Predominant characteristics of linq technology usage in education platforms

Eftâl Sehirli a, Birsen G. Ozdemir a

*Department of Computer Engineering, Doğuş University, Acsbadem - Kadıköy, İstanbul and 34722, Turkey

Abstract

LINQ is the abbreviation of “Language Integrated Query”. LINQ technology mainly allows reaching data easily from application side and making queries over data as if they are made in database systems. In this study, we aimed to share our experience of usage of LINQ technology in an education platform named as GPM. This paper mainly focuses on specification of main advantages of using LINQ in an education application. It also brings differences between LINQ and Ado.Net to the fore and discusses why any future technology will be successful if it includes the main advantages of LINQ technology.

© 2012 Published by Elsevier Ltd. Selection and/or peer review under responsibility of Prof. Dr. Hüseyin Uzunboylu
Open access under CC BY-NC-ND license.

Keywords: LINQ, object oriented programming (OOP), lambda expressions, Ado.Net;

1. Introduction

New technologies are launched and existing ones are improved day by day. Some new technologies replace old ones. LINQ has this opportunity when it makes a name in education platform. Since LINQ is not generally preferred in schools, students do not use LINQ and improve themselves regarding it. However, LINQ has some predominant characteristics such that it can be used more frequently. In addition, LINQ has some differences from Ado.Net. Hence, this study mainly mentions main advantages of LINQ, differences between Ado.Net and LINQ. Finally, it concluded as LINQ will be successful to realize expectations of users regarding applications.

2. General Properties of LINQ

“Language Integrated Query” is abbreviated as LINQ. This technology came out when .Net Framework 3.5 emerged. One of the components of LINQ is LINQ to SQL that allows easily reaching data in SQL Server databases from applications. For other database systems, there are studies to work with LINQ. In application side, programmers can make dynamic queries over data as if they are made in database systems.

* Eftâl Sehirli Tel.: +90-505-503-3550
E-mail address: 2k731026@dogus.edu.tr
E-mail address: eftalsehirli@gmail.com
3. Main Advantages of LINQ to SQL

One of the main advantages of LINQ technology is using object oriented approach. Some advantages of object oriented programming (OOP) that are also beneficial to LINQ are reusability, simplicity, modifiability and maintainability. Reusability provides that objects are able to be used again in other programs and platforms. Simplicity decreases the complexity of understanding software objects and so increases the readability of codes. Modifiability maintains that data type or attributes can be altered. Therefore, classes can be reused by joining changes or new features. Maintainability provides that objects can be protected by encapsulating, and safety level increases dramatically. In LINQ technology, programmers can rejoice in these advantages of OOP. That is to say that, every table existing in a database is automatically transformed to a class in application side by LINQ. Each column of database table is defined as an attribute of the created class (Figure 1). If data in database tables are desired to reach, an instance of related table class must be created and programmers can reach data by using that instance/object (Figure 2).

DataContext type is automatically added to database names whose example is shown in Figure 3 and ensures to translate LINQ queries operators to SQL queries. DataContext uses IDbConnection of Ado.Net to interact with databases and is automatically initialized by LINQ while dbml class settings like determining username and password of SQL Server are done (Box, D. & Hejlsberg, A., 2008).

Another advantage of LINQ is to possess lambda expressions. Lambda expressions use an operator shown as “=>”. Left side of the operator has inputs that may be more than one. Right side of the operator has only one output. Lambda expressions or statements can take any data type and as a result of expression, they can return any data type. In this way, any variable definition is not needed and covered in memory. Moreover, since lambda expressions
contain shorter code statements than standard LINQ and SQL queries, they are compiled in a fast manner. An example of code statement taken part below shows how to obtain student’s all information whose id number is 1.

```csharp
// LINQ code lines (standard)
GraduationProjectManagementDataContext db = new GraduationProjectManagementDataContext();
var rs = from c in db.Students
         where c.Id == 1
         select c;

// LINQ code lines with Lambda Expression
GraduationProjectManagementDataContext db = new GraduationProjectManagementDataContext();
Student student1 = db.Students.Single(p => p.Id == 1);
```

LINQ enabled languages such as .Net languages, Java, PHP and JavaScript provides type-safety property for LINQ technology. Programmers write queries by seeing what they can write in a faster manner (Figure 3). Furthermore, development tools can provide full intellisense and rich refactoring property when writing LINQ code. Full intellisense means code completion. That is, written code is completed by pressing space, tab or enter buttons. In addition to this, ctrl + space are pressed in order to both see list of intellisense and complete half-written code. Refactoring allows rewriting by editing code lines, improving them and using shorter form.

```csharp
void GetStudentProjectList() // to get the list of student’s projects
{
    try
    {
        GraduationProjectManagementDataContext db = new GraduationProjectManagementDataContext();
        object rs = from p in db.Projects
                     from s in db.Students
                     from sp in db.StudentProjects
                     from c in db.Courses
                     from a in db.Advisors
                     from u in db.Users
                     where s.UserId == int.Parse(Session["Id"]).ToString() && a.Id == p.AdvisorId &&
                     sp.StudentId == s.Id && sp.ProjectId == p.Id && sp.Deleted == false &&
                     p.Deleted == false && sp.CourseId == c.Id && u.Id==s.UserID && p.Approval==true
                     orderby c.Id
                     select new { p.Id, p.GroupNumber, s.Number,
                     StudentName=String.Concat(s.FirstName, "", s.Surname), c.CourseCode, Advisor=a, }
    }
    catch (Exception)
    {
    }
}
```

![Figure 3. Type-safety property](image)

4. GPM Application

GPM is the abbreviation of “Graduation Project Management”. GPM is designed as a web application in educational domain. It is developed to effectively manage the projects submitted to different departments of our university. A view of GPM for Settings popup menu is shown in Figure 4. In GPM, we have used LINQ technology
to interact with the database system SQL Server from the application side. In recent years, students of Computer Engineering Department have been developed different GPM applications for our university. When we compared database connection parts with previous GPM applications that were done last years, we observed that we finished these parts almost 2 weeks earlier thanks to LINQ. We detected from reports of elder GPM applications that people were spent more than 3 weeks for carrying out same tasks. Moreover, our database code lines are less than others, which is the most important reason why we have finished GPM earlier.

5. Differences Between LINQ to SQL and Ado.Net

Before LINQ, Ado.Net has generally been used to interact with database systems. SQL injection that means reaching a database by generally aiming at damaging in it from an application is a big problem for Ado.Net. Programmers must know how to prevent applications from SQL injection if they use Ado.Net for database connection. On the other hand, when LINQ is used, there will be no problem about SQL injection because LINQ uses classes and data is not assigned directly (Dave, A. & Hate, S., 2010).

Programmers need to know and use SqlConnection, SqlCommand, DataTable, DataAdapter etc. terms for essential connections between an application and a database in order to use Ado.Net. Each connection is always needed to be closed after opening. As there are lots of code lines while using Ado.Net, code readability is getting more difficult and difficult. On the other hand, LINQ ensures shorter code length and easier code writing. Connection code lines are not necessary to be closed when they are opened since it is controlled by LINQ classes. According to performance, there is no big difference between them. An example of code length difference is shown as below:

Ado.Net code:
```csharp
using(SqlConnection connection = new SqlConnection())
{
    connection.Open();
    using (SqlCommand command = new SqlCommand("Student", connection))
    {
        command.CommandType = CommandType.StoredProcedure;
        command.Parameters.AddWithValue("@Number", "200731026");
    }
```

Figure 4. GPM Settings popup menu
These two codes are doing same thing. But, LINQ is more readable than Ado.Net to understand the code and it has less code lines. Hence, refactoring is easier by using LINQ technology than Ado.Net.

6. Conclusion

LINQ is a useful technology that provides some advantages to interact with database systems from application sides. Instructors whose goal is to teach OOP approach can use LINQ technology. Moreover, they can teach database systems over same technology LINQ. Thus, students can learn related fundamental concepts by laboring less energy. For example, IEEE members teach LINQ programming bringing to the fore an object first approach for students. Next, they keep on teaching key components of LINQ framework. Finally, they teach how to use LINQ for database programming (Kan, W. & Yujun, Z., 2009). This paper discussed that LINQ has more advantages than Ado.Net such as readability, short code length, easy code-writing and easy-refactoring. Hence, LINQ offers very suitable technology for educational applications. OOP and database systems are separately taught in universities and LINQ may be a suitable technology for teachers to teach both of them. On the other hand, while technology develops, people hope that their web applications or desktop applications had better work in a fast way. If existing applications are refactoring or any future software programs are written with technologies that provides speed, enhancement, convenience and shorter code statements, programmers can realize this hope. When applications realize expectations of users, they can be successful. LINQ has a capability that helps programmers to reach these kinds of expectations.

Acknowledgement

We desire to thank CY-ICER to give us a chance to introduce LINQ technology and present it.

References

