



Service-Oriented Architecture and Business Intelligence

by Philip Wik, Lead Information Architect at Choice Hotels International

SERVICE TECHNOLOGY MAGAZINE • Issue LIII • August 2011

Abstract: *This article discusses how we can integrate business intelligence solutions into a service-oriented architecture.*

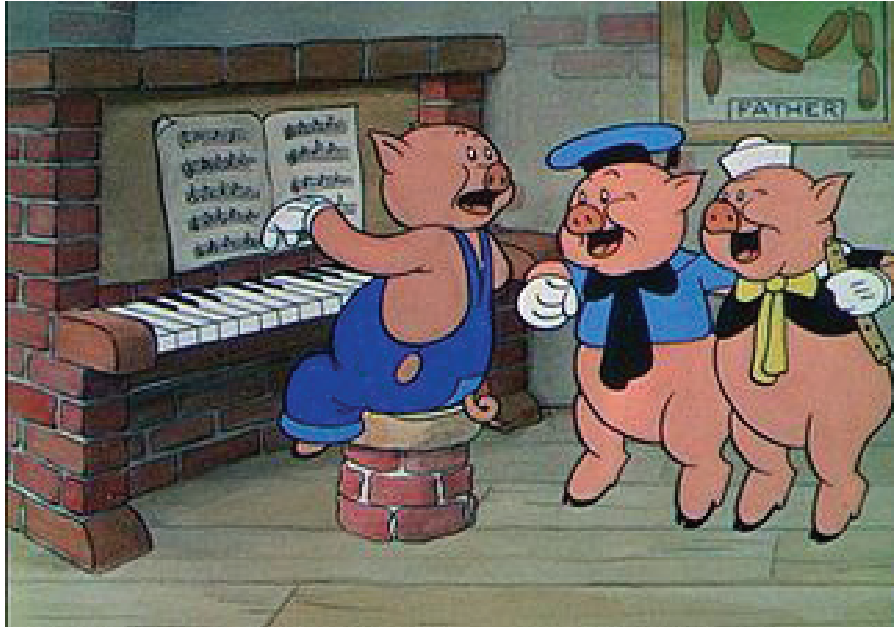


Figure 1 – Fifer Pig, Fiddler Pig, and Practical Pig (Disney, 1933)

Wikimedia Commons

Introduction

Once upon a time, there were three little pigs and the time came for them to leave home and seek their fortune. The first little pig built his home out of straw because it was the easiest thing to do. The second little pig built his house out of sticks, and the third little pig built his house out of bricks.

Many fairy tales hinder the effective use of a service-oriented architecture. But few are less justified than the notion that service-oriented architecture and business intelligence (SOA-BI) cannot co-exist because of these apparent differences:

SOA is Generally	BI is Generally
Enterprise-oriented	Subject-oriented
Middleware	ETLs and Presentation
Transactional	Non-volatile
Reliant on Web services and message-level processing of data	Reliant on Very Large Data Bases (VLDBs) and terabytes of data
Real time	Scheduled

Table 1

But there are also similarities and ways that SOA and BI can work together. The purpose of this article is to show how SOA can support BI. With a nod to the Three Little Pigs story, we'll discuss the following:

1. The Promise of Business Intelligence
2. The Promise of SOA-BI
3. A SOA-BI Reference Model
4. Implementing SOA-BI

The Promise of Business Intelligence

Business intelligence promises to turn data into insights and insights into decisions and actions. It's a tool that drives decisions with ad-hoc analysis, interactive dashboards, scorecards, key performance indicators, alerts, and geospatial visualizations.

BI promises to marginalize information technology construction in favor of client self-service. It proposes to go from well-defined questions to any answer that might arise, from what to why. It's a front-line weapon that allows us to rapidly launch promotions, address issues, and negotiate product pricing.

The Promise of SOA-BI

Business intelligence justifies SOA. Both SOA and BI can use open interoperability and reusable, loosely coupled services that can be shared across the enterprise. SOA releases BI's value when analysis is rendered immediately. Users want fresh information, such as stocks or reservations data. SOA vendors meet this desire with Business Activity Monitoring (BAM) tools. BAM monitors data elements that pass through Web service requests and then builds an image that a service can consume.

Business intelligence is evolving to embrace advanced analytics, including data mining, near and real-time measures, scalability to support terra and petabytes of data and thousands of concurrent users, 64-bit in-memory addressable space platforms, clouds, virtualization, and massively parallel processing [REF-1].

A SOA-BI Reference Model

SOA-BI could be aligned to an enterprise information integration architecture that has master data management metadata, third form normalized operational data stores, a single source of truth for all attributes, and all data scrubbed and profiled. But the reality is that most data warehouses are a wolf's breakfast of non-conforming and orphan dimensions and facts, no data lineage or metadata information, non-standardized prime/class names, point-to-point feeds, and error-infested loads and schemas. Developing a warehouse is

also a costly time-intensive effort that only coincidentally delivers what it advertised. Accordingly, business units are starting to reject old school BI solutions and are moving to alternative information delivery solutions.

Enterprise Information Integration Architecture

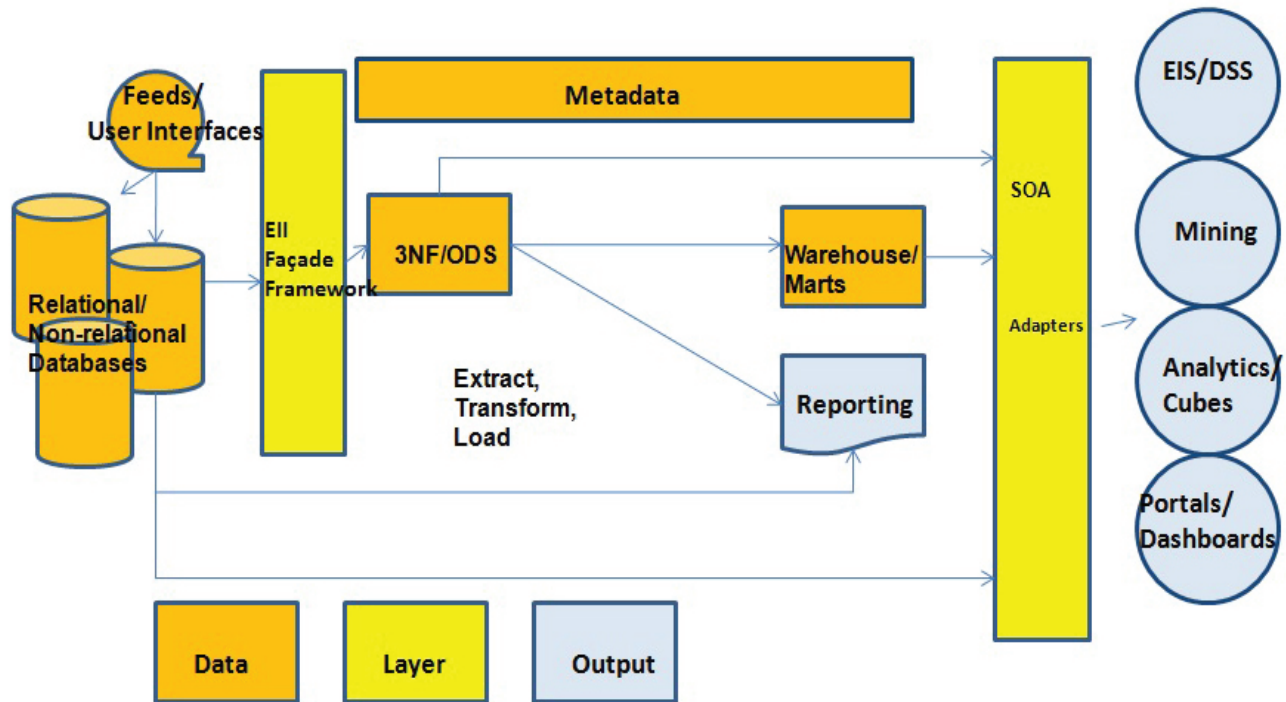


Figure 2 – SOA-BI Acquisition, Integration, and Distribution Architecture

This model shows the general data flow. The legend is at the bottom of this diagram, with data or metadata, the enterprise information integration and SOA layers, and outputs.

Enterprise Information Integration

Enterprise Information Integration is a metadata-driven infrastructure supporting access to multiple data sources through data virtualization. It provides a complete view of data as if it existed in a single, relational database with read/write SQL. The result is an enterprise-wide view of data that provides a reusable framework for information access. This addresses the problem of integration spaghetti and also creates a data layer for SOA services.

Extract-Transform-Load

Extract, transform, and load are the processes of moving data from source systems to target data warehouse fact and dimension tables. It's cheaper to buy or use an open source ETL package than to hand-craft our ETL. In Figure 2, ETLs would load the source databases, staging tables, the ODS, and the data warehouse and marts.

Metadata

Metadata is data about data. SOA metadata includes a full definition of all attributes as well as data it can get or send. A metadata management (MDM) system is a device to resolve semantic incompatibilities, declare constraints at the table level, check data flow integrity, and audit, cleanses, and recycles rejected records. A MDM will also consolidate and federate referential data.

Third Normal Form Operational Data Store

Operational data stores integrate data from different sources to meet objectives of correctness, consistency, simplicity, non-redundancy, and stability. With third normal form, every non-prime attribute is non-transitively dependent on every key of the table. Data from the ODS often comes from non-normalized staging tables.

Data Warehouse/Marts

Data from the ODS goes to a denormalized data warehouse and possibly to data marts that are built to improve response time or to silo data for groups of users.

Adapters

Web Services adapters allow Web Services interfaces to application, database, or platform systems. Adapters transform non-XML formats into XML formats. A configuration wizard helps create a service using the file adapter and defines an operation for the service, such as ReadCustomer, and an operation type including read, write, list, and synchronous read.

Executive Information Systems/Decision Support Systems

Executives and managers who use EIS/DSS are key consumers of business intelligence. Enhanced responsiveness is broadening the consumer to include clients and customers.

Expanding on the model above, the SOA-BI presentation and data entity/complex event services layers have these elements:

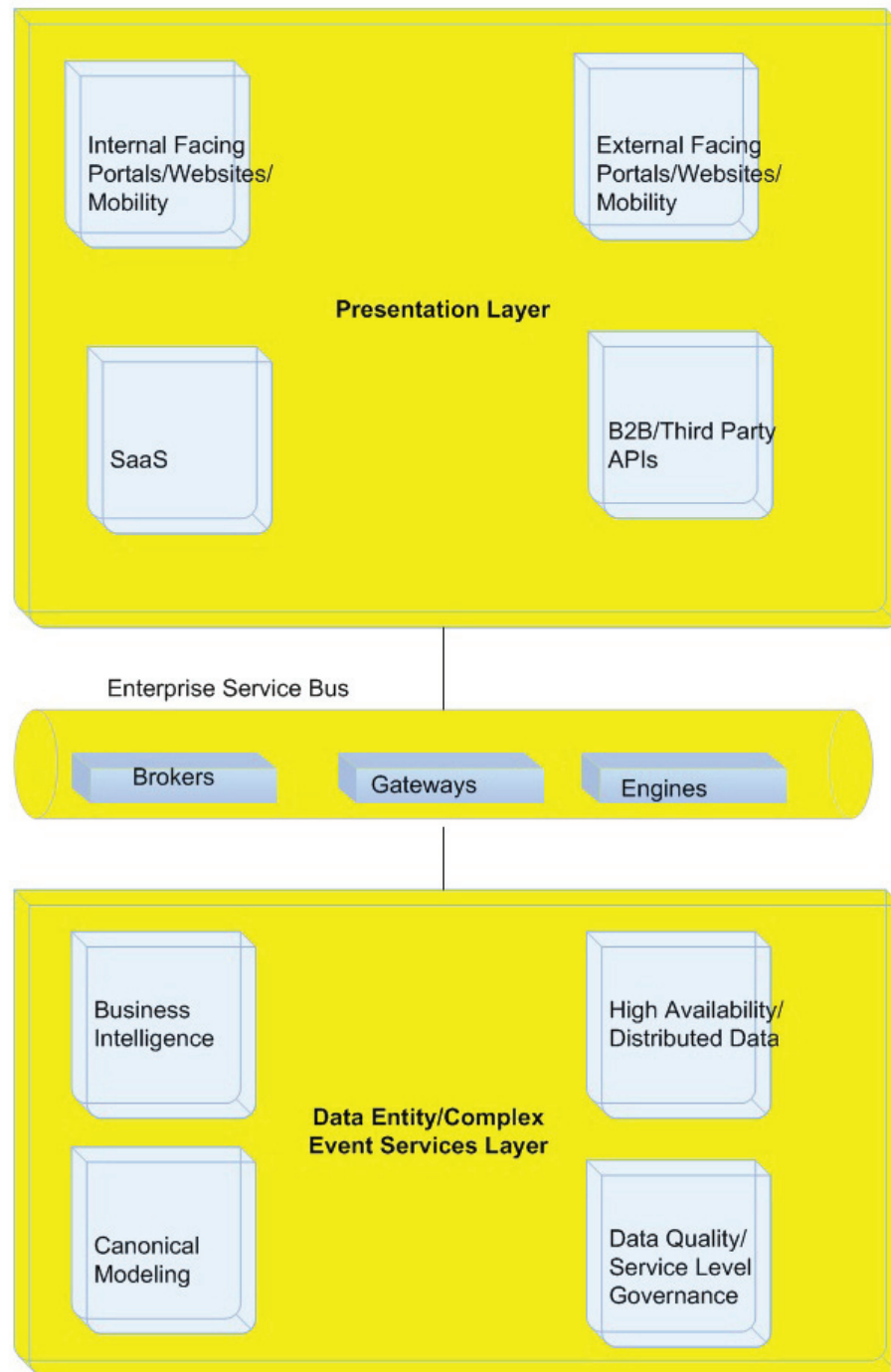


Figure 3 – SOA-BI Conceptual Model

The Enterprise Service Bus

An enterprise service bus is software architecture that provides services with an event-driven messaging engine. It supports various message exchange patterns, such as synchronous request/response, asynchronous request/response, send-and-forget, and publish/subscribe, and has adaptors, gateways, brokers, and engines to integrate with business intelligence portals, Software as a Service, and business to business interfaces.

Canonical Data Modeling

A SOA-BI implementation starts with canonical data modeling. CDM mediates semantic data modeling and defines business entities and relationships. It provides a dictionary of reusable, common objects at the business domain level rather than at the implementation level. CDM is the link, therefore, between the business context and the SOA context. Examples of CDM include Standard Universal Data Models and Industry Universal Data Models.

Complex Event Processing

Complex Event Processing is the final jigsaw piece that completes the real-time puzzle. An event ontology, a formal specification within a shared domain, defines the process for sharing real-time events between components. Here is a high-level example:

The Business	Car Sales
Business Event Ontology	Verb-Noun Maintaining Inventory Recording Inventory Storing Inventory Re-ordering Inventory Soliciting Prospects Checking Credit

Table 2

An event service can trigger an ETL update and can also monitor, correlate, or propagate events that happened on business data, in the context of event driven architecture, change data capture (CDC), and complex event driven processing. CDC capabilities can perform scheduled, on-demand, or full updates and reloads of data. It can also allow continuous loads to the data warehouse and detects changes on different data sources to pull or push data.

Implementing SOA-BI

How do we integrate business intelligence into a SOA? Here is a road map:

1. Start at the end
2. Start at the start
3. Have modest goals
4. Get services from facts
5. Be flexible
6. Develop with agility
7. Deploy with automation
8. Foster the gift of boredom

Start at the End

The end is how customers will use business intelligence. Build use cases, prototypes, and visualizations of what the customer needs and would see. This includes role-based dashboards with navigation, interaction, and drill-downs. Write test cases that users can approve before-not after-designing and constructing services using workshops and product guides.

Business intelligence is a tool like the pigs' kettle, with its usefulness that is only as good as the value from that tool. We must make sure that dollars are not spent without asking basic questions. Building cubes and services without respect to a compelling business case decouples effort to the bottom-line so as to give a meaningless return on investment. The question should not be: "Can we build real time BI?" It should be: "What business problem will we solve with real time BI?" or "What is the market for a real time BI solution?" The question should not be: "What data do we need?" but "What problem are we really trying to solve?" These questions will spin additional questions: "Who are the actors?" "Where are the markets?" "What are the workflows and state transformations?" And more deeply: "What are the hierarchies, granularity, and relationships?" If we ask the right questions, the right solutions and strategies will present themselves.

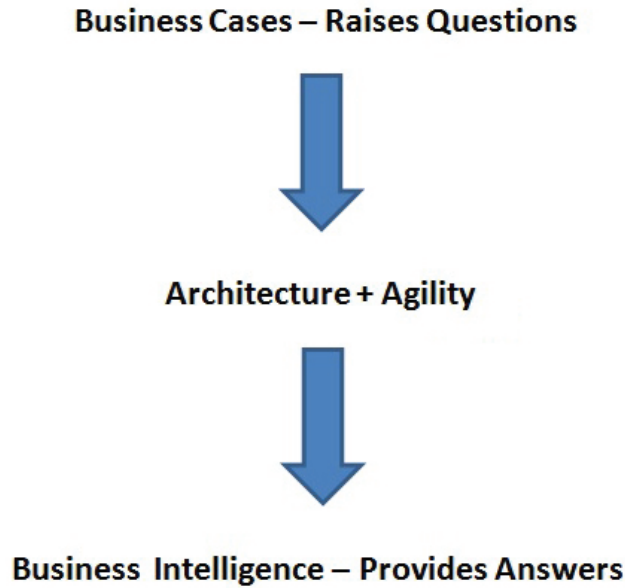


Figure 4 – Questions before Answers

Start at the Start

Rationalize and conform the data warehouse physical data model, tie the business and services domains with CDM and conceptual and logical modeling, and address metadata and data quality issues. None of this is simple, but it is necessary.

Have Modest Goals

Approach the problem SOA-BI integration economically. Our goal isn't to reinvent vendor functionality. It's to close the gap on specific BI-related problems within the context of a service-oriented architecture that can give added value to our enterprise.

Ralph Kimball suggests a dimension manager based on a Master Data Management to provide the following services to the fact table subscribers [REF-2]:

1. Fetch specific dimension member
2. Fetch all dimension members
3. Fetch dimension members changed since specific date-time with designated slowly changing dimensions (SCD) types 1, 2, and 3
4. Alert providers of new dimension release
5. Alert providers to late arriving dimension members

Other services could include the following:

1. Entity history service
2. Calculation service
3. Aggregation service

4. Data grid services
5. Master data services, including lifecycle maintenance of gold records
6. Composite data services
7. Information visualization service
8. Data quality services
9. KPI change alert services
10. Data transformation service, deployed as XSLT Factory, ETL Engine, or Canonical Mediator services.

The low hanging Red Delicious of business intelligence functionality, that can be autonomous, services are simple, repetitious, widely used, and widely overloaded functions. While XML Web services can replace extracting, transforming, and loading bulks and batches of data, they are verbose. Open source or commercial ETLs are thusly more effective at handling volume. Better candidates for services are business intelligence presentation functions, such as operational reporting, drilling, and pivoting, and analytical and windowing functions.

Get Services from Facts

Data warehousing facts reflect domains of functionality from which we can derive services, such as Orders or Authorizations. In this model, logical entity Orders is realized as an Orders ODS table, with shipping information columns and a child Orders_Detail table. Orders_Detail has measures such as Unit Price, Quantity, and Discount Percentage. In the data warehouse, these measures reside in the Orders_Fact table. The data mart Order_Dm has aggregated data from the fact and dimension tables.

Staging	3NF/ODS	Data Warehouse Facts/ Dimensions	Data Marts	Services	Business Intelligence Universe/ Classes
Orders_ Orders_Detail	Orders Orders_Detail Customer	Orders_Fact Customer_Dim Product_Dim	Orders_Dm	Orders	Orders Customer Product
Authorizations_ Auth_Detail	Authorizations Auth_Detail Customer	Authorizations_Fact Customer_Dim Product_Dim	Authorizations_Dm	Authorizations	Authorizations Customer Product

Table 3

The following conceptual model shows how services could come from facts.

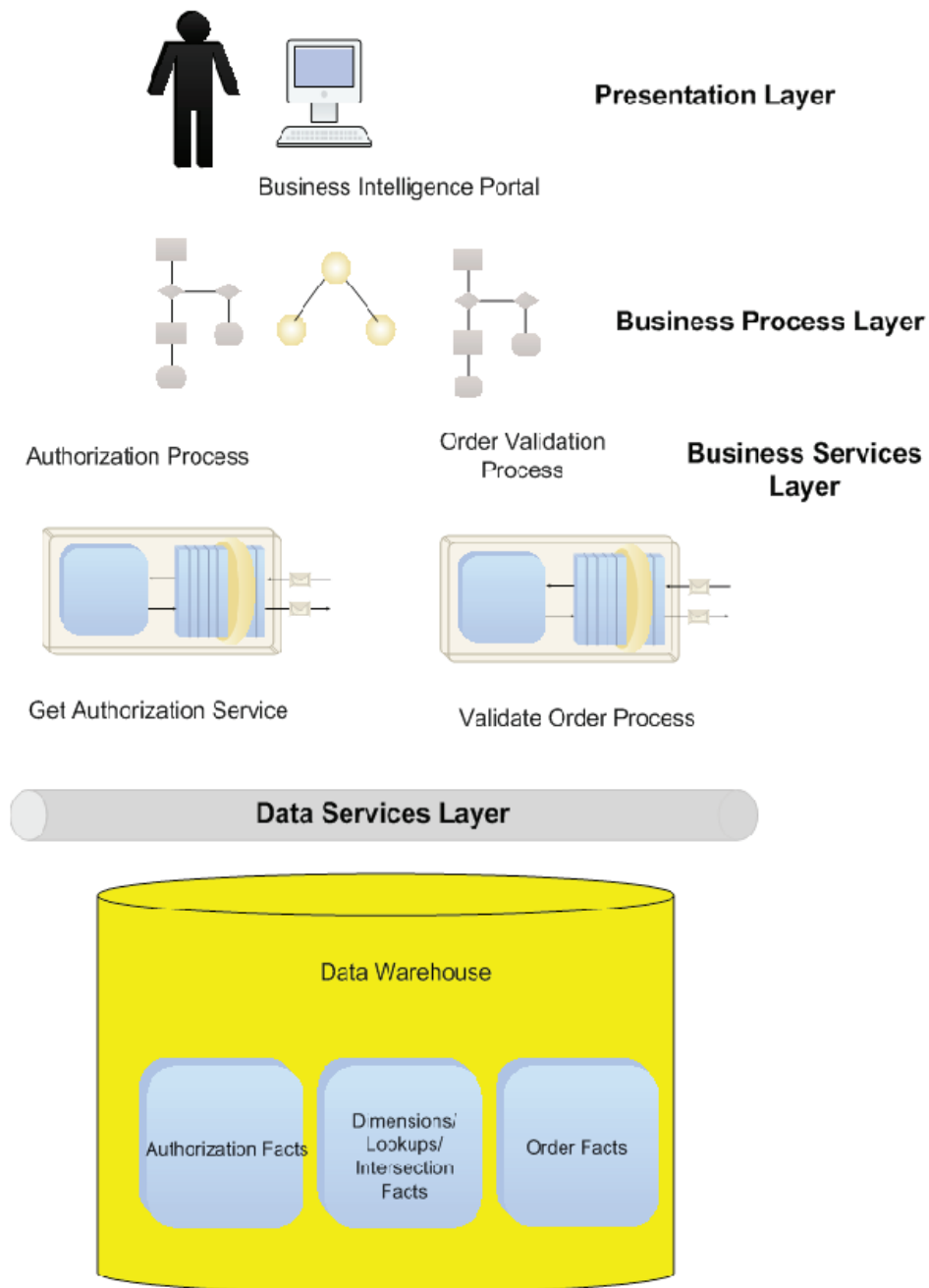


Figure 5 – SOA-BI Facts and Services

The Presentation Layer

SOA's presentation layer handles business intelligence requests and replies. SOA Web services, based on XML standards, facilitate the delivery of software applications as a service using any platform to any consumer. The presentation layer is the front-end tier that handles user requests and responses, within subject areas or universes. It consists of portals, Web 2.0, packaged applications, or user forms. Formats include charts, graphs, or dashboards. Navigation includes drill-downs, pivots, or drag-and-drops. Hosts include desktops, laptops, or hand-held devices.

The Business Process/Services Layer

The Business Process/Services Layer consists of functional builds with a rules engine, lookup tables, and workflow integration. Examples of business services include credit authorizations, inquiries, reservations, bookings, and quote generation.

The Data Services Layer

The Data Services Layer handles information exposed through autonomous services. It provides mediation between consumers and heterogeneous sources. DSL characteristics include loose coupling between the applications and data stores and quality of services features. DSL consumers include BI applications. SOA can orchestrate these services with rules, logic, and workflow in the business process layer.

Be Flexible

We must empty our minds of dogma as to open source or off-the-shelf, one vendor or another vendor, and SOAP or REST. Hybrid architectural approaches and tools can be sound approaches. REST and SOAP are not interchangeable for all architectural decisions although they do overlap. In general, REST is simpler and is oriented to the presentation layer whereas SOAP is more flexible and is oriented to middleware.

REST and WS-* Conceptual Comparison [REF-3]

The architecture has this aspect	
The architecture does not have this aspect	

Architectural Decisions	REST	WS-*
Remote Procedure Call		
Messaging		
Contract-first, last		
Contract-less		
Do It Yourself URI Resource Identification		
XML Schema		
Request-Response		
One Way Message Exchange		
HTTP/S Transport		
Waka		
TCP, SMTP, JMS, MQ, BEEP, HOP		
XML (SOAP)		
XML (POX, RSS), JSON, YAML, MIME		
URI Service Identification		
WS-Addressing		
XML Schema, WSDL		
WADL		
HTTPR Reliability		
WS-Reliability, WS-ReliableMessaging, Native		
HTTPS Security		
WS-Security		
WS-AT, WS-BA, WS-CAF Transactions		
Do It Yourself Transactions		
WS-BPEL Service Composition		
Mashups Composition		
Do It Yourself Composition		
UDDI Service Directory		
Do It Yourself Service Directory		
HTTP As Application-Level Protocol		
HTTP As Transport-Level Protocol		
Loose Coupling: Time Availability		
Browser Wars		
Enterprise Computing Middleware		

Table 4

SOAPiness

Simple Object Access Protocol (SOAP), a stateless, one-way message exchange paradigm, handles data transfer and messaging. It provides the framework that conveys application-specific information in an extensible manner. The business object, implemented as a session bean, entity bean, or other object, represents the data client and requires access to the data sources to obtain and store data. Web services expose logic as functions that are published, discovered, and developed as loosely coupled components. The service layer, consisting of Java Architecture for XML Binding (JAXB) and WSDLs, interlocks the business logic/persistence layers and the presentation/process layers. Business intelligence portlets within the presentation layer invoke Web services that run inside the application server.

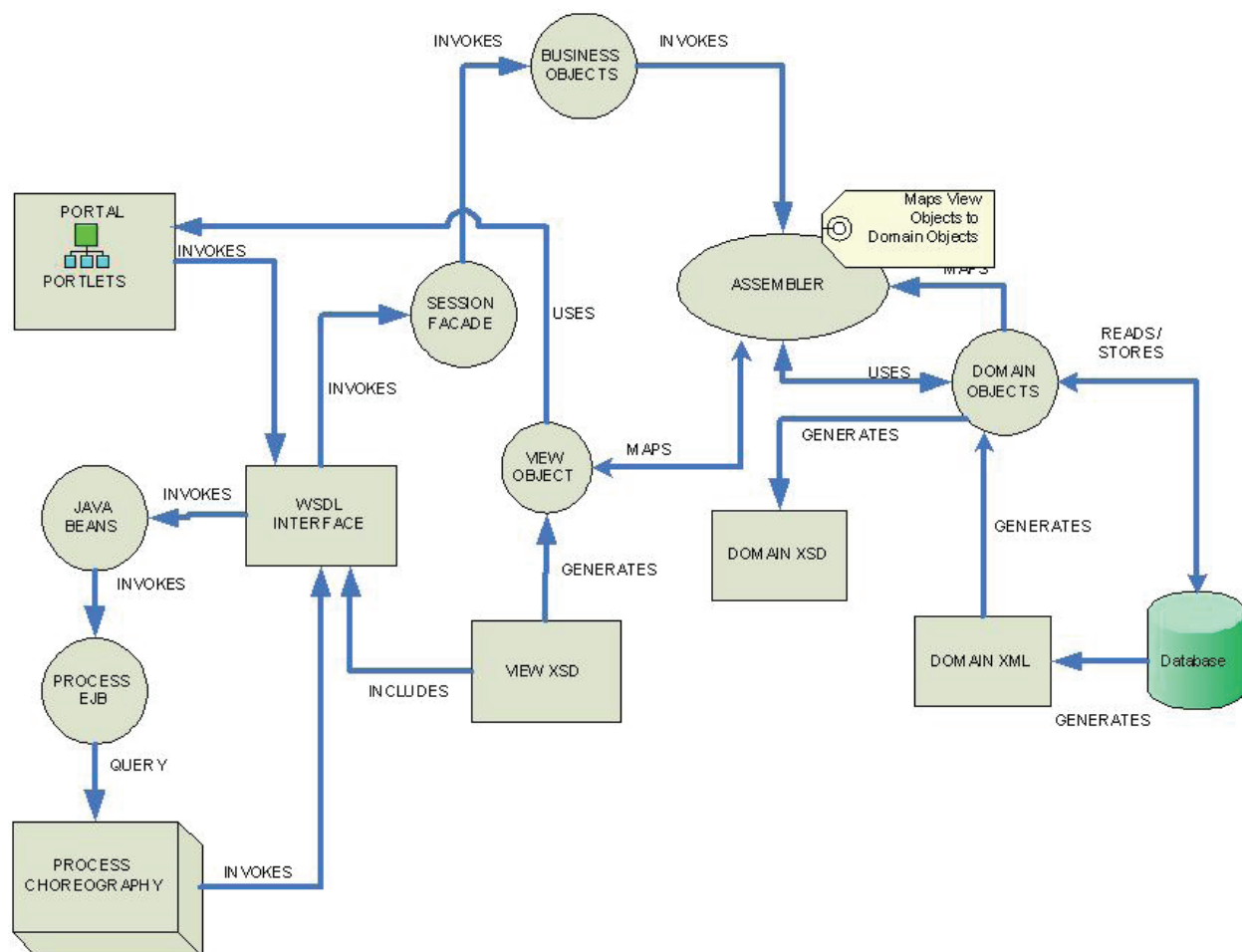


Figure 6 – SOAP-BI Model [REF-4]

Web Services Definition Language (WSDL) implements Web services by specifying functional signatures of each available service.

RESTfulness

An alternative method for building Web services for BI is with Representational State Transfer Representational State Transfer (REST). REST is a set of constraints, such as Uniform Resource Identifiers (URIs), the Hypertext Transfer Protocol (HTTP), and markup languages such HTML and XML. URIs and typical responses are in XML/JavaScript Object Notation (JSON).

REST APIs can provide intelligence on demand Web services using open interoperability and platform independence. We can use REST for data navigation, analysis, visualization, reporting, aggregations, cleansing, and scheduling. It can exploit BI Web data and offer services on top of the data. REST uses URIs with GET, POST, PUT, and DELETE verbs.

REST	CRUD	Description
POST	CREATE	Initialize a new resource state
PUT	UPDATE	Modify a resource state
GET	READ	Retrieve a current resource state
DELETE	DELETE	Clear a resource state

Table 5

Hyperlinks define relationship resources and state transitions of the service interaction. For example, exposing a BI orders service can return a result set value by scripting the following URI:

1. Query:

```
SELECT COUNT(*)
FROM ORDERS ORD,
      CUSTOMER CUST
WHERE ORD.ACCOUNT_KEY = CUST.ACCOUNT_KEY(+)
AND CUST.REGION = 'NORTH';
```
2. REST:

```
GET/search?region='NORTH'
```
3. URI

```
http://www..com//search?=region=NORTH
```

Develop With Agility

Power, the capacity to effect change, is the ability to say “no” and to make it stick. Truth, the capacity to recognize what is real, is the ability to say “1 = 1” and to make it stick. We cannot build a SOA-BI without saying “no” to what is obsolete and “yes” to what is true. A good truth test for those who are accountable to the development of a SOA-BI is to raise this question: “Can a BI system development life cycle be agile?” Any response other than a simple “yes” is wrong. It’s now beyond dispute that increasing numbers of enterprises around the world are succeeding in automatically implementing working business intelligence software, including objects of star and snowflake schemas, every two weeks into production on projects that once took months and sometimes years [REF-5].

A low total-cost-of-ownership BI solution is to leverage prebuilt out-of-the-box analytic applications using an agile SDLC. These prepackaged suites include the full range of enterprise resource planning and customer relationship management objects. They will accommodate forty percent or a more of a mid-sized company's core BI functionality. Agile customization on the remaining effort will slash time-to-market duration up to eighty percent as compared to that of a typical waterfall life cycle.

Deploy With Automation

Enterprises have solved the problem of agile automation along these lines:

Automatic Deployment

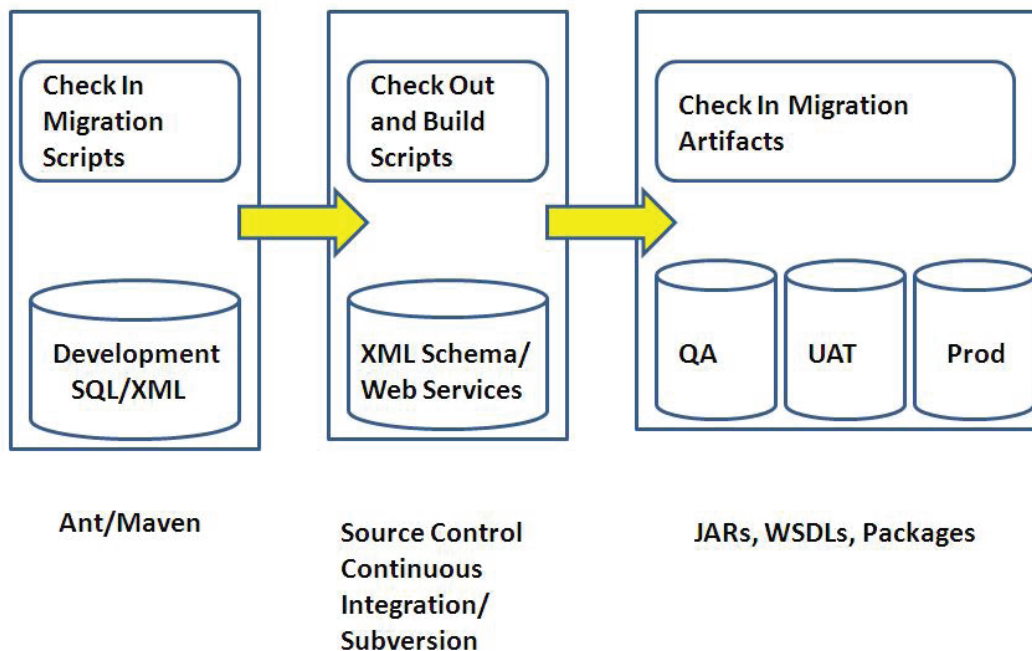


Figure 6 – SOAP-BI Model [REF-4]

Foster the Gift of Boredom

SOA-BI in itself is not the answer to corporate transformation or profitability. Framing such goals as a function of architecture or technology is akin to pigs efficiently and effectively building homes of straw. Nothing is more challenging than making the cultural changes that must occur if SOA-BI is to succeed. But it must be done. Just as there are Navaho or Hindu cultures, so too there are corporate cultures with their shared traditions, values, and creeds that speak to how they have become successful. But success is never final. The values that drive an enterprise to achieve may not be those values that sustain achievement. Culture eats technology for breakfast, and an inability to align culture to the market can gobble those crumbs that drop from the table of technology.

In our fable, a democracy of pigs in this fable would have meant the deaths of the three pigs. It takes top-down insight and will to envision and blaze a new path. That new path is often a culture that fosters bottom-up

innovation. Leadership isn't enough. Leaders must encourage a spirit of entrepreneurship to build SOA-BI solutions and similar SOA projects. This top-down bottom-up meet-in-the-middle dance is almost as rare as a wolf in a chimney as is leadership in its most enlightened sense. The typical corporate dynamic is a dance macabre with extremes of either anarchy or authoritarianism. Leadership is rare because leadership threatens established routines and roles with creative annihilation. It would seem that some managers prefer to fail slowly than to change and succeed. In the prison of personality, a diffident or rigid CTO or BI manager will extend the walls of that prison to engulf his fiefdom. At some level that only psychoanalysis can penetrate, some companies are stuck with eroding effectiveness because they want to be stuck, preferring the devil they know of inertia to the risks and the rewards of leadership. Sometimes new management is the only sure way to remove those walls, break the spell of conservative torpor, and get out of the mud.

All of the conditions exist for everything to remain as it is. Yet, change is the perpetual reality of the marketplace. How do we know what new conditions are needed to snap the malaise and spark change in the right direction? How do we know if our SOA-BI is a profit enabler or if it's just one more abortive science project? A bunch of services are not SOA, BI is far more than reporting, and buying an enterprise service bus could still amount to little more than buying a fax machine with an impact that is both trivial and overdue.

Privately-held firms, non-profits, and government agencies may have the luxury of long time horizons and deep pockets. But failing SOA-BIs will reveal themselves not just with internal metrics, but also by opportunity costs, project velocity issues, and broken management. "Non-profits need management even more than business does precisely because they lack the discipline of the bottom line," Peter Drucker notes [REF-6].

For publically-owned firms, there is but one sure measure of SOA-BI success: rising share price. The price of a stock is a collective bet on value based on all that is known or can be inferred, from wild hunches to deep analysis using vast databases. Like weather patterns or ocean tides, stock prices ebb, flow, and trend. Fundamental analysis- balance sheet figures as well as SEC 10-K footnotes and technical analysis- fluctuations of price and volume can only be understood in the context of such non-rational impulses as lust, foolishness, and fear. A stock trend will persist for any reason or no reason, like a herd of pigs stampeding over the African savannah. That thundering herd will fork into a new direction if it sees a hyena or if it thinks it sees a hyena. It's not a SOA-BI that adds stock value. It's the shareholders' belief that SOA-BI has added stock value by delivering a competitive edge. Stock price is both a thermometer, a measure of value in a moment of time, and a barometer, a forecast of value in coming and comparative quarters. General market conditions or sector weakness will pressure stock price, but managerial excellence can balance those trends. The market is quick to absorb new information including architecture and tools such as BI. "To see what is in front of one's nose needs a constant struggle," George Orwell observed, and it's easy to be tricked by the spin of enterprise suites and power suits to lose sight of what is really happening, to accept as normality or even as badges of honor the dripping pipes and the wine stains on the carpet that become part of our corporate houses of straw or sticks. Stock price forces us to look beneath the surface of things in the same way that good doctors can look at a person and see a cadaver. It delivers a stern verdict not so much on whether the SOA is good or bad or whether the BI is good or bad but on whether the management is good or bad. A stock that drops from \$60 to \$45 is a quarter million dollars in lost equity in a one million dollar company or a quarter billion dollars in lost equity in a billion dollar company. An imploding stock may seem peripheral to myopic integrators with their shiny toys in their SOA-BI sandbox. But for business-centric enterprises that are service and thusly customer driven, nothing is more demonstrative than stock price.

A company may respond to its sliding stock by cutting its SOA commitment. A wiser approach is to focus less on SOA-BI and more on its cultural values. This will above all take the gift of boredom-time to think deeply, broadly, critically, and honestly about the company's belief system and the execution that flows from those assumptions. These reflections may be rendered into mandates that shape a new consensus: "Thou shalt never say that's the way we've always done things". "Thou shalt always have a daily stand-up", "Thou shalt admit failure joyfully, fearlessly, and promptly" and so on. The idea is to develop a rank and file consensus that bridges ideals to execution, from what to how. All failing companies, for example, claim they value

communication but few of those companies insist on daily scrums and scorn for knee-jerk conservatism. These rules will have the effect of enhancing core values such as accountability, transparency, and productivity, addressing systemic organizational flaws, and also providing a counter-weight against pushback from middle-management and the siloing tendencies of agile teams so as to create a unified enterprise vision. These are some of the conditions to ensure that a company's SOA-BI will make a profitable difference in the market place.

Conclusion: The Tails of the Three Little Pigs

Well, the wolf huffed and puffed but he could not blow down that brick house. But the wolf was a sly old wolf and he climbed up on the roof to look for a way into the brick house. The little pig saw the wolf climb up on the roof and lit a roaring fire in the fireplace and placed on it a large kettle of water. When the wolf finally found the hole in the chimney he crawled down and KERSPLASH! Right into that kettle of water and that was the end of his troubles with the big, bad wolf.

In the parable of the pigs, a brick home failed the use case. The use case was to shelter pigs from rain, cold, and wolves, but the wolf was able to invade the home. It was only the kettle that saved the pigs. Doing nothing, doing the wrong thing, or not doing enough can fail the use case. It's like putting pink ribbons on the pigs' curly tails and hoping that the pigs will prevail. A perfectly working SOA-BI may not be enough. Good SOA must not just work. It must also meet clear business goals in a real-world, human context. Integrating business intelligence with SOA can help meet those business goals.

And they lived happily ever after.

Not really, it's a stretch to suggest that is the future of enterprises with SOA-BIs. To the contrary, the journey to close the gap between the ideal and the real when it comes to SOA or BI and especially SOA-BI is hard and exacting. But success came to the three little pigs through sound architecting, resourceful tooling, and collaborative flexibility. These principles endure for those striving to build SOA-BI.

References

[REF-1] Philip Russom, "Next Generation Data Warehouse Platforms", TDWA Best Practices Report, 2009, 22. This appears to be the future of business intelligence:

1. Third party SOA-BI products increasingly address ETL issues of volume and throughput.
2. Cloud computing and Software-as-a-Service (SaaS) are ubiquitous.
3. In-memory processing, 64-bit processing, and data warehouse appliance architectures are standard.
4. Operational applications have callable BI components, with improvements in response time, scaling, and concurrency. Near or real time BI analytics is a baseline expectation.
5. Open source BI software replaces vendor offerings. "Open source DBMSs are available from Infobright, MySQL, and PostGres. Open source data integration tools are available from Apatar, JitterBit, Pentaho, and Talend. Open source reporting or analysis tools are available from Actuate and Jaspersoft". Also consider Hive, an open-source data warehouse system that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in Hadoop-compatible file systems. <http://hadoop.apache.org/>

[REF-2] Ralph Kimball, Kimball University, "Design Tip #108: Can the Data Warehouse Benefit From SOA?", Number 106, October 10, 2008. Kimball is regarded as one of the original architects of data warehousing and is an evangelist for dimensional modeling.

[REF-3] Cesare Pautasso, "Rest vs. SOAP: Making the Right Architectural Decision", Faculty of Informatics, University of Lugano (USI), SOA Symposium, 2008, Amsterdam. (Adapted from "Conceptual Comparison Summary").

[REF-4] State of Texas Health and Human Services Enterprise Architecture SOA Reference Model, 2006, pp 16-19. See also <http://www.w3.org/TR/wsdl20/> for WSDL core language specifications.

[REF-5] Wayne Eckerson, "The Secrets of Creating an Agile, Adaptable BI Environment," Keynote Address to the TDWI conference, August, 2010, San Diego. Much of the conference aimed at promoting this theme. I discuss SOA agility further in "Effective Top-Down SOA Management in an Efficient Bottom-up Agile World, SOA Magazine, April-May, 2010 <http://www.soamag.com/I38/0410-1.pdf>

[REF-6] Rossabeth Moss, Kanter, "What Would Peter Say: The Continued Relevance of the Drucker Perspective," November 2009, Harvard Business Review, 67.

Acknowledgements

I wish to thank the following individuals who reviewed and critiqued my paper: Steve Wisner, Director, IT, Genworth Financial, Errol Ryland, Director, MSS Technologies, Inc., and Atul Sharma, Senior OBIEE Developer, Ascentt Business Systems, Inc.

About the Author: Philip Wik

Philip Wik is a Senior Consultant for MSS Technologies. Philip has worked for JP Morgan/ Chase, Wells Fargo, American Express, Honeywell, Boeing, Intel, and other companies in a variety of applications development, integration, and architectural roles. He has published two books through Prentice-Hall: How to Do Business With the People's Republic of China and How to Buy and Manage Income Property.

www.servicetechmag.com/contributors/philipwik

