Saimaa University of Applied Sciences
Faculty of Technology Lappeenranta
Double Degree
Information Technology

Martin Stoklas

Feature component attributes analysis and Data synchronization

Thesis 2014

Abstract

Martin Stoklas
Feature component attributes analysis and Data synchronization, 23 pages
Saimaa University of Applied Sciences
Faculty of Technology Lappeenranta
Doubly Degree
Information Technology
Thesis 2014
Instructor: Yrjö Utti, Saimaa University of Applied Sciences

The topic of this thesis is parsing and subsequently analyzing feature component attributes. The second topic is data synchronization between the company's two applications. The thesis was done for a worldwide telecommunications equipment and data networking company.

Feature component attributes analysis is used for the unification of company's workspaces, which should use the same attributes for storing information.

Automation of data synchronization is needed for saving time, which can be then used more profitably. Some tasks can be done automatically, but still some manual checking was necessary.

Keywords: Feature component attributes, XML, data synchronization, SQL

List of terms

A3 - Structured problem solving approach

API - Application programming interface

CSV - Comma-separated values, file in plain-text form.

DTD - Document Type Definition, language for defining document structure.

HTML - HyperText Markup Language

MS - Microsoft

ODBC - Open Database Connectivity, software API for accessing database systems.

PDF - Portable Document Format

SQL - Structured Query Language, programming language used for managing database systems.

UML - Unified Modeling Language, graphical language for visualization, designing, specification and documentation of a system.

VBA - Visual Basic for Applications, programming language developed by Microsoft.

W3C - World Wide Web Consortium

WYSIWYG - What You See Is What You Get

XML - eXtensible Markup Language, markup language with simple and flexible text format.

Table of contents

1 Introduction

For more efficient work and better business planning and operations, it is needed to have consistent data structure, which provides all necessary information in a simple and effective form. In a big company it is even more requested to have all processes monitored and it is much easier to work with information through all different workspaces, when they are using and storing all important data in a same way.

Many times some company is merging with a bigger one, in this case usually both companies are using different strategies and their structures vary. It is essential to unite all data and processes.

Another main property for a properly functioning company, like in this case company, is data synchronization between systems. It is crucial to have these activities automated as much as possible. Automation helps to reduce consumption of time and allows focusing on more important tasks.

2 Used technologies

This chapter describes technologies, which were used for accomplishing the thesis tasks.

2.1 eXtensible Markup Language (XML)

XML is a markup language with a simple and very flexible text format, developed and defined by the consortium World Wide Web Consortium (W3C).

Document format is both human-readable and machine-readable. XML language is nowadays highly used for data exchange between applications, publishing documents, saving settings and many other purposes, thanks to universal XML format (1).

For validating the right structure of XML document e.g. Document Type Definition (DTD) or XML Schema can be used, which then guarantees document valid structure. This includes one of the main key attributes for data exchange (2).

Working with XML document is simple and many applications and tools are well prepared for automatic converting and processing XML documents.

2.2 Structured Query Language (SQL)

Programming language SQL is used for managing database systems and manipulating data in a specific database.

The most common operations in SQL are queries, which can perform retrieving, inserting, updating or deleting data. The query can of course manage database tables and e.g. changing user's rights for accessing database data (3).

There are different versions of SQL language for different database systems and every of them can offer varying functionality and have some changes in query structure.

2.3 Visual Basic for Applications (VBA)

Visual Basic for Applications is an event-driven programming language developed by Microsoft (MS). The language was derived from another programming language named BASIC.

Visual Basic enables access to databases and manipulating the data, rapid application or graphical user interface applications development, creation of ActiveX controls and objects (4).

The language was developed to be easy to learn and use, that can be seen in option to use the components provided by the Visual Basic itself, which can simplify and accelerate application programming. A programmer can also use the Windows Application programming interface (API), which can also make programming much easier (4, 5).

VBA is present in most Microsoft Office applications, but is also implemented in other applications.

2.4 Microsoft Office

Microsoft Office is the Microsoft's office suite, primarily intended for processing text documents, excels and graphs, presentations and database oriented applications.

2.4.1 Microsoft Word

Microsoft Word is a highly used and famous text document processor. Nowadays the MS Word is using standardized XML format for saving the documents with extension .DOCX. The application works on WYSIWYG ("What You See Is What You Get") principle, that means what the user sees, is what he gets (6).

MS Word offers easy Portable Document Format (PDF) creation, can easily share data with other MS Office applications or create simple text or XML format outputs.

In MS Word the user can not only work with basic text, but can also create simple tables, graphs, equations or pictures and use some of the predefined effects.

2.4.2 Microsoft Excel

MS Excel is a spreadsheet processor offering many statistical functions, automatic graphs creation, aggregation functionality and many other options (7).

2.4.3 Microsoft Access

Microsoft Access is a database management system which combines graphical user interface with relational Microsoft Jet Database Engine (8).

Data can be accessed from any database through Open Database Connectivity (ODBC) interface or from its own data storage (8).

MS Access can be used to develop an application software, for that the Access is supported by programming language Visual Basic for Applications. Usually it is used for creating forms and reports, but can be used for creating complicated applications with wide functionality.

2.5 Visual Studio 2013

Visual Studio is a development environment used to develop console or Windows Forms applications, web pages, services and applications, Microsoft Silverlight applications, mobile applications and much more also due to third-party add-ons.

Applications and services can be developed on various platforms such as Microsoft Windows, Windows Mobile, Microsoft .NET Framework and Microsoft Silverlight (9).

Visual Studio includes a code editor with a real-time help tool IntelliSense, code refactoring, debugger and many designers for creating graphical interfaces, webs, classes and databases. Tools and functionality can be expended by plugins (9).

Also features for recommending improvements in code for better performance, testing and analyzing smooth run of application and automatically generated code from Unified Modeling Language (UML) design are included in Visual Studio.

The programmer can choose from many programming languages, which can be used for creating specific software, because the code is then compiled to a platform-neutral language called Common Intermediate Language.

2.6 SAP Lumira

SAP Lumira is an easy-to-use software for data visualization. Data can be combined from multiple sources, filtered and modified with some simple functions (10).

Software is automatically doing some common operations like summations, creating graphs and offers an easy way to represent conclusions from the found information.

3 Analyzing feature component attributes

The following sections are focused on elaboration of the first task, which was analyzing feature component attributes.

3.1 Introduction to situation

In the company there are several workspaces, where every workspace is collecting and manipulating data (Figure 1). This data is needed for future reporting, statistics, manipulating, etc. The problem is that there is no strict structure, in which data must be collected and what format must be used. So every workspace can do it in a different way. Because of the data structure incompatibility and the need for processing this data altogether appears a request for creating a template, which declares all necessary attributes and the way how to store them.

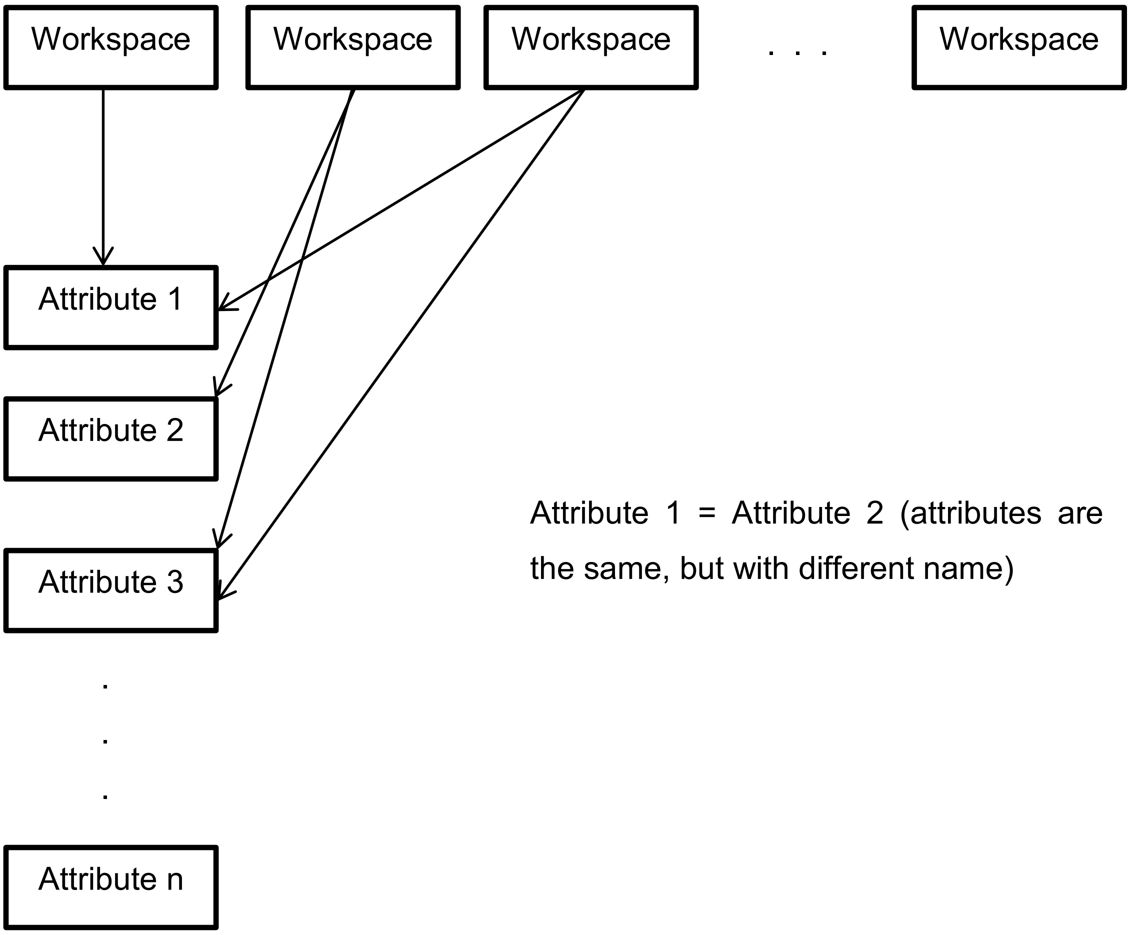

Figure 1. Current situation draft

3.2 A3 report

After understanding the problem in the current situation the second step in this task appeared, creating the plan how to analyze all used feature component attributes through all workspaces and describe the problem with A3 approach, which is commonly used in the company.

The example of the result is described here.

Theme

Analyzing feature component attributes from all designated Focal Point work-spaces for harmonized reporting view needs.

Background

- Get important information by analyzing stored data
- Important for understanding the processes and detecting problems
- Discovered by need of supervising the work of many project groups
- Impact for company is in better understanding the situation in project groups and the potential problems which may occur

Current Condition

- Project group has its own data structure, which takes time to understand
- No easy way to find, if the project is going as it was planned and get statistical information from all project groups of each project phase
- The end result will be measured by getting credible information, which can help to improve the company’s business

Goal Statement

- Get important information from data analysis
- Compare project groups and find the connection between them
- Have information from which the state of the project is visible and future decisions about the project can be done

- Get statistical information, how long each phase of the project takes through the different project groups
- Needed feature screening process information can be collected via harmonized and commonly used Focal Point attributes

3.3 Input data

Based on the company’s system a Word document containing all workspaces with assigned feature component attributes and specification about these attributes was created. The goal was to take all this information and convert it to a better format, because in a Word format processing this information was almost an impossible task. The word document has about 3 500 pages, so automated conversion to another format was necessary.

The question was, which new format would be the best option for the easiest future processing and analyzing. Because of the previous good experiences with analyzing data with MS Excel and easy to use application SAP Lumira, Excel file was the best option.

3.4 Data conversion

A problem occurred with converting a Word document to an Excel file. It was not possible to just select all data in a Word document and easily cope-paste it to an Excel file, how it is normally done. In this case, there was so much information, that MS Word was unable to handle all the information at once. Copying the data with smaller blocks helped only for the first workspace, in trying copying the rest workspaces separately occurred to a MS Word continuously loading the data to memory and after a while deleting them and loading them again, like in a never ending circle. MS Word 2013 had even more problems with manipulating big amount of information, than older version MS Word 2010.

Another option, how to export all data to Excel file was at first exporting Word document to XML format, parsing XML file and as output saving data to Comma-separated values (CSV) file, which can then be easily loaded with MS Excel. The problem with this approach was losing important information containing the comparison between workspaces contained in the Word document. Every workspace was distinguishable by a color from the attributes and their information. After exporting Word document to XML format, the converting became more complicated than converting Word document itself.

The best and easiest way to convert all data was by creating a HyperText Markup Language (HTML) file, which would contain the same information as Word document. Parsing HTML file is really easy and would be the fastest solution, unfortunately we have not reached this possibility.

At last, after many hours spent with copying Word document data separately to a new Excel file, finally all data was there, but still in a not easily processed form. Fortunately export from MS Excel file to XML file was more convenient than in MS Word case, because the distinction of individual workspaces did not become a big issue.

3.5 XML parsing

For parsing XML file and converting data to a requested format it was necessary to create a parser.

There were at least two approaches. The first one was going through every row and look at the row structure. If the row contains only one column, it is the name of the workspace or the name of the attribute. Row with two or more columns contains information about the attribute. Resolution of the name of the workspace and the name of the attribute was possible by founding, if there are two rows consecutively, in that case the first founded row was the name of the workspace and the second one the name of the attribute. This way it was possible to continue until there were again founded two rows consecutively, that was a sign for the next workspace.

The second approach, which was implemented, was by finding out if the current processed row contains a specific id attribute. One id was used for rows containing the name of workspace and other ids for the name of attributes. All rows without any id attribute contained information about the attribute. Both approaches were almost as complicated, anyway the second approach was more logical.

In parsing data, one problem occurred. Every row should represent one specific detail, but there were cases, in which information for feature component attribute contained a table with options. So it was necessary to find a solution, how to distinguish these cases and then process it in a correct way.

After loading all data, it was needed to save this data to an easily workable and transparent form. At first, a CSV file containing all information from Word document was created. However, after analyzing this file it was obvious, that all information is not needed for the first analysis, so export data was a little bit changed and only the most important information was exported to a CSV file.

3.6 Analysis

For analyzing MS Excel and SAP Lumira were used. The task was to take as much information as possible. Because of that there were a lot of tries and different approaches to work with the information. As a result of the analysis graphs (Figure 2), tables and reports were created.

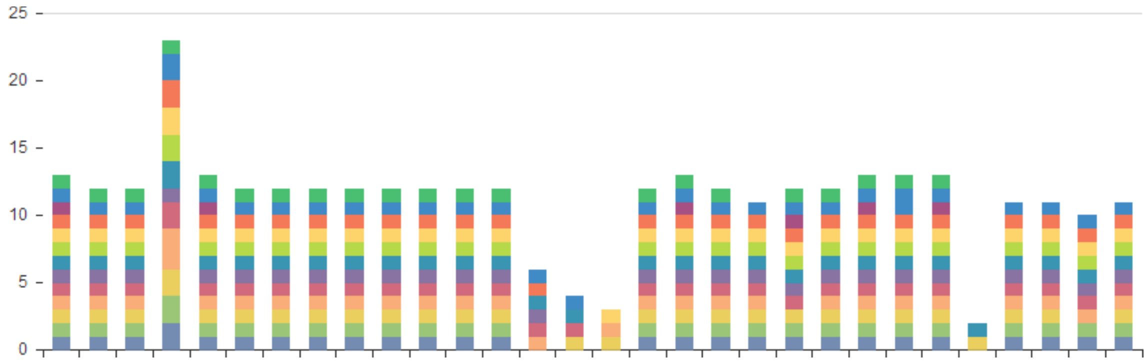

Figure 2. Analysis example

One important thing containing the analysis was merging the attributes with the same meaning, but different names. So at first it was important to find out, which attributes exist in all workspaces (illustrated by different colors in Figure 2) and then compare them by their meanings. If the meanings were the same, even if the formats of the attributes were different, it was necessary to merge them together to determine in which workspace the attribute is missing.

3.7 Analysis result

After the analysis (Figure 3) was done a new task came. One workspace was specific, different than others and for this reason it was necessary to look on this workspace in another manner and do the analysis separately.

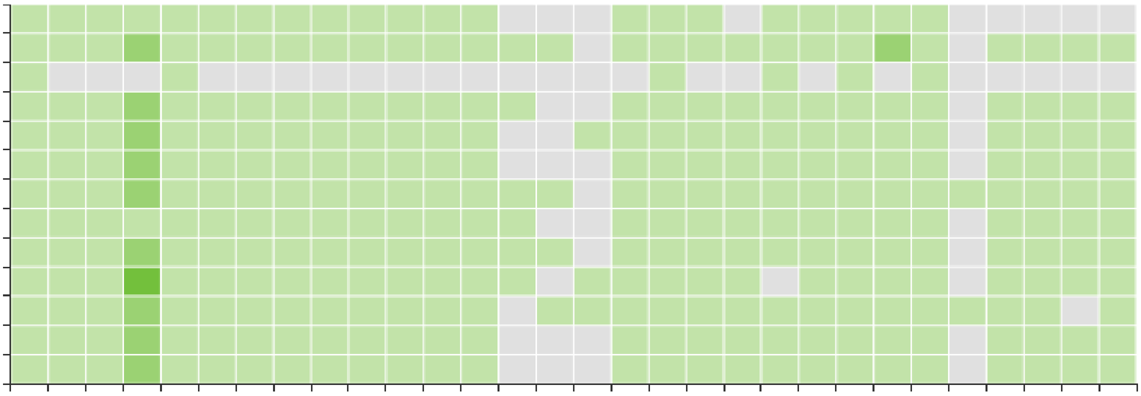

Figure 3. Analysis example 2

From the analysis it was obvious, that not all workspaces are using the same feature component attributes (illustrated by gray squares in Figure 3) and it was necessary to make specific changes to bring a common structure.

3.8 The second analysis of specific attributes

The first analysis was done on all available attributes in the system, even with the ones which were not used and visible. Because of that second analysis was done for only visible and editable attributes.

It was necessary to create a new data parser, because the data was provided in a different form. This time the data were provided in an Excel file. Every workspace had its own list, where the visible and editable attributes were contained. But this time some of the workspaces contained different views, which were needed to process separately.

3.8.1 Visible attributes

The analysis (Figure 4, Figure 5) of visible attributes provided a better view on the current situation, because in the analysis only visible and used attributes from each workspace were processed.

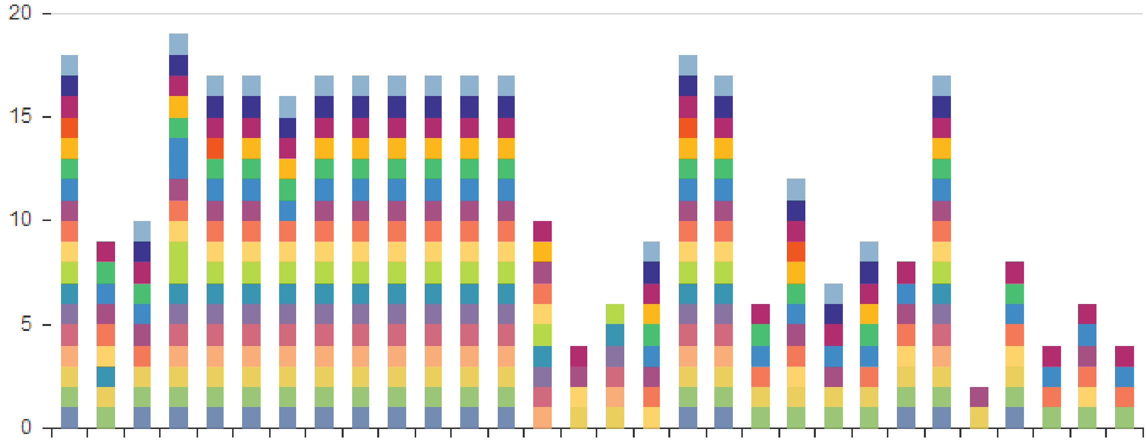

Figure 4. Visible attributes analysis example

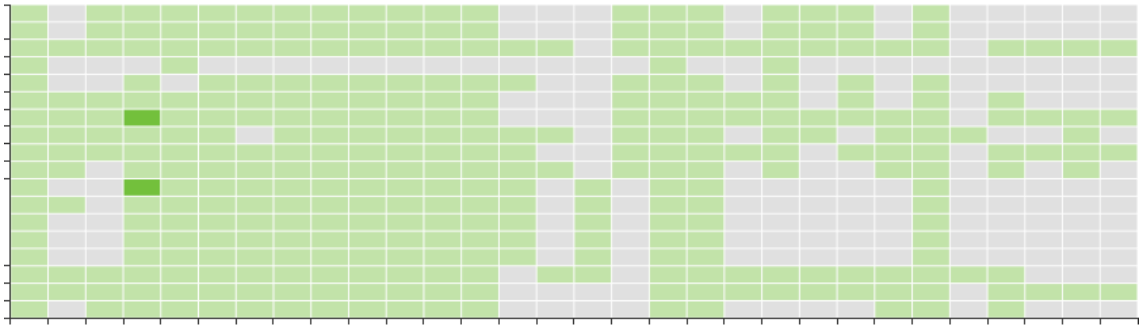

Figure 5. Visible attributes analysis example 2

The result showed big differences between the workspaces.

3.8.2 Editable attributes

Like in the request to analyze visible attributes, there was a request to analyze in the same way editable attributes. After the data was parsed, the analysis could be done and an example of the result can be seen in the following Figure 6 and Figure 7.

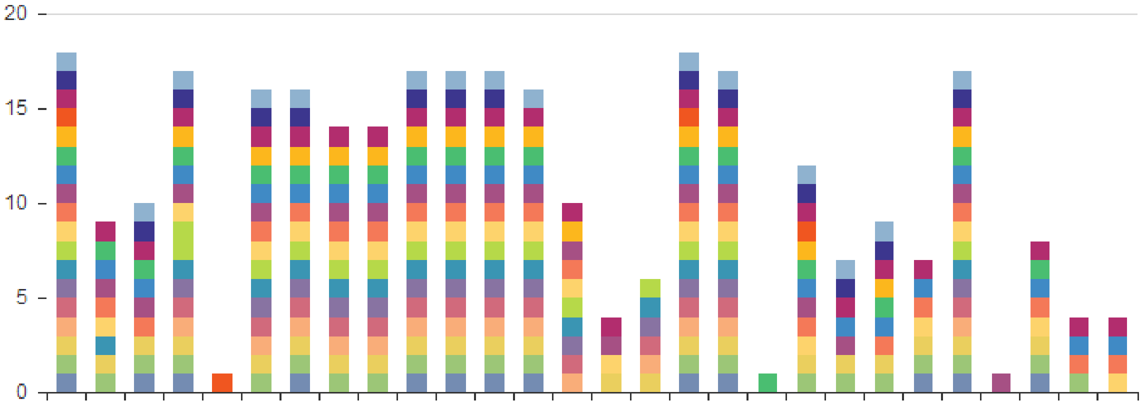

Figure 6. Editable attributes analysis example

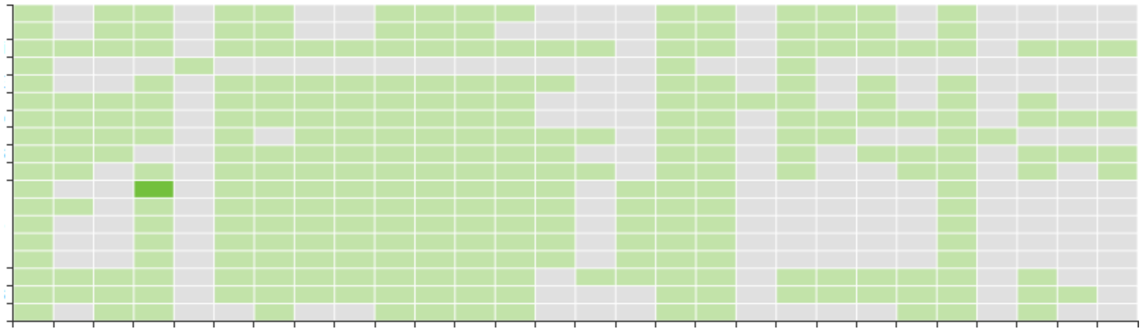

Figure 7. Editable attributes analysis example 2

The results illustrate the missing attributes through workspaces. Based on all information it was possible to manage the necessary steps to unify the data structure.

4 Data synchronization

This chapter describes the second task, which was data synchronization between the systems.

4.1 Introduction to the problem

In the company's departments there are working teams and every team has its own team leader. The problem is that the team leader can be somebody from a different department and can have his own team, which belongs to his department. In this case the person is called "acting person". Acting person is usually only a temporary role and is replaced in future by somebody, who belongs to that team.

The system which contains all information about employees does not handle these situations and for this reason an application in MS Access was created, which is able to show the actual structure of the company with acting relations.

The application is independent and works outside the normal system. Because of that it was necessary to create synchronization between this application and the system containing all information.

Acting relations in the application were handled manually by managers and all changes were then reported and changed in the local system. A problem occurred after making some structure changes in the local system and every time when it was needed to refresh data. When some employees left the company, new persons entered the company or even when somebody changed team or department, it was necessary to make these changes visible.

In the case of changes, it is needed again manually to go through the data in the application and update records correspondingly. This is a time-consuming operation and there is a need to solve it automatically as much as possible.

4.2 Deleting organization and country codes

The first task which was needed to be done was deleting the organization code from the organization name. Because of the changes in the local system, in newly exported data, all records in the organization name attribute contained at the end the code, in which organization and country the employee is working.

But the previous application’s data was without these codes, so for data correctness it was needed to create a script, which will delete these ending codes.

It was necessary to be careful and make the script, which will handle situations, when there is no existing code at the end of the organization name. Other situations were that the organization name can contain only one country or organization code or both of them.

As a result the VBA function was made, which worked as planned.

4.3 Finding changes in acting roles

As a second task it was needed to select records, in which changes were made. This was an important step to reduce time spent with looking on irrelevant records.

SQL queries were created to select only the desired records from the database and possibly mark suspicious records which were needed to check manually.

4.4 Creating possible scenarios

After the desired records were successfully selected, the analysis part started. It was required to create all possible scenarios, which cases may occur, e.g. acting person can be replaced by another person, who is going to be part of the team.

Thinking of all these scenarios it was necessary to create the following steps to automate some obvious changes in the database. This can save a lot of time instead of manual checking record by record.

4.5 Automation of changes

From the previously created possible scenarios it was decided which cases can be automated and which must be still processed manually. For automation scripts were created which perform specific actions depending on the programmed patterns.

5 Summary and discussion

My tasks were to help with feature component attributes analysis. It was requested to get information about the system, convert it to an easily workable form and then perform the analysis to find differences between workspaces.

The task was accomplished with success, all important information was found to help managing the necessary changes.

The second task was also successfully done. Improving data synchronization helped finding particular records and automation of possible processes helped to save time to the managers.

For me personally training was beneficial. I earned valuable experiences, got better knowledge about how a big company works and was part of the meetings to find good solutions and solve many problems.

References

1. W3C, Extensible Markup Language (XML).
http://www.w3.org/XML/.
Accessed on 11 April 2013.

2. W3Schools, XML Tutorial.
http://www.w3schools.com/xml/.
Accessed on 11 April 2013.

3. W3Schools, SQL Tutorial.
http://www.w3schools.com/sql/.
Accessed on 11 April 2013.

4. Microsoft, Book Landing Page: Beginning Access 2007 VBA.
http://msdn.microsoft.com/en-us/library/dd897493(v=office.12).aspx.
Accessed on 11 April 2013.

5. Microsoft, Get started with Access programming – Access.
http://office.microsoft.com/en-us/access-help/get-started-with-access-programming-HA001214213.aspx.
Accessed on 11 April 2013.

6. Microsoft, Microsoft Word – document and word processing software.
http://office.microsoft.com/en-us/word/.
Accessed on 11 April 2013.

7. Microsoft, Microsoft Excel – spreadsheet software.
http://office.microsoft.com/en-001/excel/.
Accessed on 11 April 2013.

8. Microsoft, Microsoft Access – database software and applications.
http://office.microsoft.com/en-001/microsoft-access-database-software-and-applications-FX010048757.aspx.
Accessed on 11 April 2013.

9. Microsoft, Visual Studio | MSDN.
http://msdn.microsoft.com/en-US/vstudio.
Accessed on 11 April 2013.

10. SAP, Data Visualization | Business Intelligence & Analytics | SAP.
http://www.sap.com/pc/analytics/business-intelligence/software/data-visualization/index.html.
Accessed on 11 April 2013.

Figures