SOA Best Practice Report –

Governance Led Practices to Deliver Effective Semantic Integration

There is increasing understanding that consistent, high quality data is the critical success factor to delivering agile, responsive business processes. Yet delivering this is widely perceived as being very difficult. For larger enterprises the existing application and data landscape has diverse architecture and ownership, is hugely complex and constantly changing. In this environment enterprise wide data initiatives are often high risk. What's required is a practical approach that facilitates short term business delivery projects working in a coordinated manner that progressively delivers enterprise level semantic integration with minimum short term cost or time overhead. This paper describes how a governance led approach to data and information services is necessary to deliver effective semantic integration.

Introduction

A semantic integration layer can deliver consistent data to the business process without necessarily altering the vocabularies and formats used by existing systems, data stores and services

Many projects today are still driven by application and business process goals and the data perspective is largely constrained by the scope of the delivery project. Whilst this may be acceptable in context with project specific objectives, many organizations recognize the need for data and information services architecture to support a wider scope. The results of narrowly focused data architecture are evident in all enterprises – where inter application integration has grown to become a major cost and an inhibitor to rapid response to changing business requirements. The true cost of poorly managed data is a significant business issue.

The business challenge is perhaps best explained by considering the customer perspective. In a government context the case is dramatically illustrated in the UK where a report[1] highlighted that following bereavement, a citizen was required to make 44 separate contacts with government departments. Each interaction of course being a separate process requiring repetition of information, often in a completely different format.

In the commercial context most enterprises have a huge legacy asset base of duplicate or overlapping systems, often resulting from years of narrowly focused application projects that

added new modules rather than consolidating old and new capabilities. A very common result is that different technology and business channels are managed by separate application systems with different semantics that do not allow inter channel switching or cross channel business opportunities. Inevitably, customer service and visibility of the process suffers.

Perhaps even more important is that in both commercial and government contexts narrowly focused data reduces the opportunity to create responsive business processes. The opportunity to dynamically alter response to events is frequently dependent upon a broader set of data than would conventionally be needed for a specific business process.

These issues reinforce the need for data to be loose coupled, managed independently of the applications and business processes. There are various alternative strategies that may be adopted. Perhaps most common is to establish a canonical enterprise data model that provides single definitions of all data items, relationships, attributes and rules which all inter application interfaces are required to comply with. However, while this may initially appear to be an elegant and attractive approach it may also be a very difficult task to define the entire enterprise data in such a detailed manner and yet support a rapidly changing business.

Another common strategy is to define core business services such as Party, Customer and Reference Data and to deliver these as shared services which are mandated to be used across the entire enterprise. This strategy has much in common with MDM initiatives which will often publish the shared services. However in most enterprises there will be a much wider set of data and contexts that should be addressed in order to ensure business processes work with consistent information. Whilst this strategy may be effective for a narrow data set and or process domain, a broader scope may lead to exponential increase in complexity because of varying stakeholder views and change management issues.

Neither of these strategies is optimal. In this report we will explore how a semantic integration layer can provide high levels of loose coupling for data and reduce the complexity inherent in most integration architectures. This is the second report in a three part series[2] exploring best practices in service and data architecture in conjunction with the Progress DXSI and Sonic ESB tools.

Architecture Principles for Semantic Integration

The Open Group architecture framework TOGAF[3] provides a set of example data architecture principles that are widely used as a starting point by many enterprises. In general, principles should answer important questions and choices; they should not be bland statements that have no bearing on how the enterprise is managed. Rather principles should guide the development of policies that can provide clear direction to the continuous decision making and governance processes that ensure appropriate outcomes of delivery projects that do not compromise architectural integrity. TOGAF principles relevant to data are as follows:

- Data is an asset
- Data is shared
- Data is accessible
- Data is owned (Trustee)
- Common Vocabulary and Data Definitions
- Data Security

These principles are a good starting point, but they are very generic and not sufficient in data policy setting and governance. A good place to start extending this list is with a common

vocabulary – that is to differentiate between the data and the meta data. A common vocabulary should establish a set of standardized meanings for all types of data that will be used by more than one business process. We may phrase this principle as follows:

Principle: Common vocabulary for data exchange and reporting

We have introduced a significant qualifier to this principle to make it explicitly supportive of data exchange and reporting. This is not to say that implementations should not become compliant with the common vocabulary when it is cost effective to do so, rather it is less important than enabling interaction using common semantics. The practical implications of this are of course very significant. Existing silos can continue using silo specific vocabulary.

The TOGAF principle that data is shared is of course important, but it may be an unrealistic objective for broad implementation. It will be highly appropriate for certain core business types such as Customer, Reference Data etc. but on a broader scale implies widespread invasive action by all silo implementations using the shared image. This is certainly one very good reason why shared data initiatives have a slow uptake in many enterprises. So we suggest a more practical principle would be:

Principle: Shared data for core business types

And all data, including shared data, that is used by more than one business process should comply with the principle:

Principle: Common data is decoupled from business processes, applications and databases

That is the data that is common to more than one business process should be transformed to the common vocabulary for use in business process layer, and not be owned, nor specific to a particular business process, application or database; rather the meta data view of the common data establishes an independent vocabulary that creates consistent meaning. Implementations can of course comply with that common view but access by business processes would always be loose coupled to allow change management and alternative sources over time.

Last and not least there is the question of data domains. In many enterprises there is a sharp delineation between transactional, business process related data and the data that supports analysis, business intelligence and reporting. This is commonly observed where data warehouses are updated on a delayed, overnight basis. But equally even real time data warehouses may comply with a different data model to the transactional processes and are only accessible using specialized tools as opposed to being easily available to business processes. Similarly complex events processors are frequently served by very specific data capture and analysis capabilities. We suggest the principle:

Principle: Integrated, real time business process and business analysis and intelligence data

It will be evident that business processes are becoming more and more real time in operation and increasingly deliver differentiated services to consumers based on dynamic rules. To support this trend it is vital that the design of data takes a broader view beyond the

conventional scope of the business process. Complex events particularly will reflect dependencies that reach way beyond conventional boundaries.

Architecture for Semantic Integration

Let's consider how these principles impact architecture.

Data is decoupled from business processes, applications and databases

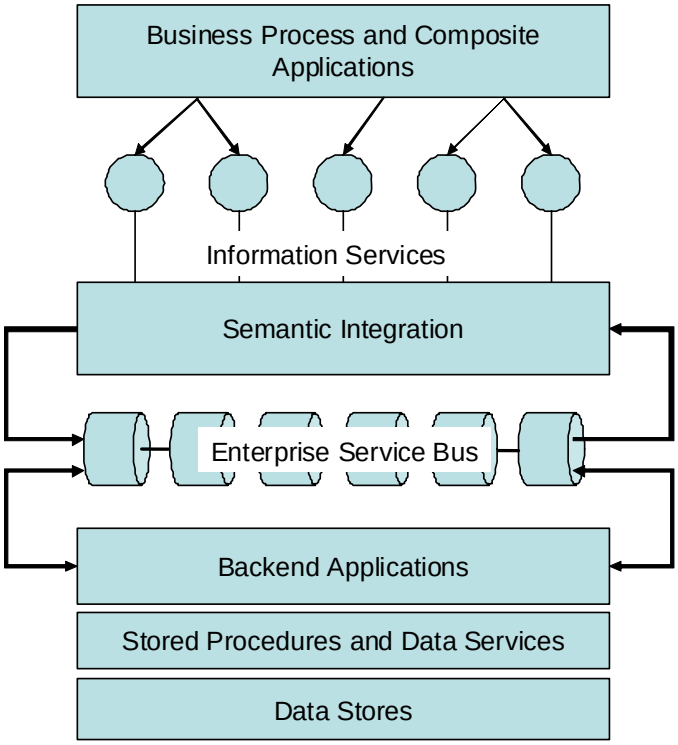


Figure 1 – Semantic Integration Layered Technology Architecture

Decoupling is a principle we are all familiar with. It separates concerns and potentially limits the impact of change to one layer. As shown in Figure 1, this has been widely used over many years in the back end application layers with stored procedures and separation of data bases from applications. More recently, stored procedures have evolved to data services that usually provide a platform independent data access method; but at this level the data will still be specific to the application. Also the broker pattern has become widely used with the ESB layer separating the business process and composite applications from the back end systems. The ESB layer is important because it establishes clear separation between consuming business processes and the back end applications and lessens the impact of changes in the back end landscape or business process layer.

Information services should provide a further level of decoupling by offering an interface contract that hides all details of the back end applications and data stores. The information service should also use data descriptions that transcend individual applications and are free from design and implementation details. We refer to this as the Specification View as opposed to the Implementation View.

The semantic integration layer also works with the Specification View data and introduces a further level of decoupling by integrating multiple vocabularies. Semantic integration effects transformation from one vocabulary to another according to a set of rules. Whilst semantic integration "could" be achieved by transforming disparate vocabularies directly, it is recommended that a common vocabulary provides a standard against which all transformations are made, so that progressively the enterprise (or ecosystem) moves towards a coherent set of semantics.

Figure 2 illustrates a logical view of the integration layer, in which source data (source model) is transformed using rules (exchange model) to a common vocabulary (common model) to deliver a message (data service) compliant with the declared standard (common vocabulary).

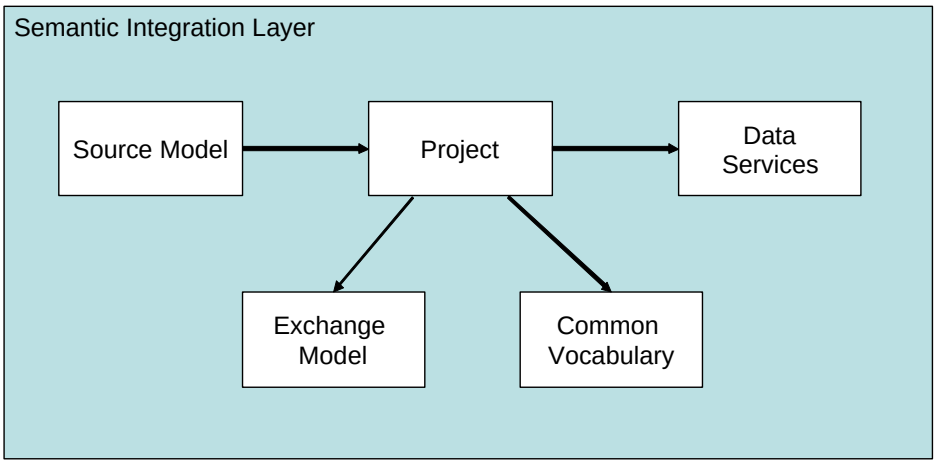


Figure 2 –Semantic Integration Layer - Logical View

There are a number of relevant patterns which will all use semantic integration techniques, but may address different business issues:

- Resolve vocabularies of multiple back end applications into one common set of semantics.
- Act as a switch between disparate unique message formats and common message formats.
- Act as a switch between common message formats and unique business process and composite application message formats. In some cases this may be in support of transition strategy, in others cases it may be to integrate business processes that are inseparable from enterprise application environment(s) that are unable to comply with the common formats.
- Act as a switch between unique or specific external message formats and common internal formats.
- Intercept application specific point to point messages to comply with common vocabulary based shared service.

Using these patterns, the semantic integration layer can deliver a consistent view of data to business processes without necessarily altering the vocabularies and formats used by existing systems, data stores, services and business processes.

Common vocabulary for data exchange and reporting

As discussed in the introduction, a consistent view of data is an important enabler of improved business models, customer services and responsive business processes. In some industry sectors there are existing industry models and these can be powerful accelerators to defining common concepts and vocabularies. But even where industry models exist, they typically will not provide complete coverage, rather addressing core business domains, particularly those that support ecosystem collaboration.

So the question applies to all enterprises - how to define common vocabulary in a practical manner without embarking on a lengthy, resource intensive effort? Inevitably there are pressures from momentum projects that create demand for specific concepts and message types. But the individual project perspective should be regarded as an inappropriate place to start defining common vocabularies. The scope and context will almost certainly be too specific.

The best way to approach this is to establish a portfolio based approach to data architecture that can coordinate program activity within a broader context. The data portfolio view should be at a high, but technically accurate level of abstraction and can be delivered relatively rapidly. It provides an overview of data concepts expressed as Business Types and their interaction with various dimensions of enterprise business model. No rocket science techniques are needed here. Simple matrices outlined in Figure 3 that identify and map the relationships between business types and business processes, business domains and important simple and complex events will provide a backdrop for planning just in time data architecture.

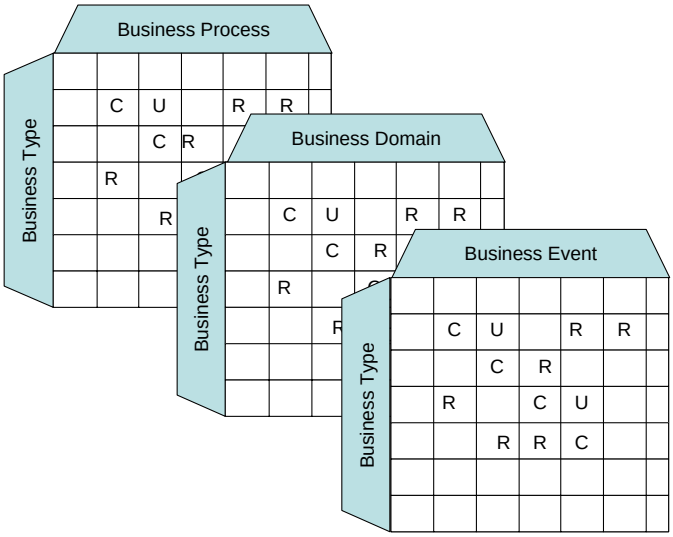


Figure 3 – Data Portfolio Deliverables

The program/project scope is mapped onto the above matrices to provide an architecture level impact analysis. The objective should be to architect and design common artifacts in such a manner that they will provide a stable platform for evolution to support portfolio requirements, but delivered to meet immediate delivery program needs. With this portfolio perspective just in time architecture can be undertaken for programs and or projects.

Why Business Type? The Business Type concept is a specification level artifact that will provide a methodological basis for information service and schema identification and definition. Business types are defined as a category of business object, the instances of which an organization needs to keep track of in its information storage system.

Business types should be modeled formally at the portfolio and specification level of abstraction, both models rigorously excluding implementation details. A fuller treatment of Business Type Modeling and explicit guidance on modeling levels of abstraction was provided in the first report in this series[4].

This section has described a process that is strongly influenced by immediate business priorities but drives out a broader, more stable data foundation that serves both shorter and longer term business goals. As the shared vocabulary grows we might refer to this as an evolving canonical model. Although conventionally canonical models may have generally been developed in a more strategic, top down manner, in practice there is no difference in the final outcome.

But in many cases it might be preferable to refer to the common vocabulary simply to avoid misunderstanding and to ensure that the essence of the principle is adhered to – to establish support for data exchange and reporting, which is profoundly different to attempting to set enterprise standards for data that implies enterprise wide compliance at some future, unspecified date.

Integrated, real time, business process and business intelligence data

The task of creating a detailed view of the data required to support a project, program or an enterprise is always important. There are some ways to short circuit this such as using industry models as mentioned previously. Another important mechanism is to use knowledge discovery techniques and tools to analyze, consolidate, rationalize and restructure the data requirements of existing systems. However both of these approaches are driven by the business process perspective and will inevitably be focused on transactional data – and deriving vocabularies that support business process execution.

In most enterprises the data that supports analysis, business intelligence and reporting is managed separately from the transactional business conventionally in data warehousing, analytics and reporting systems. But this total separation is an unsustainable practice in the future as transactional and business intelligence behaviors converge in responsive business processes.

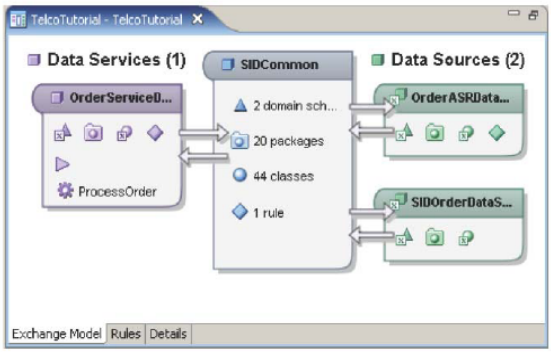
In mature service architecture the broader spectrum of business transactions and events are managed on a common bus in near real time. This opens up many opportunities to explore relationships between simple and complex events across a much broader scope and permits business processes to customize responses for business advantage.

For example in the financial sector identifying and correlating trades by the same corporate customer across diverse product domains in a given time window may signal the need to introduce altered terms and conditions either to reflect higher risk or provide enhanced customer service. This cross product domain scenario may involve data managed by disparate applications and require rationalization of product based vocabularies in order to permit correlation. This could of course be specified in a complex events rules engine, but a more repeatable and reusable approach would be to integrate the disparate product group based vocabularies in the semantic layer so that the converged view is defined not simply in respect of a single event, but as part of a wider understanding that will allow the business to monitor and respond more rapidly and intelligently to important situations.

Policy Setting and Governance

Attempts to develop consistency of data and process are usually controversial because they have the potential to conflict with delivery project objectives. The just in time architecture approach outlined above is designed to mitigate this, but realistically conflicts will arise in which delivery program managers prioritize the immediate delivery objectives.

The Progress Software DataXtend Semantic Integrator (DXSI) provides a model based approach to data integration. The central concept underlying DXSI is that common semantics or vocabulary are managed independently of existing data sources and facilitate rules based mappings between one or more existing data sources against the common view. This approach breaks the dependencies created by point to point integration and allows consistent data services to be defined and published from disparate sources. The common view approach is also effective in limiting the impact of change in existing data sources and in the common view.



DXSI generates run time data services that can be used anywhere a Java object can be invoked. To implement the data services into the target environment the DXSI Workbench generates wrapper code that instantiates the entities and invokes methods in an exchange model and Java classes. The SI Workbench generates this wrapper code for JavaBeans, SDOs and XMLBeans, which can be deployed in the application and Web servers including Apache Tomcat, BEA WebLogic, IBM WebSphere Application Server, IBM WebSphere Process Server, RedHat JBoss Application Server, Oracle Application Server and Progress Sonic.

The DXSI Workbench includes a test environment that provides import of test data and snapshots of transformations that aid debugging. Multiple versions of models can be managed.

Arriving at solutions that deliver compromise is often unacceptable, and what's required is clarity on policy relating to compliance with the semantic integration architecture. This can be agreed by key stakeholders from business and IT domains, in order to agree an approach in advance that will be applicable to most projects and programs and represent a sensible compromise between individual program and project goals and the broader business objectives.

The use of the term policy is appropriate. Policies are a set of decisions that have been agreed by a cross stakeholder forum that form a framework within which programs and projects can

plan and manage. It will be expected that policies apply to the majority of scenarios, but there will always be some requests for policy waivers.

So what are the policies that will facilitate semantic integration? Table 1 provides some examples. The policy examples are organized within policy area which is a useful way to think about the policy requirements as well as ensuring the correct stakeholders are involved in both policy setting and ongoing governance activity.

Policy Area	Example Policy Instances	Comments and rationale
Planning	All programs and projects supporting or integrating with defined business domains to align with or map to common data architecture via the semantic layer	Semantic layer integration is mandatory for certain areas of the enterprise that will benefit from semantic integration in a given timeframe
Architecture	All messages and services provided and or consumed by defined business domains to use the common ESB	Guarantee that private bus structures do not compromise the enterprise in correlating events across all business domains
Architecture	All back end data access to be via information services aligned with common vocabulary	Mandated separation and loose coupling of business process data from back end systems and data stores
Architecture	All solution architectures within defined business domains required to comply with data portfolio plan, either by semantic integration or alignment High level data architecture defined Definition of standard data concepts, schema	Explicit deliverables required to be approved by Governance Board
Sourcing	Significant weighting of procurement decision making to be given to products that: a) offer all functionality and capability through service architecture b) comply with relevant industry data models	That allows ease of integration with the ESB and semantic integration layers
Best Practice	Change management practices mandate life cycle for changes to the common data architecture	Coordinated change practices
Organization	Central data architecture team to be responsible for development and maintenance of data portfolio, architecture and common vocabulary	Clarity of responsibility
Organization	Program and project architects responsible for managing compliance or waivers with data architecture	Delegation of responsibility to delivery teams

Table 1 - – Policy Areas and Instance Examples

Conclusions

In this paper we have explored the concept of loose coupled data as an approach to resolve what are for many enterprises, high priority problems. We have discussed that consistency of data is not solely a technology centric matter; rather it is essential to enable business processes to provide a more coherent interaction with customers in terms of both efficiency and minimum interactions to complete a complex transaction with cross channel effectiveness.

The loosed coupled approach described is a realistic alternative to a more conventional canonical enterprise model approach. Loose coupled data as we have described it reduces the pressure to rationalize back end systems. It allows the enterprise to work with a consistent vocabulary to better support customers and to resolve the back end issues in a manner that reduces, not increases complexity. Of course loose coupling of data requires technology to make it happen, but the critical success factor will always be the ability of the enterprise to organize around different ways of working. In this paper we have outlined an approach that is governance led – that is clearly articulated in terms of principles and policies that can be translated into concrete actions that facilitate and not impede delivery projects.

David Sprott

Everware-CBDI Inc. January 2011

This paper was commissioned by Progress Software Inc. References

Progress Software: Progress Software Corporation (NASDAQ: PRGS) provides application infrastructure software for the development, deployment, integration and management of business applications. We strive to increase your business' effectiveness by offering you the most effective and open set of infrastructure products possible.
http://web.progress.com/index.html
Progress DataXtend - http://web.progress.com/dataxtend/index.html

Everware-CBDI: Everware-CBDI is the premier, independent source of Service Oriented Architecture and Engineering guidance and implementation. Our mission is to help organizations fundamentally change the way they respond to business needs through innovation in modular architecture and service engineering practices Research Portal: www.cbdiforum.com Corporate: www.everware-cbdi.com

---

[1] Service Transformation: A better service for citizens and businesses, a better deal for the taxpayer. Sir David Varney, HMSO, 2006
[2] Part 1: Architecture and Specification of Information Services with Progress DataXtend Semantic Integrator
Part 3: Information Services Architecture for Responsive Process Management
[3] TOGAF reference
[4] For more on Business Types refer to the first report in this series – Architecture and Specification of Information Services with Progress DataXtend Semantic Integrator

About CBDI

CBDI Forum is the Everware-CBDI research capability and portal providing independent guidance on best practice in service oriented architecture and application modernization.

Working with F5000 enterprises and governments the CBDI Research Team is progressively developing structured methodology and reference architecture for all aspects of SOA.

CBDI Products

The CBDI Journal is freely available to registered members. Published quarterly, it provides in-depth treatment of key practice issues for all roles and disciplines involved in planning, architecting, managing and delivering business solutions.

Visit www.cbdiforum.com to register.

Platinum subscription – A CBDI Forum subscription provides an enterprise or individua with access to the CBDI-SAE Knowledgebase for SOA delivering ongoing practice research guidance materials, eLearning, tools, templates and other resources.

Everware-CBDI Services

At Everware-CBDI we enable large enterprises and governments to become more agile by modernizing their business systems. We have repeatable processes, resources, tools and knowledge-based products that enable enterprises to transform their current applications in an efficient, low risk manner, into an optimized service-based solutions portfolio that supports continuous, rapid and low cost evolution. Our consulting services range from providing practices and independent governance to architecture development, solution delivery and service engineering.

Contact

To find out more, and to discuss your requirements visit www.everware-cbdi.com or call

USA and Americas: 703-246-0000 or 888-383-7927 (USA)

Europe, Middle East, Africa, Asia, and Australasia: Telephone: +353 (0)28 38073 (Ireland)

www.everware-cbdi.com