A Model-Driven Method to Generate Visual Analytics System

Ying Qian[1, a], Shuai Zhao[2, b]

[1]School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

[2]School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

[a]qianying@cqupt.edu.cn, [b]sdzsh@126.com

Keywords: Visual Analysis; System-generated; Model Driven; Development Method

Abstract. With the increasing development of requirements and frequent changes in business needs, the total costs of visual analysis systems for commercial applications increase continuously in recent years. To develop a visual analysis systems, the developers are required to have academic background knowledge in data visualization. However, the demanders of these system in commercial field often do not have such backgrounds. To solve this problem, this paper proposes a Model - Driven Visual Analytics System Generate Method (MDVAM) and gives application examples of this method in the framework of WinForm and the C# language as well. These examples proved that the visual analysis generation system using the method can make the non-expert users complete system development tasks quickly and efficiently.

1. Introduction

Visual analysis is an important method of data analysis and a new cross-fusion research area of scientific visualization, information visualization, data mining, human-computer interaction, cognitive theory and data management formed [1]. Visual analysis highlights assimilating the visual perception of human into knowledge discovery [2]. Its analysis process includes two main lines, one is automatic analysis and another is visualization analysis [3]: The system presents the data results of automatic analysis to the user by visualization technology, the user understands the data by visual interface and changes the data automatic analysis of the system by human interface, then the system presents the improved data analysis to the user. The process is cyclic until the user achieve their analyzed goals.

The application of visual data analysis techniques to analyze business process, human visual perception of complex patterns will be integrated into value discovery, this will improve the quality and efficiency of data analysis largely. However, business needs in business data analysis system changes frequently, develop a visual analysis system of business function fixed has been unable to meet the needs of business users, and because the business needs change repeatedly development system, so development costs will substantially increase.

The main work of this paper has two aspects: First, it proposed a visual analysis method MDVAM model-driven system generates, Its core business is abstracted business needs into a platform-independent interface domain model and the automatically generated from model to code; the second is to build a visual analysis generation system according MDVAM in C # language and Winform framework.

2. Related Work

Visual analysis involves scientific visualization and information visualization. Starting from commercial property data, it even more biased in favor of information visualization. Card SK et al. divided the information visualization technology into one-dimensional, two-dimensional, three-dimensional, multi-dimensional, hierarchical, network, and timing information visualization depending on the difference of data attributes [4]. In recent years, researchers around the classified

proposed by Card SK et al, the visual analysis technique has been applied in different fields, gave birth to several types information of distinctive features, Including text, web and maps, temporal and multidimensional data [5]. Corresponding to the type of information, there are Word Clouds [6], Tree-map [7], Heatmap [8] and Parallel Coordinate [9] and other common visualization model. In addition to these three models, common data histograms, line, pie, area, and scatter charts and other visual data analysis model is an indispensable part, In the actual analysis tasks, it usually used for the above four categories of assistance in explaining the model.

Human-computer interaction is an important part of the visual analysis. The interactive technology of information visualization can be summarized as: dynamic filtering technology, overall and detailed technical, translation and scaling technology, the focus and context-sensitive technology and multi-view coordinate technical [5].

In recent years, researchers have proposed and developed a series of methods and tools, such as ManyEyes [10] and InfoVis [11] and so on. Its main purpose is to generate a visualization application quickly, but it does not provide a common development methodology of visualization systems.

3. MDVAM

Model-Driven Development (MDD) is a model-oriented analysis and design methods. In this method, use model to driven software development process, achieve the purpose that the separation of business and technology. In 2003, the Object Management Group (OMG) has developed a standard Model-Driven Architecture (MDA). Its core is the transformation between model and model [12, 13]. Model-driven software development refers to abstraction modeling for the practical problems of high-level, and generate executable code through model transformation [14].

In order to meet the changing and growing of business needs data analysis, this paper presents a system generates method of visual analysis MDVAM which is driven by model. The core of the method is composed by interface domain model, the model description language and code generation automatically.

3.1 Interface Domain Model

The development of visual analysis system involves visualization and human-computer interaction technology. Only experts studying visual techniques can master it. And business decision makers do not have the expertise and system development capacity. Interface model oriented development method is one model-driven development approaches [15]. It is mainly focus on users who doesn't have expertise and system development capabilities. To help them be able to develop business-oriented application quickly in an easy way.

According to the thought of interface model oriented development method, this paper defines 5 elements for interface domain model, including information, characterization, correlation, incident and layout.

3.1.1 Information

Information element is basic data for data to be analyzed, it contains three types of basic attributes as shown in Table1.

Table 1 The basic attributes of Information elements

Attribute	Definition
DataSource	Source of data to be analysed, such as memory, file or database
DataSet	A collection of data to be analysed, including data type and size
DataConstraint	Describe the relationship between each data set

3.1.2 Presentation

Presentation element is the description of the visualization model, it contains four basic element attributes as shown in Table 2.

Table 2 The basic attributes of Presentation elements

Attribute	Definition
Type	Types of visual model, such as the Text Visualization model or the

	Multi-dimension Data Visualization model
FeatureStructure	Visual model description of characteristics, such as the structure of the model, the layout rules, rendering method and so on
Parameter	Visual model parameter descriptions, such as color and line thicknesses
DataCategories	Visual model of the supported data dimensions and types, etc

3.1.3 Relevance

Relevance element is the description of human-computer interaction techniques. Usually, a data set include more than one information sides, called information polyhedron [5]. Each information side of the information polyhedron shows the different angle of the same data set. By establishing multiple visualizations of the same dataset, multi-view correlate technology defines the dynamic relevance mechanism between various views, and provides support to information polyhedron. Using correlated coordination technology in human-computer interaction, MDVAM defines the relevance rules between visualization models which are chosen by users.

3.1.4 Event

Event element describes the collection of user interactions, such as click, select, drag, drop and so on. It includes computing-related and computing-unrelated events, which is used to distinguish the users' interactions whether they are in demand of the support of data calculation.

3.1.5 Layout

Layout element contains visualization model layout and interface container layout. Interface container layout is used to define the size of containers. Visualization model layout defines both the size of the visualization model and its position in the interface container.

3.2 Model Description Language

Model Description Language is an important part of the MDA. It provide basic support to the conversion between models and automatic generation form model to code. MDVAM developed description language for the interface domain model, which is used to record the content model in detail. The description language is platform independent, and the details are stored as XML files. Its structure is shown in Figure 1.

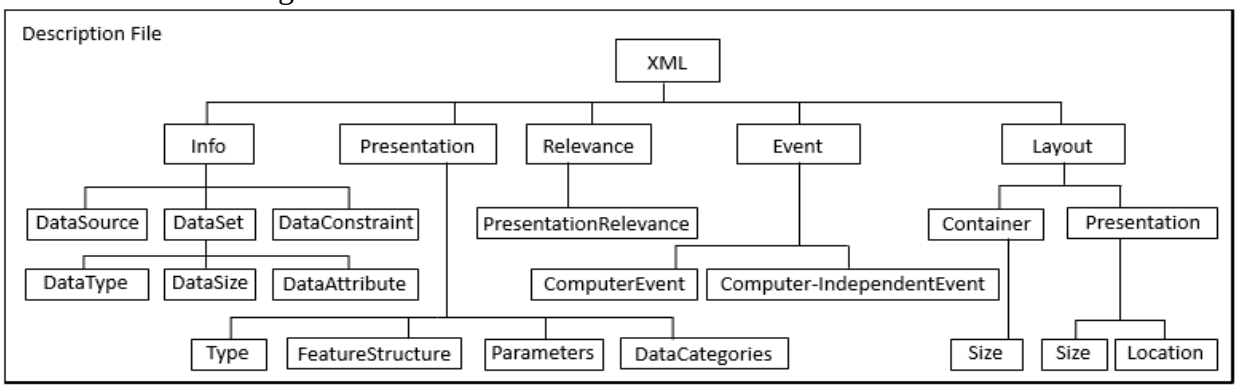

Fig. 1 The structure of XML description file

3.3 Automatic generation of code

Automatic generation of code is a mapping process helps transfer the model description language to executable code. Depending on different functions of the production code, it can be divided into three categories, including data mapping, graphics mapping and logical mapping. Data map contains the conversion from source data type to system data type and data calculation. Graphical mapping is a process converting the description language related to drawing into platform usable code. Logical mapping is used describe the response of system behavior to user interactions. The mapping relations of code is recorded as a XML file. It can be mapped to the same language according to different target system. Figure 2 is a part of its mapping relationship.

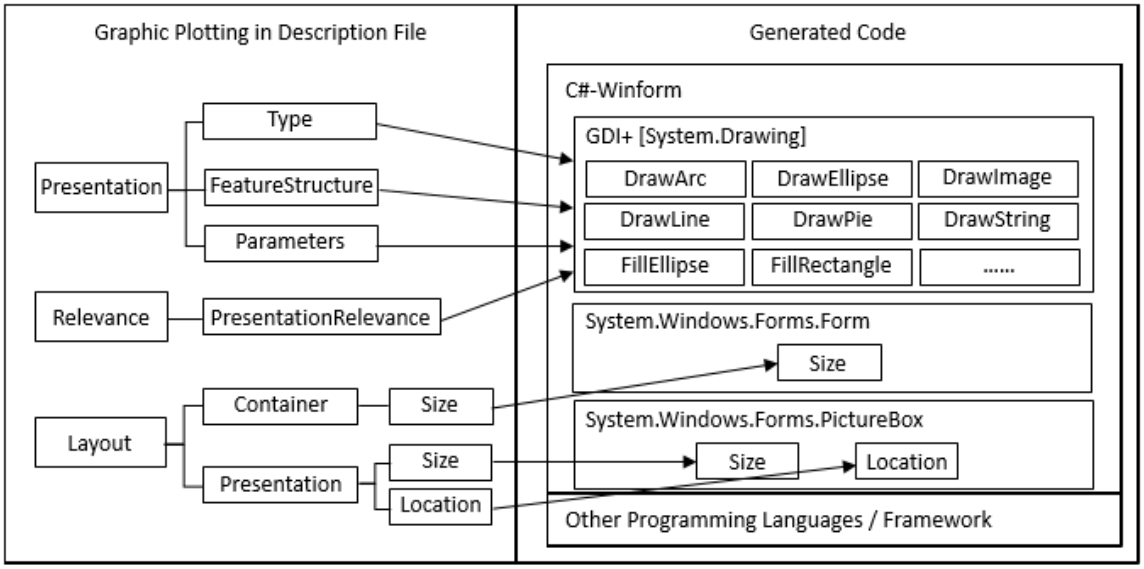

Fig. 2 Mapping relationship for part of the model description language to executable code

3.4 Approach Architecture

As shown in figure 3, there are 3 parts in the architecture of MDVAM, including the configuration, interface domain model and visual analysis system. The operation of system begins with user data source and the visualization configuration. Data source can be designated as a local file or a database. Visual configuration is oriented to business domain, users need to select the corresponding visualization model set according to different data type, and set the parameters of selected visualization models, correlation, interactive events and layout interface. Interface domain model is the core of the system. This model uses the description language generator provided by the system. It maps the user profile of data and visualization to model description file. Using language parser model, the model description file is transformed to executable code according to different development platform and language. Finally, the code is embedded into system framework to generate visualization analysis system.

Data processing, graphics rendering and event handling are the three functions engine of the system. The data processing engine contains data preprocessing and data calculation. Data preprocessing includes data extraction, data cleansing and data conversion. In order to improve the efficiency of data calculation, depending on the data storage medium and the volume of data, data needs to be extracted in two ways. Firstly, the small volume data is loaded to memory from a file or database. Secondly, the target data is extracted to database or file from the data warehouse. Data cleansing is used to filter error data. Data conversion is transferring the raw data to a data type which is supported by development language according to user-selected language. Data calculation is divided into memory computation and database computation according to different data mediums. It includes calculation type under different languages and database. Graphics rendering engine is used to draw the user interface. Its work is using the data obtained by data processing engine to draw user UI according to the visualization configured by user. Event processing engine is responsible for the interaction between user and UI. Depending on the characteristics of user's operation, the event handler is divided into two categories. One does not involve the operation of the data, only needs to call the visualization rendering engine to redraw the system UI, such as user visualization container pan and zoom. The other one operation, the event processing engine will call data processing engine and visualization rendering engine to redraw UI based on user operations focus on the data.

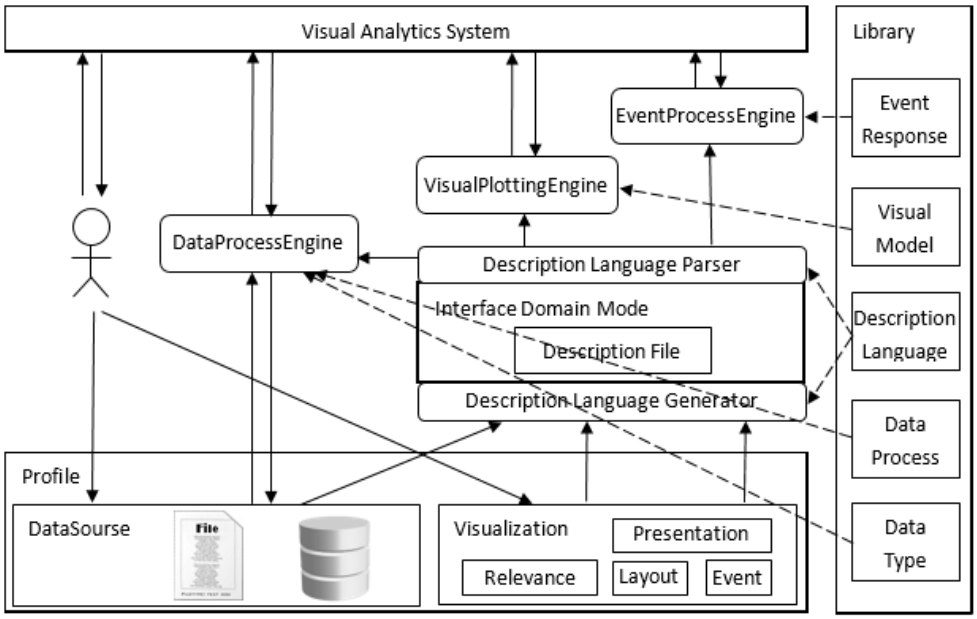

Fig. 3 System Architecture

The normal operation of the system is depends on the support of component library. The component library includes incident response, visual model, description language, data processing and data type. Incident response library includes user event monitoring and processing operations, which includes monitoring classes and processing methods of all operation events of users, serving for incident processing engine. Visualization model library includes various detailed visualization model parameters and supported incident response type. It defines model drawing methods under various types of language services, serving for graphics rendering engine and event processing engine. Description language library contains the map rule form user configuration to model description language and the map rule from model description language to various development languages, serving for model description language generator and parser. Data processing library includes the preprocessing and computation methods of data, serving for data processing engine. Data type library contains the conversion rules off raw data under various languages, serving for data processing engine.

4. The System Implementation

In order to verify MDVAM, this paper intercepted 847 accounts of a bank in Chongqing in June 2014, which contains 5268 transactions of 200 customers, and implemented a complete prototype system with C# and Winform framework. The system includes three function modules, which are data processing, business configuration and interface domain model description of file generation and analysis.

4.1 Data Processing

The source data exists in the remote database. Considering the relatively small volume of data and network delay and other factors, the paper decided to load the source data into memory in order to improve the computational efficiency of the data. Then the system provides extraction and configuration to the data source, as shown in Figure 4 and Figure 5. To further reduce the data volume, the system supports user-selectable data fields, and provides configuration data associated with the function. Data cleansing and data conversion are also are important part of treatment. As shown in the table 3, due to the bank sub-business systems too many, different definitions to the different subsystems business so that there are hundreds of different kinds in the type of business transactions , the next step in order to carry out effective analysis of data, the need for these data reasonably clean. Save the data type library as an XML file, provides support for the data conversion, the contents of the structure. Systems based on XML description file, the source data field type is mapped to the system data type, and allows the user to redefine the system data type field mapped.

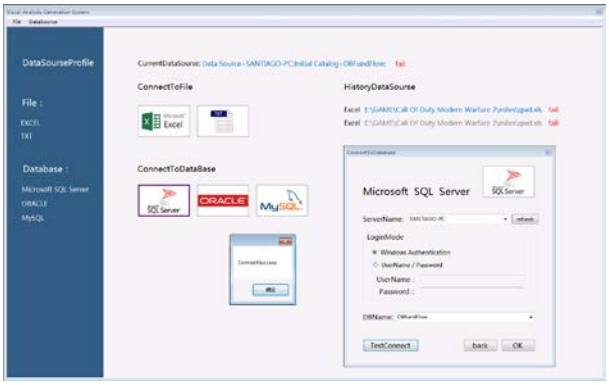

Fig. 4 Choose DataSourse

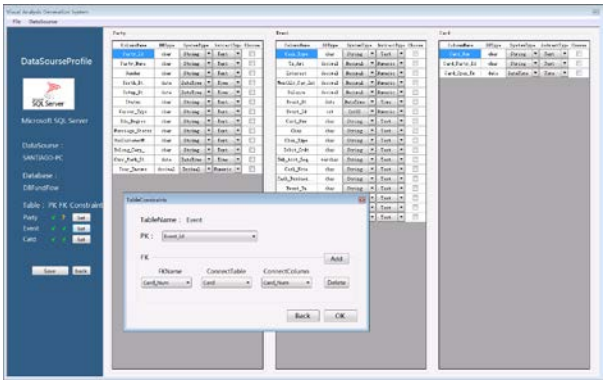

Fig. 5 DataProfile

Table 3 Types of business transaction data

BusinessType	Content
Transfer	Net transfer, transfer, ATM transfer, POS transfer,..., agent transfer, POS transfers
Remittance	City remittance, return,..., transfer, remitted outside, borrow, deposit, remittance
Deposit	Card deposit, existing , adaptation,..., counter, ATM deposits, accounts, batch accounts
...	...

4.2 Visualization Configuration

Visualization configuration is the process of user for business modeling. The system provides users with business modeling capabilities. The main interface function is shown in Figure 6: the left is the data interface panel, which contains the data field of the user selected; the middle is the visual interface to configure the main panel; the right of the interface is the visualization model kit, providing a visual model library. First the users will need to visualize a model container from the right panel and drop it into the main panel, and interface layout. Visualization model container includes a data area, a model and Configuration area of three parts. The user data field of the data model panel to drag and drop the data area of the vessel, the system will be the model for visualization toolbox model to filter the data type and structure, and exhibits available models. The toolbox can be used to model the model drop zone after a container, the model can be configured to operate. Model configuration interface shown in Figure 7, contains the model parameters, and events associated with vessel response in three parts, can be configured to operate a vessel within the model: the model parameters provides the user visualization model size, location, color , set the grid and axis parameters; associative containers set by the user to provide a multi-view association between models within each container that is the main interface, visualization; incident response model provides a supported event types, the user can select the desired event needs, by defining event actions associated with the container, you can view more complete linkage semantic definition.

4.3 Interface Domain Model Description File

Interface domain model describes file is an important bridge between the user service configuration and visualization analysis system. The system provides a description language library, use the profile generator configuration and user data into a visual configuration mapping XML description file and platform-independent, its generation rules shown in Figure 8.

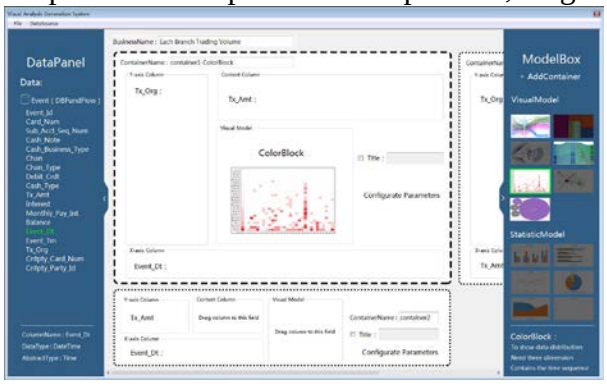

Fig. 6 Interface of business configuration

Fig. 7 Visual model parameters, correlation

and event configuration

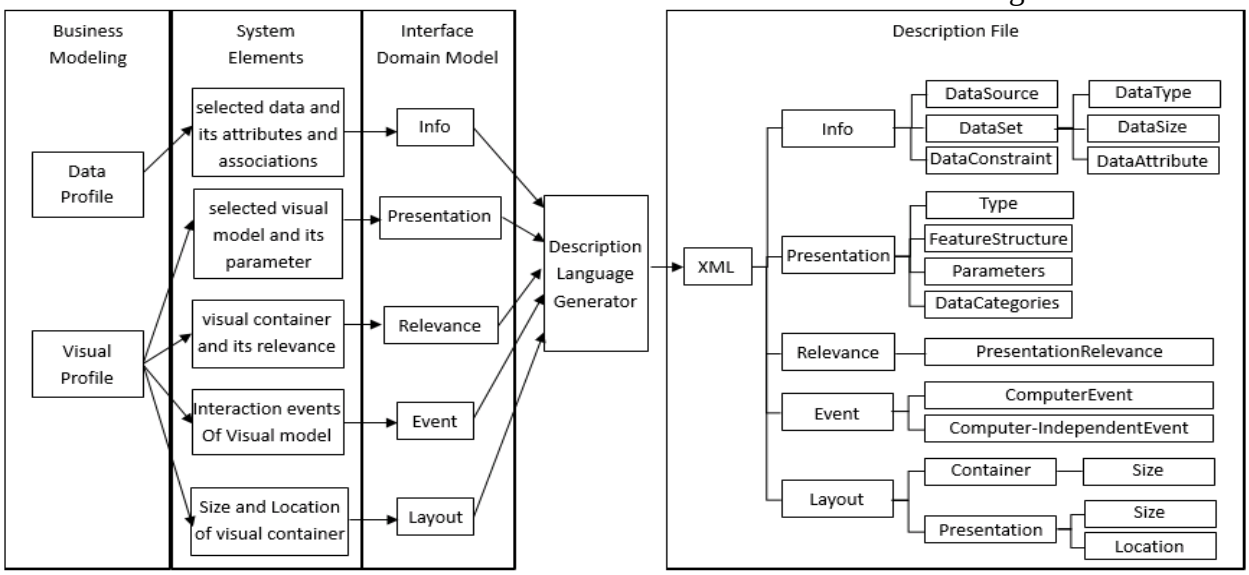

Fig. 8 Description file generated rules

The system automatically generates the interface is domain model description file parsing process. The system provides for description file parser of the C # language and Winform framework. It can map model description language into executable code, embedded into the system and framework, that visual analysis system generated for the result. Analysis of description file and system generation process shown in Figure 9.

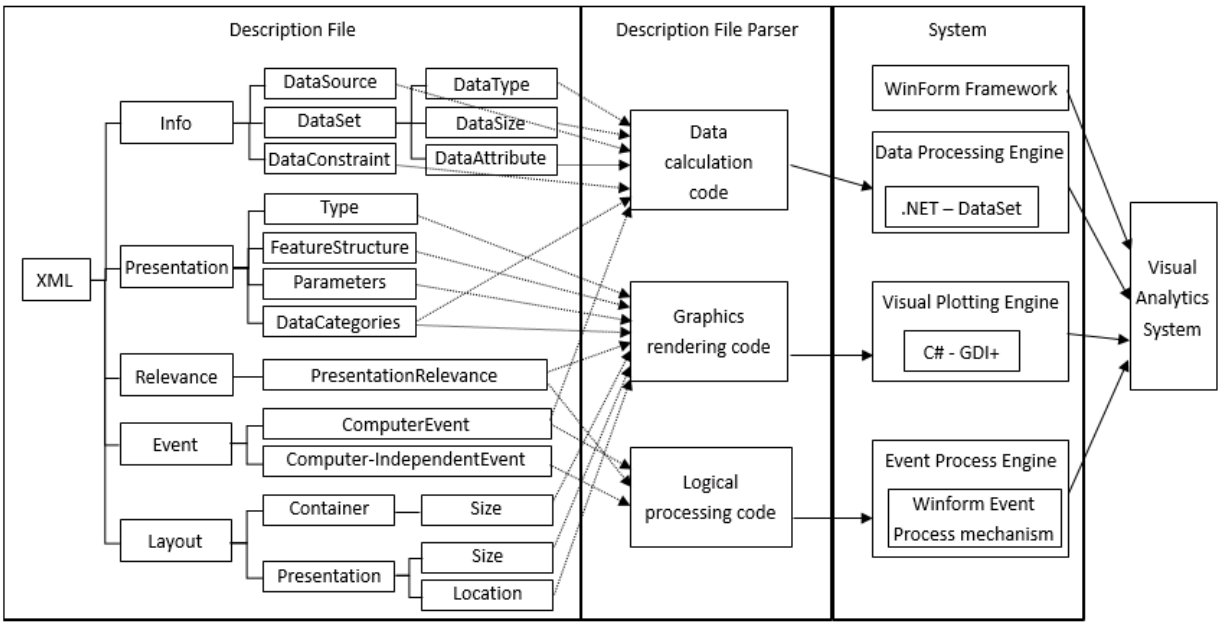

Fig. 9 Process of description file parsing and system generation

4.4 Visualization Analysis System

Visual analysis system is generated by the description file parser as shown in FIG. 10 and FIG 11. Providing analysis function which for the trading volume of the user is shown in interface Figure 10: the top left of the interface is the main model, the Color Block shows different branch transactions distributed over time, and the right side of the histogram below the bar graph master model is in line with the main models of assistance in explaining for the data, histograms and pie chart on the right interface describes the different channels of trade in the constitution. Interface master model is associated with around statistical model, the data is shown from different perspectives. The time range of interface analysis FIG. 10 is in June, in which among the Color Block model, there is obvious highlight of longitudinal data in No. 21, when the user clicks the Timeline tab 21 that can be achieved like the interface as shown in Figure 11, and then specific analysis on transactions of No. 21.

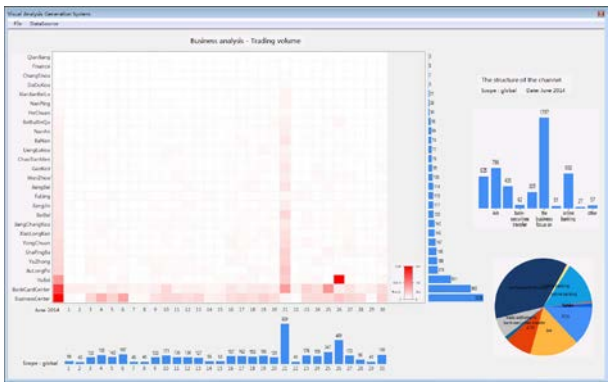

Fig. 10 Visual analysis of the Banks full month interface

Fig. 11 Analysis of 21 volumes of business

Implementation of the system to verify the feasibility of MDVAM approach. By supporting MDVAM method and system toolbox, uses do not have the ability to encode and visual analysis for business backgrounds can rapidly complete the development of the visual analysis system, thereby greatly reducing the threshold of the visual analysis system and improve the development effectiveness.

5. Summary

Based on the model-driven development methodology theory, this paper generating a method of visual analysis system MDVAM, gives the definition of the interface domain model and an XML file structure that describes the described language. In this paper, it gives supporting method of MDVAM in C # language and Winform framework, even verify the feasibility by an example method. MDVAM still exist shortcomings: The visual analysis system that MDVAM generated with the lack of analysis and the reasoning process management tools, cognitive theories in the analysis process is not fully reflected. This is also the focus of this future work.

References

[1] Keim D, Andrienko G, Fekete J, et al. Visual analytics: Definition, process, and challenges[M]. Berlin: Springer Berlin Heidelberg, 2008. 154-175.

[2] Keim Daniel, Qu Huamin, Ma Kwan-Liu. Big-Data visualization[J]. IEEE Computer Graphics and Applications, 2013,33(4): 20-21.

[3] Keim D, Konlhammer J, Ellis G, et al. Mastering the Information Age: Solving Problems with Visual Analytics[M]. Goslar: Eruographics Association, 2010. 1-168.

[4] Card SK, Mackinlay JD, Shneiderman B. Readings in Information Visualization: Using Vision To Think[M]. San Francisco: Morgan-Kaufmann Publishers, 1999. 1-712.

[5] Ren L, Du Y, Ma S, Zhang XL, Dai GZ. Visual analytics towards big data[J]. Journal of Software, 2014(9): 1909-936.

[6] Wu Y, Provan T, Wei F, et al. Semantic-Preserving word clouds by seam carving[J]. Computer Graphics Forum, 2011,30(3): 741-750.

[7] Zhang Xin, Yuan Xiaoru. Treemap Visualization[J]. Journal of Computer-Aided Design & ComputerGraphics, 2012,24(9):1113-1124.

[8] Bojko A A. Informative or misleading? Heatmaps deconstructed[M]. Berlin: Springer Berlin Heidelberg, 2009. 30-39.

[9] Heinrich J, Weiskopf D. Continuous Parallel Coordinates[J]. IEEE Trans. on Visualization and Computer Graphics, 2009,15(6): 1531-1538.

[10] Viegas FB, Wattenberg M, van Ham F, et al. ManyEyes: a Site for Visualization at Internet Scale[J]. IEEE Trans. on Visualization and Computer Graphics, 2007,13(6): 1121-1128.

[11] Fekete J. The InfoVis Toolkit[C]. IEEE Symp. on Information Visualization (INFOVIS 2004), 2004:167-174.

[12] Brown A W. The model driven architecture: Principles and practice[J]. Software and Systems Modeling, 2004,3(4): 314-327.

[13] Favre L. Foundation for MDA-based forward engineering[J]. Journal of Object Technology, 2005, 4(1):129-153

[14] Ren L, Wang WX, Zhou MJ, Teng DX, Ma CX, Dai GZ, Wang HA. A Model Driven Development Method for Interactive Information Visualization. Journal of Software, 2008(8):1947-1964.

[15] Puerta A, Eisenstein J. Towards a general computational framework for model-based interface development systems[J]. Knowledge-Based Systems, 1999,12(8): 433-442.