FH – HES

Fachhochschulen – Hautes Ecoles Spécialisées

Chimia 52 (1998) 740–742

ISSN 0009–4293

BridgeVIEW and Chemistry Automation

Olivier Naef*

1. Introduction

This paper discusses how to build an application to control a chemical reactor with BridgeVIEW. The reactor is very simple, but it contains all the features that can be found in an industrial installation controlled by a computer.

The example given in *Fig. 1* is a chemical reactor with a mixer, a temperature indicator, and a level indicator. The reactor has a jacket for controlling the mixture's temperature. It uses steam (from 1 bar to 13 bar) to heat the contents. The temperature of the jacket varies between 100° (212°F) and 180° (356°F). The reactor has two other features:

1) We can fill the reactor with water, and a proportional valve adjusts the flow of water into the reactor.
2) We can empty the reactor with an on-off valve. A sensor indicates if the valve is really closed.

After the design of the installation, we connect the reactor to the computer with a FieldPoint interface. The application to control this reactor will be written with BridgeVIEW, a graphical programming environment. The application generated is as generic as possible, and as a result, the code can be reused for other industrial automation applications with a minimum amount of modification.

2. Generality

2.1. *BridgeVIEW*

BridgeVIEW is a graphical programming language for automation application. BridgeVIEW is based on LabVIEW that use the graphical programming language G from *National Instruments*. With BridgeVIEW, you can easily develop Human Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) solutions for manufacturing and process control applications.

For more information, see the Web side: www.natinst.com, or contact *National Instruments Schweiz*, Sonnenbergstrasse 53, CH–5408 Ennetbaden (Tel. +41 56 200 5151, Fax: +41 56 200 5155, E-Mail: ni.switzerland@natinst.com).

2.2. *FieldPoint*

FieldPoint is an innovative, modular system with versatile analog and digital I/O capabilities, an unmatched set of usability features, and full suite of software that tightly integrates with LabVIEW, BridgeVIEW, Lookout, and other software packages. You connect the FieldPoint with a serial cable (RS232 or RS485). You can have many (up to 25) stations that are connected to one PC. Each station controls from 1 to 9 I/O modules.

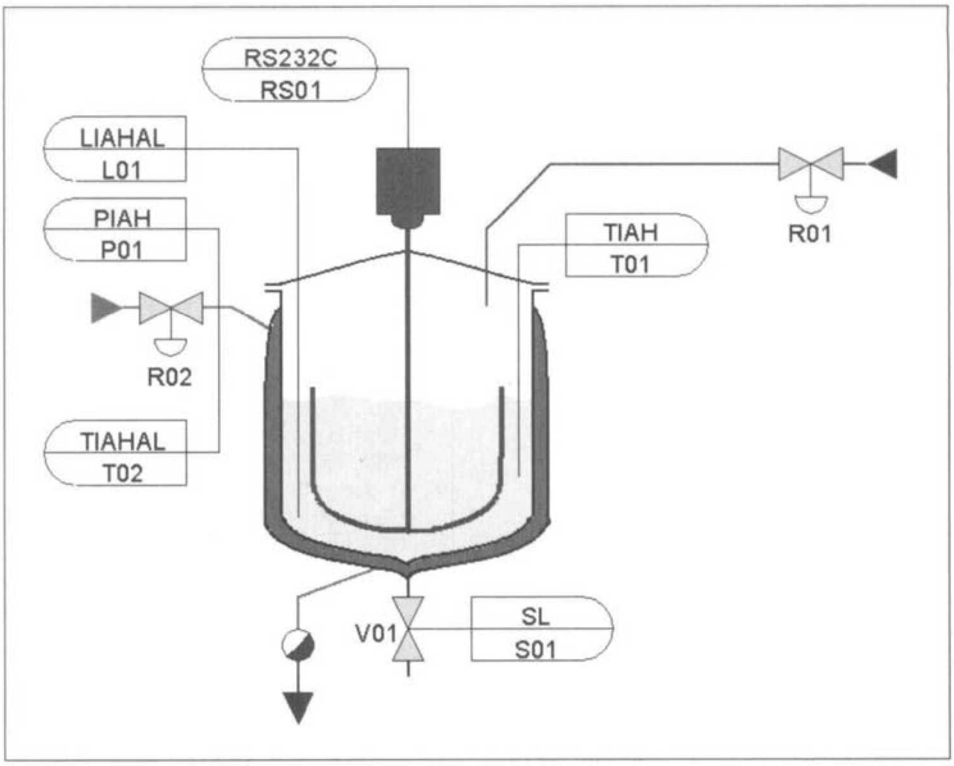

Fig. 1. *The simple reactor*

**Correspondence*: Prof. O. Naef
Swiss Engineering College of Fribourg (EIF)
Department of Chemistry
Computational Chemistry
Blvd Perolles 80
CH–1705 Fribourg
Tel.: +41 26 429 66 11
Fax: +41 26 429 66 00
E-Mail: olivier.naef@eif.ch

3. HMI

3.1. *Introduction*

The Human Machine Interface (HMI) is a simplified representation of the installation and an interface for the operator to control and supervise the system (*Fig. 2*). In an industrial installation, we find many HMIs that represent different aspects of the installation. In an HMI, there is always an active window. From the HMI, we can control the link between the operator and the installation effectively. The operator can work in manual mode (direct access to the elements of the installation or with automation functions) or in batch mode (automatic sequencing of functions to realize a product).

3.2. *Main HMI VI*

In this system, we have the main HMI VI (the main HMI VI is the primary HMI that controls all the other HMI sub VIs) that represents the global installation and other VIs that provide details on the level system. The main HMI VI starts all other HMI sub VIs. All HMI sub VIs monitor the state of the main HMI and can run only if the main HMI is running. If you close the main HMI VI, all associated VIs will be closed. The main HMI VI can be aborted (abnormal termination) by the main application VI.

3.3. *Tags*

Another problem addressed is the access rights of the tag. In manual mode, the operator can access the tag to write a value to an actuator. But if the operator starts a function that needs an actuator, the function must disable the write flag of this actuator. The HMI VI can only read the value of the actuator. Hence, the control must be disabled so that the user is not allowed to write to the tag.

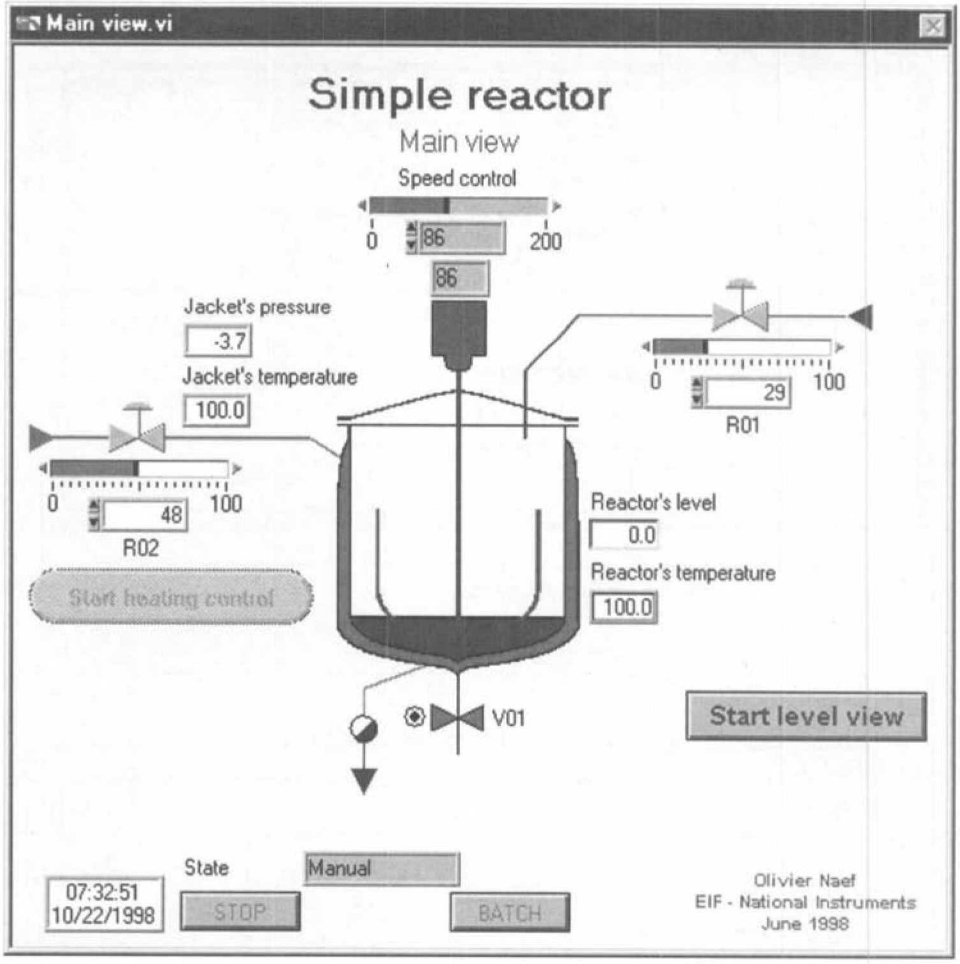

Fig. 2. *Main HMI VI*

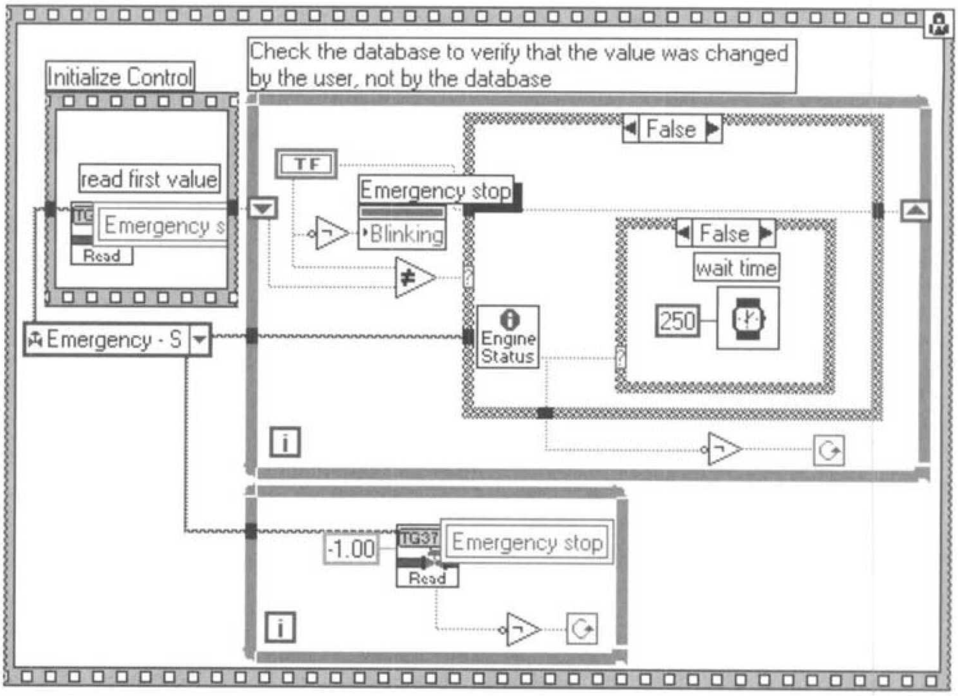

Fig. 3. *Write and read tag using the G Wizard*

4. Simulation

If you have hardware available, this part is not necessary. But, if you don't, it is helpful to have a simulation VI so that you can emulate the installation and see how it will be working in reality. The simulated VI emulates the response of the sensors. The VI reads the values of the actuators on the FieldPoint or the simulated FieldPoint and calculates the response from the sensors. All sensors are emulated on the FieldPoint interface with an output (DOT or AOT). The simulation of the reactor's level depends on the proportional input valve and the switch output valve. The closing action of the output is simulated with a delay of two seconds. The simulation of the jacket pressure depends on the aperture of the proportional jacket valve. We introduce simulation tools for the jacket pressure: a time shift (delay function) and a time constant (first-order function). The simulation of the jacket temperature is directly proportional to the pressure. The reactor's temperature is calculated by the *Euler*'s integration of the differential equation for thermal balance.

CHIMIA 52 (1998) Nr. 12 (Dezember)

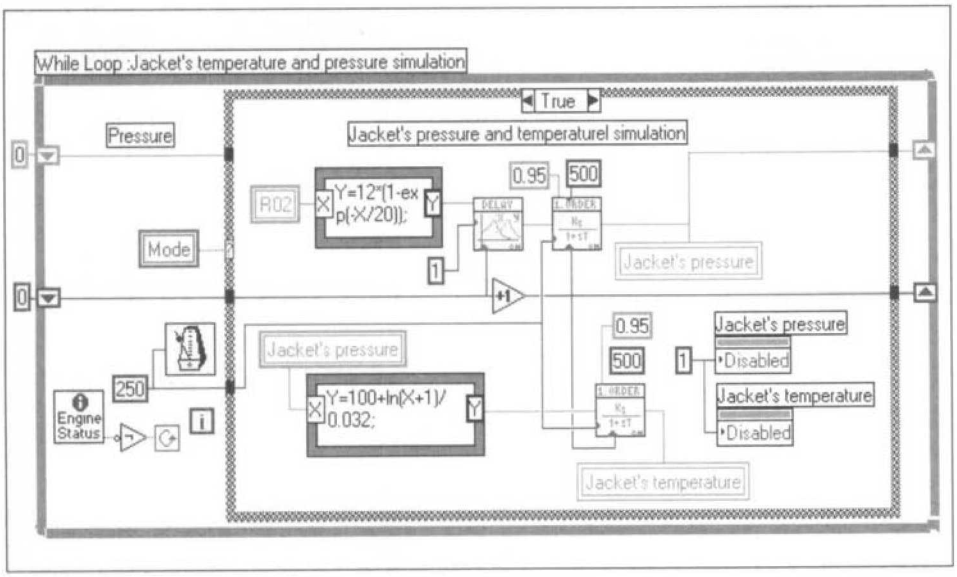

Fig. 4. *Jacket's temperature and pressure simulation*

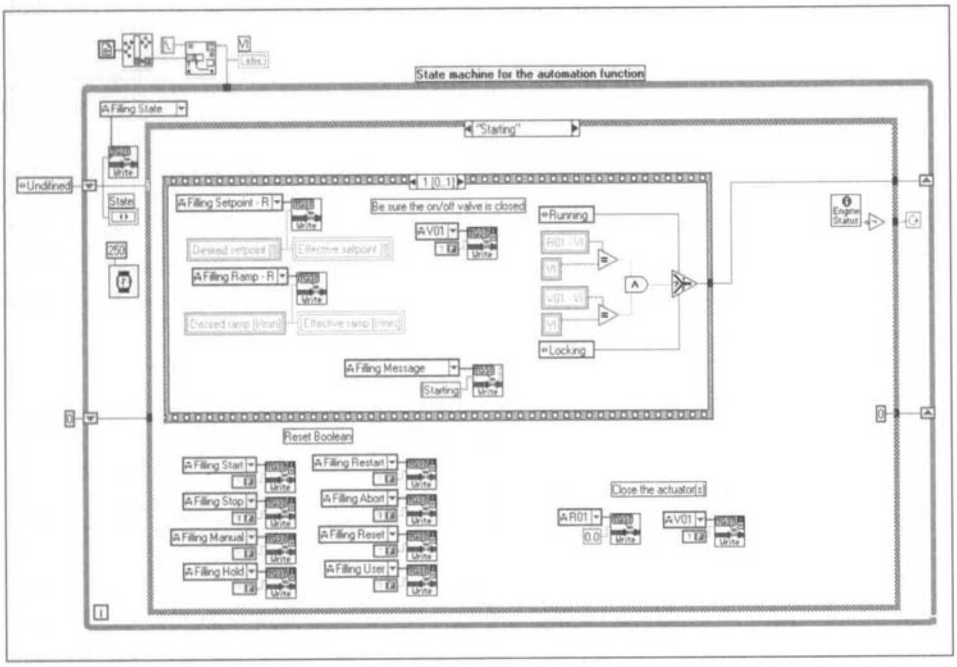

Fig. 5. *State machine for the automation function*

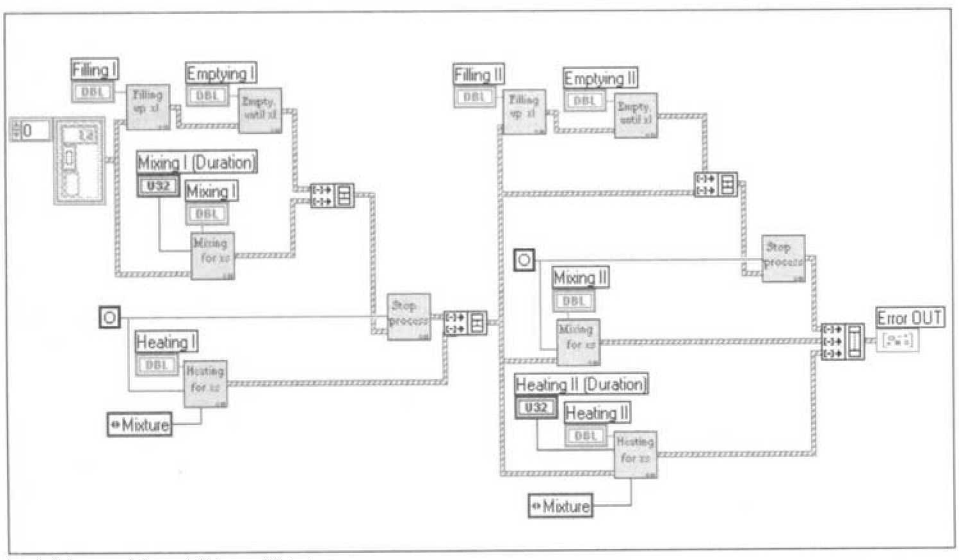

Fig. 6. *Batch process VI*

The program's structure is parallel loops. For the tag, we use the simple G Wizard that automatically creates the code (*Fig. 3*). Another example can be seen in *Fig. 4* with the loop for the jacket's temperature and pressure simulation.

5. Automation Function

Until this point, we have been running the application in 'manual mode'. We can only write values to actuators and read values from sensors. But you can imagine how difficult it would be to set up an industrial installation without a minimum set of automation functions. In this chemical reactor, the automation function we need is a temperature controller. The temperature controller adjusts the jacket's valve. The operator gives the set point, and the controller works in different modes: reactor's temperature constant, jacket's temperature constant, and a difference between reactor and jacket constant. We build two other automation functions: a filling function and an emptying function. These two functions have an exclusion rule: if you fill the reactor, you cannot empty it at the same time.

The automation function contains two loops: the first loop reads and writes the values to the tags, and the second loop is a state machine that you can see in *Fig. 5*.

6. Batch

To create a product in a reactor, you can use the automation functions of the installation. If you need to create exactly the same product more than once, the best way is to have the installation runs in an automatic mode. This is called batch processing. Batch processing uses the automation functions. In our example application, we will start with a continuous function: the temperature control (jacket's temperature control). In parallel, we will fill the reactor with water. When the reactor is filled with two liters, we will start the mixer. When it has eight liters, we will stop the introduction of water and change the temperature control (mixture temperature control) for 2 min. After this time, we will stop the temperature control and the mixer. We will then empty the reactor. The batch process is stopped when the emptying function is finished.

The batch uses a main VI to control the batch process, a batch process VI that contains the effective rules of the batch (*Fig. 6*) and batch functions.

Received: October 15, 1998