GUI Migration Case Study: Aucon BV

By Peter Scholz

hen Aucon BV, a Dutch software consulting company specializing in computer automation for the automotive industry, first contacted BASIS and my company, Soft Service Computer GmbH, we spent almost six months just in preparation for the migration. We had to define and agree on the goals of the project and the possible ways to achieve those goals.

The Approach

The initial approach that Aucon had in mind was to first do a one-to-one conversion of the programs from Siemens Nixdorf (SNI) Basic to BBx®, resulting in a character-based application that would run under PRO/5®. Then, once the reliability of PRO/5 was proven, the character-based programs would be migrated to GUI.

Very soon, we all realized that converting the application and creating a new graphical user interface simultaneously would have several advantages. From a marketing standpoint, we could satisfy new and existing Aucon customers. The current customer would retain the functionality of the character-based program but would have an upgrade and support path that was no longer available from Siemens. (Siemens had decided to kill any further development of the Nixdorf application.) And the new prospective customer gets the GUI interface he or she wants.

Also, it was advantageous from a technical viewpoint because the SNI Basic handles screen input/output differently than PRO/5, which would have meant extra effort to simulate the old SNI behavior.

We defined a project team and assigned specific tasks and areas of responsibility based on expertise. One engineer is responsible for designing the screens. Another is a programming expert on the old Nixdorf dealer management system. His responsibilities include analyzing, adapting and implementing the Nixdorf business rules into the new application, dubbed Easy Car Connect (ECC) by Aucon. A third engineer came into the project to concentrate on the help system. Other tasks include the maintenance of a library with common subroutines and user-defined functions requested by the programmers.

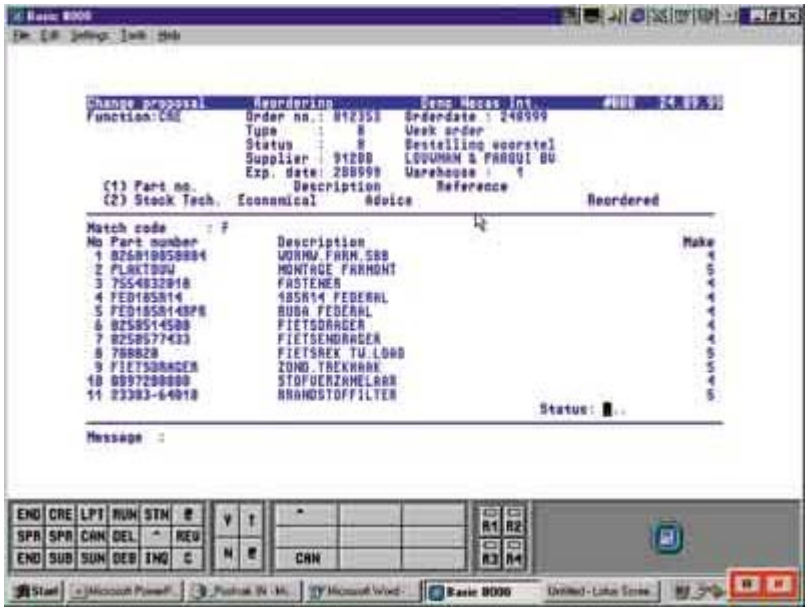
We also had to establish standards and create style guidelines and naming conventions for field names, variable names, labels and user-defined functions. Although as yet we don't have a tool that automatically enforces these conventions, merely having a convention for every sort of object helps in finding easily remembered names - an important feature in a major software project. If the standards are intuitive and well documented, it is easy to enforce them.

Of course, having the team agree on the standards also helps. The hours we spent in proposing and discussing our standards were well invested because people tend to follow the rules they have set themselves. Keeping good documentation about what's in the library is important. Also important is setting up a system by which your programmers can file function requests that can be fulfilled by a dedicated team member. This is much more efficient than having every programmer write his or her own function set.

The Tools

Once we had some standards defined, we then had to choose the tools. We automated the conversion of the Siemens Nixdorf Basic to BBx by building a conversion tool, SNIPit! Using SNIPit!, about 95 percent of the code could be converted automatically before we began designing the interface.

The introduction of the new grid features within Visual PRO/5® 2.0 had a strong influence on our decision to do the migration project with BASIS products. The implementation of the grid within the language allows for a lot of possibilities, far more than in many other languages.



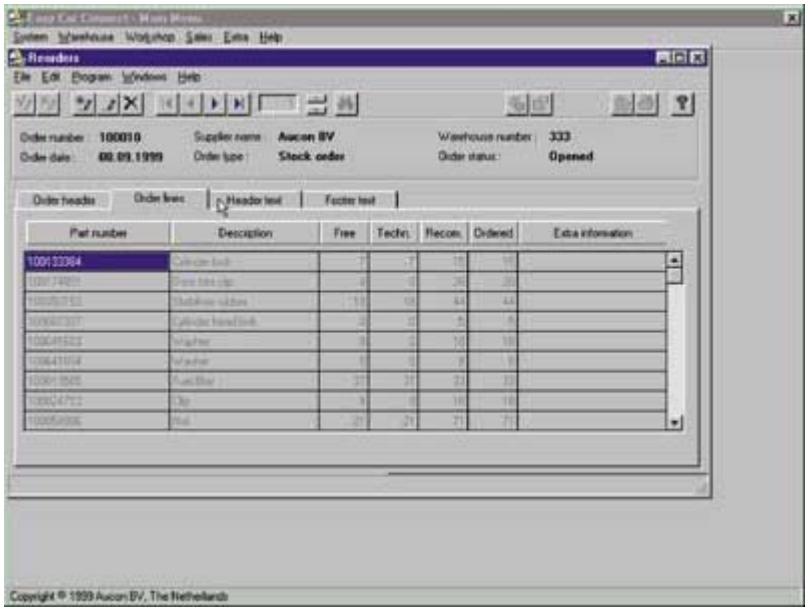
A Part Reordering Screen From the Old Character-Based Siemens System.

To facilitate the use of the grid objects, we are using a Grid Management Library, developed by SCS Consulting Services Inc. in close cooperation with the Aucon project team. This Grid Management Library is now available through BASIS. With these library routines, the programmer can easily manage the contents and appearance of the grid cells without the need for cryptic Sendmessage or Notify parameters.

We used Resbuilder™, a tool in Visual PRO/5, to design the screens. This was an easy decision because the current version has a lot of features that support uniform-looking resources, for example, the ability to set default properties for all objects. To support the multilingual capability that we need, the resources will be converted later to ASCII resource files and then fed through a translation parser to create a new set of resources for different languages.

We decided that our main programming tool would be GUIBuilder™, the development workbench in Visual PRO/5. It frees the programmer from maintaining events for objects within the resource. With a click of a button, a programmer can choose a resource and start maintaining the code that he or she wants to implement for given events. We were also influenced by the fact that GUIBuilder keeps the code in ASCII files.

As the editing tool, we chose TextPad from Helios, a company based in the United Kingdom. TextPad offers a lot of support for the programmer, like elaborated grep-like search-and- replace routines, a clip file feature that allows us to paste Visual PRO/5 keywords and phrases into our program text, automated



A New GUI Screen For Part Reorders in Aucon's Easy Car Connect Application, Built Comletely With Visual PRO/5

indentation, bracketing and even syntax highlighting. Another strong feature is the ability to define customized commands, so the programmer can compile and test a Visual PRO/5 program right from the editing interface. If you want to take a look at the TextPad tool, visit: www.textpad.com

The entire Easy Car Connect migration project involves about 500 program modules and accompanying files. To facilitate tracking changes to the modules and who is working on which module, we use Visual Source Safe, a source code development application.

Automating the Migration

Moving from linear, character-based code to event-driven GUI screens meant that we had to create a lot of drag-and-drop operations to implement the former programs' business rules in the events and functions predefined by GUIBuilder. This was where the graphical user interface of the TextPad editor was a relief.

SNIPit! was enhanced to eliminate line numbers from the original SNI code and to replace line number references by symbolic labels. That made it easy to implement the converted code in the appropriate place within GUIBuilder's predefined GBF structure.

We added some elements to GUIBuilder to automatically generate some features in the programs. Several, for example, facilitate testing and debugging. At run time, these features can be switched on and off by the use of global variables, making it easier to test and support the application. We added some mechanisms for controlling and logging program use, and we added a preprocessor to allow C-like features, like #define, #include and multiline IF statements for indentation, to make the code more readable and easier to maintain.

We've completed about a third of the total project so far, having migrated all the parts- and inventory-related programs to GUI. The response from Aucon's customers has been overwhelming. They want the rest of the migration to take place immediately. Right now, we anticipate a September 2000 release date for the entire Easy Car Connect application.

Peter Scholz is president of Soft Service Computer GmbH, a German software consulting firm, and the creator of SNIPit!, a conversion tool for Siemens Nixdorf Business Basic.