On the Generation of Dynamic Business Indicators

Fábio Alexandre Pereira dos Santos, Rui César das Neves and Joaquim Belo Filipe
Instituto Politécnico de Setúbal, Escola Superior de Tecnologia de Setúbal, Setúbal, Portugal

Keywords: Dynamic Generation of Data, Business Intelligence, OLAP, Data Representation, Infographics, Entity Framework.

Abstract: While information is rapidly gaining relevance to the organizations, systems that help companies analyse that information need to improve their effectiveness in several layers. We address, in our on-going research work, mainly the presentation layer and business layer of software systems development. The aim of this position paper is to discuss how to develop a system that is as flexible and configurable as possible, which allows multiple methods of analysing and visualizing the relevant data to an organization, allowing them to define certain views on business data with appropriate graphics. One of the values from this system is the domain of technology in a controlled environment, because it helps solving a specific problem, since it is both a generic tool, autonomously and with a high degree of adaptability. Flexibility will go hand-in-hand with ensuring the consistence of the data model, i.e. the data being analyzed by an user must respect syntactic and semantic constraints when relations to each other created in the organization's logic. This feature prevents the user to attempt to connect data that aren't related. Based on the existing data types described on metadata in the data model, the system provides the users with a list of possible graphical representations for the selected information type. This list will be filtered in order to allow the user to select only graphical representation types that are appropriate to the selected data types. This is an innovative feature, in the sense that the system constrains the selection of the visualization elements thus avoiding potential conceptual errors.

1 INTRODUCTION

In recent times an increasing attention has been given to take advantage of information and knowledge in organizations in order to gain competitive advantage.

Thus, the information about organizations is now an essential component of support for their operations, since it allows creating internal and external favourable conditions to reduce costs and to provide innovative services. It also facilitates the construction of knowledge that is required to plan and implement solutions for problems and challenges that arise everyday on an organization.

In organizations, are agents there are responsible for decision making and in this process they require rigorous tools to analyse the data and further improve the performance of the organization. Those agents need to access and process different types of relevant data.

Unfortunately the development of information systems is, usually, more concern in the development of the information input areas, creating system that collects information and guarantees the information coherence and leaving the outputs areas with limited capabilities.

The main goal is the development of a tool that could be connected, in a standard way, independently from the application domain, to an already developed system which gives information extraction capabilities to the end user. These capabilities will allow the drill down in all information available, maintaining the coherence and relations defined in the input area.

With this in mind, the system presented in this position paper is a tool to help companies mining their own data and obtain new relevant information, which is not accessible from the traditional information systems.

Actually there are a lot of systems doing similar things, but most of them are not as specific and configurable as desired. The systems that are configurable often have features that most organizations do not need.

Alexandre Pereira dos Santos F., César das Neves R. and Belo Filipe J..
On the Generation of Dynamic Business Indicators.
DOI: 10.5220/0004199603900394
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 390-394
ISBN: 978-989-8565-29-7

2 IMPORTANT TERMS AND SYSTEM MODEL

In this section, it is important to clarify a few critical terms that are needed to set the stage for everything else that follows.

The first term is **entity**, which is defined as a concept in the domain of application from which a data type is defined, where data type is an element that represents the structure of a top-level concept and is a template for instances of entity types in a system.

Another important term is **indicator**, which is defined as a set of data belonging to one or more entities which are related to each other in the data model. It's not only necessary to be related, but also that they have the same meaning in the organization's business logic.

An important aspect of the system that we will implement based on these ideas is its generic and configurable structure, independent of the application domain and requiring only a set of rigorously defined terms in order to allow navigation through all information.

Thus, the aim is to develop a web-based system that allows structured access to information, according to the organization's business logic. This gives to the end user the opportunity to do queries using a configurable and appropriate interface. The integrity of these queries will be guaranteed through the modeling process required by ORM (Object/ Relational Mapping), because this, ORM, will guarantee that the end user has restrictions on data access, avoiding mistakes, unlike conventional systems in which the ORM is applied on developer side.

Based on this, the system is built around the approach of the Entity Framework (EF) from Microsoft, ADO.NET. There are other ORM technologies available in the market, but we choose this one because it comes from a major software supplier, is the ORM embedded in.NET Framework, and is the approach that we are studying.

It is also important to mention that changes in the data model shouldn't force the rewrite of the whole system and should allow the reuse of all available information.

The idea is to provide to end users a list of possible graphic representation forms for an indicator, because the aim is to constrain possible actions by the user, thereby reducing the probability of mistakes on selecting a graphical representation.

We select that kind of representation, although there are several ways to display and process these types of data, but the most simple is through graphical representation, being this conclusion given by (Tufte, 2001) "(…) *graphics are instruments for reasoning about quantitative information. Often the most effective way to describe, explore, and summarize a set of numbers – even a very large set – is to look at pictures of those numbers. Furthermore, of all methods for analyzing and communicating statistical information, well-designed data graphics are usually the simplest and at the same time the most powerful*".

These systems, which help decision-making, must have the simplest forms of data representation, i.e., it should be able to easily illustrate the graphical representation of sums, dates and tables.

It is also expected to illustrate facts that have occurred in time, i.e., present time series, for a given data, because with only one dimension this kind of representation allows to display time series, with an appropriate scale that can't be achieved with another type of graphical representation.

Another important feature is the presentation of indicators regarding space, because organizations typically have data that is related to locations / regions and are of great importance to the organization's business model.

The graphical presentation is one of the key points in this system, because according to (Tufte, 2001) *"(...) No doubt some graphics do distort the underlying data, making it hard for the viewer to learn the truth*". Thus, it is important that the system can filter out these types of representations, according to the types of data that compose the indicator; this feature will be described in section 3 of this paper.

The system that we are implementing is clearly placed in the scope of the so-called OLAP, i.e On-Line Analytical Processing.

OLAP provides the ability to analyse different aspects of information in a fast and dynamic way, where large volume of data are contained within a data warehouse according to (Thomsen, 2002) "(…) OLAP is meant to contrast with OLTP (On-Line Transaction Processing). The key aspects are that OLAP is analysis-based and decision-oriented".

Until recently these systems were known as DSS (Decision Support Systems), but now it is common to refer to them as Business Intelligence (BI) systems, which according to (Rud, 2009), "(…) BI is defined as the ability for an organization to take all its capabilities and convert them into knowledge (…)".

The concept of BI is relatively recent and BI systems are usually composed of a set of tools that enable report generation and allow users to extract useful information from the stored data.

BI systems have also a set of tasks associated, which according to (Cardoso, 2011) *"(…) can be set into four groups:*

- *Make predictions based on historical data, based on past performance and current;*
- *Creation of scenarios that demonstrate the impact of changes to existing variables;*
- *Allow ad-hoc access to data;*
- *Analyse in detail the organization, thereby ensuring a deeper knowledge about the same.*"

With the sets of tasks and features shown above, the traditional systems could work connected to the BI systems yet independently, allowing to apply specific techniques, which according to (Santos and Ramos, 2009) *"(…) BI systems have implemented the functionality, scalability and security of existing systems which manage the database for building data warehouses that are analysed with techniques for OLAP, Data Mining and Query Report*".

The system being developed cannot be considered a BI tool as it doesn't cover the first two groups identified by the (Cardoso, 2011), just covering the third and fourth points - allow ad-hoc access data and in detail analysis of the organization.

At the data source level, the system needs a model that contains information about the present data and the relationships between objects that compose this data.

The next section will describe forms of presentation for indicators.

3 DATA AND DATA TYPE REPRESENTATION

With the information available from the EF, the system must advise users what types of common graphical representation can be used to see the data. After analyzing the major database vendors, namely Oracle, SQL Server and MySQL, it can be concluded that all kinds of data may be grouped in 4 groups, which we call Numeric, Date and Time, Strings and Other Types of data that we don't want to represent, like "XML", "Cursor", "BLOB", "BFILE" types, etc..

The most common forms of representation of these groups of information are "Column", "Lines", "Pie", "Bar" and "Scatter".

The scatter plots (X, Y) usually shows values in both axes X and Y, while the other graphs show usually in the Y axis, using "categories" in the X axis.

This information is very important because it allows the system to tell the user which data representation can be selected. Better yet, according to the data types of the selected information, we can tell the user what kind of graphical representation shouldn't be selected, avoiding the user to select forms of graphic representation which may not be suitable to the selected data.

Initially we want to use simple graphics, which can be represented by 2D representation, because we need to find an efficient way to help users avoiding errors on select data representation.

Those simpler graphic representation forms can be divided into two groups, depending on the data types that we are working on. Those groups are Categories vs. Values and Values vs. Values, as we can see on Table 1, where categories are like objects and values are like the object values. The second one is a traditional graph, where values for both axes are available.

Table 1: Forms Representation.

Forms Representation															
Categories Vs. Values															
Data Types	Numeric					String					Date and Time				
Numeric			x			x	x	x			x	x		x	
String			x	x									x		
Date and Time	x			x				x							
Graphical Representation	C	L	P	B	S	C	L	P	B	S	C	L	P	B	S
Values Vs. Values															
Data Types	Numeric					String					Date and Time				
Numeric					x										x
String															
Date and Time															
Graphical Representation	C	L	P	B	S	C	L	P	B	S	C	L	P	B	S

Legend:
C - Column
L - Line
P - Pie
B - Bar
S - Scatter

The idea of this mapping, between the existing data types and forms of graphical representation, is to show that there is dependence between the existing data types and the ways to represent them. According to the table above, there are types of data that cannot represent certain forms of information and that when displayed on screen are shown in the wrong way and therefore lose their meaning.

4 DEVELOPMENT

In the following sections, the main focus is to describe how we want to solve the problem, explaining the specifications and highlighting some implementation details.

4.1 Requirements Analysis

The requirements for the project are divided into two different groups: technical and user interface.

4.1.1 Technical Requirements

Regarding technical requirements, it is expected that the system is capable of accessing information types at runtime and validate new data relations that the user may wish to create.

In the EF, the conceptual model, storage model and the mapping between the two is defined on an ***.edmx*** file. This file is updated when either the database or the model changes. According (Corporation, Modeling and Mapping, 2011), "the EDM Generator, which is included with .Net Framework, generates the ***.csdl***, ***.ssdl*** and ***.msl*** files from an existing data source".

These files, XML-based, describe the conceptual model, storage model, and mapping, are known as metadata.

This means accessing the metadata which was represented in .csdl and .ssdl files and was loaded into instances of the "***System.Data.Metadata.Edm.EdmItemCollection***" and "***System.Data.Metadata.Edm.StoreItemCollection***" classes, which are accessible by using methods in the "***System.Data.Metadata.Edm.MetadataWorkspace***" class.

These instances allow the system to understand the structure of data, thus, allowing navigation through all information to create query's dynamically as well as an appropriate representation for the user.

4.2 User Interface

The second group of requirements defines the user interface: users will use the application to analyse data which originated from the data source.

The system should enable its users to create and display dynamically a set of indicators through graphical representation, which can be customized for each user.

The user should be able to select or even create data visualization models at runtime, in a highly configurable way: the user could add a new graphic representation for an indicator, he/she could also remove certain graphical representation forms or modify parameters of others.

If the user wants to create a new graphical form to represent the indicator, user should select which properties from entities wants to relate and after that the system will process the selected data. After that, the system will show to user the output. If the user wishes to add more constraints to this process, may at any time add it.

This process, the selection of properties from entities, should operate in a cycle, and this cycle is summarized in Figure 1, allowing refining the output, which will be shown on the user interface.

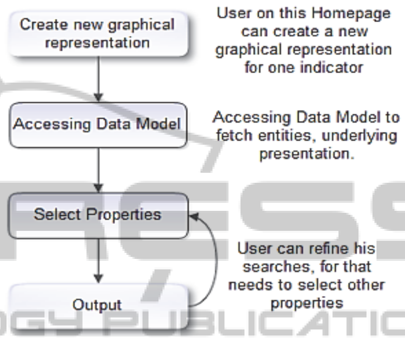

Figure 1: Operating Workflow.

As the user can have a large amount of graphics and indicators representation, the idea is to give the ability for to personalize her/his interface, deciding the order in which the content should appear on the page, for example. The user should be able to minimize, move items and presentation forms around the page as if it were a design surface and to remove items as needed.

The system will be required to maintain the user page customization, in order to avoid repeating the visualization settings each time the user visits the system. This can be done using Web Parts, a technology that provides an appropriate way to build a modular Web Site that can be customized, with dynamic settings, on a per-user basis. This customization is provided by one provider model – Personalization Provider, and according to some authors (Evjen et al., 2010) *"(...) this provider makes associations between the end user viewing the application and any data points stored centrally that are specific to that user"*, giving exactly what we need.

The approach used in the proposed implementation follows a similar approach to the traditional providers, according to the same source (Evjen et al., 2010) *"a provider is an object that allows for programmatic access to data stores, processes and more"*, and in this systems' context means the independence of a data model.

5 CONCLUSIONS AND FUTURE WORK

In this position paper, we presented a comprehensive set of criteria for the development of a data visualization system. This system is based on Microsoft EF and it has, explicitly or implicitly, a constrained conceptual data model that describes the various elements of the problem domain.

The data model represents concepts and also the relationships between concepts, constraints, and so on.

Since currently most applications are written on top of relational databases, they will have to deal with data represented in a relational form. The programming paradigm is typically some form of Object-Oriented Programming (OOP) including features such as data abstraction, encapsulation and inheritance, and these features are fundamental to the flexibility that is intended to characterize this system.

Typically a higher-level conceptual model is used during the design phase, and that model is not directly executable, so its need to be translated into a relational form and applied to a logical database schema and to the application code.

With this flexibility guaranteed by EF, our system differs from many other BI systems on this point as well, because the great majority run on top of traditional databases, i.e. other systems are based on multiple data sources but ours just needs to connect to a schema from EF.

Another feature of our system is the ability to help users to visualize the information that was identified as relevant business indicators, based on the selection and combination of data types or dynamic relations and constraining the choice of possible graphical representations.

This feature is also important because it avoids user errors when selecting the graphical representation forms for the selected information. The system that we are developing will use an approach like the providers approach because, as we mentioned above, this feature allows the user to access data via a web browser, in readable form, without the need to use a complex system.

It is intended that the system will evolve to an interface that allows the user to create a configuration of graphical representation forms, as well as the possibility of adding new types of graphical representation and mapping them to existing data types.

ACKNOWLEDGEMENTS

We would like to thank the Polytechnic Institute of Setúbal, School of Technology of Setúbal, for supporting the research work reflected in this paper, presented at KDIR 2012 in the scope of the RETE project.

REFERENCES

Cardoso, E. (2011, Setembro). *Introduction to Business Intelligence.* Lisboa, Lisboa, Portugal.

Corporation, M. (2006, June). *The ADO.NET Entity Framework Overview*. Retrieved 04 2012, 04, from MSDN: http://msdn.microsoft.com/en-us/library/aa697427(v=vs.80).aspx

Corporation, M. (2011, 8 29). *Modeling and Mapping*. Retrieved 5 5, 2012, from MSDN: http://msdn.microsoft.com/en-us/library/bb896343

Evjen, B., Hanselman, S., & Rader, D. (2010). *ASP.NET 4 in C# and VB*. Canada: Wrox.

Inmon, W. (2002). *Building the Data Warehouse.* Willey.

Rud, O. P. (2009). *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy.* Hoboken, New Jersey: Wiley & Sons.

Santos, M. Y., & Ramos, I. (2009). *Business Intelligence.* FCA.

Thomsen, E. (2002). *OLAP Solutions - Second Edition.* Willey.

Tufte, E. R. (2001). *The Visual Display of Quantitative Information 2ª Edition.* Graphics Press LLC.