

## **THE SIMPLE APPLICATION FOR THE VISUALIZATION**

**Ing. Pavel Pokorny, Ph.D.**

**Thomas Bata University in Zlin, Faculty of applied informatics  
Mostni 5139, Zlin  
Czech Republic**

**Ing. Martin Sysel, Ph.D.**

**Thomas Bata University in Zlin, Faculty of applied informatics  
Mostni 5139, Zlin  
Czech Republic**

### **ABSTRACT**

*The modern control systems are composed from the more layer structure. For example, in the 2-layer structure can be the first layer programmable machine in the industry environment connected to the measured and controlled process. And the second layer can be personal computer with visualization and control software. Both phases are connected via the serial or wireless connection. In this paper there is described the simple software application for the visualization of the technological processes. This application is in a first phase, now. It contains a support for the basic control loops in simulation.*

**Keywords:** visualization, application, programming

### **1. VISUALIZATION**

The control system of the technological process can be the one or more layer structure. The one level structure is composed from the personal or industry control computer. This computer has implemented a technological card and through this card this computer communicates with the controlled process. The modern control systems are composed from the more layer structure. For example, in the two layers structure can be the first layer programmable machine (for example PLC) in the industry environment connected to the measured and controlled process. And the second layer can be made with personal computer with visualization and control software. These phases can be connected via the wireless or serial connection. The three layer structure can have added an intranet or internet application and with this application we can control process with any computer, which is connected to the internet.

We can use the commercial applications for the visualization of control systems, for example Control Web or Wonderware Intouch. These applications have many benefits. For example, significantly improve operational efficiency, dramatically increase engineering productivity, efficiently propagate changes to multiple applications with Wonderware SmartSymbols, visualize and control operations with impressive ease of use, create flexible, agile, scalable applications, keep our options open with unrivaled connectivity, benefit from the convenience of pre-integrated software and hardware, certification from Microsoft "Designed For Windows® XP" and many more. [8]

These described benefits are good for the large visualization control systems. They have a universal construction and high level variability. But this software is very costly and for the simple control systems uselessly complicated. In this paper there is described a very simple application for the visualization of control system. This application is developed on our department and its benefit is simplicity, efficiency and low hardware requirements.

## 2. PROGRAMMING

We write about a programmed application in the previous chapter. With the good design of conception and selecting suitable programming language is creating of this application not very difficult. In this chapter we describe more details.

### 2.1. Programming language and graphics library

A programming language is a set of instructions and a series of lexical conventions specifically designed to order computers what to do. When choosing a programming language to make a project, many different considerations can be taken. First, one must decide what is known as the level of the programming language. The level determines how near to the hardware the programming language is. For our application we select C++ language. The C++ language is in a middle position, since it can interact directly with the hardware almost with no limitations, and can as well abstract lower layers and work like one of the most powerful high level languages. C++ directly supports object-oriented programming (allows the programmer to design applications from a point of view more like a communication between objects rather than on a structured sequence of code), portability (we can practically compile the same C++ code in almost any type of computer and operating system without hardly making any changes), brevity (code written in C++ is very short in comparison with other languages, since the use of special characters is preferred to key words, saving some effort to the programmer), modular programming (an application's body in C++ can be made up of several source code files that are compiled separately and then linked together) and C language compatibility (C++ is backwards compatible with the C language).

For the graphics interface we selected library OpenGL. OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

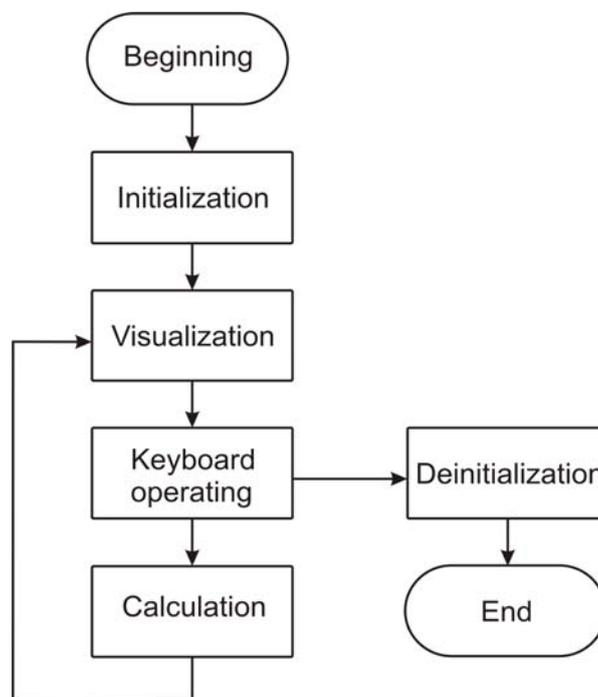


Figure 1. Conception of the visualization application.

## 2.2. Conception of the program

The block conception our application is shown on the Figure 1. Our program is a Windows application. At first we must create a window. It contains creating a Window class structure. This structure has information about our window (name, size, style, etc.). After register this class we show this window on the screen of computer. This showing makes OpenGL, which was initialized before.

In the next step is running a control loops with processing a generated window messages. We process same messages only, for example quit message which causing the program to quit or pressed keys on keyboard. Each defined keys causes defined. When we press Esc key, before the program is ended, we must call routine to release the rendering context, the device context and finally the window handle.

The communication with this application is with the keys and mouse too. This application contains a buttons for the Start and Stop of calculation the control loop, Save for the saving calculated parameters and Exit for the quit. With mouse click on these buttons we can manipulate with this window and work with external file.

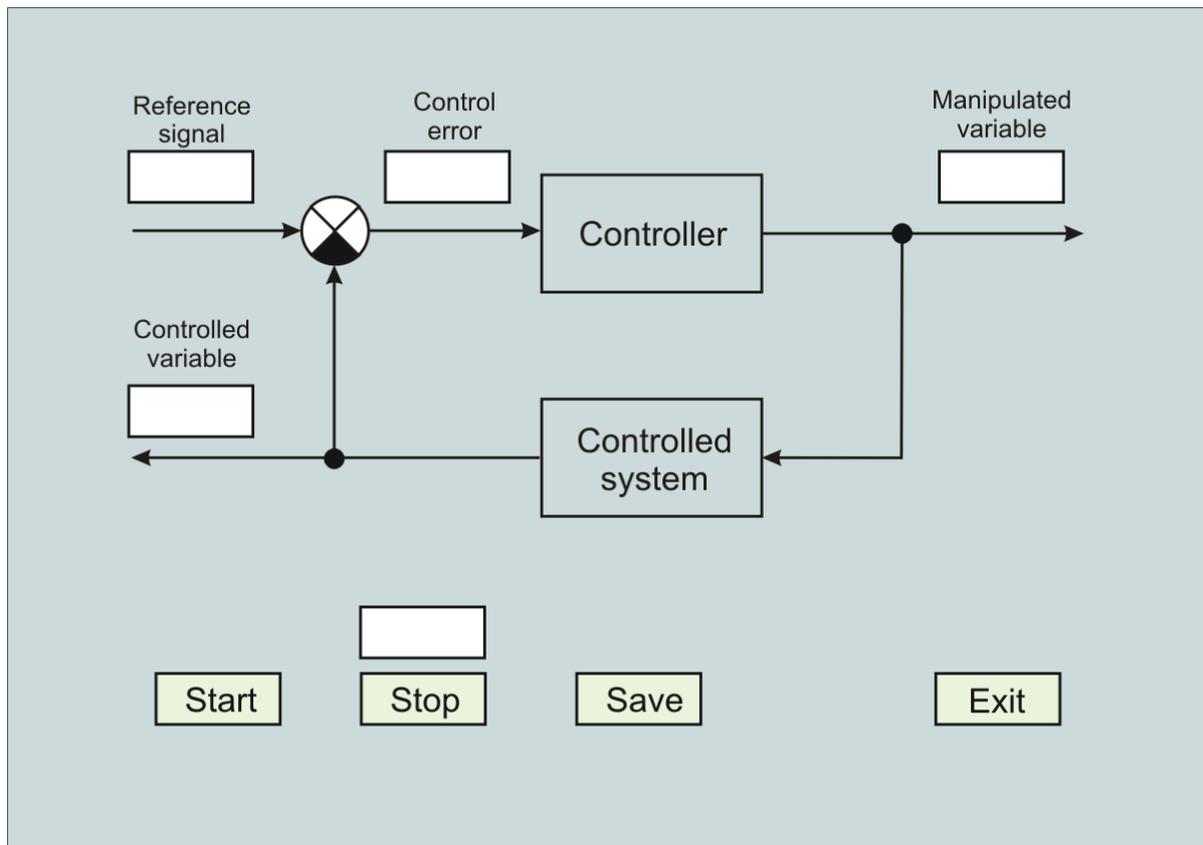


Figure 2. Conception of the visualization application.

## 3. DESIGN OF THE VISUALISATION ENVIRONMENT

Our application is in a first phase. Now it contains support for the simple control loop only. The window design of our application is shown on the Figure 2. We have 2 systems – digital controller and continuous controlled system. In the actual version our application their parameters (with sample time period) are set directly in the source code, but in the future will be set directly in this application. The order of these systems is adjustable too.

We have some parameters in the control loop. They are reference signal, controlled value, control error (difference between reference signal and controlled value) and manipulated value. These numerical values are directly displayed on the screen in the actual time.

Four buttons are on the screen down. With mouse clicks on the Start we start simulation of controlling. Control time is parameter set directly in the source code. But we can stop the simulation

with click on the button Stop. Above this button is text array. Here we can to write a time, when we want stop the controlling automatically. In others text arrays (all arrays are white and the texts are black) we can see actual value of all variables in the control loop. When the control is calculated, we can save all variables into the external text file. These variables are formatted and with this, we can easy read them into others applications (for example Microsoft Excel) and show it, in graphics dependence. The latest button (Exit) ends our application.

This application is very simply. It supports only the basic control loop, but we will more modify it and enchant it to the more variability, functionality and user comfortable. Our main plan is to create a visualization control system for the technological processes, which are in our laboratory. Here we will test it and remove relevant errors. After this, this application can be used in real environment. One of eventuality is use it for the control of process enzymatic hydrolysis. This technology is composed from three parts (fermentation, filtration, evaporation) and each part controls various technological variables (temperature, Ph, underpressure, etc.). For these all three processes we can create special visualization control system. These systems can be created very easy with modification presented application.

#### **4. CONCLUSION**

We have presented a problem of visualization control systems. For the visualization we can use commercial software or we can make it with programming tools. The benefit of programmed application is simplicity, efficient and low hardware requirements. A simple visualization application is created on our department. This application is based on a programming language C++ and graphics library OpenGL. Now, it simulates a simple control loop only. But this application will be more enchanted in the future, as necessary. One of these extensions will be routine, which communicates with the serial port. With this, we can read a process data from others computers or programmable logic controllers, which support the serial communication. For the specific control we can easy modify our application for the better visualization.

#### **5. ACKNOWLEDGEMENT**

This work was supported by the Ministry of Education of the Czech Republic in the range of research projects No. MSM 7088352102.

#### **6. REFERENCES**

- [1] Balate J.: Automatické řízení (in Czech), 1<sup>th</sup> edition, BEN – technická literatura, Praha, 2004, ISBN 80-7300-148-9.,
- [2] Petzold Ch.: Programming Windows, 5<sup>th</sup> edition, Microsoft Press, 1998, ISBN 1-57231-995-X.,
- [3] Pokorný P.: The control possibilities at the hydrolytic processing of albuminous secondary raw materials. Dissertation work. Zlín: FT UTB Zlín, 2003.,
- [4] Pokorný P., Vasek V., Stalmach J.: Visualisation program of graphics algorithms. In: Proc. 15<sup>th</sup> International DAAAM Symposium '2004, Vol. 1, Vienna University of Technology, Vienna, Austria, November 3-6, 2004, 363-364 ISBN 3-901509-42-9.,
- [5] Prata S.: Programování - mistrovství v C++ (in Czech), 1<sup>th</sup> edition, Computer Press, 2001, ISBN 80-7226-339-0.,
- [6] Shreiner D., Woo M., Neider J., Davis, T.: OpenGL programming guide: The official guide to learning OpenGL, version 1.4, 4<sup>th</sup> edition, 2003, ISBN 0-32117-348-1.,
- [7] Wroblewski P.: Algoritmy – datové struktury a programovací techniky (in Czech), 1<sup>th</sup> edition, Computer Press, 2004, ISBN 80-251-0343-9.
- [8] Commercial www pages: Wonderware Intouch [on line]. Place of edition: Company Headquarters, Lake Forest, CA 92630 U.S.A., 2006. [cit. 15.5.2006]. Available on: < <http://www.wonderware.com>>.