Redesign Accounting and Budget System Using LINQ Framework and Web Service

Rekik Asefa
Cybersoft Plc., Addis Ababa, Ethiopia
rekikasefa@yahoo.com

Mesfin Kifle
Department of Computer Science, Addis Ababa University, Ethiopia
kiflemestir95@gmail.com

Abstract

When companies need to have a system, they have two options to choose from. The first one is to make an agreement with software companies to study their workflow and to develop a system for them from scratch and the other option is to get a package application of the shelf. There are many reasons companies do not prefer software development from scratch; it takes relatively longer time, it is more expensive and users can not be sure what will be the final result of the project, whether it succeeds or fails.

A package application, unlike a custom or home grown application, reflects the fact that some business processes are universal. Nevertheless, some package solutions had been developed years ago and are not compliant with the current state of the art technological solutions. The newly emerged technologies can improve application qualities in terms of performance and integration.

The approach used to come to the solution of the problems takes the form to explore the data access technology of Accounting and Budget System package applications which were developed in Ethiopia. So in order to materialize the aim, different methodologies were used such as conducting interviews to collect input from various organizations that develop package applications, performing an extensive literature review, make experimentation and developing a prototype that best illustrates the objectives of the work.

In the experimentation, LINQ is compared to ADO. Net and comparison is made to see which is easier and faster. In effect, LINQ is easier because some functionalities coming from the development environment are already available or built-in. In terms of performance, six different comparisons are made.

After the experimentation and prototype application implementation, we concluded that most companies have to convert their package application data access layer from ADO.Net into LINQ to improve the performance of the application and therefore there is a need to develop an application that converts ADO.Net framework into a LINQ framework at the data access layer.

Keywords: Package Application; LINQ; ADO. Net; Data Access Layer

1. Introduction

The dynamic nature of today's business world demands institutions to be more alert to external environment and manage their resources accordingly. Different professionals have come up with a variety of theories on how to manage these resources and achieve one's organizational goals. Even though these theories have great uses, it can be argued that there is no standard solution that will solve every one's needs. Thus, organizations try to fuse the principles they employ to their actual needs.

A software application is simply the interface to that data, the thing that helps to dig up, organize, and present it so that it becomes actionable information [1]. When companies need to have a system, they have two options to choose from; the first one is to make an agreement with software companies to study their workflow and develop a system for them from scratch and the other option is to purchase a package application of the shelf. In order to attract more companies to use package applications, we need to solve problems of package applications like performance and integration with other systems.

A package application, unlike a custom or home grown application, reflects the fact that some business processes are universal [2]. So it makes a lot more sense to create a single application that covers

most of the needs of most businesses than it would for every business to spend the time and money to develop such applications from scratch.

In today's business trend, using package applications in Ethiopia is increasing. When we see, for example, Point of Sales (POS) applications, different organizations or institutions are required by the government to use POS applications, and almost all of these applications are cost package applications.

There are criticisms regarding these package applications. One thing is their performance related issues. In order to solve performance issues, the data access technology we are choosing for the data access layer as well as how data transfers within the modules are important factors.

Over the last twenty years, Data access technology providers have been developing ever-more powerful and flexible data access solutions [1]. Some are very specialized, others are general-purpose, but all of them share two common goals; to help applications get the information they need and to help developers spend less time dealing with difficult details of data storage and more time creating sophisticated software the utilizes that data for real benefits to their customers.

In the historical overview, there are as many different data development technologies as there are different types of data sources. Some technologies have been retired, but most of them are still active and keep being developed. In order to benefit more from packaged applications, the data access technology has a great impact on the performance of the application. So choosing appropriate technology is essential.

After using package applications, institutions start complaining about the performance and other related issues of the software. We can increase the performance of a software by choosing appropriate data access technology and creating a suitable way for data transfer between modules.

2. Background

The importance and necessity of business applications is well recognized by the countrywide software industry and by concerned government agencies. This enlightens us to allot a considerable effort to better the development procedures of business applications. There are many challenges that developers face in the process of developing business applications. As software developers, we have been experiencing many of these challenges. The following list shows the summary of the problems to be tackled and reasons for the initiation of this work:

- Developers and users face difficulties on the performance of package applications.
- Developers face difficulties when they want to integrate package application modules with other applications.
- Usually too much effort is spent to promote code reuse, especially in the area of data access layer.
- The need to facilitate team projects that would constitute modules which are easily broken down to member developers.

The ultimate aim of this work is showing a mechanism to tackle the above mentioned problems that are encountered during package application development.

Hence, the objective of the work is to redesign Accounting and Budget system at data access layer using LINQ framework and for integration and data exchange purpose using Web Service. In the process, it was attempted to gather requirements that will serve as input to the design and development of an appropriate system, introduce appropriate technology to redesign the package application software in the data access layer to improve the performance, clearly show the advantage of web service in integrating modules in the application and with other web applications, evaluate to what extent LINQ framework enhances an application's performance and decrease developer's effort, come up with an architectural solution that will show the package application integration in the development of web applications by using web service, and develop an 'Accounting and Budget Web Application System' using Web service and LINQ to SQL framework.

3. Experimentation/Prototype

One of the objectives of this research is to clearly show the benefit of introducing LINQ framework at

the data access layer. So before suggesting that this framework enhances an application's performance, a comparison needs to be performed with other data access technology frameworks. For this purpose, the experimentation was done by comparing LINQ framework with ADO.Net framework. The reason why we choose ADO.Net framework is because during the study of package applications which are developed here in Ethiopia, most of them use ADO.Net framework at the data access layer.

CRUD (Create, Read, Update, and Delete) operations are basic operations for most business applications. So by investigating the behavior of data access framework for each operation, we can say that this framework increases or decreases the performance of the application in this operation. The following scenarios are selected to be experimented.

- Performance of reading data from an SQL Server 2008 using ADO.Net and LINQ using stored procedure.
- Performance of reading data from an SQL Server using ADO.Net and LINQ by directly executing Select SQL Statement.
- Performance of inserting data into an SQL Server 2008 using ADO.Net and LINQ by directly executing inserts SQL Statement.
- Performance of updating data from an SQL Server 2008 using ADO.Net and LINQ by directly executing updates SQL Statement.
- Performance of deleting data from an SQL Server 2008 using ADO.Net and LINQ by directly executing deletes SQL Statement:
- Performance to establish connection to an SQL Server 2008 using ADO.Net and LINQ.

Figure 1 shows one of the test results when the application inserts data using direct select SQL statement from the database. The x-axis shows the database visit from the application by using the specified data access technology and the y-axis shows the total elapsed time measured by current instances in timer ticks/100.

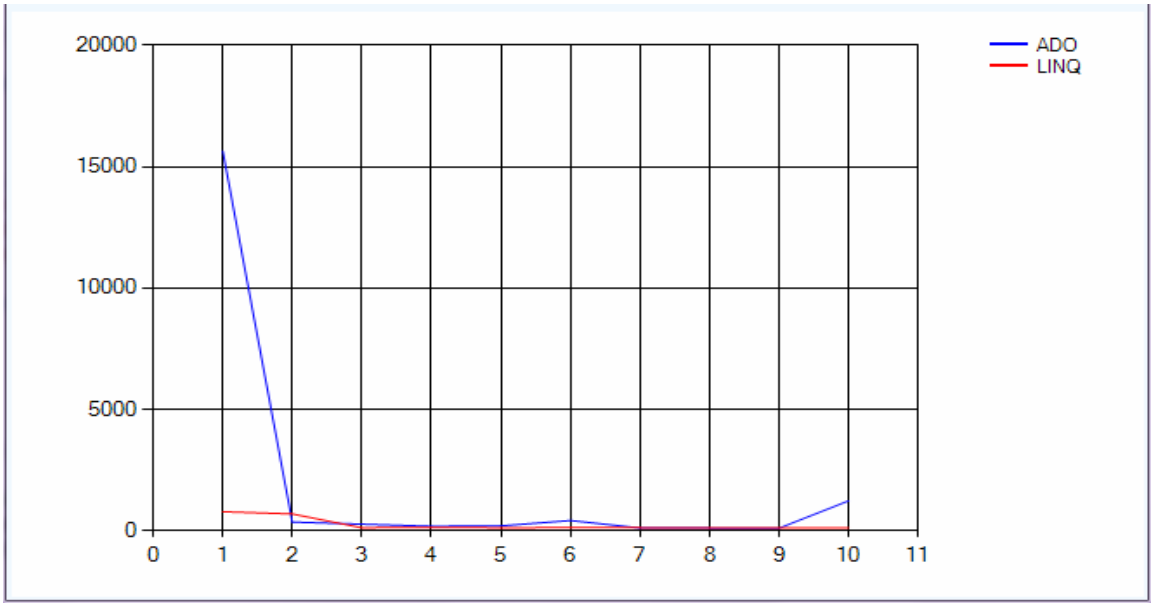


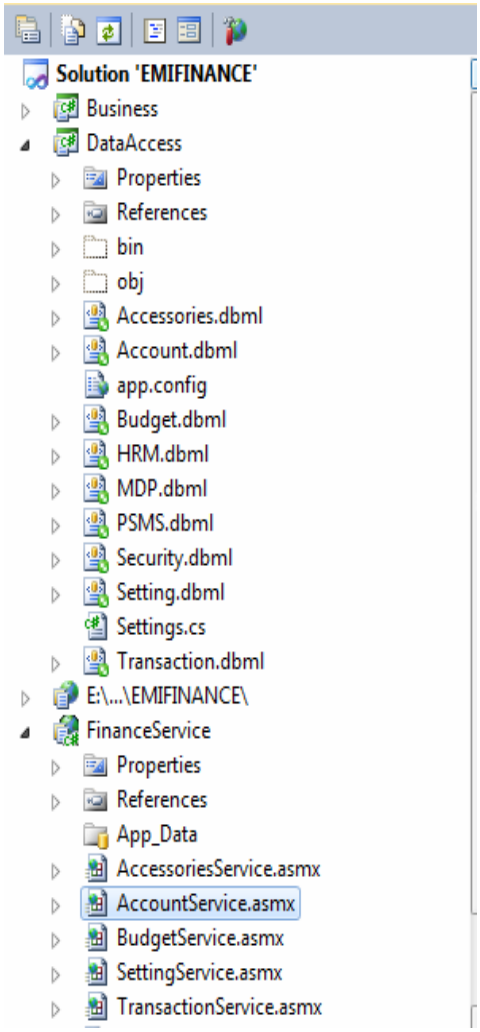
Figure 1: Test result when the application inserts data using direct select SQL statement

The implementation of the Accounting and Budget prototype for this work was taken care of using the following technologies.

- Microsoft Visual Studio 2010 (Programming Language C#)
- LINQ to SQL Framework
- Web Service
- SQL 2008 database

The structure of the Accounting and Budget system prototype application is composed of four main modules. These are the Business module, Data Access module, Presentation module, and the Accounting and Budget Service module which is the web service part. Let us discuss how our modules fit into the architecture we used. We followed the three tier architecture. The business module contains the work flow and the business rules, the Accounting and

Budget service will make the business class methods to be available as a service and these two modules are the Business layer part. The Data Access module part is composed of different LINQ dbml files. They contain different kinds of database tables, and they create a mapping between visual studio and the database. So it is Data Access layer, the final module which is the presentation module that contains different pages of the application, which is the Presentation layer.



4. Related Work

Over the course of research for this work, it has been noticed that not much resource is available that concentrates on the performance and integration aspects of package applications. It was very difficult to find literature that emphasizes upon data access frameworks, package application architectures, and current application design best practices. The main sources of literature for this research were interview, books, and forums. But this doesn't mean that there was absolutely nothing that was available and relating to the focus of this work.

Package application software is licensed by a third party software provider to fulfill a specific business purpose. It is generally designed to support commonly performed business functions to multiple types of user organizations [3]. Examples of package applications include, Accounting System, Human Resource, and Enterprise Resource Planning.

Package applications must be integrated into in-house developed applications [2]. In-house programming teams have had to modify application packages and this had resulted in difficulties in reintegrating vendor upgrade back into the package software.

As presented in [4], integration is the process of bringing data or a function from one application program to that of another application program. Web Service will play a major role in Function/Method oriented integration, which is one of the most commonly used pattern for enterprise and business to business application integration [5]. Because web services are accessible using URL, HTTP XML application running on any platform and in any language can access XML web service. Web Service is a software system designed to support interoperable machine to machine interactions over a network [6]. In a simple way, web service is defined as a piece of business logic located somewhere on the Internet that is accessible through standard Internet protocols.

Language Integrated Query (LINQ) is a Microsoft.net framework component that adds native data querying capabilities to .NET languages [7]. LINQ to SQL is an Object Relational Mapping (ORM) framework that allows the direct one to one mapping of a Microsoft SQL Server database to .NET classes, and query of the resulting objects using LINQ [8]. CRUD (Create, Read, Update, and Delete) supports in LINQ to SQL which is generally the most compelling features of an ORM tool.

5. Conclusion

This paper constitutes a result that attempts to show what challenges exist in the effort of developing applications, especially in the area of data

access layer. The main purpose of the paper was to uncover what the challenges were and providing sound methods (technology) that solve them. As a result, the identified challenges have been listed down and were the constituents of the problem statement of the work. The methodology employed, to go about collecting data and come to conclusion, included interviews, literature review, experimentation, and prototyping. The experimentation done for this work clearly shows the advantage of LINQ over ADO.Net. The application selected for the case study was an Accounting and Budget system. It was developed as a prototype in order to show how we can introduce LINQ framework and web service for the selected architecture of the work and to show how we can integrate with other applications using web service but we didn't introduce web service into our architecture. If another application wants to interact with the Accounting and Budget system, it needs to access the database of the Accounting and Budget System but that is not the recommended approach for many reasons.

The discussion on the findings suggests that through the process of developing the experimentation and the case study application, most of the challenges identified in the problem statement have been successfully solved.

As a future work, problems related to web service in the area of data exchange and performance can be compared with other latest service technologies. Even if the horizon of this work could not come up with a feasible approach to show how to deal with these challenges, package applications which are found in Ethiopia need to redesign their data access layer to use LINQ framework. Therefore there is a need to develop an application that converts ADO.Net framework into LINQ framework in the data access layer.

References

[1] http://msdn.microsoft.com/en-us/library/ ee730343.aspx, Microsoft Data Development Technologies: Past, Present, and Future, Last-accessed- Nov 22, 2012.

[2] http://www.ameinfo.com/103688.html, why should I choose packaged business applications? Last-accessed- Nov 22, 2012.

[3] http://www.cutter.com/content-and-analysis/ journals-and-reports, November 24, 2012.

[4] http://en.wikipedia.org/wiki/Data_access Data access, Last-accessed Nov 12, 2012.

[5] http://www.techrepublic.com/article/using-web-services-for-application-integration/1045211, Using Web Service for application integration, last-accessed- June 12, 2012.

[6] http://www.w3schools.com/webservices/ default.asp, Online Library "Web Services", last accessed June 12, 2012.

[7] http://en.wikipedia.org/wiki/Language_ Integrated_Query, Language Integrated Query, last-accessed- Sep 12, 2012.

[8] http://msdn.microsoft.com/enus/netframe work/aa904594.aspx. Microsoft LINQ NET Framework Developer Center, last accessed, October 12, 2012.