December 2001

A Set of Components to Include Security, On-line Translation and Other Features in Information Systems

John Bauer-Mengelberg
Colegio de Postgraduados, Mexico

A SET OF COMPONENTS TO INCLUDE SECURITY, ON-LINE TRANSLATION AND OTHER FEATURES IN INFORMATION SYSTEMS

John Bauer-Mengelberg
Colegio de Postgraduados, Mexico
jbauerm@colpos.colpos.mx

Abstract

The most important attributes of reusable components are probably good and extensively applicable functionality, and the ease of use by its potential users. Productivity and applicability to several platforms usually dominates research issues, as a result of the increasing demand for very powerful components to reduce costs and decrease labor needs. This project places emphasis on the selection of components meant to empower less experienced or knowledgeable developers to include features which will improve their systems. Protecting a system and its data against unauthorized use by the system developers and other IT personnel is a growing concern which also played a role in the design of the components and the distribution scheme. Market considerations were never part of the selection process. The toolset was transformed from the author's personal productivity tool into a teaching device. The only one of its modules which is described here is the methodology which makes it possible for an end user to translate the finished system, interactively, into any other language. Translation is done without recompilation or access to the design or source code of the system. It is not just another way to distribute the product in several languages, though it performs that task. An example outlines this process, and some technical information is included about the method. Translation of messages and table values or catalogues is performed interactively in other modules, even though their main objectives are other issues such as productivity, cost reduction and quality of the systems. The modules do not address the main functions of a system, such as data update and usage, although they often support them in some ways. They concentrate on complementary functions, such as tight data security and confidentiality, controlled access to functions and effortless ways to associate multimedia materiels to objects.

The Toolset

The most important attributes of reusable components are probably good and extensively applicable functionality, and ease of use by its potential users. Productivity and applicability to several platforms usually dominates research issues, as a result of the increasing demand for very powerful components to reduce costs and decrease labor needs. This project, on the other hand, places emphasis on the selection of components meant to empower less experienced or knowledgeable developers to include features which will improve their systems.

A set of tools meant for system developers and Information Systems students is being designed and implemented, and will be distributed free of charge essentially as a teaching aid. Its functions have proved to result in savings and increased quality of many systems. It will probably be delivered in an electronic book format, or at least as a voluminous manual. Its name JBMINFRA refers to its functions as the foundations of the systems, their infrastructure. Its components could serve as examples or suggestions to be used, totally or partially, in systems to be built. In most cases they are complete subsystem components, not merely software. The design of the detailed functionality plans the very likely partial use of the facilities or models. If this were not very carefully planned, it could easily lead to violation of the crucial demand we make on our components: that their users will be able to use them *well*. Flexibility is another complicating factor. Greater flexibility can increase the number of potential users, but should do so without decreasing the number of real ones, especially through complication or confusion.

The definition of the features which are included in the toolset is based on many years of constructing systems and evaluating them from different points of view. Diagnostics of many systems that failed or were constantly in trouble showed that often this was not due to faulty model, function design or programming. We focused our attention on the other factors which could increase risks. Although we have searched the literature many years we have failed to find papers documenting similar projects, meant to help many developers who have little experience in certain facets of their future systems. The huge market for such aids of any type suggest the convenience of investing some effort in the matter. Many of the package's modules have benefited from published research; however, we found no help at all for the translation module we shall describe. This single module will have to point out some general objectives of the toolset as well as somehow outline its types of solutions and functionality.

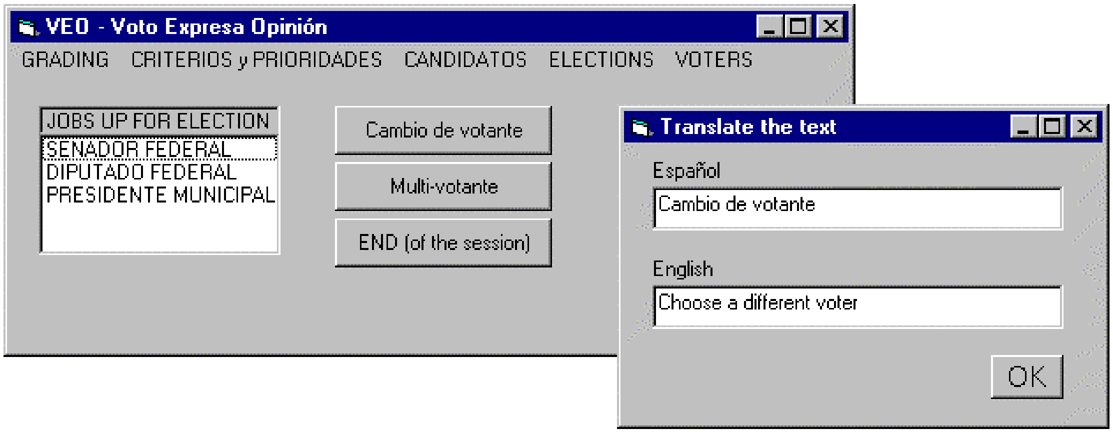
Interactive Translation of a System

Regarding the translation of a system, the function addressed by this toolset is often confused with other functions. Internationalization, or producing versions in order to distribute a system in several languages, is performed in-shop and then packaged with tools of the trade. Automatic translation is of course something totally unrelated to system translation in this sense. Though both are widely researched areas, they differ from the possibility offered to any user of the system, to produce a version in any language he chooses – with the obvious alphabet restrictions since we do not offer a variety of these. No technical knowledge nor tools are necessary.

Translation into other languages is not the only use of the module described: many systems may profit from different versions of certain texts and messages. First name basis messages were included as an option in some systems, but a normal application of this feature would consist in adapting the terminology of a system to local usage, whatever the context ("shipping language" vs. "logistics language", Mexican vs. Spanish.)

Our first version of an interactive translation method was developed for VEO, Vote Expresses Opinion (Bauer-Mengelberg 2000), a system where users introduce the candidates for one or several elections and formulate evaluation criteria categories, items of these categories and default weights for both of them. Then future voters grade candidates and optionally reflect their priorities in the weights, and a linear model provides ratings. Since the system was designed for its use in many parts of Mexico, in some of which people only speak their indigenous languages, we prepared the system to be translated locally into whatever language was needed (schools were special targets of the project). The table of languages could be extended dynamically to include any new one: every language has its own name plus what it is called in the default language, which is an installation parameter of the system to be developed. We show a simulated fragment of a screen being translated in this interactive manner – indicating the operations in the text since we cannot show more screens. It should be pointed out that the English version is being made by a user during normal execution – just as he would use any other part of the system.

The language used to display a screen depends on the user's preference, available at log-on. In VEO we included a "pull-down menu": it presented a list of the available languages and upon selections, switched to the chosen language.



SENADOR FEDERAL
DIPUTADO FEDERAL
PRESIDENTE MUNICIPAL

Español: Cambio de votante

English: Choose a different voter

Figure 1. A Screen During the Process of Its Translation

The user is translating the label of a "button" on the screen. He has already translated *some* menu items and other captions or texts. The default language versions of all texts are used when no translation text is available. Note that the system data (in this case, the POSTS UP FOR ELECTION) continue in the language in which they were introduced. System data cannot be translated using the toolset. Of course a system can provide translation facilities for its data, which is a lot easier than the topic we are explaining. The developer may even copy the methods used to enable users to translate messages or table values from our other modules.

To introduce this feature into VEO, we not only had to design the methodology, files and routines, but also control the actual preparation of every object for its eventual translation. In the new version included in the package, a developer will provide a tag for each graphic object. He will probably have to use an object naming standard if his platform does not support such tags. His system will use programs and files from JBMINFRA's modules and automatically create everything he needs to offer translation and use the translated texts, actually creating the records for all default values. He must, however, include a way to invoke (our) translation feature for a given object – we suggest a special key in conjunction with authorization for the current user to perform the task. Details of special situations are of course included in the toolset but cannot be discussed here. Finally, the method complements this way to translate with others, for example a list of all objects, with filters to limit the size of the list, based on several selection criteria starting with the "form". A list can be obtained during the "tagged objects translation process" described before, for example: show all non-translated items and prompt for translation. Messages are translated as a part of a message module, another component of the toolset, as are elements of catalogues or lists of items in the tables module, which for example handle tables of Countries. Brevity forced their exclusion from our present description. Naturally these modules address other needs, especially productivity oriented, besides offering the interactive translation of their elements.

Some technical data follows, mostly about the translation module. Tagged objects translation uses a single file. It may be a database table, but we are considering other alternatives as well. Records (or rows) of this table have a unique primary index or key, formed by the value of the tag assigned to the object, the language and the version. The row TAG-ID= F33A1, Lang=1,Version = 1 (1 is default) will be created automatically for the object tagged "F33A1", where the text field will contain the original text of the appropriate attribute (taken from the form or the equivalent object). Our programs add other data for use in the filters described before. Every translation or change of version will produce a new row, or update one if both the language and version coincide. Translation facilities include list translation, without using the screen at all, with many options. Several features are offered to the developers who must include the scheme in their systems, as well as many suggestions.

A search and study of comparable available tools produced no results. Since VEO was developed in PROGRESS SOFTWARE's 4GL (PROGRESS SW 1998) which is strong in the translation of systems for distribution, their method of internationalization provided suggestions in both senses, positive and negative, as to how to do things. No real help was provided to produce the translatability in our terms. Other development tools did no better in this sense.

The package is being prepared for use in several platforms: the work to determine how to encapsulate it and the alternatives as to file managers is still in progress. One of our objectives is to broaden applicability by eliminating constraints regarding development platforms and environments.

Conclusions

With the toolset described partially in this paper, a student or developer will have the added power provided by the routines and files. But he can also use many suggested features which might enhance or improve the systems he will design, and which he might not have considered for some reason – including limited experience or knowledge, besides the obvious productivity and cost constraints, which were not discussed out of respect to the audience of this paper. We think this particular subject of helping developers who are somewhat unprotected provides an ample field for research. Not only new features and expansion to an increasing variety of technologies, but ways to make them useful to persons without the up to date knowledge.

The method presented for interactive translation by the users themselves is new. Our previous versions not only required more labor by the developer and the future translators, but also some degree of organization on their part unnecessary in the current scheme. Since it does not perform all tasks which should be offered to the translators, for example automatic or manual resizing and relocation of graphic objects depending on the length of texts, more work and research can lead to even better tools.

As was pointed out, another major module provides security and controlled access to functions of systems based on the toolset. Brevity of the proceedings did not allow discussion of this topic, though it was initially planned, as the title reflects. Until published elsewhere, some information is available at the author's website.

References

Bauer-Mengelberg, J.R. "VEO – a system to decide for whom to vote", www.geocities.com/jbauerm/home.html, March 2000.
Progress Software. *PROGRESS TRANSLATION MANAGER GUIDE*, April 1998.