RAPID, COST-EFFECTIVE DEPLOYMENT OF DATA SERVICES IN A SERVICE-ORIENTED ARCHITECTURE
IMPLICATIONS FOR AN NCES MEDIATION SERVICE

2 INTRODUCTION
2 THE DATA ACCESS CHALLENGE
3 THE SERVICE-ORIENTED ARCHITECTURE
3 LEVERAGING ENTERPRISE DATA ASSETS
  4 Enterprise Information integration (EII)
4 MODEL-DRIVEN DATA SERVICES
5 THE METAMATRIX SOLUTION
7 LEVERAGING DATA MODEL INTERCHANGE STANDARDS
9 SOA ARCHITECTURE CONSIDERATIONS
  9 Converting data to XML
  9 Semantic mediation language
  9 Optimal location for mediation processing
10 IMPLICATIONS FOR AN NCES MEDIATION SERVICE
10 CONCLUSION
11 ABOUT METAMATRIX ENTERPRISE
11 ABOUT RED HAT

www.jboss.com
INTRODUCTION

The service-oriented architecture (SOA) and its most common implementation, web services, represent a new evolution in application architecture for the enterprise. Organizations can significantly increase their agility in the face of change, improve their operating efficiency, and greatly reduce the cost of doing business. However, many organizations with significant investments in data collection and storage technologies are still struggling with how best to embrace and deploy the new architecture. One reason is the vast amount of data in existing data stores—organizations can’t afford to reengineer or replicate all of them. Another concern is balancing the need for information sharing with the need to protect and secure critical information systems.

The challenge for the enterprise is to achieve data, business, and process integration rapidly, cost-effectively, and incrementally while ensuring immediate and ongoing benefits. In real-world deployments, companies also face the challenge of providing a service-oriented solution that loosely couples data sources with applications while providing integrated access to an organization’s data assets and an acceptable level of control, performance, and scalability.

The MetaMatrix Enterprise Data Services Platform from Red Hat provides a data services management system that allows organizations to rapidly deploy their legacy, relational, or other data sources into a service-oriented architecture. MetaMatrix can cost-effectively expose their varied, distributed, and heterogeneous data assets as federated, scalable, secure data services in an SOA architecture. These data services can fully support standard data models, particularly as those models evolve and change.

This paper describes the numerous challenges faced by federal agencies seeking to embrace SOA and details a metadata-driven, model-based approach to addressing them. It also discusses the implications of constructing a standard data mediation service as required by the Department of Defense’s Net-Centric Enterprise Services (NCES) initiative.

THE DATA ACCESS CHALLENGE

Today’s organizations are compelled to share their data assets with users and communities of interest (COI) for whom those assets were never originally designed. The business needs driving these requirements include supporting new users and use cases as well as enabling greater integration and reuse of existing data resources and applications. Meanwhile, financial constraints dictate that enterprises maximize the use of existing systems, rather than make broad investments in new information assets.

Enterprises typically maintain information in multiple types of systems. Often, these systems have grown organically, rather than in a structured manner, in order to respond to rapidly-changing business needs. Information silos—disconnected and seemingly unrelated to each other—are the most common result. And yet much of the information is related. The enterprise that can integrate and leverage these disparate data sources gains tremendous value.

The challenge is to achieve data, business, and process integration rapidly, cost-effectively, and incrementally while ensuring immediate and ongoing benefits. New architectures, technologies, and products have emerged to make it possible.
THE SERVICE-ORIENTED ARCHITECTURE

Enterprises are increasingly looking to service-oriented architectures (SOA) to integrate their systems. An SOA is a set of policies, practices, principles, and frameworks that allow for the encapsulation of data and processes as a set of software services. It uses standard interfaces and protocols that can be accessed by a growing and ever-changing community of information consumers. By leveraging the industry-standard XML, SOAP, WSDL, and UDDI protocols, services can be published, discovered, and used in a technology-neutral way. Such an architecture can begin to bridge the gaps between the information silos.

This approach has some key advantages, including:

- Enabling reuse - using services, not copying code or implementations
- Published interfaces - the service is made available and its interface is explicitly and precisely described
- Formal definitions - the provider and consumer understand and accept the rules of interaction
- Enforcing abstraction - all aspects of the service implementation are hidden
- Functional relevance - functionality can be presented at a granularity that is appropriate and useful for the consumer

Those advantages give companies increased agility to cope with changing business needs and requirements. They get increased visibility into current operations by identifying and encapsulating key business functionality, and reduced complexity, risk, and cost w integrating systems.

LEVERAGING ENTERPRISE DATA ASSETS

Even with all the components of an enterprise-ready SOA architecture in place, creating the services to populate that architecture can be an enormous undertaking. The siloed information assets may be managed and maintained in multiple locations. Examples include modern and legacy relational databases, proprietary applications and services, text files, spreadsheets, ERP and CRM systems, and data warehouses and data marts. There are significant roadblocks to accessing and integrating the data in these systems, because of the issues with:

- Dislocation - the access technology varies from system to system
- Security - with typically source-specific protocols and policies
- Disparity - with wide variation in data semantics, syntax, and structure
- Interfaces/reuse - information is not reused or shared, and often there is no complete knowledge of the information that is available within an enterprise
- Data quality - not all data is good data

It is essential to realize that these roadblocks to data and information sharing will always exist. When finding a means to integrate the disparate data assets, it’s important to also recognize and maintain their autonomy.
**ENTERPRISE INFORMATION INTEGRATION**

Enterprise Information Integration (EII) has emerged as a powerful new technology for building SOAs. EII enables seamless access to existing data sources, both singly and with the ability to combine data from multiple sources to create new integrated business artifacts. This better enables the reuse of existing information data sources as reusable services, giving the enterprise tremendous agility for adapting to changing business needs and requirements. Companies benefit from large cost reductions and increased operating efficiencies because existing data sources and services can be repurposed and reused, avoiding the need to build more siloed data stores to serve changing requirements and evolving communities of interest. EII also allows them to discover and potentially eliminate redundant or duplicate data stores. Integration can be accomplished in less time and with fewer resources, particularly if the solution makes use not only of the data resources in the enterprise, but of the metadata describing them.

**MODEL-DRIVEN DATA SERVICES**

In April 2007 Red Hat acquired MetaMatrix, the developer of the market-leading EII offering. Now part of the JBoss Enterprise Middleware family of products, the MetaMatrix suite has added substantial new functionality that enables the creation of data services, façades encapsulating existing information resources. They allow seamless access, federation, and deployment of disparate and legacy data sources into a service-oriented architecture. Data services go beyond the traditional concept of a federated data object and include rich semantic and syntactic mediation capabilities, as well as realtime mapping and transformation to structured XML documents for use in a web service.

An integration solution needs to provide a practical approach to solving both bottom-up (data driven) and top-down (business process driven) information problems within a SOA. For the bottom-up problems, the solution needs to provide access to information in the existing physical data sources as quickly as possible and with the least amount of effort. For the top-down challenges, the solution must also be able to provide higher levels of data abstraction to support the longer term goals of SOA and decouple the data from the applications.

In the short-term, the challenge is to find a way for other parts of the organization to share existing information, regardless of the physical limitations imposed by the data source's structure and technology. This short-term problem addresses the real-world situation most organizations face as they attempt to make quick progress in making their information assets more accessible and reusable. Data services can solve this problem and can provide a much needed abstraction layer that enables eventual migration from existing data environments to new ones.

To truly align IT with business requirements and make SOA initiatives more useful and agile, data services must become part of an organization’s long-term strategy. Web services would ideally be based on coarse-grained business processes, rather than fine-grained components. To achieve this agility, organizations need a distributed development process—business analysts drive process articulation, architects conceive of aggregate services, and developers easily discover, publish, and implement business services. To do so, enterprise-wide business data must be associated with each service without being hard-coded into the service. Data services provide the ability to enable a collection of both data and application services that are consumable by the business.
THE METAMATRIX SOLUTION

MetaMatrix Enterprise uses a model-based, metadata-driven approach to creating data services that addresses both bottom-up and top-down requirements. By using the knowledge enterprises have of the underlying structure and semantics of their data—both their data sources and the applications that consume them—data services can be rapidly and cost-effectively defined and deployed in a web services architecture. With the power of a metadata-driven architecture (rather than costly and brittle programmatic solutions), MetaMatrix provides extensible and scalable infrastructure software for defining, managing, and accessing enterprise information sources (EIS).

The MetaMatrix product offering consists of three components: the MetaMatrix Designer, the MetaMatrix Repository, and the MetaMatrix Server, shown in Figure 1.

Complete metadata descriptions of data producers and consumers are captured in the Designer, managed in the Repository, and deployed to the Server to enable real-time, federated SOA data services.
The MetaMatrix Enterprise Data Services Platform consists of three components:

- MetaMatrix Designer, an Eclipse-based GUI modeling environment
- MetaMatrix Repository, a full-featured metadata store
- MetaMatrix Server, an enterprise-capable run-time environment

These three integrated tools allow users to capture, model, store, and manage metadata for enterprise information systems. They then dynamically deploy the models to drive optimized, high-performance, real-time data integration. Using the Designer, representative models of the physical data sources are created, often automatically. New models representing combined, transformed views of those sources are then created, building abstractions of the underlying data sources that are application and service-oriented, rather than data-centered. Because of their abstract nature, they are called virtual models, but they can appear just as concrete as physical models to a user of the system when they are deployed to the MetaMatrix Server. Mappings between the models are then defined, including unions, joins, selection criteria, procedural functions, and transformations. This is where the details of the data integration are defined, including commonly needed data reconciliation functions like name, attribute, and datatype conversions. This is also where transformations, mappings, and reconciliations to various standard data models can be specified (see following section).

Once the models and their associated transformations are defined, they are deployed to the MetaMatrix Server, where they can be directly executed to provide rapid, real-time access to data across the enterprise. These data services can be automatically deployed as web services by registering the WSDL describing them into any industry-standard UDDI registry. They can also serve as reusable components for creating new services and functionality.

The Designer demonstrates a particular strength of this metadata driven, model-based approach. It can be used to define the mappings from relational and other sources into the structured XML documents required for web services. For example, in Figure 2, an XML schema representing a portion of the Global Force Management data model has been imported into the Designer for use as a template to create compliant XML document responses to a web service request. The data fields comprising a fragment of the hierarchical document, in this case the Equipment object, are automatically mapped to a specialized virtual model called a mapping class. A transformation node (the yellow “T”) is associated with each mapping class, and defines the syntactic and semantic mediation necessary to populate the data fields from the back-end data sources.

The models representing physical sources are shown as blue in the Designer. Note the Input Set parameter that contributes to the transformation. It allows identifiers and other fields higher up in the hierarchical document to be propagated to the XML fragment. This in turn allows the user to select the parts of the XML document to return. That way mappings can be performed to highly complex XML schemas (including simple or complex datatypes, sequences, choices, association groups, and recursions) without any custom coding. As the sources or schemas change, the mappings can be easily and rapidly modified, leading to remarkable agility on the part of the enterprise. The ANSI-standard SQL that comprises the transformation (which is generated automatically by the Designer) is shown in the Transformation Editor, where it can be examined or edited.
FIGURE 2

The Enterprise Designer, showing a mapped portion of a structured XML document to the underlying data sources via a user-specified transformation. The Transformation Editor shows the SQL comprising the transformation, which can perform semantic as well as syntactic mediation.

LEVERAGING DATA MODEL INTERCHANGE STANDARDS

Using models to drive integration has another substantial benefit—rapidly and seamlessly using data model standards that may be defined by a COI to assist data exchange and integration. Organizations are realizing the great benefits of standardizing the terminology and semantics they use to describe important entities within their business processes and workflow. Examples of such models in the federal government include C2IEDM, GJXDM, JC3IEDM, GFM, DDMS, and NIEM. By creating domain data models with agreed-upon standards for data element names, datatypes, and semantics, organizations can deploy a common semantic layer that can be used to rationalize and reconcile disparate contributing data sources, as well as help define interfaces to various consumers and COIs.
However, there are challenges both in defining such a standard and in applying it to the enterprise systems. One of the biggest challenges is determining how to mandate adherence to the new (and typically evolving) standard from the contributing sources and systems. Rather than force every data source in the enterprise to change to adopt these standards, MetaMatrix allows for the semantic mediation and transformation necessary to occur in the integration layer. Because of the model-driven approach, MetaMatrix is uniquely positioned to allow for the rapid use of such data models (e.g., XSDs, UML, etc.) to serve as templates for data integration, reconciliation, and data service creation. The data exchange model is represented by a virtual model layer in the Designer where mappings and transformations between sources and services can be easily specified. The models and transformations that are thus created can be thought of as a “future-proof” abstraction layer - new mappings can be quickly modified to adapt as current standards evolve and new standards appear.

**Figure 3**

Figure 3. Standard data models can be imported directly into the MetaMatrix Designer for semantic rationalization and integration without requiring changes in the source data.

For example, in Figure 3 the C2IEDM data model is in Designer and being used both to rationalize the differing semantics used by the underlying data sources and to consolidate the disparate data in those sources to COIs. This system can form the basis of a core mediation service as described by the NCES architecture.
SOA ARCHITECTURAL CONSIDERATIONS

A common approach when architecting a new SOA architecture is to convert all underlying data sources to a common XML format and then expose them as fine-grained data access services. While useful for some applications, this strategy has unfortunate implications both for network traffic and overall performance. The problem is exacerbated by vendors of more XML-centric products who tend to encourage this approach.

CONVERTING DATA TO XML

Most enterprise data is stored in relational tables, and even those data sources that are not explicitly relational (such as spreadsheets) have an inherent relational structure. Transforming these tables into XML documents can be time-consuming. It may also cause an explosion in the amount of data being transferred, as the amount of additional data comprising the associated XML tags is substantial. The performance penalties of turning all enterprise data into XML do not end with the initial serialization, but propagate through the system as the XML must be continually reparsed and repackaged. Depending on the size of the datasets involved, the impact on network bandwidth can be severe. It’s not uncommon to see data sizes increase by orders of magnitude.

SEMANTIC MEDIATION LANGUAGE

If all of the enterprise’s data has been recast into XML, then all subsequent processing must be performed using XML-specific techniques such as XSLT. While highly general and very powerful, XSLT has severe limitations, both with respect to performance and capability. The amount of data that needs to be processed expands tremendously due to the need to capture the hierarchical relationships between elements (which are natively preserved if the data is kept in a relational form). Then the bloated data stream must be successively processed by general purpose engines. These problems impact performance.

But if the federation and semantic mediation is instead performed close to the data sources themselves using industry-standard SQL as the transformation language, then the highly specialized and tuned databases that support this processing can be directly invoked. On the other hand, for certain applications that involve a single source transformation to single target architecture, XSLT provides an elegant solution.

OPTIMAL LOCATION FOR MEDIATION PROCESSING

It’s highly preferable to delegate processing to those resources that are best suited to perform it, rather than spool large amounts of data to general purpose processors. In addition, very common mediation activities, such as federation of multiple sources, is not easily supported by XSLT-style processing. For example, consider an FBI investigator who wants to perform a query on a suspect. The pieces that make a complete picture of a person, including a physical description, address(es), name/alias information, picture, rap sheet, and so on, aren’t likely to come from one data source. Nor are those sources likely to agree on the semantics used to describe each of their data elements (although the FBI is addressing this with their GJXDM and NIEM data models). Substantial federation and semantic processing will be required to return a complete set of information to the investigator.

Now imagine that the investigator is performing the query for someone named Jones with a particular birth date and his last known residence. Contrast the approach of pushing these search criteria to the various underlying sources and returning only the pertinent records via a standard XML interchange format, versus surfaced every entry on “Jones” as an XML document and performing the searching/merging/semantics on a massive number of very large documents.
Certain types of mediation and semantic conversion are better done at different layers in the architecture, depending on the number of sources, complexity of queries, application requirements, and so on. An integration engine needs to be able to make optimization decisions about where to best perform the required processing.

**IMPLICATIONS FOR AN NCES MEDIATION SERVICE**

As the Department of Defense’s service-oriented architecture, NCES must be applicable to and useful for a wide variety of potential uses. Therefore, data for ever changing and evolving COIs needs to be exposed while providing a robust and stable information abstraction layer that is independent of COI-driven requirements.

As outlined in the previous sections, where mediation occurs and the technologies used to enable data for web services (i.e., create data services) are important considerations in implementing an SOA. MetaMatrix Enterprise, by capturing these requirements as virtual models that can then in turn be used to directly drive integration, can provide federation and semantic capabilities that are highly efficient. It does so by performing mediation processing in the architectural layers best designed to perform it and by exposing the resulting consolidated and rationalized services as either reusable business objects or as SOAP-based XML data services.

**CONCLUSION**

IT organizations are eager to find practical, incremental, and enterprise-ready technology to help transform their enterprises to service-oriented architectures. With MetaMatrix’s metadata-driven, model-based technology for rapidly and cost-effectively turning existing data assets into SOA data services, enterprises now have the capability to address today’s challenging integration needs.

Data services require a different way of thinking about the performance implications of web services technology. The ability of EII technologies to provide high-performance query federation lends itself to solving this problem. As with any architecture, the performance of an organization’s data services will be crucial to SOA success, so architects will want to minimize the volume of data that flows “over the wire.” The architecture underlying the MetaMatrix product suite is ideally suited to addressing this challenge, with the capability of applying selection criteria and transformation logic to data sources before the results of the data service are converted to XML.

The MetaMatrix Enterprise Data Services Platform enables companies to rapidly and cost-effectively build successful data services from the valuable corporate data already maintained by the enterprise. Information services at the granularity appropriate for the information consumer can be provided, abstracting the information consumer from the implementation and dynamically the changing data landscape. Domain data models used by the enterprise to assist in conforming can be seamlessly incorporated into the architecture via MetaMatrix’s unique model-driven approach to integration and mediation.

MetaMatrix makes distributed data accessible and manageable, breaking through the traditional barriers of location, structure, semantics, and context.
The MetaMatrix Enterprise Data Services Platform from Red Hat is the only solution that combines the ability to access and manage all forms of information across the enterprise in a centralized platform. The platform provides a flexible, secure information layer for service-oriented architectures and application development. With MetaMatrix, organizations improve corporate performance by extending timely, contextual information to the people who need it.

MetaMatrix Enterprise enables companies to:

- Access all information sources, regardless of source or storage format.
- Transform data assets of any structure into SOA-enabled data services.
- Cut application development time, cost, and risk.
- Deploy business-critical applications more quickly and cost-effectively.
- Improve ROI on existing IT investments.

MetaMatrix Enterprise is built using open standards and offers the advanced flexibility and scalability needed to meet any company’s enterprise information integration needs. Current customers include numerous U.S. and foreign government agencies, as well as Fortune 1000 customers like Merrill Lynch, SAP, Credit Suisse First Boston, and Motorola. See jboss.com/products/platforms/dataservices for product details, whitepapers, analyst reports, and information on how to arrange a free trial.

Red Hat, the world’s leading open source solutions provider, has headquarters in Raleigh, NC with over 50 satellite offices spanning the globe. Red Hat provides high-quality, affordable technology with its operating system platform, Red Hat Enterprise Linux, as well as applications, management, and service-oriented architecture (SOA) solutions, including JBoss Enterprise Middleware. Red Hat also offers support, training, and consulting services to its customers worldwide.

Learn more at redhat.com.