8. Case Study 1: A Book Management System

Here I give an example of using the downloadable architecture and the architecture transformation process in a WinBIS. Moreover, I hope the example can provide evidence to support the feasibility of such architecture and process.

8.1 About the System

This is a typical two-tier WinBIS (see Figure 3) which supports the management of books. The single client program of the system is written in Delphi 3.0. The database of the system, on the other hand, is managed using SQL Server 2000.

The client program provides fifteen functions that can be grouped into three categories: reader service, daily operation, and data management. From the form perspective, the client program is composed of ninety forms that can be classified into six types: splash form, main form, about form, function-specific primary form, function-specific secondary form, and function-specific design-time-only form. Additionally, each function is

implemented as a function-specific primary form, zero or more function-specific secondary forms, and zero or more function-specific design-time-only forms.

Although the system is not too complex, it possesses all the characteristics of a WinBIS. Further, keeping the example simple will help you understand the downloadable architecture and the transformation process.

8.2 Improving the Deployability of the System

In order to automate the deployment of the system, I go through the architecture transformation process (see Figure 5), to transform the client program from a Windows program into a program warehouse.

First, I make a home page, a menu page, and a message page for the client program. The home page splits up the browser window into three frames. The menu page is loaded into the left frame; the message page into the top-right frame; while the end-user selected package carrier is loaded into the bottom-right frame (see Figure 11).

Next, I make an ActiveX component (with Delphi 2006), an INF file, a component package, a package carrier, and DLL packages for each function of the client program.

Detailed accounts of the processes involved in making four of the fifteen ActiveX components are given below.

8.2.1 The Log In/Out ActiveX Component

The Log In/Out function allows an end-user to login or logout of the system. In the old system, this function is implemented as three forms: the log in/out form (primary form), the warning form, and the information form. The log in/out form is custom-made; the warning form and the information form are made with existing dialog subroutines. In the new downloadable system, this function is implemented as an ActiveX component, which is composed of an embedded form and two pop-up forms (see Figure 11).

To create the ActiveX component, I first create a new ActiveX component project. I then open the log in/out form, and copy and paste all its components and code into the embedded form (except the Delphi auto-generated code). I modify the global variable related code in response to the change from global variable to session cookie. Following this, I write code to refresh the message page. I then specify the file version of the ActiveX component. Finally, I compile and link the project to produce the ActiveX component.

The size of the package carrier is 1.60 KB; the size of the ActiveX

component's package is 24.47 KB; the total size of seven dependent common DLLs' packages is 1284.79 KB. Hence, an end-user will download 1310.86 KB of code, when he/she first uses this function, if I cannot pre-install the DLLs on his/her machine.
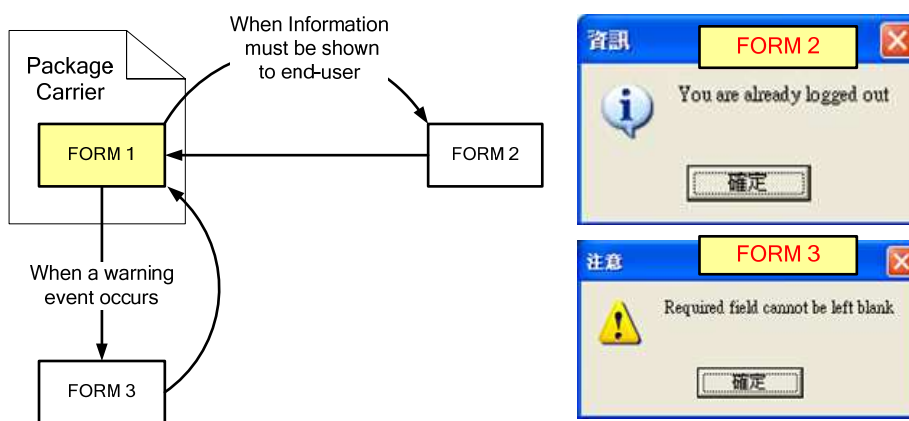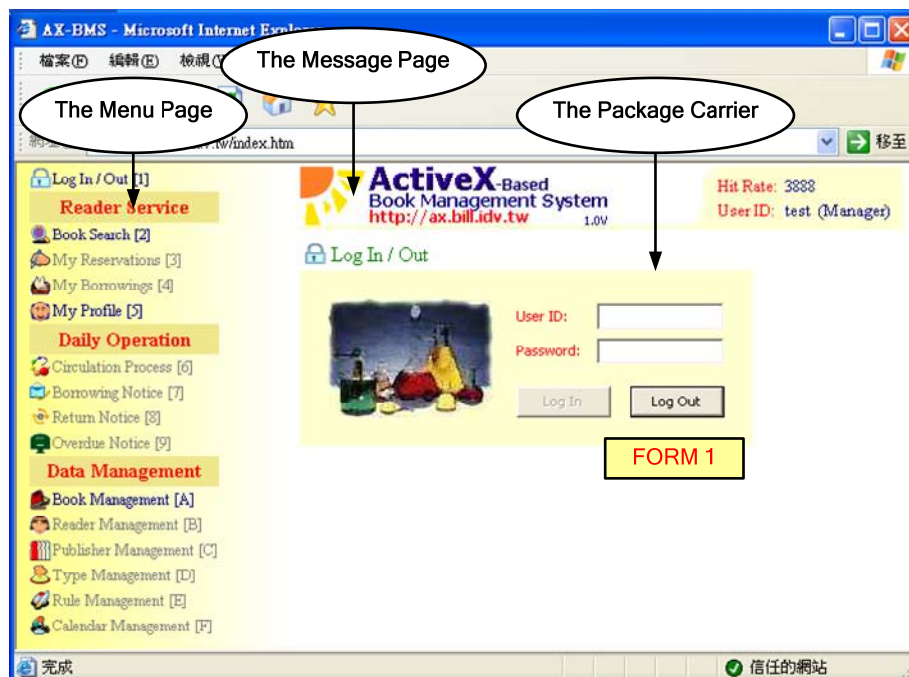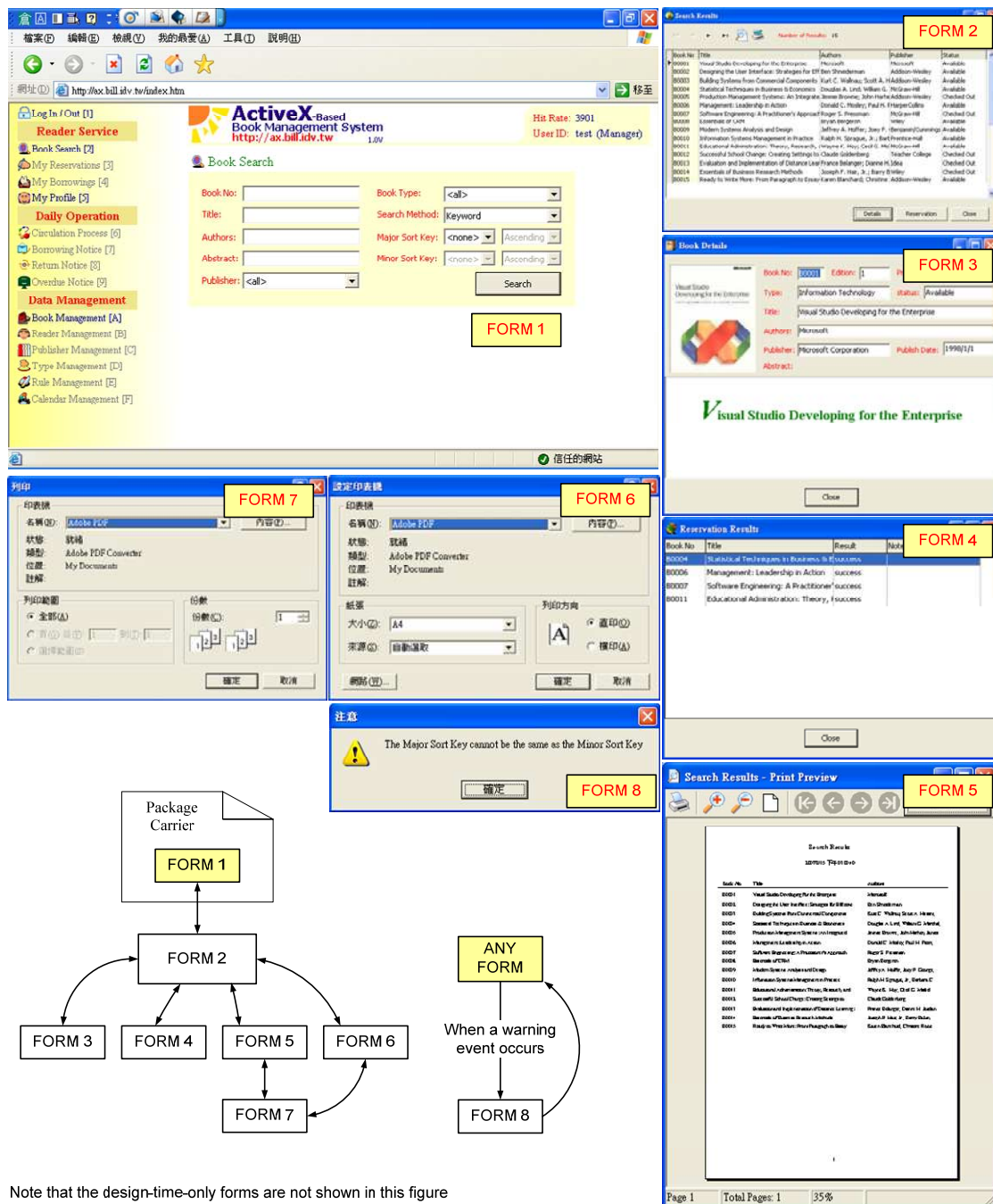

Figure 11. The Log In/Out ActiveX Component of the Downloadable Book Management System

8.2.2 The Book Search ActiveX Component

The Book Search function allows an end-user to search and reserve books. In the old system, this function is implemented as ten forms: the search form (primary form), the search results form, the book details form, the reserve results form, the print preview form, the print form, the printer setup form, the warning form, the data module form, and the QuickReport form. The print preview form, the print form, and the printer setup form are made with existing dialog components; the warning form is made with an existing dialog subroutine; the data module form and the QuickReport form are design-time-only forms; the remainder are custom-made. In the new downloadable system, this function is implemented as an ActiveX component, which is composed of an embedded form, seven pop-up forms, and two design-time-only forms (see Figure 12).

To create the ActiveX component, I first create a new ActiveX component project. I then open the search form, and copy and paste all its components and code into the embedded form (except the Delphi auto-generated code). I write code to check if the end-user is logged in (using session cookies), and if not, to disable the ActiveX component. Following this, I attach the two design-time-only forms and the other three custom-made forms to the project. I then specify the file version of the ActiveX component and finally, I compile and link the project to produce the ActiveX component.

Figure 12. The Book Search ActiveX Component of
the Downloadable Book Management System

The size of the package carrier is 2.16 KB; the size of the ActiveX component's package is 43.52 KB; the total size of seven dependent common DLLs' packages is 1284.79 KB; the total size of four dependent report-related DLLs' packages is 437.54 KB. The Log In/Out function must be performed before this function. Hence, an end-user will download 483.22 KB code, when he/she first uses this function, if I cannot pre-install the DLLs on his/her machine.

8.2.3 The My Profile ActiveX Component

The My Profile function allows an end-user to change his/her profile. In the old system, this function is implemented as five forms: the edit form (primary form), the load picture form, the export picture form, the warning form, and the confirmation form. The edit form is custom-made; the load picture form and the export picture form are made with existing dialog components; the warning form and the confirmation form are made with existing dialog subroutines. In the new downloadable system, this function is implemented as an ActiveX component, which is composed of an embedded form and four pop-up forms (see Figure 13).

To create the ActiveX component, I first create a new ActiveX component project. I then open the edit form, and copy and paste all its

components and code into the embedded form (except the Delphi auto-generated code). I write code to check if the end-user is logged in (using session cookies), and if not, to disable the ActiveX component. Following this, I modify SQL commands in response to the change from global variable to session cookie. I then specify the file version of the ActiveX component and finally, I compile and link the project to produce the ActiveX component.
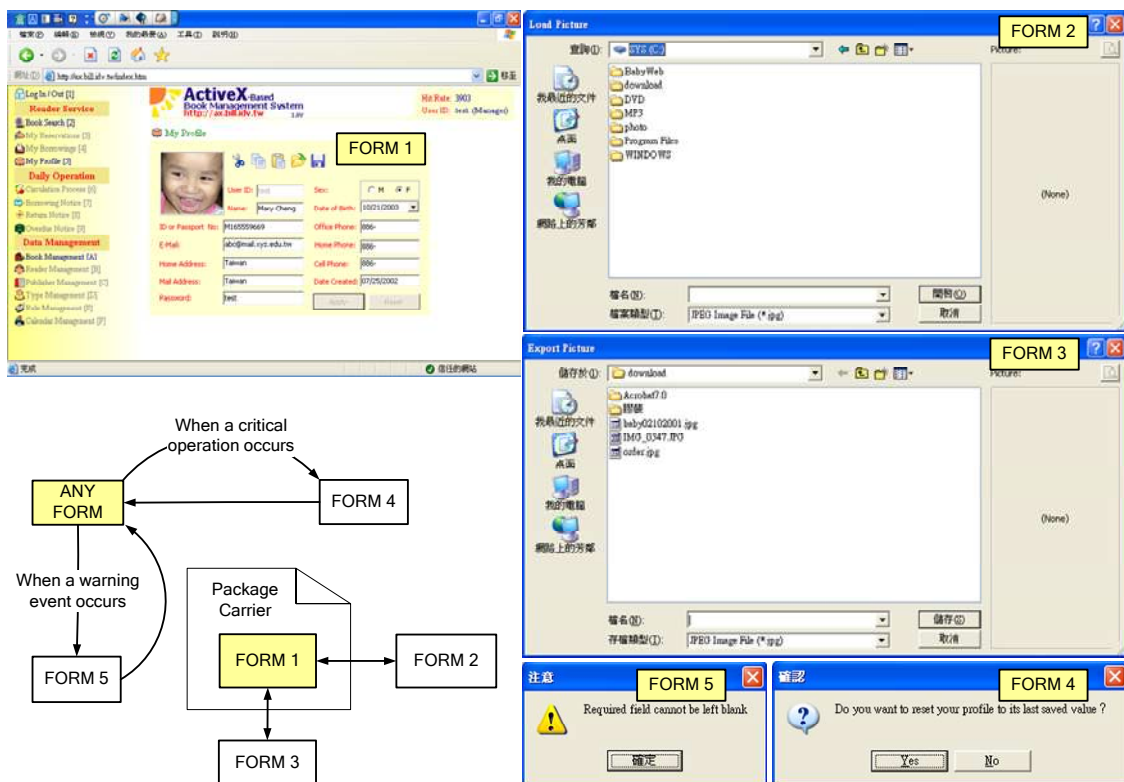

Figure 13. The My Profile ActiveX Component of
the Downloadable Book Management System

The size of the package carrier is 2.16 KB; the size of the ActiveX component's package is 24.81 KB; the total size of seven dependent common DLLs' packages is 1284.79 KB. The Log In/Out function must be performed before this function. Hence, an end-user will download 26.97 KB code when he/she first uses this function.

8.2.4 The Book Data Management ActiveX Component

The Book Data Management function allows an end-user to insert, delete, modify, and retrieve book data. In the old system, this function is implemented as fourteen forms: the search form (primary form), the search results form, the book edit form, the delete results form, the load picture form, the export picture form, the abstract edit form, the find form, the replace form, the font form, the print form, the warning form, the confirmation form, and the data module form. The search form, the search results form, the book edit form, and the delete results form are custom-made; the warning form and the confirmation form are made with existing dialog subroutines; the data module form is a design-time-only form; the remainder are made with existing dialog components. In the new downloadable system, this function is implemented as an ActiveX component, which is composed of an embedded form, twelve pop-up forms, and a design-time-only form (see Figure 14).
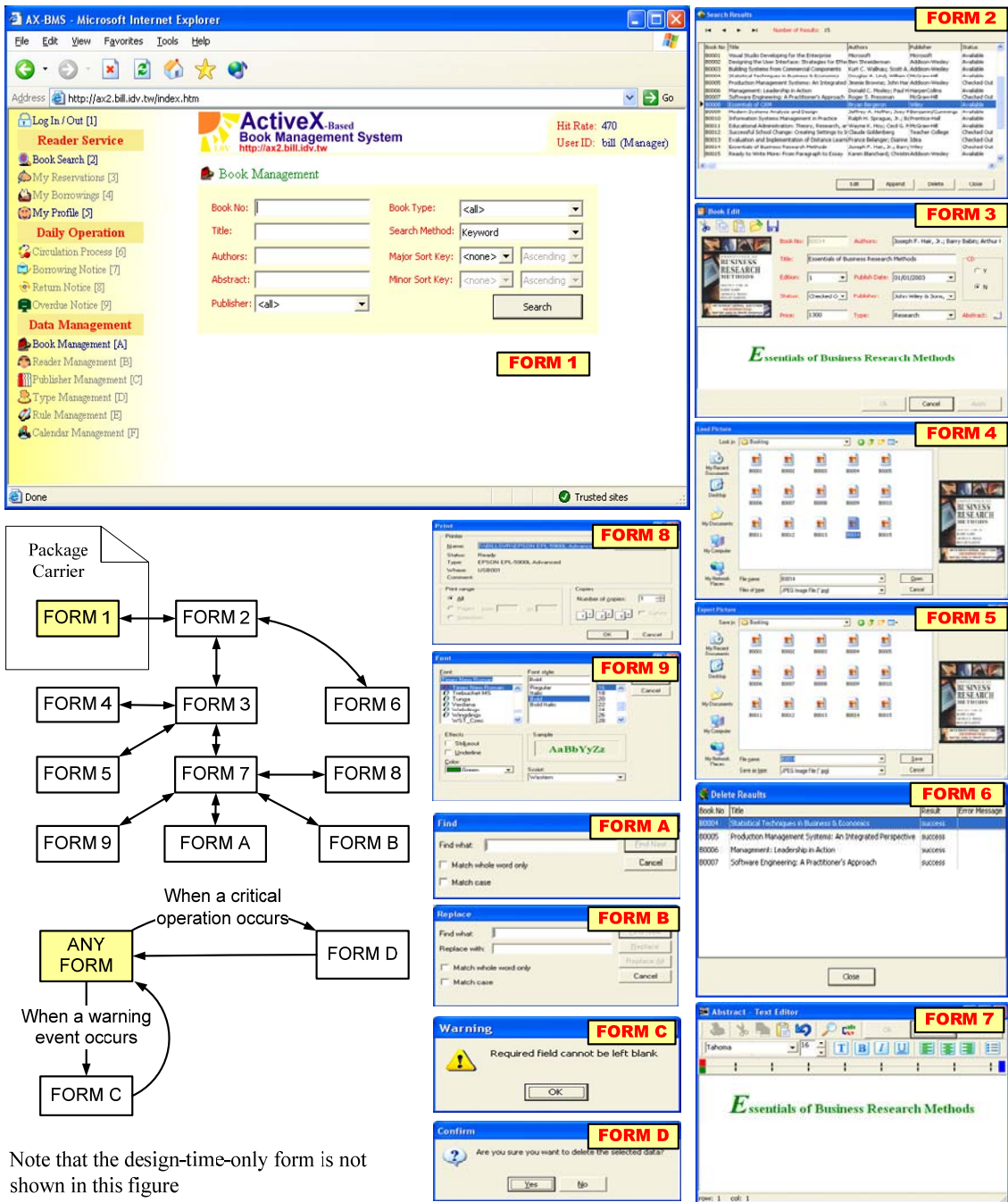
Figure 14. The Book Data Management ActiveX Component of the Downloadable Book Management System

To create the ActiveX component, I first create a new ActiveX component project. I then open the search form, and copy and paste all its components and code into the embedded form (except the Delphi auto-generated code). I subsequently write code to check if the end-user is logged in (using session cookies), and if not, to disable the ActiveX component. Following this, I attach the data module form and other three custom-made forms to the project. I then specify the file version of the ActiveX component and finally, I compile and link the project to produce the ActiveX component.

The size of the package carrier is 2.33 KB; the size of the ActiveX component's package is 47.29 KB; the total size of seven dependent common DLLs' packages is 1284.79 KB; the size of dependent text editor DLL's package is 17.82 KB. The Log In/Out function must be performed before this function. Hence, an end-user will download 67.44 KB code, when he/she first uses this function, if I cannot pre-install the DLLs on his/her machine.

8.3 Testing the System

You can test the downloadable book management system via the Internet. The URL is http://ax.bill.idv.tw/. The id is test; the password is test again. However, please note:

1. You have to configure the browser's security settings to enable ActiveX components and session cookies. Moreover, if behind a firewall, you have to configure the firewall to open the TCP port 2433 and UDP port 1434.

2. Downloadable BISs are restricted to use in Intranet or Extranet only. In fact, you will manipulate the downloadable book management system via the Internet. Bear in mind that file download speed and database access efficiency on the Internet are very different from the Intranet/ Extranet.

3. The Web server (Internet Information Services 5.0) and the database management system (SQL Server 2000) both operate in a humble environment: (a) AMD Athlon 1.0G CPU; (b) 768 MB SDRAM; and (c) 2 Mbps/256 Kbps ADSL. In general, businesses will provide more desirable operating environment for downloadable BISs.