Technologies used in Enterprise Application Integration and Business-to-Business Integration processes

Anca Mehedintu, Ion Buligiu, Cerasela Pirvu
Faculty of Economics and Business Administration,
University of Craiova, Romania
ancamehedintu@yahoo.com, buligiu_ion@yahoo.com, ccpirvu2006@yahoo.com

Abstract
Enterprise computing has progressed enormously in just the last few years. Especially with the advent of the Web, not only is it possible for diverse organization to automate and integrate their businesses and computer operations, it is imperative that they do so. Suddenly, as more and more corporations become Web enabled and find themselves relying on a myriad of applications, the ability to evolve and integrate existing applications becomes significant. Virtually all enterprise organizations at some time face the problem of integrating different applications and database systems. In addition, enterprise organizations must constantly evolve. This need to evolve occurs as enterprises strive for competitive advantages. In today’s economy, it is rare for an organization to continue to be successful by merely maintaining the status quo. In a sense, enterprises are forced to evolve to stay at the forefront of their industries. Enterprises frequently find themselves having to merge with other enterprises, reorganizing their internal structure, and adopting new technologies and platforms as they strive for competitive advantages.

The e-Business model is particularly useful for managing purchasing and supply-chain issues, managing customer relationships and providing customer service, and providing Web-based applications and services.

Since it is imperative that enterprises adapt to business and technology driven changes, they need an e-Business model more than ever to adapt their existing business processes, applications, and enterprise systems to these changes.

In these situations, enterprise application integration assumes a great importance. Enterprise application integration (EAI) enables an enterprise to integrate its existing applications and systems and to add new technologies and applications to the mix. EAI also helps an enterprise to model and automate its business processes.

Keywords: e-business, business-to-business, enterprise application integration (EAI)

Introduction

Enterprise computing has progressed enormously in just the last few years. Especially with the advent of the Web, not only is it possible for diverse organizations to automate and integrate their businesses and computer operations, it is imperative that they do so. Suddenly, as more and more corporations become Web-enabled and find themselves relying on a myriad of applications, the ability to evolve and integrate existing applications becomes significant.

Virtually all enterprise organizations at some time face the problem of integrating different applications and database systems. In addition, enterprise organizations must constantly evolve. This need to evolve occurs as enterprises strive for competitive advantages. In today's economy, it is rare for an organization to continue to be successful by merely maintaining the status quo. In a sense, enterprises are forced to evolve to stay at the forefront of their industries. Enterprises frequently find themselves having to merge with other enterprises, reorganizing their internal structure, and adopting new technologies and platforms as they strive for competitive advantages. More and more, they are adopting an e-Business strategy.

In these situations, enterprise application integration assumes a great importance. Enterprise application integration (EAI) enables an enterprise to integrate its existing applications and systems and to add new technologies and applications to the mix. EAI also helps an enterprise to model and automate its business processes (Figure 1).

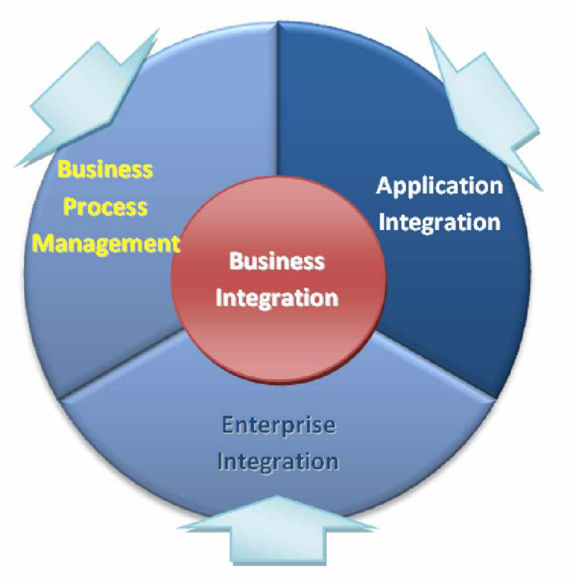

Figure 1: Enterprise application integration

Fundamentals of enterprise application integration

Enterprise application integration (EAI) entails integrating applications and enterprise data sources so that they can easily share business processes and data. Integrating the applications and data sources must be accomplished without requiring significant changes to these existing applications and the data.
Typically enterprise applications grow into silos of data and application functionality. System disparity precipitates an abundance of costly manual reporting and data entry processes. Enterprise Application Integration creates valuable information flow, reduced costs and allows for faster response to marketing opportunities (Giurca Vasilescu, 2007, p.665).

Before EAI, integrating applications and data within a corporate environment has been an expensive and risky proposition. As we noted previously, companies were trying to combine applications that often ran on different hardware platforms and had no protocols for communicating with other software packages outside of their own narrowly defined realm. In a sense, companies had “islands” of business functions and data, and each island existed in its own, separate problem domain (Figure 2).

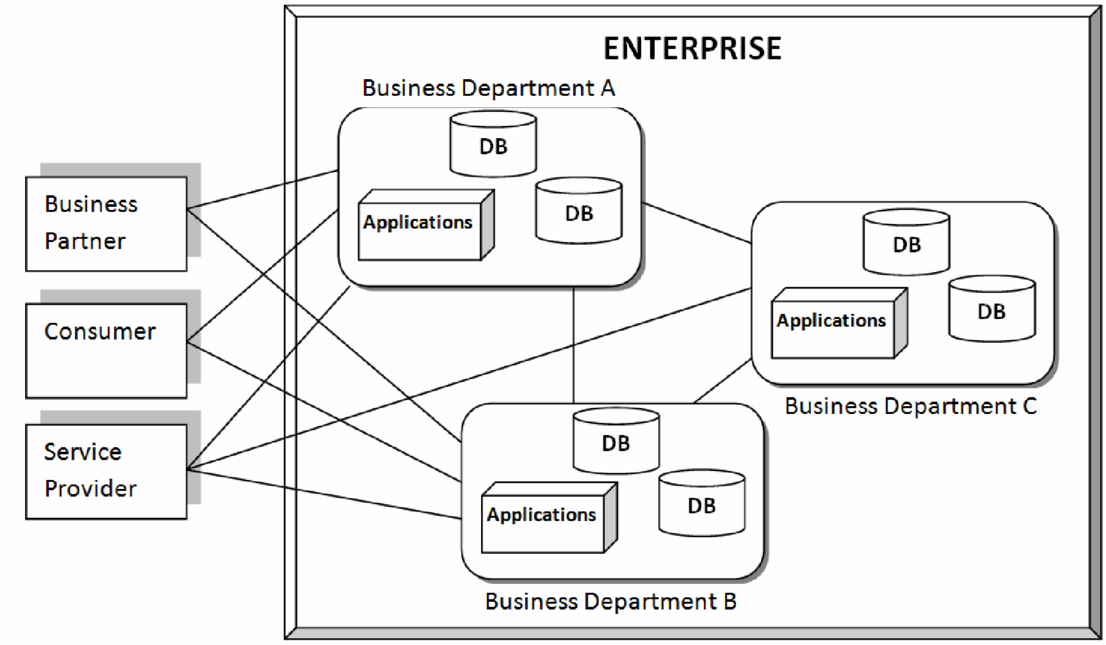

Figure 2: A Typical Enterprise Domain

Supply chain management applications (for managing inventory and shipping), customer relationship management applications (for managing current and potential customers), business intelligence applications (for finding patterns from existing data from operations), and other types of applications (for managing data such as human resources data, internal communications, etc) typically cannot communicate with one another in order to share data or business rules. For this reason, such applications are sometimes referred to as islands of automation or information silos. This lack of communication leads to inefficiencies, wherein identical data are stored in multiple locations, or straightforward processes are unable to be automated.

Enterprise application integration (EAI) is the process of linking such applications within a single organization together in order to simplify and automate business processes to the greatest extent possible, while at the same time avoiding having to make sweeping changes to the existing applications or data structures. In the words of the Gartner Group, EAI is the unrestricted sharing of data and business processes among any connected application or data sources in the enterprise (Hohpe, 2003, p.685).

One large challenge of EAI is that the various systems that need to be linked together often reside on different operating systems, use different database solutions and different computer languages, and in some cases are legacy systems that are no longer supported by the vendor who originally created them. In some cases, such systems are dubbed "stovepipe systems" because they consist of components that have been jammed together in a way that makes it very hard to modify them in any way (Linthicum, 1999, p.39).

EAI can be used for different purposes:

- **Data (information) integration**: ensuring that information in multiple systems is kept consistent. This is also known as EII (Enterprise Information Integration).
- **Process integration**: linking business processes across applications.
- **Vendor independence**: extracting business policies or rules from applications and implementing them in the EAI system, so that even if one of the business applications is replaced with a different vendor's application, the business rules do not have to be re-implemented.
- **Common facade**: An EAI system could front-end a cluster of applications, providing a single consistent access interface to these applications and shielding users from having to learn to interact with different applications.

Multiple technologies are used in implementing each of the components of the EAI system:

- **Bus/hub:** This is usually implemented by enhancing standard middleware products (application server, message bus) or implemented as a stand-alone program (i.e., does not use any middleware), acting as its own middleware.
- **Application connectivity:** The bus/hub connects to applications through a set of adapters (also referred to as connectors). These are programs that know how to interact with an underlying business application. The adapter performs two-way communication, performing requests from the hub against the application, and notifying the hub when an event of interest occurs in the application (a new record inserted, a transaction completed, etc.). Adapters can be specific to an application (e.g., built against the application vendor's client libraries) or specific to a class of applications (e.g., can interact with any application through a standard communication protocol, such as SOAP or SMTP). The adapter could reside in the same process space as the bus/hub or execute in a remote location and interact with the hub/bus through industry standard protocols such as message queues, web services, or even use a proprietary protocol. In the Java world, standards such as JCA allow adapters to be created in a vendor-neutral manner (Zahavi, 1999, p.21).
- **Data format and transformation:** To avoid every adapter having to convert data to/from every other applications' formats, EAI systems usually stipulate an application-independent (or common) data format. The EAI system usually provides a data transformation service as well to help convert between application-specific and common formats. This is done in two steps: the adapter converts information from the application's format to the bus's common format. Then, semantic transformations are applied on this (converting zip codes to city names, splitting/merging objects from one application into objects in the other applications, and so on).
- **Integration modules:** An EAI system could be participating in multiple concurrent integration operations at any given time, each type of integration being processed by a different integration module. Integration modules subscribe to events of specific types and process notifications that they receive when these events occur. These modules could be implemented in different ways: on Java-based EAI systems, these could be web applications or EJBs or even POJOs that conform to the EAI system's specifications.

- **Support for transactions:** When used for process integration, the EAI system also provides transactional consistency across applications by executing all integration operations across all applications in a single overarching distributed transaction (using two-phase commit protocols or compensating transactions).

Methodologies used in Web-driven application integration

With the advent of the Web, enterprise application integration has taken on a larger significance beyond that of merging application systems solely within an enterprise.

Enterprise servers now handle and maintain huge amounts of data and business logic. Furthermore, because the Web enables easy information and service access, it has become a principal means of communication. An enterprise must be able to make its business data accessible to others, from internal employees to external partners, suppliers, and buyers. Employees require access to the enterprise data to keep abreast of company policies and developments, and to carry on the internal business of the company. For example, employees file their expense reports through a Web interface. Business partners may be communicating important technological information.

Buyers and suppliers need access to enterprise data to facilitate the parts ordering and delivery process.

As more and more businesses establish a presence on the Web, Web-driven EAI becomes more and more essential. Enterprises need to integrate their existing applications and enterprise systems to drive their business-to-consumer and business- to-business interactions, plus their other Web services. In fact, success in e-Business is driven by an enterprise’s ability to integrate existing applications and extend the reach of these applications to Web-based access.

Up to now, applications were classified as either front-office or back-office applications. Front-office applications are considered to face the customer or end user. Front-office applications include applications for customer relationship management and marketing automation. Back-office applications provide the information infrastructure for running the back-end business processes of an enterprise.

Applications provided by an enterprise resource planning (ERP) system are good examples of back-office applications.

Traditional EAI focused on integrating the front and back office applications. However, traditional EAI is becoming Webdriven EAI. Rather than being targeted to the front end or the back end, most EAI applications are now integrated for the front and back ends and Web enabled. Just as it is imperative for an enterprise information system (EIS) to move to a web-based architecture, there is also a need for enterprise applications to be deployed on widely adopted, standard application platforms. Enterprises now regard application servers as mature platforms for developing Web-based applications. As Figure 2 shows, application servers are particularly appropriate for the B2C and business-to-business (B2B) areas that place so much stress on application integration.

The application server provides a natural point of integration between an enterprise’s existing enterprise information systems and the Webbased applications. The application server also helps handle transactions and can be scaled as needed.

The J2EE application platform is the technology of choice for enterprises and application vendors. Figure 3 illustrates this Web direction to which enterprises are currently moving. The success of the Java programming language and the J2EE platform are also responsible for this Web-driven application integration, in large part because they make it easier to develop and implement Web-based applications.

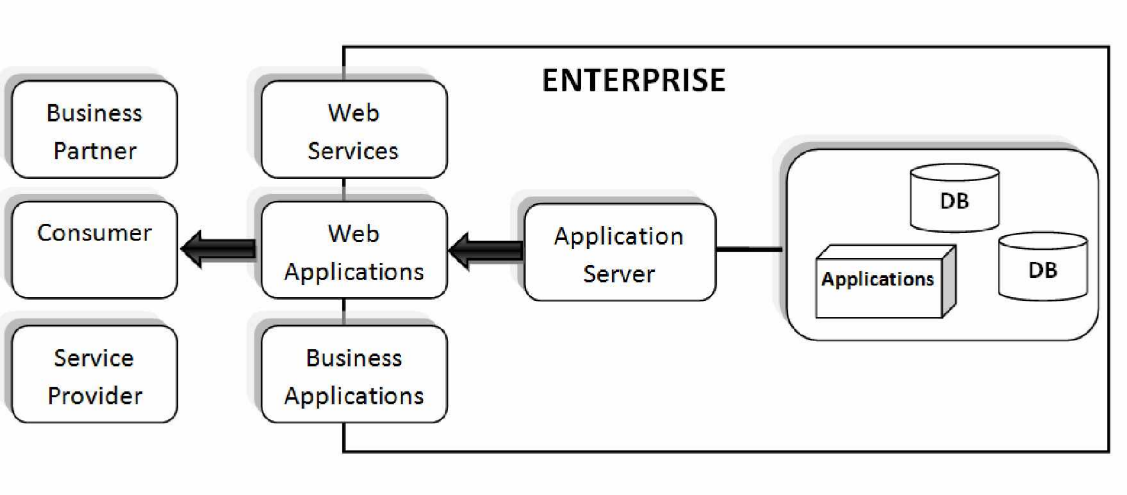

Figure 3: Web-driven Application Integration

To maximize this Web-driven application integration, enterprises are turning more and more to the Java programming language and the J2EE platform. Java is a platform-independent computer language that is designed for the Web, and it is a successful, widely adopted platform for enterprise application development (Sharma, 2001, p.90).

In addition to the Java platform, enterprises are using XML to exchange corporate data across application domains. XML is a platform-independent way of representing data formats, and it is invaluable for exchanging data among different entities. There is a synergy between XML and Java. XML is to data what the Java programming language is to application services. Because of XML's platform-independent features, it serves as a foundation for the current generation of Web technologies.

Automating business interactions between suppliers, customers, and trading partners allows companies to better share information and optimize processes across the value chain from better demand forecasting to streamlined manufacturing to more responsive customer service:

- Reduce large value added network (VAN) fees.
- Reduce costs, processing time, and errors through automated processing of invoices, purchase orders, and payments.
- Synchronize price, product, and promotion information between trading partners and reduce disputes.
- Improve collaboration and visibility across the value chain.
- Achieve compliance with government and customer mandates such as HIPAA, RosettaNet, and EDI-INT.

Business-to-business (B2B) integration represents a significant area of opportunity for using technology to drive tangible ROI and fundamentally change the way businesses communicate and interact with each other (Naick, 2000, p.53).

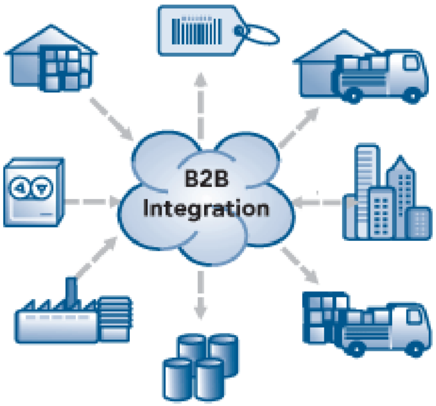

Figure 4: Business-to-business integration

- Manage the secure execution of transactions over the internet using accepted industry standards.
- Streamline inter-enterprise processes that go from your back-end systems to your partners' systems.
- Manage a large and diverse community of trading partners with minimal overhead.

Enterprise information systems

Before delving into the details of EAI, it is useful to understand the definition of an enterprise information system. An enterprise requires certain business processes and underlying data to run its business. An enterprise information system encompasses these business processes and information technology (IT) infrastructure.
Typically, the enterprise business processes include applications for handling payroll processing, inventory management, manufacturing production control, and financial accounting (accounts payable and accounts receivable).

We define an enterprise information system as an application or enterprise system that provides the information infrastructure for an enterprise. Typically, an EIS consists of one or more applications deployed on an enterprise system. An EIS provides a set of services to its users. Services exposed to clients may be at different level of abstractions—including the system level, data level, function level, and business object or process level. Graphically, this might look as shown in Figure 5 (Dumas, 2005, p.65-67).

In this EIS environment, the applications reside on the application server. The application server has a vendor-specific infrastructure, particularly regarding such services as transaction processing, security, load balancing, and so forth. Different vendors may supply the applications that sit on the server, or they may be developed by the IT department in house. Applications have been written in various languages, such as COBOL, C, and C++. There are application programming interfaces (APIs) for clients to access the different applications. An API is some routine that allows a client to do such operations as create a purchase order or update a customer record. The data access interface represents the means of access to the legacy datastores or relational databases. The business object interfaces are abstractions representing the business-specific logic for accessing functions and data (Dunn, 2004, p.43).

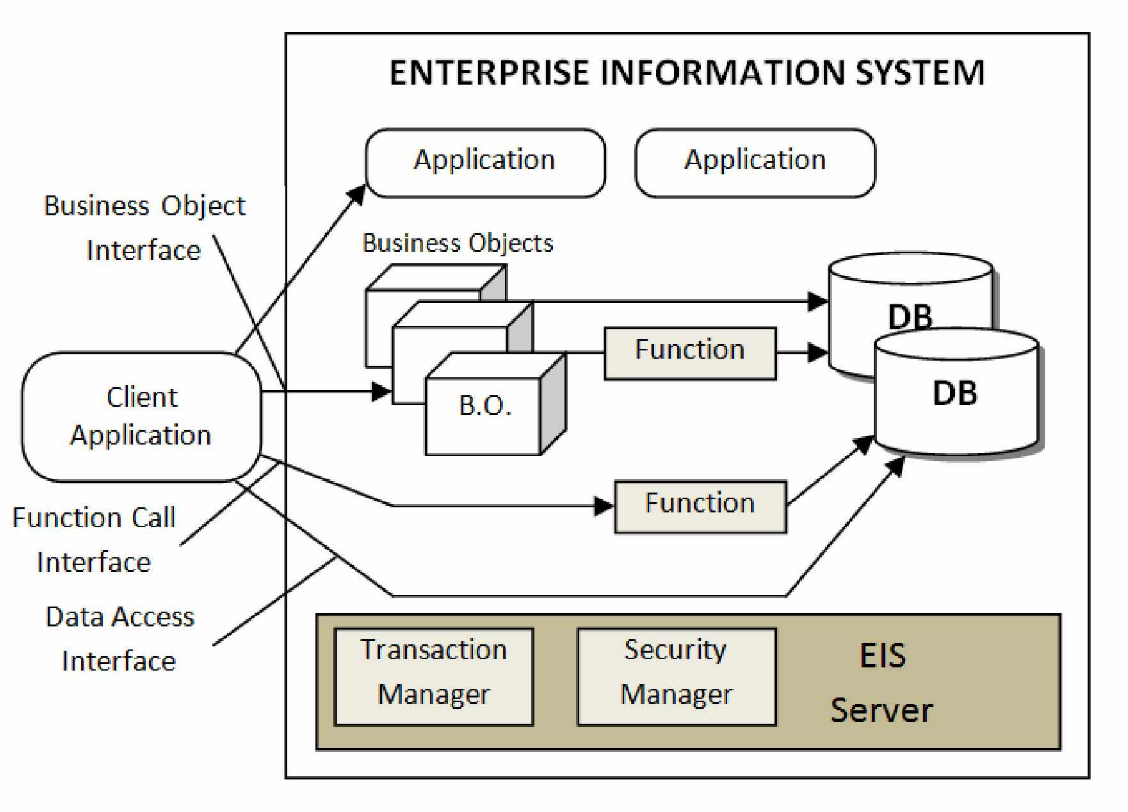

Figure 5: Enterprise Information System Environment

Typically, EISs include the following:

- Enterprise applications that have been developed by an enterprise specifically to meet its business needs. These are considered to be custom applications. Typically, legacy applications run on different computing environments. In addition, they are developed using different programming languages, such as C and COBOL.
- Applications that are part of an ERP suite of applications. ERP applications cover a wide range of functions, including inventory management, production control, human resources. Logistics applications are another set of ERP applications.
- Transaction programs running on a mainframe transaction processing system.
- Legacy databases that manage data critical to the business processes of an enterprise.

For a variety of reasons, EISs vary greatly even within the same enterprise. Usually, EISs vary because of the following:

- Enterprises purchase or implement different EISs over a period of years as their business needs grow.
- Enterprises deploy enterprise applications on different platforms or architectures.

- Enterprises customize an EIS to fit their own unique business needs.

Typically, an enterprise develops EISs over time, as a need for a particular EIS arises. For example, an enterprise may start out by purchasing a manufacturing system. Over the years, as its business grows, it incrementally adds different accounting packages, customer support, human resources, and so forth. It may be able to add some systems to the platform that hosts its manufacturing operations.
However, other packages require different platform capabilities, or have only been developed for a particular platform or architecture. Not only does the enterprise add the new software systems, it also buys additional hardware that may be completely different than its original configuration.

It is easy to see that when an enterprise has been in business for a long time, it may very well have EISs in use that have been developed and installed on different computing platforms and architectures. It is not uncommon for a large, established enterprise to have a few applications that run on a mainframe transaction processing system. These mainframe based systems may have been purchased years ago.
The same enterprise runs other applications that may be part of an integrated ERP suite of applications. In addition, it is typical for an enterprise to customize its applications to its own enterprise-specific business processes.

This level of customization can vary greatly. For example, an enterprise may purchase an off-the-shelf ERP application, and then customize the application so that it addresses its specific business processes. At the same time, it may develop other applications internally, using its own employees or consultants. These internally-developed applications are completely custom applications, designed to specifically meet the enterprise’s business needs.

Conclusions

There is a definite trend among enterprises towards integrating their existing enterprise applications and information systems with Web-based applications and services.
More and more, enterprises must establish a Web presence and make their business services available to Web clients. However, at the same time, an enterprise cannot afford to discard its existing systems and applications, but must leverage these existing assets to be successful.
This Web-driven application integration is a process that closes the gap between existing applications and Web-based applications and services.
Ultimately, Web service and wireless clients, in a B2C or B2B context, will be able to initiate business processes that act on critical information maintained in EISs.

References

Dumas*, M., W.M.P.* Van Der Aalst*, A.H.M.* Ter Hofstede*, 2005,* “Process Aware Information Systems - Bridging People and Software through Process Technology”, Wiley, John & Sons, Incorporated.

Dunn, C., J.O. Cherrington, A. Sawyer Hollander, 2004, “Enterprise Information Systems: A Pattern-Based Approach”, 3rd ed, Irwin/McGraw-Hill.

Giurca Vasilescu, L., S. Tudor, D. Tapordei, 2007, “Economic integration of Romania. Evolutions and prospects”, School of Business and Economics, TEI Larissa, Greece, *MIBES 2007 Proceedings*, 664-675.

Hohpe, G., B. Woolf, 2003, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”, The Addison-Wesley Signature Series.

Linthicum, D.S., 1999, “Enterprise Application Integration”, *Addison-Wesley Information Technology Series.*

Naick, I., M. Berkhoff, D.V. Bosnich, 2000, “Business-to-Business Integration Using MQSeries and MQSI Patterns for e-business Series”, IBM Corporation, *International Technical Support Organization*.

Sharma, R., Stearns, B., 2001, “J2EE Connector Architecture and Enterprise Application Integration”, *Addison-Wesley*.

Zahavi, R., 1999, “Enterprise Application Integration with CORBA”, Wiley, John & Sons, Incorporated.