Design and Implementation of Parameterized Report Builder component

Yingchun Li
School of Software,
University of Science and Technology Liaoning,
Anshan, 114051, China
E-mail: as_lyc@163.com

Abstract—At present, people usually uses visual report design components to simplify the report design. But this means reducing the reusability of software development and working efficiency. Aiming to raise the efficiency and reusability of software developing, this paper introduces a new parameterized Report Builder idea, and also gives the concrete design and implementation for this report generator based on .NET component technology. In this paper, the basic design idea, primary data structure, format calculation and printing algorithm for the report generator are also discussed in detail. And an applied instance of the generator is also given in this paper.

Keywords-.net component; parameterized; report builder; format calculation; algorithm for printing

I. INTRODUCTION

In general statements, we designed a variety of interactive report component, such as AC Report, QReport Maker Chinese report component statements generated print control package and the famous crystal report component Crystal Report [1][2]. These statements of components, for people to provide a rich statements (Chart) to generate and print statements, reduces the complexity of software design. It makes statements programming like editing an electronic document as simple. But in the process of software development, these components also makes statement of the design process is very complex and personalized, report design and repetitive development work is still very great, serious impact on the efficiency of software development and maintenance efficiency [3].

At present, most of the reporting tool main working mode is based on graphics and interactive [4], the advantage of the method is reduced to the programmer in the algorithm analysis and design requirements, simplifying the design of report, but its main drawback is the report design personalized, not to report design reuse, increased. Design of the workload, reduce product maintainability.

Combined with practical project data used in the report, report model study of a simple [4] basic format and structure generation algorithm, and gives a non-interactive .NET component based design and implementation of the statement generator.

II. THE BASE IDEA

The base idea put forward in the paper is to design a case program by programmers, people can design report by run this program instead of using interactive report design method, and the report's format and contents can be providing to the report component object through the program in the form of parameters, the report system automatically generates and prints report by analysising the report data and the report format data.

This method is characterized by changing the interaction with the user, significantly reducing the user's workload during develop report program in practice, improving the efficiency of software development and maintenance. In a variety of application systems which use forms it has very important real sense.

III. A REPORT MODEL

People gives a number of different data models[5] about data report based on different classification. To simplify the issue, this paper only discusses the most common model that has tree-like list of table header as show in figure 1. Obviously, the list model consists of header, body and footer. Header is a description of reporting data structures data, it has the typical characteristics of hierarchy. Table is a collection of all data which has the same type in the report, the form of data structure and form are exactly the same. Footer is defined as multiple separate the rows collection of the table.

![Figure 1. List report model](image.png)

IV. DESIGN FRAMEWORK OF REPORT BUILDER

The report builder component framework based on the .NET component technology is as shown in Figure 2, its
work is driven by case procedure, each case program corresponding to a type of data report, design a report is design a case program. In usecase program, the user designs a process extracting report data from a database or other data source, and format into a form required by the report component. According to user's needs to design report (including the header, footer and text format, etc.), then submit them to the report component object.

**Figure 2. Framework of report builder**

**A. Design of Component Class**

Design class diagram of component as shown in figure 3. Class GeneralReport provides interface to users, and its main operation includes the establishment of statements, calculation, preview and print, etc., figure 4 shows its structure.

**Figure 3. class diagram of component**

**Class GeneralReport**

- A. Design of Component Class
- B. Design of External Interface Component
- C. Key Algorithm

**Figure 4. Class GeneralReport**

Class Document is the control class of the report component. It is used in various parts of an aggregate report to describe the report structure and generate algorithm. It consists of a header (Class TableHead), body (Class TableBody) and footer collection (Class TableTails), which respectively used to store the header, body and footer data.

**B. Design of External Interface Component**

Design of external interface components as shown in table 1. The user can according to the interface and the contents of a specific table, design a case program to implement the specific report, in practice, the workload of design, commissioning and maintenance such a program is far less than a interactive report design[6]. The problems may be further simplified if design a series of related classes according to the thought of design patterns [7].

**Table 1. External Interface Component**

<table>
<thead>
<tr>
<th>Serial number</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>void StartToPrint()</td>
</tr>
<tr>
<td>2</td>
<td>void PrintPreview()</td>
</tr>
<tr>
<td>3</td>
<td>TreeNode AddTitleItem( String Title, Boolean IsVertical, StringAlignment Alg, TreeNode Parent)</td>
</tr>
<tr>
<td>4</td>
<td>void FinishTitleItem()</td>
</tr>
<tr>
<td>5</td>
<td>void SetItemFormat( int index, int Weight, StringAlignment Alg)</td>
</tr>
<tr>
<td>6</td>
<td>DataRow GetNewRow()</td>
</tr>
<tr>
<td>7</td>
<td>void AddNewRow( DataRow r)</td>
</tr>
<tr>
<td>8</td>
<td>int GetColumnNum ()</td>
</tr>
<tr>
<td>9</td>
<td>void AddTail( DataTable t, DataFormatItem f())</td>
</tr>
</tbody>
</table>

**C. Key Algorithm**

Algorithm of the problem consist of the initialization data, format and print, key algorithms is format generation algorithm. Initialization section is used to populate data to header, body and footer of report, and it also used to populate formats data, that Lay the foundation for format generation calculation. Format generating algorithm used to calculate the position of each print item in the report. Report printing algorithm control overall print, such as paging control.

**Figure 5. Class DataItem**

MStartx, MStarty, MWidth and MHeight respectively meaning the start of x, y coordinates, width and height. These properties determine the print position. The calculation of print item width is based on the power distribution strategy. The user only needs to define the relative width of print item the in each data lines. Print item height is defined according to the item level, content of the print line number, character height, the row spacing index.

- **Generation algorithm of Report form**

To generate a report form, the first work is calculate form of table according to the input format data, the second work is to generate header format according to the table format, at last generation of end table format. The body is the core part of table.

The body consists of a plurality of table rows, each row of the table is composed of a plurality of data items. In this model, different table rows having the same data structure and format, height of table row defined according to the item level number, character height, the row spacing index of every row.

Header consists of the report header and a group of table items with the hierarchical structure, in which the leaf node set have same structure with table body. When to calculate
form. First work is to calculate the width of each table item by postorder traversal, the width of leaf nodes are assigned directly to table column width, corresponding the width of the branch node assignment into the sum of widths of lower layer node, and the height of each table item is calculated postorder traversal.

In this paper, end of table is defined as multiple independent table rows, and its generation algorithm are simila with header format generation algorithm.

generation algorithm of form is as follows:

set width of table is RegionWidth, it can be the difference of width of the paper and the left and right margins, can also be specified by user, width of each table column have the weights \(W_1, W_2, \ldots, W_n\), width of table column for \(W_1, W_2, \ldots, W_n\).

calculation table column width use follow formula,

\[
C_i = \text{RegionWidth} \times \frac{W_i}{\sum_{j=1}^{n} W_j} \quad (i=1,2,\ldots,n)
\]

Then correction of rounding error use formula as follow, \(C_n\) is the width of the last print item.

\[
C_n = \text{RegionWidth} - \sum_{i=1}^{n-1} C_i
\]

calculation of the body height of table line.

\[
H_i = \text{Font. GetHeight()} + 2 \times \text{Linespace} \quad i = 1,2,\ldots,n
\]

\[
H = \max_{i \in \{1,2,\ldots,n\}} \{H_i\}
\]

Printing Algorithm

The printing algorithms uses the event-driven printing model[8] provided by .Net. The main idea of algorithm is to create an instance of the Class PrintDocument, and set the properties describing of it, then its Print method can be called to start the printing process. PrintDocument's object will repeatedly activate the PrintPage event, until the user changes the complete flag of the current printing point to true. Accordingly, this print algorithm is designed to be the PrintPage event of the response process, it is clear that this process is a process of reusable.

In order to achieve report paging control, in the body and tail of report, an internal cursor is set up respectively, used to point to the first row of the table data which has not yet been printed, during printing we can achieve paging control by moving the cursor.

V. EXAMPLE OF APPLICATION

In the real software development projects, we has used the report component. Compared with interactive components, the efficiency and reusability of software developing has been considerably improved. Figure 6 shows the components of a specific application instance.

<table>
<thead>
<tr>
<th>Student number</th>
<th>Student name</th>
<th>Course 1</th>
<th>Course 2</th>
<th>Course 3</th>
<th>Total grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Zhi-dong</td>
<td>90</td>
<td>90.5</td>
<td>90</td>
<td>270.5</td>
</tr>
<tr>
<td>2</td>
<td>Jian-xue</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>270</td>
</tr>
<tr>
<td>3</td>
<td>Li-wei</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>240</td>
</tr>
<tr>
<td>4</td>
<td>Xiao-mei</td>
<td>60</td>
<td>85</td>
<td>95</td>
<td>240</td>
</tr>
<tr>
<td>5</td>
<td>Xiao-lin</td>
<td>85.5</td>
<td>90</td>
<td>90</td>
<td>265.5</td>
</tr>
<tr>
<td>6</td>
<td>Feng-xiao</td>
<td>70</td>
<td>70</td>
<td>69</td>
<td>209</td>
</tr>
<tr>
<td>7</td>
<td>Average</td>
<td>70.9</td>
<td>80.25</td>
<td>75.3</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 6. Application example of the Report Builder

VI. CONCLUDING REMARKS

Object-oriented programming techniques and .Net component programming techniques has been used in the design and implementation of This report generator, that enhanced scalability, and maintainability of the software, and also enhanced reusability of software.

From the perspective of simplifying the problem, this article only discusses the issue of list model design and implementation which with hierarchy header. For other types of reports, the method presented in this article also can be further modified and expanded. Undoubtedly, further research and analysis of these report models has a strong theoretical and practical significance.


