

Chapter 1: Overview and Architecture

Overview

The Microsoft Office Information Bridge Framework is a set of development tools, a metadata server, as well as a run-time component that connects the Microsoft Office system to a company's back-end line of business (LOB) systems. It was released in February 2003 and presented by Steve Ballmer at the Office Developer Conference during his keynote.

The driving force behind IBF is the ability to link your company's LOB systems together in such a fashion that information workers can be ignorant of the underlying infrastructure. For businesses that use the Microsoft Office System for information gathering, reporting, analysis, and collaboration, providing a method for working within this familiar interface has many benefits.

- Lower training investment
- Single point of access to data for information workers
- A new way to view existing data that provides context to the information worker
- A new way to add and update data so the information worker can make better decisions.

IBF does not attempt to replace the role of smart clients and other enterprise applications in the business environment. It simply provides another set of views into the back-end data, and the actions they contain within the familiar Microsoft Office system.

Business Case

The Problem Today

The Information Worker Dilemma

- Information workers send and receive e-mail messages that are neither "live", or "actionable"
- Information workers create documents in Microsoft Word 2003 and Microsoft Excel 2003 that are also not "live" or "actionable"
- Information workers often perform heavy copy-paste operations to build a view of multiple systems
- "Occasional users" of back-end systems must learn and re-learn the applications to work with a reasonably small set of data

The Solution Developer Dilemma

- Solutions are developed using smart documents that require custom code and a profound understanding of the back-end systems
- Each of these new applications becomes an information silo with little or no reusability
- Service-Oriented Architectures (SOA) exist in many companies today, but there are few Business-Oriented Architectures (BOA) providing context to these services

Almost every document contains a reference to data that resides in back-end systems. An example could be the word "Contoso", which could relate to "Contoso the customer", which could be found in the company's CRM system, Order Entry system, or possibly other data sources.

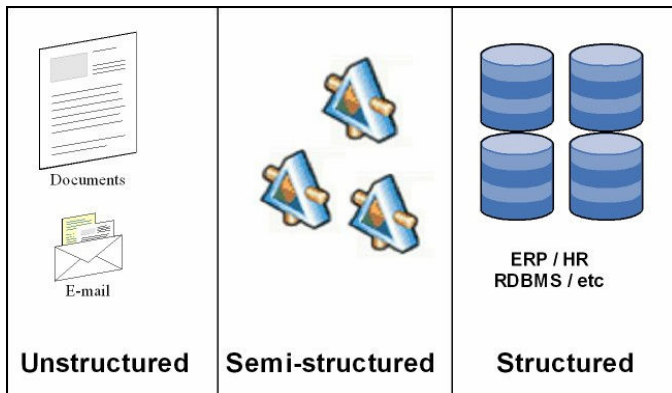
IBF can provide a complete view of the "Contoso" entity, from within the Office system.

How the Information Bridge Framework Works

IBF builds on the XML support in Microsoft Word 2003, Microsoft Excel 2003, and Microsoft Outlook 2003. In this way it can deliver a standardized, metadata driven approach to:

- Expose entities (like customers)
- Relate these entities across multiple back-end systems
- Display this data in a contextually aware manner using Office documents
- Allow the information worker to act on this data from their Office documents

Data exists in several different formats today.



Unstructured data are an information worker's Word documents, Excel spreadsheets, or Outlook e-mail messages. These are static, non-living, and non-related silos of information.

Semi-structured data are considered xml documents, web services, or existing smart documents that were custom developed. They are semi-structured because data could be inter-related among the data stores, but often in a clumsy or tightly-coupled manner.

Structured data are commonly back-end databases, and custom solutions. This data is inter-related within the same system, but not related between databases or systems. Accessing this data in new formats often requires deep knowledge of the system, or a strong relationship with the vendor.

The Information Bridge Framework provides another approach to connecting the user to the back-end systems using web services and associated metadata to describe these services.

Figure 1-1 shows the information workers view, including context-sensitive data and actions - from within Microsoft Office

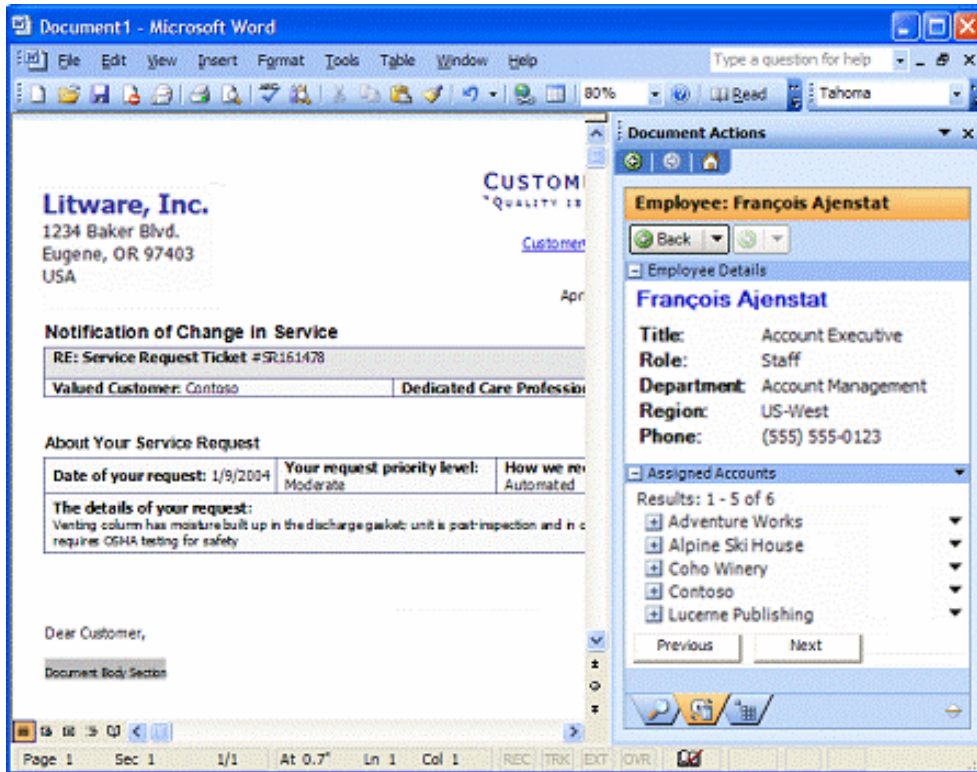


Figure 1-1. The Information Worker's View

Technologies

XML

This ubiquitous language can be found in software by many vendors and technologies. It is platform independent and consists of only text, w/ tags to describe elements. XML files can describe virtually any type of document or data structure possible.

Microsoft Office Professional Edition 2003 supports XML Web Services enabling the IBF client to connect to the Information Bridge Framework Metadata Service.

XML Schemas (XSD)

XML Schemas define the structure and type of data that each element within an XML file can contain. The schema frequently resides close to the XML file, and its location is defined at the top of the related XML file.

The XML Schema is used to validate the XML file.

XML Style Sheets (XSL)

The eXtensible Stylesheet Language (XSL) is a family of languages which allows one to describe how files encoded in the XML standard are to be formatted or transformed. The family has three parts:

- XSL Transformations (XSLT): an XML language for transforming XML documents from one syntax to another
- XSL Formatting Objects (XSL-FO): an XML language for specifying the visual formatting of an XML document
- the XML Path Language (XPath): a non-XML language used by XSLT, and XLink, to access or refer to parts of an XML document.

Simple Object Access Protocol (SOAP)

SOAP is a standard for exchanging XML-based messages over a computer network, normally using HTTP. SOAP forms the foundation layer of the web services stack, providing a basic messaging framework that more abstract layers can build on.

Web Services Description Language (WSDL)

WSDL describes the public interface to the web service. This is an XML-based service description on how to communicate using the web service; namely the protocol bindings and message formats required to interact with the web services listed in its directory. The supported operations and messages are described abstractly, and then bound to a concrete network protocol and message format.

Information Bridge Engine

This component requests data from the Metadata Service and renders content to the user

Task Pane

Word and Excel both contain a task pane, which hosts panes such as Help, Formatting, Style, and Document Actions. Each pane can be divided into several regions. Chapter 5 will discuss controlling the panes programmatically

Microsoft Outlook does not natively have a task pane, so the IBF client adds a floating window that cannot be docked.

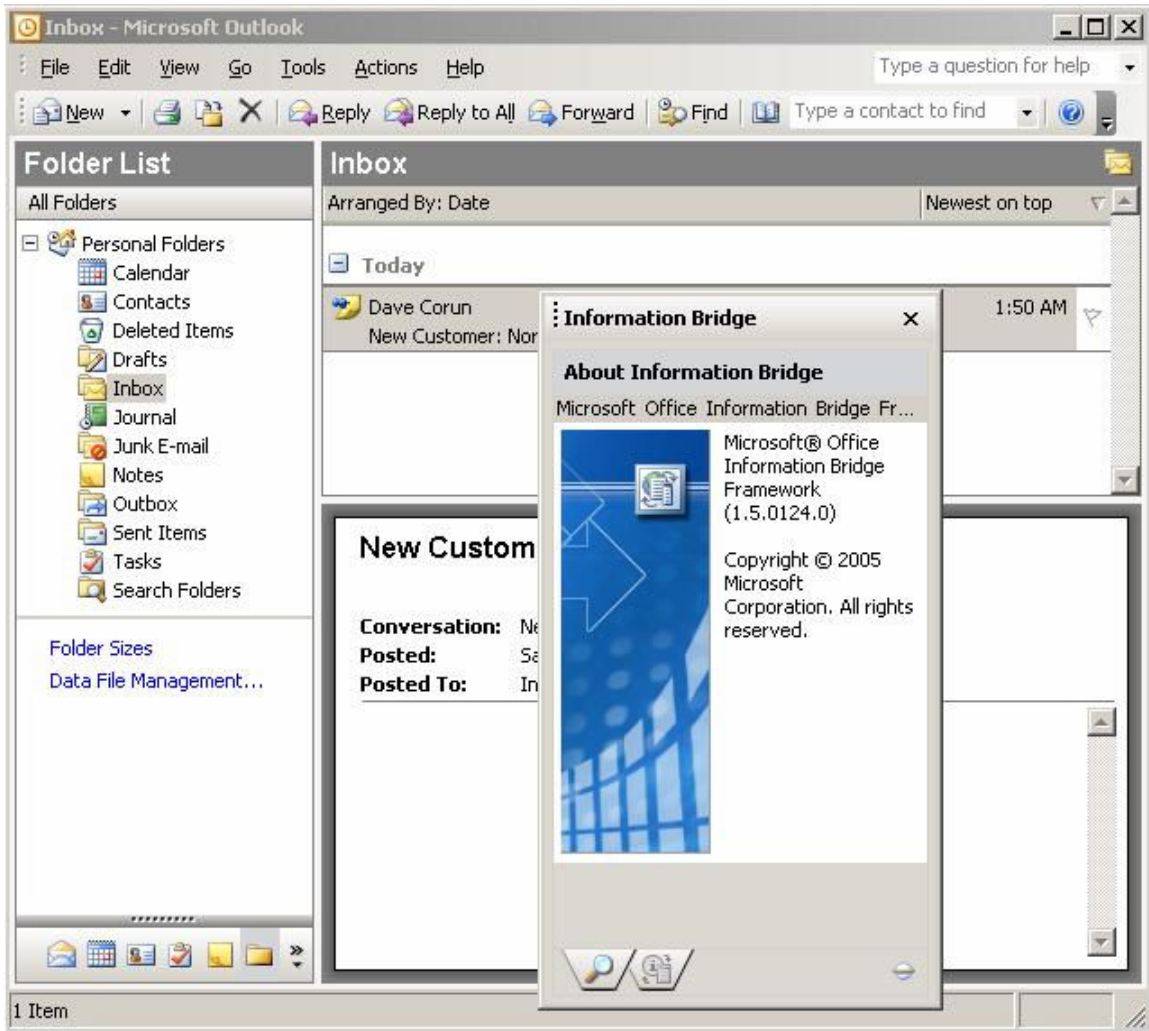


Figure 1-2. Floating Task Pane

Information Window

This is the generic term for the Document Actions pane that is part of the task pane.

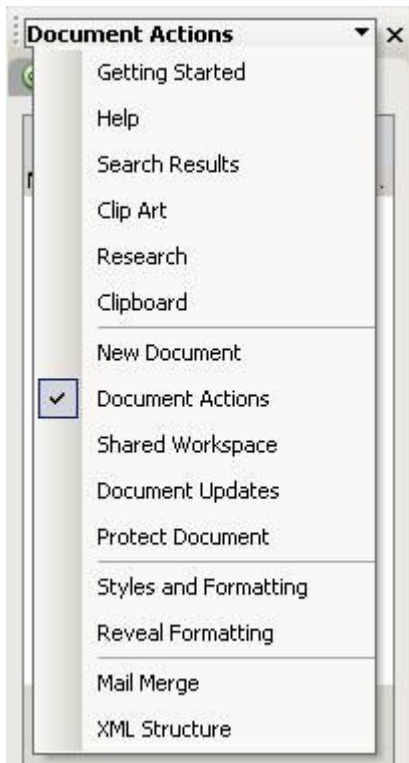


Figure 1-3. Document Actions Pane

Smart Tags

Smart tags are hidden pointers implemented inside a document, spreadsheet, or e-mail message. Text tagged with these hidden pointers appears with a dotted underline so that you know the text is of special interest to you.

Smart tags will be discussed in the context of IBF in chapter 6. If you want a deeper understanding, I highly recommend Ty Anderson's *Office 2003 Programming - Real World Applications* (Apress).

Attached Schemas

Word supports the ability to attach an XML schema, or XSD and map data to a schema. This can be used to control the syntax of an XML document. This will be discussed in more detail throughout chapter 7.

Although Excel has the ability to use attached schemas, IBF currently only supports attached schemas in Word 2003.

Application Architecture

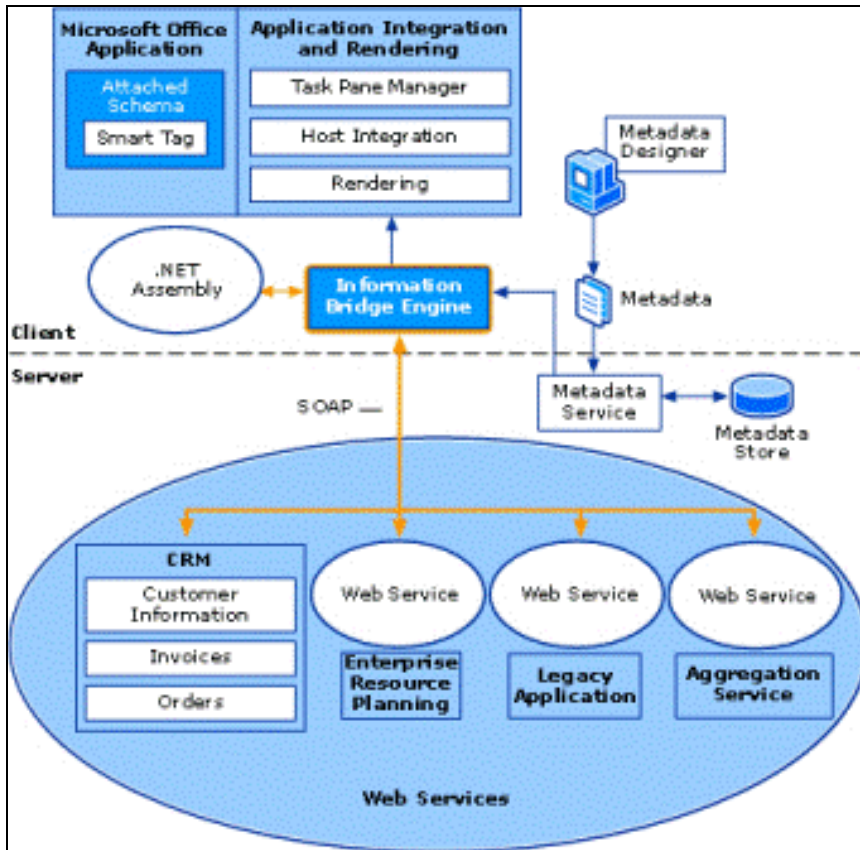


Figure 1-4. High-level Architecture

Prerequisites and Required Components

Information Bridge Framework Metadata Server

- Windows 2003 Server Standard Edition
- Internet Information Services 6.0 (IIS)
- .NET Framework 1.1 SP1
- SQL Server 2000 Service Pack 3
- Information Bridge Framework Service 1.0

Metadata Designer

- .NET Framework 1.1 SP1
- MSXML 4.0 Service Pack 2 (Microsoft XML Core Services)
- Information Bridge Metadata Designer 1.5
- Visual Studio .NET 2003 Professional
- Microsoft Office Professional Edition 2003 Service Pack 1
- Microsoft Office 2003 Primary Interop Assemblies (PIA)

Information Bridge Framework Client

- Information Bridge Framework Client 1.5
- .NET Framework 1.1 SP1
- Microsoft Office Professional Edition 2003 Service Pack 1

- Microsoft Office 2003 Primary Interop Assemblies (PIA)

Details on installation are in the following chapter.

Chesapeake Logic - Business Problem

Chesapeake Logic is a Maryland based company that gathers and ships local crabs from the Baltimore harbor to surrounding restaurants. Their business partners and customers depend on speedy delivery of the fresh goods in order to provide tasty meals to hungry clientele. They currently employ 50 information workers that do such various tasks as planning delivery schedules based on needs, responding to customer concerns, billing, and building relationships with new customers. Data is stored in back-end systems that include a custom logistics application for delivery schedules, an off-the-shelf CRM system for customer information and billing, a custom order entry system, and a custom sales database for leads.

Chesapeake Logic also uses a SharePoint Portal Server internally to collaborate on documents, including maps of the harbor and the best crabbing locations.

The data is stored in several Microsoft SQL Server 2000 databases, and one Oracle database.

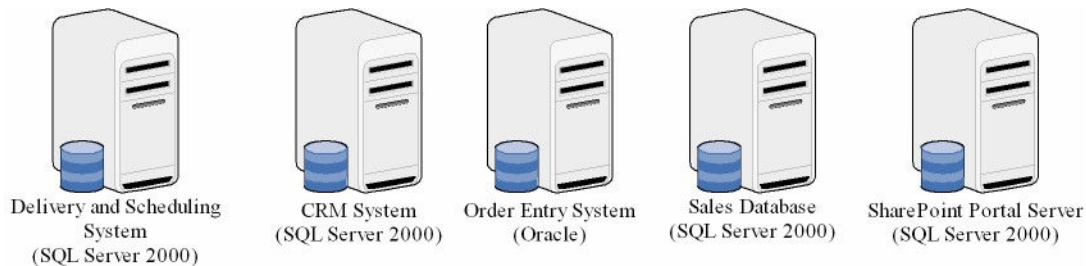


Figure 1-5. The Chesapeake Logic Infrastructure

Information workers spend the majority of their day switching between these applications and communicating through e-mail. Upon inspection of typical e-mail messages being sent and received, most included partial copies of data originally stored in the back-end systems. A business analyst observed the behaviors of information workers and found much of their time was indeed spent alt+tabbing between the various systems.

Another issue Chesapeake Logic faces is the loss of important customer communication. While the IT administrators set up e-mail aliases for incoming orders, support, and sales, it's not always clear who responded, if anyone. It's not uncommon for a customer to order a same-day shipment in the morning, and never receive it. Customer satisfaction is of high importance to senior management. They agree that while it may not be 100% possible to eliminate all of these issues, order entry personal should have an avenue to alert the dispatchers of any high priority orders.

Today they have to open the CRM system to see if customer information exists. If the customer doesn't exist, they may have to get the address and contact information from the custom sales database, and then enter it into the CRM system by hand. Usually this is done through heavy copying and pasting. After they verify the customer information for billing, they enter the order into the order entry system. If this is a high priority order, they may call and verify with the dispatchers that the order was received on their end and scheduled.

Since information workers are familiar with the Microsoft Office System, they often copy data from the various systems and massage the results in Word for presentation and Excel for analysis.

However there is no relationship between the documents they're creating and the back-end systems.

Chesapeake Logic - Proposed Solution

Senior management approved a full requirements gathering and analysis by an outside development company that specializes in unifying business data. First the development company defined scenarios that involved the order entry personnel. Chesapeake Logic sorted results in order of importance.

The development company proposed developing a small IBF solution that would expose data from the CRM system on-demand based on the information worker's context in the various Office applications. In this way, an order entry person could receive an e-mail from a customer, and view all of the relevant data from the various back-end systems. Several iterations were also planned that would increase the amount of data feeding into the information workers' Office applications. The development company emphasized that the addition of a new system and various views of data would be easier once the infrastructure was in place.

Chapters 3-9 describe the Chesapeake Logic solution in detail. Figure 1-6 displays the final infrastructure.

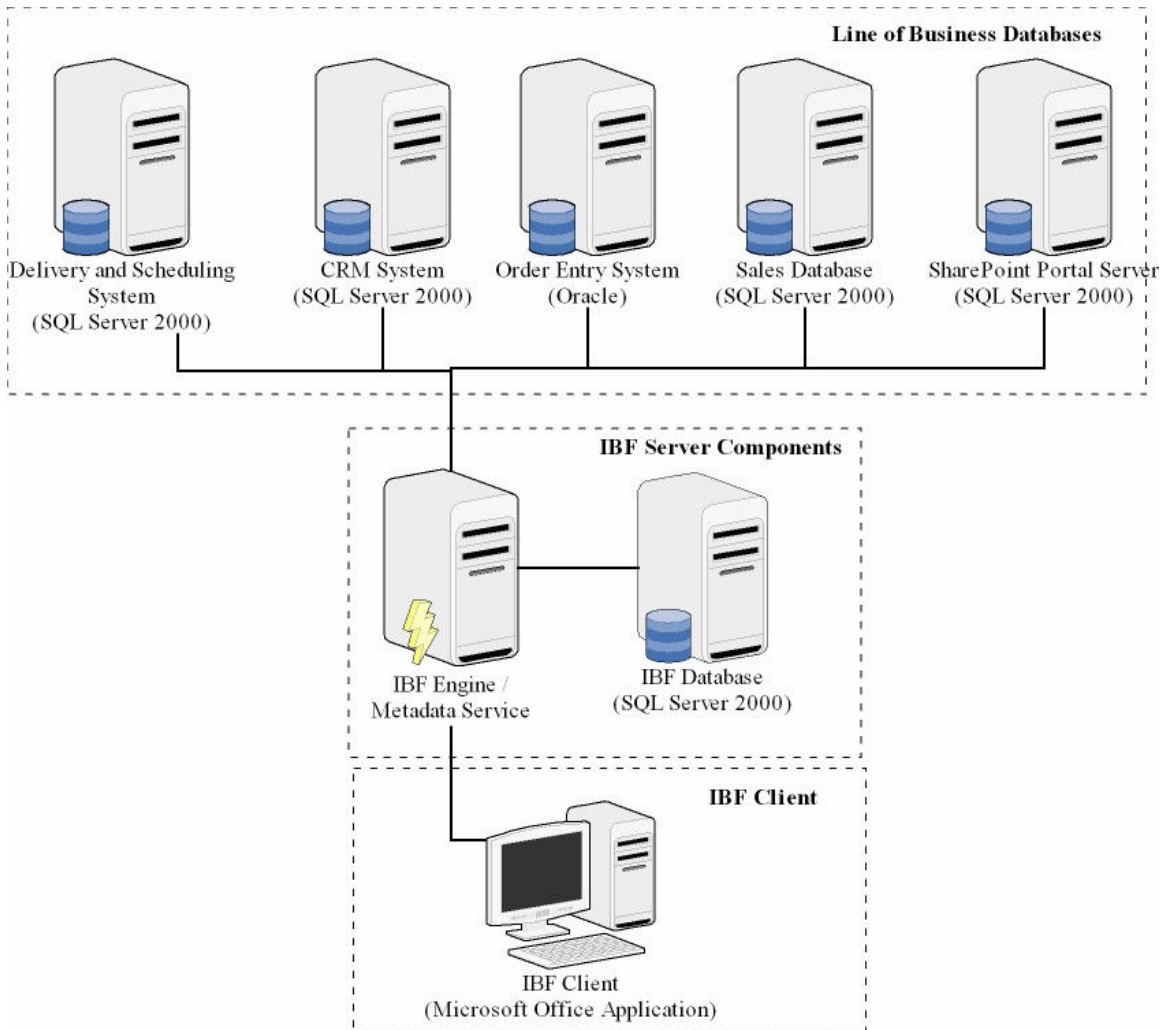


Figure 1-6. Chesapeake Logic's Completed IBF Solution