Enterprise Interceptor: A Framework for Applications Interchangeability

by
Adam Kuta

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

at

School of Computer Science and Information Systems

Pace University

May 2006

UMI Number: 3225000

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

We hereby certify that this dissertation, submitted by Adam Kuta, satisfies the dissertation requirements for the degree of *Doctor of Professional Studies in Computing* and has been approved.

_____-_____

Dr. Fred Grossman Date
Chairperson of Dissertation Committee

_____-_____

Dr. Charles Tappert Date
Dissertation Committee Member

_____-_____

Dr. Fran Gustavson Date
Dissertation Committee Member

School of Computer Science and Information Systems
Pace University 2006

Abstract

Enterprise Interceptor: A Framework for Applications Interchangeability

by
Adam Kuta

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Professional Studies
in Computing

May 2006

Business have invested heavily toward business system applications in an effort toward process standardization and having a single version of the truth data-wise to no avail. Most businesses application architecture is littered with various enterprise business application solutions without ever achieving the promise of standardization on a single application footprint, or ever achieving the single version of the truth.

We can walk into Home Depot today and buy a standard bathroom sink faucet that we like from any manufacturer, bring it home and install it on our bathroom sink. This is the same with practically all the components used in our homes, from the windows, to the tubs, to the sinks and so on. This is achieved obviously because of standards, but the standards do not limit one to the point where they are difficult to implement. For example, you can have bathrooms of all sizes using standard components. The standards in this case do not limit the size of the bathroom that one desires. We have made great strides in Information Technology (IT) in the area of object reusability applications and data integrations. However, we are constantly building and rebuilding databases/applications, moving data from one database/application to another over and over again.

This research will show how to we can utilize the Structured Query Language (SQL) Interceptor to help rational data movement among business application, and a mechanism toward business applications functionality interchangeability. The Interceptor deployed as a web services application, provides the middleware interoperability framework we are looking for to seamlessly integrate disparate business applications without impinging on the sources application or the target application. The Enterprise Interceptor: A Framework for Applications Interchangeability can be utilized to transform not only the way we build applications today, but the way we build applications in the future and how to better integrate our existing disparate business applications. By introducing the same principles and concepts as used in the construction of our homes, we can streamline applications deployment to the assembly of components from different vendors. Business application architect can then map the data to the targeted base repository.

Acknowledgements

This Word dissertation template was created by Dr. Lixin Tao for all of our DPS students based on the spirit of *One for All and All for One*.

Thanks to Dwight Pitter, IT Principal, Avon Products Inc, for providing all the coding expertise in building and testing the Interceptor and integration with the Oracle Service Provider.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In their quest for a single version of the truth and a 360-degree view of the enterprise, businesses have invested in projects to improve efficiency, effectiveness and productivity. They have built IT systems for Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Supply Chain Management (SCM) and systems for budgeting, planning and forecasting. They attempted to integrate data with data warehouses, data marts and business intelligence (BI) projects. They relied on enterprise application integration (EAI) software to integrate all their applications, but it failed to integrate the data. They tried enterprise information integration (EII) software to create virtual data warehouses, but it too failed to provide the silver bullet for avoiding the hard work of data integration. Despite all the investments, data continues to be inconsistent across the enterprise [20].

ERP and CRM applications systems dominate the business applications landscape. Their architecture continues to entrench us in applications development methods that are tightly coupled. The applications are built with presentation layer depending on the business layer, and the business layer depending on the data layer of the respective applications. The most important challenge with this development paradigm is how dependent the layers are on each other. By tightly integrating the layers, these applications' architecture stymie the flexibility the business requires to continuously integrate new business models

and processes. While we are looking to change from our current legacy environment by implementing ERP and CRM applications, the ERP and CRM applications lead us automatically to a new legacy environment. ERP and CRM applications instantly relegate us to the vendor's chosen technologies and provide very difficult mechanisms of integrating with other business applications, or interchanging a function from different vendors.

Packaged applications are natural stove pipes (silos). This reality not only places them squarely within the problem domain of most application integration projects, but it also makes them the most challenging applications [14]. The problem has always been and will always be in the way we build business applications. The packaged solutions are built in their own silos, always looking at data from the packaged solution point of view, as opposed to the way the solution is going to be implemented from a business prospective. Software applications must coexist in a business environment with other business applications. Human resources applications coexisting with Supply Chain applications, and Supply Chain applications coexisting with Financial Planning applications; different applications from different vendors intertwined in support of various business processes.

The common underlying way the applications communicate is through the exchange of data. The data is often duplicated and represented differently in the solutions data schemas. There are numerous software available in the market today that provide the mechanisms and transport layers for moving data from one application to another. Some of the software tools provide functionality that allows for multiple databases to be updated simultaneously in a synchronous manner. While these integration tools are

extremely helpful, the end result typically always results in a copy of data moved or duplicated in another applications database. Vendors offer business performance management (BPM) packages -- bundles of BI tools (MicroStrategy, Cognos, Brio, Oracle, etc), extract, transform and load tools (Informatica, WebMethods, Assential, Oracle, etc), analytic engines and pre-built data warehouses (Oracle, MicroStrategy, Rogue Wave, etc)-- for analyzing business data. Many of these packages provide great business value, but they also have the potential to create even more data silos inconsistent with ERP, data warehouse and other applications [20].

The promise of Enterprise Resource Planning (ERP) was to obtain functionality from a single vendor, which would eliminate the need for custom integration and management of multi-vendor application environment. Most of the users realized that this was a false promise during functional evaluations. The next best solution was to get as much as you could from a single vendor and augment it with point solutions. Buying best solutions from the same vendor increases the chance that the solutions are integrated among the vendor solutions. This is usually true with applications built by the vendor, but considerably more difficult with solutions the vendor acquired themselves. Many users have implemented this arrangement; however, this type of environment only reduces the need for custom integration and does not eliminate it [7].

The proliferation of applications and instances of a common application is a main concern of organizations trying to control total cost of ownership and aligning the IT function with business strategy and objectives. Most of Gartner's multinational client organizations are going through a process of rationalization of business application portfolio, usually involving an instance consolidation strategy. The proliferation of

business applications or multiple instances of a common application through-out the organization is typically a result of historical growth through acquisition or a decentralized management philosophy. [6]

1.1 Problem Statement and Solution Strategies

Today's business environment transcends the boundary of linking manufacturing department with sales department to human resources department. We are in an environment where vendors as well as consumers are demanding more insight of data, information and business logic to help shape the way they do business. The business requirements are fluid and in constant flux.

Imagine the number of times we have to design Customer tables, Product tables, City tables and so on. The bulk of the attributes are the same e.g., Name, Address, City, State, Zip, Zip Plus 4, Date of Birth, and so on. Pertinent information about the customer is defined in many ways. While there may be some nuances related to a specific country, or industry or business segments requiring some additional custom information, the bulk of the attributes are the same. The same is true for business rules and process functions. For example, Materials Requirement Planning (MRP) is a technique which assists a company in the detailed planning of its production, and is used by many Enterprise Resource Planning software vendors, as well as custom applications. In another example, all municipalities use tax Revenue Mills Calculation, to calculate property taxes. Yet we continue to define and redefine the calculation over and over again. One can go on and on with such examples, yet every software development goes through the process of developing and maintaining these components. We are constantly gathering

requirements, coding, testing and debugging these business functions over and over again. We are constantly moving data from one business application to another business application, always transforming the data to suite the target application view of the data.

There is nothing magical about these things. An MRP calculation is an MRP calculation; an independent body that monitors the manufacturing industry defines the elements of MRP . The key critical element is how the concepts behind Enterprise Interceptor will work and how does one go about making it happen. How do we assemble and deploy the best pieces of functionality from various vendors without moving lots and lots of data from one application's repository to another.

1.1.1 Problem Statement

Most companies' application portfolio is littered with numerous business applications supporting various functional areas in the enterprise. In the majority of these companies, multiple applications are used to support similar functions in different geographies or businesses. To achieve this, data is constantly being moved from one application repository to another, creating a complex data integration architecture, and challenging the time it typically takes to run the overnight schedule of events to ready the environment for processing the next day. The combination of legacy business applications and new applications implemented to support the changing business needs leads to a very complex, hard to manage, and difficult to support business application architecture.

These complex business application environments relegate the companies to embrace multiple expensive business applications integration architecture that usually include

multiple vendors and software. How can we build a business application framework that addresses the need to integrate our current disparate and legacy business applications architecture, while providing a platform for business application framework for future application development that is responsive to the changing business needs? How do we provide a framework that significantly reduces the movement of data among disparate applications, while stratifying the unique needs of each business application?

One recent study of six corporate data centers revealed that most of their 1,000 servers were using just 10% to 35% of their available processing power [1]. Desktop computers fare even worse, with IBM Corporation estimating average capacity utilization rate of just 5% [3]. Gartner Inc., the research consultancy based in Stamford, Connecticut, suggests that between 50% and 60% of a typical company's data storage capacity is wasted [11]. An overcapacity is by no means limited to hardware. Because software applications are highly scalable – in other words, able to serve additional users at little or no incremental cost- installations of identical or similar programs at thousands of different sites create acute diseconomies in both upfront expenditures and ongoing costs and fees. The replication from company to company of IT departments that share many of the same technical skills represents an overinvestment in labor as well. According to a 2003 survey, about 60% of the average U.S. Company's IT staffing budget goes to routine support and maintenance functions [8]. When overcapacity is combined with redundant functionality, the conditions are ripe for a shift to centralized supply [5].

1.1.2 Solutions Strategies

This research will focus on how to we can utilize the Structured Query Language (SQL) Interceptor to help rationalize data movement among business application, and a mechanism toward business applications functionality interchangeability. The Interceptor is deployed as a web services application, and provides the middleware interoperability framework necessary to seamlessly integrate disparate business applications without impinging on the source or the target applications. The Enterprise Interceptor Framework for Applications Interchangeability (Enterprise Interceptor) can be utilized to transform not only the way we build applications today, but the way we build applications in the future and how to better integrate our existing disparate business applications. By introducing the same principles and concepts as used in the construction of our homes, we can streamline applications deployment to the assembly of components from different vendors. Business application architect can then map the data to the targeted base repository.

The research demonstrates how to extend application integration tools (Loosely Coupled Applications Architecture Design, Web Services, Middleware, Interoperability, Reentrant Programming techniques) within the Enterprise Interceptor to bring something to the business software development approach that is currently available in an industry such as the construction industry. The finished product will be a virtual Enterprise Interceptor that will allow for applications functionality interchangeability from various vendors without impinging on the base code.

The intent of the Enterprise Interceptor is to have the bulk of the application configured by the applications architect through a components (design, tables, attributes, business

logic, user interface) mapping of the data to the source data repository, with disparate applications targeting a specific base repository, without changing or making modification to the disparate business applications.

Enterprise Interceptor provides a better mechanism of integrating current disparate business applications. The framework supports the development of new business application solutions as well as supporting the current legacy environments.

1.2 Research Contribution

The major contribution of this research are:

- Define a framework that can be used to support existing and new business applications – Enterprise Interceptor a Framework for Applications Interchangeability (Enterprise Interceptor).
- Show how to extend Loosely Couple Applications Architecture Design, Web Services, Middleware, Interoperability into the Service Oriented Architecture Business Applications Framework, to better support business applications.
- Build the basic components required to operate an Enterprise Interceptor environment.
- Show how the application can be used to map source and target applications.
- Show how Enterprise Interceptor is deployed in a real business environment.

1.3 Dissertation Outline

Chapter 2 gives an overview of most global organization applications architecture landscape, and the integration challenges faced by these organizations. We also explore

the implication of Sarbanes Oxley Act on the current architecture. Chapter 3 examines Service Oriented Architecture strategies to get a basic understanding of the architecture and how it is currently used. We also examine the issues surrounding why we have so much difficulty in integrating applications from various vendors. Chapter 4 defines the Enterprise Interceptor a Framework for Applications Interchangeability (Enterprise Interceptor), showing the components and how it integrates better than the existing integration techniques. In Chapter 5, we take a deeper look at the Interceptor. The Interceptor is a critical component of the framework, and here we explore all of the nuances associated with the Interceptor and how it integrates to the environment. In Chapter 6 we apply Enterprise Interceptor to real business examples, comparing the results first in a legacy environment and second when deployed using the Interceptor. In Chapter 7 we look at performance implications of the Interceptor. A series of tests are executed to compare environments with and without the Interceptor. In Chapter 8, we look at maintenance and version control. Maintenance and version control are most challenging components of an application development life cycle. We will look at how we are proposing maintaining and controlling the versions in the framework. Chapter 9 looks at how the open source community and other open source initiatives pave the way for a Enterprise Interceptor environment. Chapter 10 summarizes the dissertation and provides some general observations.

Chapter 2

Current Business Applications Environment

In this chapter we look at business applications environment in most global companies.

2.1 Current Business Applications and Integration Issues

Application systems are built at different times by different groups operating independently of each other. Organizations are stuck with incompatible architectures and hard-to-maintain, but harder-to-eliminate legacy applications. Organizations are also embracing buy before build strategy that favors purchased applications packages over internal development. The purchased applications vendors operate in their own silo, marching to their own beat.

These issues lead to quality problems when external packages are integrated. Data quality in mapping and re-mapping different data models, inconsistent data model due to the silo mentality, data ownership and stewardship is left up to the business organization when versions change, semantic knowledge and behavior challenges that must be managed by the business organization to ensure integration among various vendors. There is latency amongst most of our integration solutions for high volume transaction applications, as real time integration is often difficult if not impossible, because there is high level of transaction consistency required in most packaged applications.

Most companies' enterprise portfolio is littered with numerous package business applications supporting various functional areas in the enterprise (Figure 1). Efforts have been made by companies to integrate their disparate business applications, but the bulk of the integrations have resulted to nothing more than moving data from one disparate business application to another disparate business application.

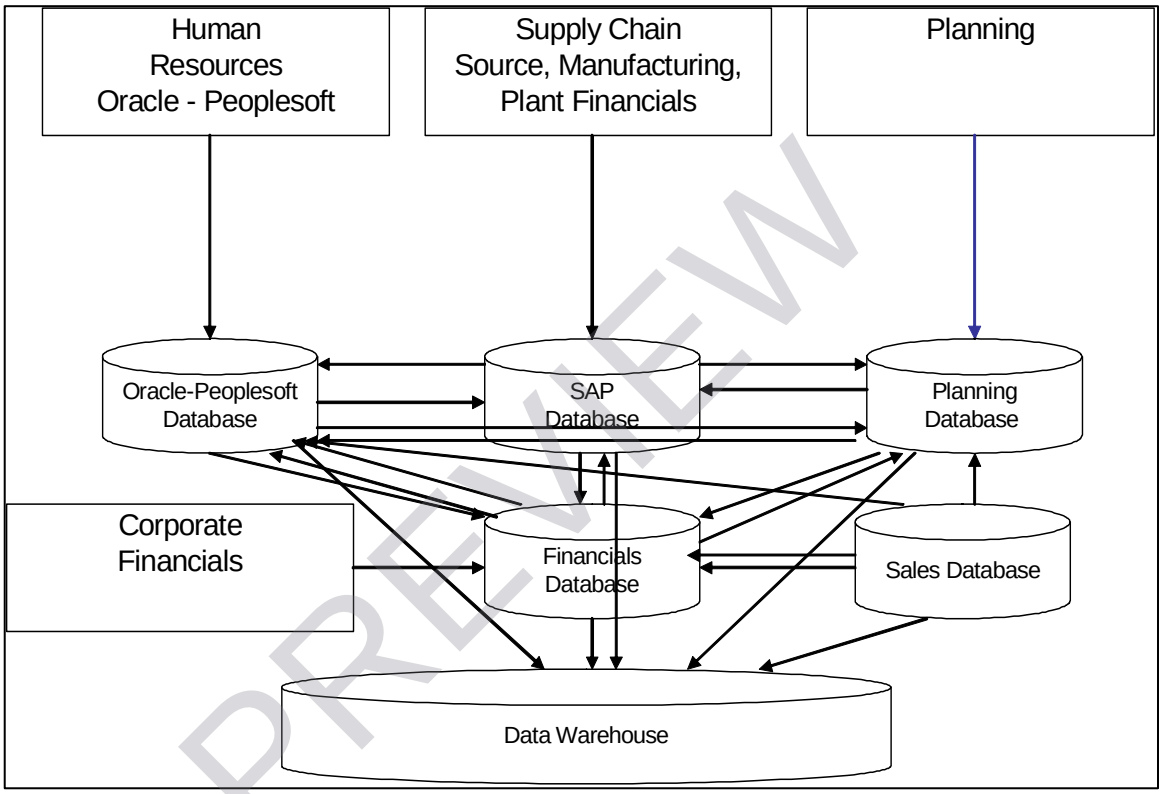

Figure 1 Typical Corporate Application Portfolio

In another related Gartner research material on financial systems, most multinationals companies have a patchwork of financial management systems. This is often the result of regional purchasing of applications to comply with local generally accepted accounting principles (GAAP), combined with merger and acquisition activity that added more financial applications to the portfolio. The more disparate the finance systems and

instances are, the more challenging it will be to ensure compliance with regulations, such as Section 404 of the Sarbanes-Oxley Act of 2002. [17]

2.2 Sarbanes-Oxley Act Requirements and Data Integrations

Sarbanes-Oxley Act (SOX) section 404 requires "an internal control report, which shall:

1. State the responsibility of management for establishing and maintaining an adequate internal control structure and procedures for financial reporting; and
2. Contain an assessment, as of the end of the most recent fiscal year of the issuer, of the effectiveness of the internal control structure and procedures of the issuer for financial reporting

The section focuses specific attention on the controls and procedures for managing the financial reporting process. It is important to understand that section 404 and 409 compliance depends primarily on the quality of existing and future business processes, particularly financial reporting processes. Section 404 mandates the production of a new kind of compliance report concerned with process quality and section 409 mandates communication of material events to the marketplace in a timely manner.

Although business management applications are essential to provide support for Sarbanes-Oxley Act compliance, there are a number of technological issues which can complicate and frustrate compliance efforts, including the use of:

- Multiple, non-integrated CRM/ERP/MRP/SCM/Project Accounting systems within a single organization
- Disconnected spreadsheets to reformat and "massage" information extracted from the transaction processing systems