

Distr.
GENERAL

CES/SEM.47/30
31 January 2002

ENGLISH ONLY

**STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

**COMMISSION OF THE
EUROPEAN COMMUNITIES**

CONFERENCE OF EUROPEAN STATISTICIANS

EUROSTAT

**Joint UNECE/Eurostat Seminar on Integrated Statistical
Information Systems and Related Matters (ISIS 2002)**
(17-19 April 2002, Geneva, Switzerland)

Topic III: Object-oriented technologies, component architecture

COMPONENT DEVELOPMENT AT STATISTICS SWEDEN

Contributed paper

Submitted by Statistics Sweden¹

I. VISUAL BASIC COMPONENT DEVELOPMENT

I.1 INTRODUCTION

1. The production environment for applications at Statistics Sweden used to be to in the mainframe environment. Today Windows is the primary platform and the applications that run are created using Visual Basic 6 (VB6). It is a great number of Visual Basic applications that has been converted for the Windows environment since the mid 1990s. Because it did take 4-5 years before the platform change was completed, those applications were done in VB3, VB4, VB5 and VB6. Today, almost every application has been converted to VB6. The quality of those applications differs a lot. The main factors that had an effect, on that in some times bad quality and lack of standards was:

- We had a time limit that said that all of the statistical surveys had to be moved to the new platform before year 2000.
- Statistics Sweden had a distributed IT-organization with IT-personal not sitting together, and often by themselves.
- The central IT unit in those days worked as consulting agency. The main business idea was not to work with standardization or help the rest of IT personal to build better solutions. Instead they also worked with application development.
- We also have the geographical Stockholm/Örebro issue. Statistics Sweden is located in two different places. It is not always easy to have a good contact with colleagues in the other locality.

¹ Prepared by Henrik Sandberg (henrik.sandberg@scb.se) and Markku Ruonakangas (markku.ruonakangas@scb.se).

- Individual programming skills differ also a lot. We also have employees that are part time programmers, and part time production personal.
- VB3 and VB 4 was not the easiest tool to use to build client/server solutions.
- The use of different versions of Visual Basic.

2. The positive things that did help the situation in some way is the fact that we have had our own Visual Basic training courses and that we developed one Visual Basic project template with standardized forms, dialogs, and menus etc. We also did get a name standard for Visual Basic programmers. That comprehends VB-files, VB procedures, components and all type of objects that you can use in Visual Basic.

3. Today, in 2001, we have taken a new step with the SCBWizard and the SCBComponents.

I.2. THE SCBWIZARD

A. Key features

4. The SCBWizard is a production tool for programmers at Statistics Sweden. It has been programmed in VB6. To use the tool the programmer must have VB6 (SP4) and MDAC 2.5 installed and registered on his or her machine.

5. When the programmer runs the SCBWizard it generates a VB6 project. The project type that is generated is Standard EXE. It also copies the latest versions of the SCBComponents to the local disc and registers them. The generated project consists of:

- Standardized forms and other VB-files
- MDI Window (Multiple Document Interface)
- MDI menus
- Ini-file for the application
- Template for a help-file
- References to the SCBComponents
- A standardized template structure for an Visual Basic application
- Standardized object names

6. The focus of the generated code by the SCBWizard is in the GUI (Graphical User Interface) and how to make database interaction, not in the structure of the application (business-tier, multi-tier etc). We have certain recommendations how a GUI should be implemented to make it easier for the end user to interact with the application.

7. The main reasons for the upgrade from the project template mentioned earlier to the SCBWizard are:

- The wizard benefits over the template in that it can generate a project dynamically in the following aspects:
 - The path and the name of the Visual Basic project
 - Names of the included files
 - What data provider to use (ODBC,OLEDB)
 - Editable menus
 - Include the SCBComponents, create templates of VB-files and include existing VB-files
- It's possible to save a template with the wizard which is based on the generated project.

B. How to use the SCB wizard

8. The wizard is available for all Visual Basic developers at Statistics Sweden on a central fileserver. Information of the SCBWizard is available on a Visual Basic homepage on the intranet. It is also included in our own Visual Basic training courses. It has also been introduced, as the SCBComponents, in seminars at Statistics Sweden.

9. The SCBWizard should be used when the programmer wants to begin a new Visual Basic project. The purpose of running the SCBWizard is to generate code. The generated code consists of some part of a code that's ready to use and some part that is a template. The template parts are where the programmer has to start coding and to build the application.

C. Conclusions/experiences

10. The SCBWizard is quite new in the organization at Statistics Sweden. The first version was shipped in autumn 2001. It has been used in quite many projects already and the experiences are very good. It's easy to use and it has been proved to be stable. In fact the development team did a lot of tests on the SCBWizard before shipping. It was also tested on Windows 95, Windows NT and Windows 2000 platforms.

11. The next step to the SCBWizard is that it should be possible to use it to generate multi-tier applications. As it is today the programmer has to make ActiveX-dll projects from scratch. We think that a multi-tier solution is the right way to architecture most Visual Basic applications at Statistics Sweden. If you choose that type of solution you can easier develop and maintain the system because it's divided into logical separated parts.

12. But of course it requires more skills from the programmer to do a multi-tier solution. To do that type of solution, as it is today, you must have some understanding of COM (Component Object Model) and have done a good architecture of the system before starting the coding process. Most programmers today at Statistics Sweden are not used to work with multi-tier solutions.

13. The future of SCBWizard is probably to migrate to the .NET platform and to generate code for VB.NET. When this is written Statistics Sweden has no experience of the .NET platform (it is not been released yet). Because of the good experiences of the SCBWizard and the Visual Basic project template, mentioned earlier, we think that we also will have this type of tool for the .NET platform.

I.3. THE SCBCOMPONENTS

A. Key features

14. The SCBComponents are COM components developed in VB6 at Statistics Sweden. The key feature of the components is to encapsulate functionality. Visual Basic programmers, Office programmers and other environments that support COM can then use the components. The components can be used without the SCBWizard but if you use the wizard they are automatically generated into the Visual Basic project.

15. The SCBComponents can be classified in 3 categories:

- Dll-files that encapsulate functionality like file interaction, communication with databases, communication with the registry etc.

Every programmer often uses this type of functionality, and it can in many times be done in many different ways. By using these components we can receive a faster development speed and more standardized solutions. The programmer does not need to spend a lot of time searching and testing how to implement a certain functionality.

- Dll-files that encapsulate Microsoft Excel, Microsoft Outlook.

Despite the fact that it is quite easy to program against the object models of this type of tools you often have to write many lines of code to do certain things. This type of components encapsulates code lines that should be used together. In this way we can receive a faster development speed and more standardized solutions. And the programmer doesn't need to spend a lot of time searching and testing how to implement a certain functionality.

- OCX-files, that are standardized logon dialogs.

A logon dialog can normally differ in appearance despite the fact that the functionality is the same. Those components can take parameters like name, password, database, servers etc. In this way we can receive more standardized solutions and the result is that it is easier for the end user to interact with the dialog.

B. How to use the SCBcomponents

16. The components are available for all the Visual Basic developers at Statistics Sweden when they run the SCBWizard. As mentioned earlier, it copies the latest versions of the SCBComponents to the local disc and registers them. Information of the SCBComponents is available on a Visual Basic homepage on the intranet. Each component also has one own help-file. The help-file is an html-file that is installed when you run the SCBWizard.

17. Of course the end users must have the same components on their machines. Therefore, SCBComponents is a part of a standard installation package for new computers at Statistics Sweden. Newer version of the components must today be installed manually. The programmers must help the end users with the installation process. An installation package is available on a central fileserver. This type of upgrading processes will be automated when Statistics Sweden upgraded the client computers to Windows 2000.

C. Conclusions/ experiences

18. The SCBComponents were quite newly introduced in the organization at Statistics Sweden. The first versions were shipped at autumn 2001. They have also been used in quite many projects already and the experiences are very good. They have been proved easy to use and they have also been proved to be stable.

19. The only issue so far is the upgrading process, but we think that should be easier to handle with automatic installations packages, when we have migrated to Windows 2000 on the client computers.

20. Despite the lack of an automatic installation process the availability of the installation package makes it quite easy to upgrade to a newer version. Of course the newer versions must be binary compatible with the first version. But that is no problem because the developers are not allowed to break the compatibility. The components have also been developed in a Microsoft recommended style: they use interface classes implemented by co-classes. In that way it is easy to have more standardized names on methods in interface

classes and then have different implementations in the actual component. It is easy to maintain that type of component.

21. An important factor is of course that the programmers must know the existence of the components and how to use them. That can be a problem in a decentralized organisation like Statistics Sweden. We have solved it by using those components in our own training courses. That part is working well. The problem is to receive feedback and suggestions for new functionalities and for new components. Therefore we are trying to start up a Visual Basic network at Statistics Sweden. We do hope that this can give us more ideas and suggestions for future components.

22. .NET is of course a big issue that will change our way to build components. We think that these components that we develop with COM technology today, will soon be developed in .NET.

II. WEB DEVELOPMENT

II.1 INTRODUCTION

23. Since the middle of the 90's Statistics Sweden has been an actor on the Internet. At the beginning only statistical output was presented. The pages were static and implemented in HTML².

24. In the late 90's the Swedish government made it easy and cheap for employees to hire a computer from their place of work. That way many computers ended up in Swedish households. The government also invested a lot of money in broadband for the Swedish society. Those initiatives increased the importance of the Internet for Statistics Sweden.

25. In the late 90's dynamical services were added to the website. To do that static HTML pages wasn't enough, applications had to be created. These services were implemented for the Microsoft platform. The first applications used idc and htx files to fetch data from databases. Later ASP³ was used for the same action.

26. The ASP code was a mess. No application was implemented like another. Often there were Visual Basic-script, SQL, HTML and JavaScript in the same file. It was hard for the developer to control database connections and it was almost impossible to have an influence on performance and scalability. Forms and logic were in the same piece of code, so there was almost no chance for update or reusability. What we needed were guidelines for how to create good web applications.

27. In the beginning of the new millennium we started to look at the Windows DNA⁴ platform. At Statistics Sweden we have a lot of Visual Basic programmers so if it was possible to use the same development tool for web applications as for client/ server applications it would be great. Cap Gemini Ernst and Young were consulted to help us.

28. The collaboration resulted in

- A framework.
- A training course in Windows DNA 2000 and the framework.
- A wizard that support implementation of a new application.

² HyperText Markup language

³ Active Server pages

⁴ Distribute Internet Architecture

- A statistical survey as pilot project.

II.2 FRAMEWORK

29. The framework is designed for component-based development on the Microsoft DNA 2000 platform. The components require COM+, which is a part of Windows 2000. The framework is also designed for:

- Performance and scalability.
- Making it possible to reuse code and components.
- Making it possible to rotate developers.
- Standardisation of the web application development.

30. The framework is a multi-tier architecture. Presentation, business rules/logic and data are separated. The logic is implemented in Visual Basic, which makes it easy to debug the application during development. It is not necessary that the developer of the logic also design the layout of the web pages.

II.3 WIZARD

31. A wizard, based on the framework, is implemented as an add-in for Visual Basic. The wizard helps the developer to start a new application and speeds up the initial phase of the implementation part of a new project. Depending on an existing database model it generates a skeleton for components in Visual Basic.

32. The wizard also generates:

- Test forms for the generated components in Visual Basic and ASP.
- A folder hierarchy to store project files and binary compiled files in a standardised way.
- Common code, e.g., error code, uses the Statistics Sweden name standard.

A. How to use the wizard

33. The wizard is meant to support the developer. The main purpose is to help the developer to implement the framework. That means that the developer doesn't necessarily have to use the wizard. The important thing is that the application implements the framework. Sometimes it's easier to copy an old project than to generate a new one.

II.4 EXPERIENCES

34. So far only one project is implemented this way. There are others being implemented at the moment of writing this paper. The experience so far is that:

- Most developers think it's great to increase their skills in Visual Basic development and to see other ways to implement applications.
- The developers don't have to start from the beginning because they are already familiar to the development tool and the syntax.
- The framework makes team oriented development much easier because the application is developed in a standardised way.

35. However, some issues have to be taken into account:

- Most developers at Statistics Sweden are used to develop two-tier applications with most of their code in the database. Therefore many developers find it difficult to implement multi-tier applications
- Some developers don't like that they have to use a certain design pattern.

II.5 NEXT STEP

36. The next step is to upgrade the framework and the wizard to support the .NET framework. Of course action has to be taken to new experiences when other applications are finalized.