

Research on Method of Rapid Software Development Based on Component Design

Hu Jingyi

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China
e-mail: 1005607919@qq.com

Wang Yaming

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China
e-mail: 616419803@qq.com

Lei Juchao

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China

Zhao Yalin

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710032, China
e-mail: zhaoyalin6987@qq.com

Abstract—This article presents a quick way to develop your system with custom components in Delphi. Use the above method can minimize code duplication, improve work efficiency. In this paper, we apply this method to the development of information management system in a university based on the Data Snap three-tier architecture. Through practice, we find that the workload of the original three months, and we only need 20 days to complete. Therefore, adopting the rapid software development method based on component design proposed in this paper can not only shorten the programming time of management systems such as ERP and information management by more than two thirds, but also bring great advantages in system development.

Keywords—Delphi; Custom Components; ERP; Efficiency; Time

I. INTRODUCTION

With the advent of the information age, we are more and more inseparable from technology. Now developed a lot of PC software such as information management systems that can help people save a lot of manpower, material resources. Especially every year at the end of annual work statistics, the design of a set of software through the relevant settings for automatic statistical calculation, approval, access and other functions is necessary. The completion of the software can be fully realized at any time computer statistics automatically calculate the workload of a variety of teaching and research, but also can effectively avoid errors in the calculation process.

There are a lot of software that uses a lot of languages to develop the PC end now, but to the visual software development, this text uses Delphi to finish. And Delphi is a powerful, Windows-based environment, object-oriented visual application development tool. It combines the power of the traditional programming language Object Pascal with the database language for both traditional arithmetic programming and database programming. The Data Snap technology system in Delphi provides a mechanism for transferring database information between the client and the

application server. Meanwhile, the rich component library brings great convenience to the program design. Delphi has become one of the most popular visual software development tools today.

When we used Delphi to develop PC software, we developed a way to customize the components to complete the software development and make the program completed in three months and we only need 20 days to complete.

II. THEORETICAL BASIS

A. Practicality of Custom Components

In a system development, design a custom component. In the process of system development can be based on the analysis of needs sorted out the same part of the logic function, the use of custom components to complete the corresponding function. Therefore, anytime, anywhere custom components into different needs of the interface according to the corresponding function, reduce code duplication and greatly facilitate the development of programming speed.

In an ERP system to do needs analysis, we will find a lot of repetitive logic requirements, For example: doing computer college information management system, whether it is to deal with college business or teachers related to the operation, there are for the basic data to add, modify, delete and so on. That by designing custom components: Add button to complete the corresponding function, into the page you need to complete its specific functions and greatly reduce the programmer's efficiency.

B. Inheritance of Custom Components

Delphi component is an object library, but also a class. The base class for all objects in Delphi is TObject, and the most primitive base class for a component is TComponent, as shown in Figure 1.

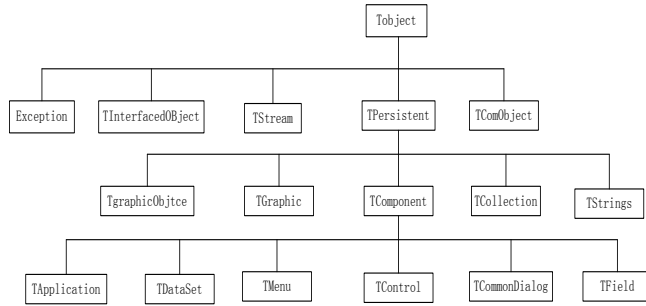


Figure 1. Example of a ONE-COLUMN figure caption.

The hierarchy of components in this structure is: TObject ----> TPersistent ----> TComponent ----> TControl.

TComponent, the base class for all Delphi components, inherits from this class for all object libraries that you want to register directly with the IDE for visualization. TComponent provides the necessary information to make the component work on Delphi's IDE. Then TComponent derives the TControl class, which is the base class for all flagship visual controls. All of our components are based on TComponent.

III. EXPERIMENT AND TEST

A. The Basis for Custom Component Development

In the development of custom components we need to analyze a page needs and then do some work:

- Data acquisition: In general, we are all custom components that design functionality, and we need to get the data or output the data. In each page there will be a lot of data-type components, how do we find the corresponding data components, you can design components by the naming rules to find the corresponding data-type components. For example: Set the settings in the component, to the components of a unified naming rules such as: cktext1_XXXX,XXXX for the database table field names, according to cktext1 can find the components needed, according to the XXXX components and database fields correspond.
- Data transfer: Passing data using JSON. JSON is a lightweight data exchange format based on a subset of ECMAScript that stores and represents data in a completely independent of the programming language. There are only two structures in JSON: Objects and Arrays JSON values consist of numbers, strings, logical values, arrays, objects, null, and so on. Using JSON as a data exchange format between the client and the server, the JSON data format can be used directly by the server, greatly simplifying the code development of the client and server, and easier to maintain. And the data format is relatively simple, easy to read and write, the format is compressed, occupy a small bandwidth.
- Mutual exclusion between functional components: When the button On Click is executed, in order to

avoid data confusion, you must make the button is not related to the button is invalid. For example, when you click the Add button in the information management system, the buttons, such as modify, delete, archive, data export, return, etc., must be inactive. The Save and Cancel buttons must be valid. 2 When the save or cancel button is clicked, buttons such as add, edit, archive and return should be valid, while save and cancel buttons should be inactive. Therefore, the entire information management system interface point of view, in addition to archiving, data printing, return no mutual exclusion, the other buttons need to be mutually exclusive with other buttons.

B. Associations Between Components in the Page

There are many custom components in a page, but we need to know the current step to the program flow, the current table, the current operator, the current name from the table, and others. Then we need to design a custom template components on each page, template components design attributes as shown in Table 1:

Template attributes	Explanation	Template attributes	Explanation
mb_lcmc	Process name	mb_prior_step	Step number
mb_now_step	The current step number	mb_next_step	Next step number
mb_tablename	The current table name	mb_prior_tablename	Pre-step name
mb_lb_field	Category field	mb_lb_name	Category name
mb_cb_tablename	From the table name	mb_next_tablename	Next table name 1
mb_next_table_nameb	Next table name 2	mb_next_lczttable	Next process status table name
mb_next_lbname	Next category	mb_next_lbfield	Next category field
mb_yzd	Data upload original field	mb_mbzd	Data download target field
mb_czr	Operator Name	mb_cjyh	root
	Template name	tag	Mark field
mb_lc_yesno	Is it a process?	Others as spare fields	

C. Implementation of custom component development

After the preparation of the above two parts, the process of custom components in Delphi contains the following steps:

- Create a library unit that contains new parts.
- Member has inherited a new type of component type.
- Add properties, methods and events.
- Register parts with Delphi.
- Create Help files for the component's properties methods and events.

Custom components such as adding components The main code is as follows:

Uses System, SysUtils, System, Classes, Vcl. Controls, Vcl. StdCtrls, Vcl. Buttons, Vcl. ComCtrls, DBGridEh, Math, Datasnap, DBClient, mysave, IniBoxes, mygd2wgd, MyIniDB ComboBox, MB, MyIni

ComboBox, Vcl. Graphics ;// The required unit file for creating the component.

```

RegisterComponents('d102', [Tmyadd]);
constructor Tmyadd.Create(AOwner:TComponent);
inherited;
self.OnClick:=myadd;
nowday:=FormatDateTime('yyyyMMddhhnnss',Now());
czsj:=nowday;
ssl:=inttostr(i);
lcbh:=czsj+ssl;
(self.Owner.FindComponent('MB1')
TMB).mb_lcbh:=lcbh;
lcmc:=(self.Owner.FindComponent('MB1')
TMB).mb_lcmc;
czr:=(self.Owner.FindComponent('MB1')
TMB).mb_czr; //Obtain name and operator letter via
template component
k:= Owner.ComponentCount; //Get the number of
components on the form
for j := 0 to k - 1 do
Begin
namey:=Owner.Components[j].ClassName;
if(namey='Tmyedit')then(self.Owner.Components[j]
as TBitBtn).Enabled:=False//Button exclusive event
else if(namey='Tmysave')thenif(namey='Tmysave')then
begin
(self.Owner.Components[j] as TBitBtn).Enabled:=True ;
(self.Owner.Components[j] as Tmysave).isadd:=True;
End
//Other mutually exclusive operation
(self.Owner.FindComponent('maindbgrid')
as TDBGridEh).Enabled:=False;
main_dataset:= (self.Owner.FindComponent('main_set')
as TClientDataSet);
main_dataset.Append;//Find the form on the main
interface with the data after the Query, and add records
main_dataset.FindField('lcbh').AsString:=lcbh; //New
process ID
main_dataset.FindField('czr').AsString:=czr;
main_dataset.FindField('czsj').AsString:=czsj;
main_dataset.FindField('lcjdmc').AsString:=lcjdmc;

```

IV. TEST

This experiment is in the computer system of information management system for experiments, the test results are as follows:

Custom Component Testing: A custom component of this article appears in the Delphi Visualization Toolbox: Figure 2

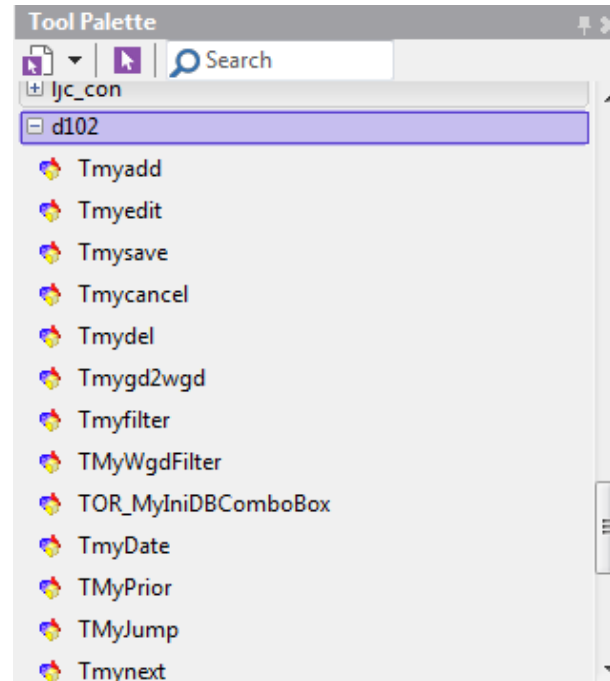


Figure 2. Custom component renderings

- In the page, drag the custom components to complete the corresponding function: Figure 3



Figure 3. System in a page renderings

V. CONCLUSION

According to the method of using custom component design in software development, the following advantages and limitations are obtained:

- Using the custom component technology, the management system model was initially realized, the idea of software reuse was realized, the phenomenon of repeated encoding was reduced, and the software development efficiency, maintainability and scalability were improved.
- The use of custom components in the information management system and the related operation of each page through the related parameter settings are feasible and safe.
- for the development of a set of software, through the development of custom components greatly reduce the development cycle, a great use of value to the actual development of software.

- This method of custom components in the development of software generally for the role of ERT system, for some web version of the software is not very effective.

ACKNOWLEDGMENT

Thanks to teacher Lei hard guidance, thanks to brothers and sisters, brothers and boys to provide help. Thanks for the correction of the expert group, to thank all those who helped me.

REFERENCES

- [1] Yang Changchun [ed]. Delphi Programming Guide Third Edition [M]. Tsinghua University Press, 2016. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [2] Gao Zhiming. Depth discussion of Delphi three-tier architecture [J]. Programmer. 2004 (09)K. Elissa, "Title of paper if known," unpublished.
- [3] Chen Dongsheng. System Analysis of Multi-tier Application Based on DataSnap Technology [J]. Computer Knowledge and Technology. 2013 (35).
- [4] He Dinghua. Three-tier client-server database application Delphi implementation [J]. Information Technology and Information Technology .2005 (02).
- [5] YIN Kaicheng, WAN Xiaodong. Application of Three-tier Distributed Structure Database Based on Delphi [J]. Tangshan University .,2003 (04).
- [6] Miaofei Hua, Zhou Qiaoshu. SQL Server 2008 database management system advantages [J]. Journal of Changchun Normal University, 2014 (06): 76-77 + 81.
- [7] Li Wensheng. Implementing Three-tier Client / Server Database Application Program in Delphi [J]. Computer Engineering. 2000 (07).
- [8] WU Xiaolin, JIANG Xian-gang, GAO Yanjin. Research on Connection Technology of Multi-tier Database Application System Based on Delphi [J]. Jia Dong Jiao Tong Da Xue Bao .2005 (01).