Code Creator for Automatic HYSYS.Plant^{ã} and MSExcel^{â} Interoperability

David Fernández, Antonio Rodríguez, Laureano Jiménez, René Bañares-Alcántara

Department of Chemical Engineering, ETSEQ, University Rovira i Virgili, Av. Països Catalans 26, 43007 Tarragona, Spain. Tel.: 977-559673; Fax: 977-559667; e-mail: rbanares, ljimenez@etseq.urv.es

1. Introduction

There are many different CAPE tools being used independently for different engineering activities, e. g. simulation, physical property estimation, process design, optimisation, etc. Hence, the opportunity to integrate, develop and customise some of these tools in a common framework is an emerging area.

In particular, the use of process simulators in conjunction with external applications is rising. HYSYS.Plant$^{©}$ offers OLE communication capabilities with a broad set of packages (MSExcel$^{®}$, Matlab$^{©}$...) creating an open architecture that allows very easy communications. The simplest integration mechanism is to use VBA™ and its available interfaces.

The main barrier to exploit Hysys$^{©}$ programmability and extensibility capabilities rely on the language and syntax used, which sometimes is complex and obscure;

in particular for beginners, who go throughout a long learning curve. The *ETSEQ Code Creator* directly provides the code string to access any object in a Hysys$^©$ simulation environment (equipment, stream or process). Nevertheless, when the user has some expertise, the VBA™ library use is recommended.

2. Results

The interoperability Hysys$^©$-MSExcel$^®$-VBA™ is well known, but its application is not common, partly because of the difficulties involved. To promote its use, the Hysys$^©$ libraries (hysys.tlb) were migrated to MSWord$^®$ where additional capabilities were added. The new library is able to classify the objects, automatically link the objects that are collections of objects (i.e. streams), to differentiate, for each object, which are their method properties (functions and subroutines) and differentiate by a colour criteria the variables that represent an object from those variables that are *enums* (options that a variable may achieve). Moreover, the application incorporates some of the cases provided with the Hysys$^®$ tutorials.

A *ETSEQ Code Creator* application was developed in VBA™ to produce data structures to facilitate the access to the different objects in Hysys$^©$. For example, this application allows the user to select any object (i. e. PFRreactor) and get the code to use it in VBA™.

The advantages of this approach are:

(a) *Provide users with the access code to Hysys© objects:* within the library you have to go step-by-step, while now the whole string is retrieved at once. For example, if we pretend to access to 'ShellSideFeed' in a heat exchanger, the instruction provided is: "Application.SimulationCases.SimulationCase. Flowsheet.Operations.HeatExchanger.ShellSideFeed".

(b) *Declare objects variables in Hysys©.* Provides the type of variable to declare and the best way to do it (in some cases there are more than one path to arrive to a certain variable). For instance, if *'hyreflux'* is the VBA™ variable name used to represent the reflux stream, the application will provide the declaration: 'Dim hyreflux As ProcessStream'.

(c) *Declare all variables used in the VBA™ application*, useful as reminder for big applications and to familiarise with third-person developments (Figure 1).
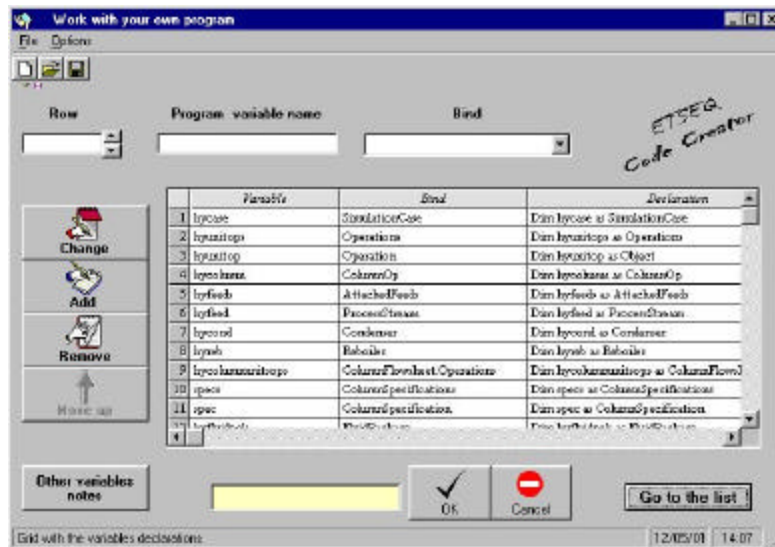

Figure 1. List of variables declared in a VBA™ application.

(d) *Provide users with the access code to Hysys© object using the variables declared.* The novelty functionality of this application is that supplies the code to reach any object using the declared names of the intermediate path, and thus shortening the syntax. For instance, if the heat exchanger in the previous example has been defined as 'heater', the code will be: 'heater.ShellSideFeed" or 'Set ShellFeed=heater.ShellSideFeed', if we had declared 'ShellSideFeed' as a variable called 'ShellFeed'. The code string can be directly copied to the VBA™ environment (Figure 2), where it will link the stream object ShellFeed once the link between MSExcel® and Hysys© has been established. Then a property of this particular stream can be sent to MSExcel® easily (Figure 3).
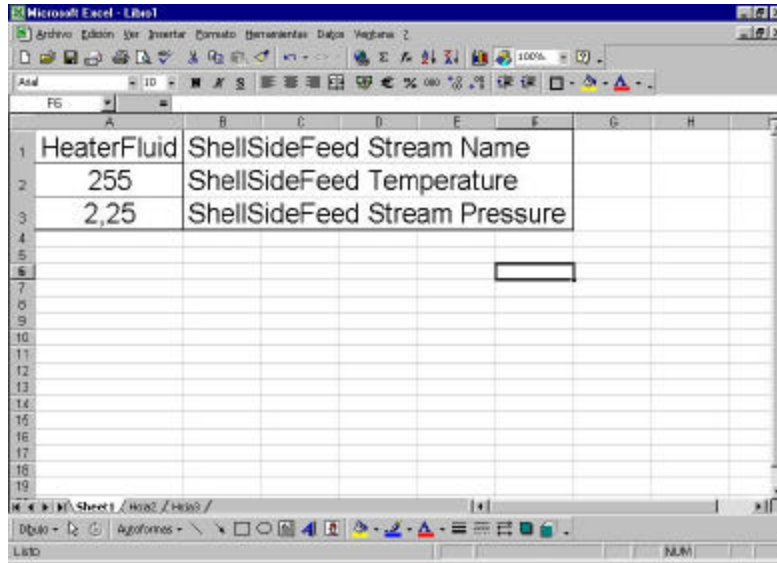

Figure 2.- Hysys© code embedded into *VBA™*.

Figure 3.- MSExcel®-Hysys© link using *VBA*™.

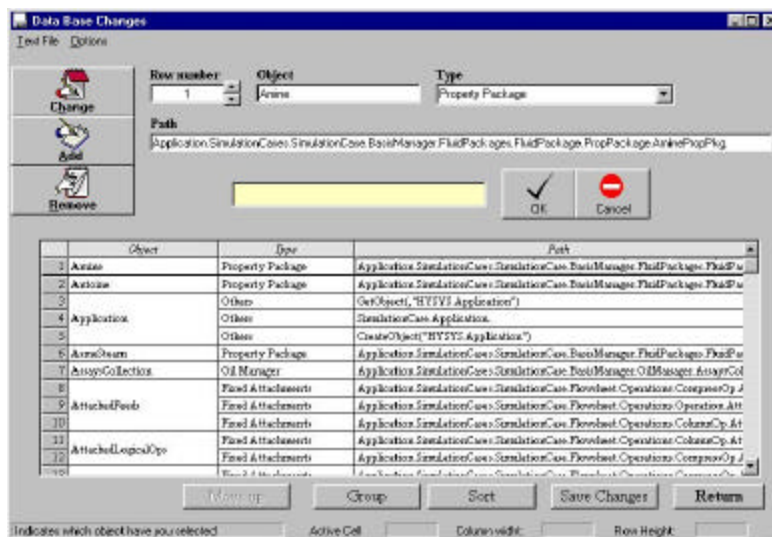(e) *Full access to delete, modify or add new structures*, thus customising the
application.

Figure 4.- Current database used for Hysys© objects.

The *ETSEQ Code Creator* performance was validated developing toy-examples. More complex cases from Hyprotech and Soteica S.A. homepages were tested.

3. Conclusions

The Hysys$^{©}$ library on MSWord$^{®}$ is useful for beginners to learn the structure of the objects. The *ETSEQ Code Creator* seems a good approach to provide the user with the whole access code. The application presented does not include all the possible structures, but has the possibility to be extended.

One of the advantages of this approach is that all the tools used are known for chemical engineers. Besides, the engineer effort can be focused on the system communications and the decision-making, but not in the development of its constitutive parts.

Finally, the communication effort is expected to be significantly reduced after Global Cape Open standardisation.

4. Files provided

Upon request, we will provide:

- *ETSEQ Code Creator* program, with the base code, including a manual and MSExcel$^{®}$ examples. Requires Hysys.Plant$^{©}$ 2.x, Visual Basic for Applications$^{®}$ and MSExcel$^{®}$.

- Hysys$^©$ libraries on MSWord$^®$.

5. Literature

1. AEA-Hyprotech Technology homepage. Available at
   ftp.hyprotech.com/Pub/OLE, Calgary, Canada. Accessed July 2001.

2. Soteica S.A. homepage. Available at
   www.soteica.com.ar/espanol/softinge/softinge.htm, Rosario, Argentina.
   Accessed July 2001.

3. Ceballos Sierra, F. J. Enciclopedia de Microsoft Visual Basic 6, Ed Ra-Ma,
   Madrid, Spain, 1999.