

The Universalization of Data Interaction Technology Research between Database and Spreadsheet of WORD Based on OLE

Qing-zhong JIA and Bin XIAO

Key Minist. Educ. Lab. of Dynamic & Control of Flight Vehicle, Beijing Inst. of Technol., Beijing, China

Keywords: Database, WORD, Data interaction, Template, Bookmark, Universalization.

Abstract. Existing document automation programs are often designed only for one WORD document and cure this document's parameters in the program. It is inconvenient to upgrade or change the document. This paper will combine program with WORD and use the template and bookmark technology in WORD. The document format and data exchange position which are variable will be placed into WORD, and the program retains only a small number of configuration elements which also can be edited. When document data do not change, user just modifies the template instead of program. When data change, only a few parameters need to be configured in program, and all other work can be completed in WORD. In this way, document automation upgrade or change will become simpler. Using above method not only reduces the difficulty of maintenance and upgrade, but also increases the versatility of program.

Introduction

Document management automation technology has been widely used. However, automated document management procedures are more or less curing document parameters. When to modify the document is required, it is unavoidable to change the original document automation program, which seriously affects the efficiency and increases maintenance difficulty. Therefore, the development of a universal document automation software has important application value.

Studies on the database and document data interaction have already been in full swing. Literature [1] generated documents controlled by program, the whole process of editing the document was written into the program. To do so, once the document format changes, procedure needs to be modified. Literature [2] was proposed by using the template form. This method assigns document format to the WORD (Microsoft Word) application to improve the versatility. Literature [3] did the data exchange based on template, but output data to report in fixed cell. Because of the data output position cured in the program, this way also results in the problem that we need to modify the program if the template is modified. Literature [4] mentioned bookmark application in document, the bookmark can help us quickly locate the target. Literature [5] used a combination of the template and the bookmark to complete data interaction. Although the literature puts forward the scheme of the database and document interaction, not mentions detailed design proposal or solution about how to exchange data if the template changed or cross industry using. The way talked in this literature also limits the universalization of the program.

Conventional document automation programs interact data between the database and document by using a combination of a fixed document form with a fixed process. This fixed way only suits for one table to interact, which will bring the problem that we have to change program for each report. This way severely limits the applicability of program. The nature of above problem is the general automation programs curing variable parameters in program. This paper will extract the variable parameters and propose a universal document automation solution.

Solution

The variable parameters of automatic document program is focused on both ends that the program connects with. One end interacting with the database reflects the data changing, the data that contains value and name will be set up, and an edit interface will be provided. Another end interacting with the WORD document reflects a change including the document format and the location of data, and we use template and bookmark to solve this problem. Then it forms the structure diagram as follows:



Figure 1. The program structure diagram.

Wherein the bookmark and the set of data interaction (later known as the tags library) are the interface linking database and document. During the interaction of the data, the program matches the tags library name and bookmark name, then the bookmark provides the location to exchange data. When the database data is outputted to a document, the template which also includes bookmarks is responsible for the document format, program parses template bookmarks and stuffs data into the document after the match with the tags library. When the document imports data into the database, the program gets data according to the document bookmarks and writes into the database after matching with the tags library. Overall framework contains two edit points, the tags library and the template, which are the configuration points of variable parameters, only if the two are unified together could the data interaction be completed by the automatic document procedure.

The document Automation program first needs to get control of WORD, this study will use the OLE programming interface to control WORD. Operations mainly include WORD open, read the table, access bookmarks, get bookmark position, read and write the cell, and WORD close. These actions can help us read bookmarks in the table of the template, and then complete data read and write.

In the user's point of view, program structure has two configurations: the tags library providing data and the template providing the document format and data location. But in the view of the procedure, it can be divided into two parts, the document format and data interaction. The document format is provided by the template, the program does not need to care about it except just filling data in the template. The remaining problem is data exchange. The data interaction consists of two parts. One part is the tags library in the program, another part is the corresponding bookmarks in WORD. The two parts complete data exchange by name match.

The program provides four-part interactive modules, including check template, format existing document, import document, and export document. The interactive structure diagram is showed in Figure 2:

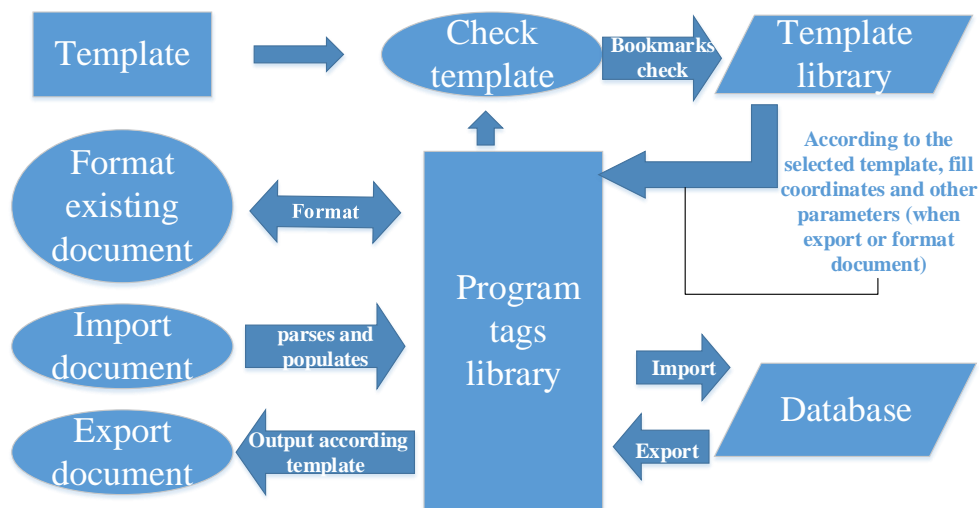


Figure 2. The interactive structure diagram.

As it can be seen from the chart, the program tags library is a reusable structure, as well as the core of the program. Each time the program is applied, the user first edits the program tags library. Then the user develops the document format, adds the appropriate bookmark in the corresponding position, and makes them into a template in WORD. Last the user use the program to checks the template. If succeed we can continue the document and database interaction. If data do not change and only the document format needs to be modified, the user just need to re-edit the template; if the data change, the user just simply edits the program tags library, and modifies the appropriate bookmarks in the template at the same time. According to the above operations, users can complete document automation upgrading or changing.

The following article will detail implementation plan from program point of view.

The Operation of WORD

Fundamental

This study uses OLE technology to operate WORD. OLE (Object Linking and Embedding) is a Windows system process interactive communication mechanism, which, in fact, is a programming interface. The OLE automation can link or embed the object of the program such as WORD into the application system. WORD is an automated server, which provides many programmable OLE Automation objects, common automation objects have CApplication, CDocuments(Collection of documents), CDocument, CSections(Collection of sections), CSection, CSelection(Cursor selection), CRange, CParagraphs, CTables, CTable, CRows, CRow, CColumns, CColumn, Cells, Cell, CBookmarks, CBookmark etc.

Concrete Implementation

Using the OLE programming interface, we encapsulate some common functions of the WORD operation as a base class. This base class will be used as an interface which the program uses to operate the WORD document. The commonly used functions in this paper are summarized as in the follow table:

Table 1. Basic class member function.

Member function name	Functional Description
CreateApp()	Create the WORD process, obtain the Application, Documents Object.
CloseApp()	Close the WORD process, release resources.
OpenDocument()	Open or create a document, obtain the Document object, initialize Tables, Selection, and other resources.
CloseDocument()	Save and close the document, release all resources

SelectTable()	Select one of the tables in the document, get the Table object
GetTableBookmarks()	Get all bookmarks from the table
AddTableBookmark()	Add a bookmark to the appropriate location in the table.
SetTable_GetCellText()	Get the text of the corresponding position of the table
SetTable_WriteCellText()	Fill in the text at the corresponding position of the table
GetBookmarkTablePosition()	Get the location of the bookmark in the table

Data Interaction

Basic Data Structure

Program Tags Library. The Program tags library is written in the program, used to match the data tags and document bookmarks. The program tags library is a multi-level structure, mainly includes the tags set of table. The base unit includes tag name, cell ranks number (hereinafter referred to as coordinate), and interactive value. The basic unit set (contains a table characteristic parameter) consists of single-level tags set, then the collection of single-level tags set is composed of the all-level tags set. Wherein, the table characteristic parameter is a special bookmark for the table identification.

The initialization of the tags library only needs to edit tags name, other parameters would be filled by the program. The program tags library is a reusable structure. It is the interface between the template library and the document, the coordinates and other parameters will be extracted from the template provided by the user, for the subsequent data export and formatting the document. It is also the interface between the database and the program, when the database is imported and exported, it is needed to interact with the data in this structure.

User Template. This part is operated entirely in the WORD software, the user provides the WORD document template as required. The user can customize the document format, the output document will follow the template style. The template is a collection of sample tables, the name of each bookmark must correspond to the program tags library. After adding bookmarks and making the template, there will be a validation program for the template, only the template that meets requirements can be imported into the template library.

Error Bookmarks Collection. If the Template verification is completed, it will generate an error bookmark collection, which consists of three parts: all matched levels error bookmark collection, template unrecognized tables collection and unmatched levels collection. All matched levels error bookmark collection contains a set of differences between the template bookmarks and the program tags library. Template unrecognized tables collection and unmatched levels collection are a pair of the different set of levels and tables.

Error Bookmarks Correction Collection. Error bookmarks correction collection is a return structure, which contains the original name and modified name of the bookmark to be modified.

Function Realization

User Template Check and Correction. To interact data depends on matching the tag name and bookmark name, the correspondence between the two is done manually, there will inevitably be errors. So, after inputting the program tags library's information and creating a template, checking alignment is essential. The function includes three parts:

Check template: First, each table uses the table characteristic parameter matching with level, if the level matching fails, the table will join unrecognized tables collection, after all tables recognizing, unmatched levels join unmatched levels collection. If the match is successful, then the program matches all the bookmarks in the table and all tags in the program tags library, after the match, the two remaining things are added to all matched levels error bookmark collection.

Error bookmarks correction: The function provides a user interface, which renders the template and program tags library different bookmarks. The user chooses to re-match, the program fills error

bookmarks correction collection. After the user is determined, the background corrects all bookmarks in the template.

Template library management: The export and delete function of the template are provided here, which is convenient for the user to view and manage the template library.

Format Existing Document. This feature (for document data import later) is mainly for the convenience of the rapid application for large-scale documents which are without the use of this paper's technology. The program formats the document in accordance with the template file, and adds all the bookmarks. Handling large table by the program can omit the manual operation.

This function requires the format of template the user selected must be consistent with the format of the document to be formatted. First, the user selects a template, the program will fill the coordinates of each tag in each level in the template tags library according to template. Second, the program uses table characteristic parameter to identify the table. Third, if the recognition is successful, the program adds all bookmarks in this level for the table.

Export Document. According to the content that the database need to output and the template that user select, the program outputs the database data. First the procedure reads the bookmark coordinate information from the template to the program tags library. Then the database fill the interactive value of each parameter which will be output in the table into the program tags library. At the same time, the program copies the corresponding sample table to the document in accordance with the template selected. Finally According to the coordinate information in the program tags library, the program outputs tag interaction value in the specified location of the table, while add bookmarks.

Import Document. According to the bookmarks in the document, the program quickly locates the data, assigns the value to the program tags library, and then calls the function import data to the database. This function does not require a template to participate and can complete the analysis of the work independently. To use this function, the document must be formatted or exported by this program.

Summary

This paper has solved the problems raised at the beginning of this paper. The integral structure has two configuration points: the program tags library and WORD template. When data does not change, the user just needs to modify the template. When data changes, both configuration points need to be configured, but the program only needs to modify the name in the program tags library without caring about the Office operation. As for other operations, the user only needs to modify the template in the WORD software. The above method can easily complete the modification or upgrade of the document automation program.

References

[1] YANG Rongqing. Using VBA and OLE automation technology to realize VB database report generation under OFFICE environment [J]. Computer Programming Skills& Intenance,2001,(09):19-27.

[2] YANG Shugeng, Ding Guolin. Producing Word Document Automatically in the Development of Database Application System [J]. China offshore Platform,2002,(02):43-48.

[3] XIAO Gang, WANG Hongen, WANG Changjian, SHAN Jihong, ZHANG Yuanming. Policy And Realization Of Data Exchange Based On WORD [J]. Computer Applications and Software,2004,(03):34-35+66.

[4] CHE Xiao-bo, WANG Zhi-xue, WANG Lei. Application of bookmark in word automatic operation [J]. Information Technology,2013,(08):148-150.

[5] XIONG Yu-rong, CHAI Yi, WANG Shu-juan, WANG Jia-qi. Word Document Automatically Generated Technology Based on VC++ [J]. COMPUTER ERA,2010,(01):52-54.

[6] LIN Gong. The data interaction study between database and WORD based on OLE and VBA [J]. JOURNAL OF FUZHOU UNIVERSITY(NATURAL SCIENCE EDITION),2006,(06):831-835.