stores. The tools enable relational data to be joined with unstructured content, providing that such unstructured content is suitably keyed. Thus, information integration provides an ideal means of accessing diverse, distributed unstructured content across the breath of the Internet or intranets. With federation operating in the background, the data warehouse can provide a continuous stream of relevant content through the executive information system without necessarily storing all of that content locally. Of course, when appropriate, DB2 Information Integrator can also cache the data locally to enhance performance.

Rationalizing government agency data marts

Many large companies today have multiple data warehouses, each supporting multiple data marts, which provide silos of business information aligned to departmental or functional reporting needs. This is probably most evident in the government sector, where diverse agencies gather and carefully guard overlapping and often conflicting sets of sensitive information about people, businesses and resources. Both practically and politically, it would be unfeasible to combine such warehouses into a single, overarching mega-warehouse.

Nonetheless, the requirements today for improved manageability, reduced inconsistency and, especially, increased security all contribute to the desire to somehow create a merged view of all this information. The third architectural configuration of information integration, as shown in Figure 5, provides a technological foundation to address these needs; although, the data archaeology and modeling efforts that will be required must not be underestimated. The cost of such efforts may dwarf the technology investment required, and is likely to be similar, irrespective of the approach taken—be it a new mega-warehouse or a federated approach.

Using DB2 Information Integrator as the federated query engine allows organizations to tackle the problem gradually. Rather than needing an understanding of the relative mapping of all of the data in two warehouses and then having to define and deliver a method for combining them, DB2 Information Integrator starts with small mappings that deliver immediate value and over time expand the solution. Thus, users of data mart A, for example, can use a federated query to join a subset of information from data mart B to their own information. Over time, further subsets of the information in data mart B are included in federated queries, and the base of metadata or mappings between the two marts grows. The same can happen in the opposite direction, as well as between other sets of marts within the environment. This growing store of metadata can, in the future, become the basis for rationalizing the storage and distribution of this information.

The programmer and database administrator views

What effort is required to achieve the benefits of information integration? For the programmer, DB2 Information Integrator actually simplifies a complex environment. With a single SQL statement, the "average" programmer can access data from several databases, be they DB2, Oracle or SQL Server, as well as from non-relational data sources. DB2 Information Integrator simplifies program complexity by eliminating the need to know multiple SQL dialects, by managing connections to multiple databases simultaneously, and by managing complex join logic to correlate multiple data sources. This not only reduces program complexity, it also reduces the level of skill needed for the programmer to cope with a heterogeneous IT environment.

It is the responsibility of the database administrator (DBA) to set up the environment to enable this. Through the DB2 Control Center, the DBA defines the critical data sources—the type of data source, the servers they reside on, user mappings and the mapping of the source fields into a relational schema, known as a nickname. In many cases, the DBA will also define simple transformations that allow data in one database to be joined in a rational context to other databases. For example, the DBA may need to define that male and female codes "M" and "F" replace "1" and "2" from one of the data sources. Once the DBA sets up the nicknames, server addressing and simple transforms, the programmers are ready to use DB2 Information Integrator.

Conclusion

As business users require access to more of their organization's data resources, integrating information becomes increasingly important. Data federation provides a number of interesting possibilities for data warehousing; however, federation does not supplant data warehousing. Nor does the federation approach dispose of the need for thorough investigation and detailed modeling. The functionality allows the extension of traditional data warehousing paradigms. In many cases, it allows organizations to access data outside of the warehouse environment as if it were available in the warehouse itself.

Leveraging DB2 Information Integrator and federated queries, data from different and diverse sources can be joined. Thus, federation can provide a cost-effective and relatively quick means of providing access to data that is not in the warehouse along with the data that already resides there. However, it must be confined to carefully selected subsets of the data, chosen to respect network bandwidth, source application performance and data quality issues. The process can also transform the format of the data, making logical and reasonable changes. However, federated queries cannot easily cleanse data of inaccuracies or inconsistencies. For data cleansing, organizations must typically go through the process of loading the data into the warehouse. In short, federation allows the extension of the data warehouse architecture to support real-time access to data and content in its original source and form. It is a powerful tool, and must be used with care and precision.

Federation complements the traditional ETL and replication approaches used to populate the warehouse. This approach to integrating information provides powerful functionality for building, maintaining and evolving the data warehouse. It will also form a solid foundation of comprehensive and consistent data for the integrated enterprise. Creating a data warehouse is an initial step toward integrated information access. Combining the warehouse with DB2 Information Integrator will help organizations realize the vision of delivering real-time data access as well as historical information in support of decision-making.

For more information

Please contact your IBM marketing representative or an IBM Business Partner, or call 1-800 IBM CALL within the U.S. Also, visit our Web site at:
ibm.com/software/data/integration

IBM Corporation
Silicon Valley Laboratory
555 Bailey Avenue
San Jose, CA 95141
U.S.A.

Printed in the United States of America
03-03

Printed in the United States on recycled paper containing 10% recovered post-consumer fiber.

GC18-7562-00