Simulation Tools for P-NET based Networks

S. Dietz*, S. Bielkovsky, M. Manninger*, D. Dietrich*, O. Nisamudtinov****

*Institute of Computer Technology, Vienna University of Technology, Gußhausstraße 27-29, A-1040 Vienna (AUSTRIA)
Tel. +43 1 58801/3704 Fax +43 1 505 38 98-14
E-Mail: {dietz, mm, dietrich}@ict.tuwien.ac.at

**Institute for Automatic Control Systems, Perm State Technical University, October Square 1, Perm (RUSSIA)
E-Mail: CEP@asu.pstu.ac.ru

Abstract

The growing complexity of industrial processes requires new tools for design, off-line testing, commissioning, and maintenance. This paper presents the concept of such a tool and a possible implementation for Windows95 written in C++. Our tool allows modelling of the technical process, network design, network simulation (especially of load on the net and on the controller) and communication with the real P-NET to test already installed segments. The system described in this paper consists of three modules: a program for modelling and simulating the technological process, an OLE server that simulates the network, and a user interface that enables the user to project and monitor the network.

1 Introduction

The complexity of modern technological processes and the cost of on-line testing require tools that allow the projecting engineers to test systems off-line and optimise the use of hardware. The tools should allow to simulate the complete system :

- Modelling of the technological process
- Simulation of the technological process
- Planning of the Network structure
- Simulation of the behaviour of the Network
- Progressive installation of the System (connection to the real P-NET)

Today's tools are not designed to work together and require the integration of the different steps by the designer of the system. Often the only possibility to integrate the components is to write additional programs that link the different modules, which is time consuming and requires deep knowledge of the tools.

Another severe problem is the need to test the design on the real system, which again is very time consuming, does not allow any mistakes and makes optimisation difficult due to a necessary security margin. For the expansion of existing systems, sometimes the whole production has to be stopped for longer periods of time during the installation.

We present a graphical system that includes all the functionality to develop complex automation systems in form of a RAD (Rapid Application Development) tool. In this way the engineer can concentrate on the problems of optimising the network and start the installation of the network with an already tested system. He can test e.g. a chemical plant, simulate operation at maximum load, optimise the use of resources (e.g. number of masters) and test the response on exceptional situations (e.g. failure of a machine, tests on software redundancy, etc.).

2 System description

The user is able to virtually develop the complete system by linking blocks contained in libraries or created by himself. Fig. 1 shows the architecture of a typical project, with a workstation used to run the simulation of the technological process, the simulation of the network and to connect the virtual network to existing P-NET segments.

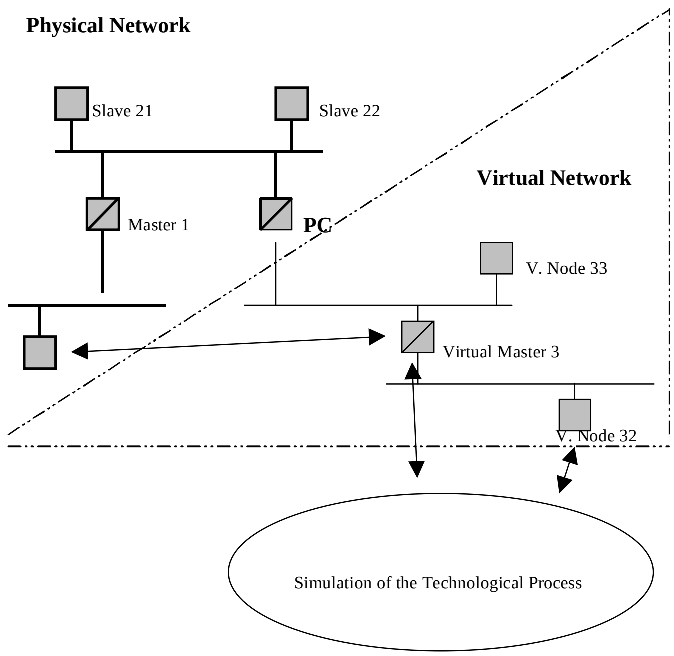

Fig. 1: Typical Structure of a Project

The main advantage of using this system is that the engineer can work as if he was installing the real system, and the work can be used in the real installation without big changes. The user

also has the possibility to test ideas he could not test on a real system that needed to be programmed on-line, because of the risk of damaging any part of the installation.
Diagnosis tools also help to find errors in the programs and can monitor a variety of parameters. These tools are useful e.g. to find out if a value is read more often than needed, or to see if all parameters requested arrive within a certain period of time.

3 System Architecture

3.1 Software architecture

As described in section 1 the tool consists of several modules that work together. These modules allow to project and test a complete automation system without writing a single line of code, but by graphically designing all parts of the system.

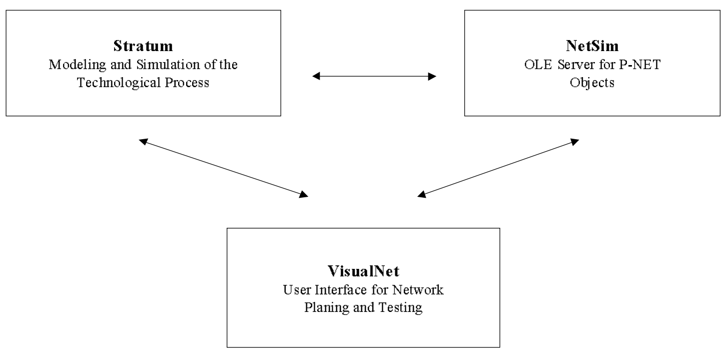

Fig. 2: General Architecture

Figure 2 shows the main components of our tool.

- Stratum is a software package that allows the user to create models of physical objects and simulate their behaviour. It has an interface that allows other applications to access the variables. This makes it suitable for planning and testing a variety of industrial processes (e.g. chemical plants).

  The tool includes a library of components that are often used in automation projects like PT-100 temperature sensors, models of motors, pumps, etc. A manufacturer of automation products is able to develop easily models for his products and include them in the libraries, and the user can develop own models that suit his needs in an easy C-like programming language.

  In the modelling phase of the project the engineer is able to link these objects without the need of writing additional code. This is done in a graphical environment (see Fig. 3) by linking variables and defining dependencies. The starting parameters and configuration of the objects are also set using this graphical environment.

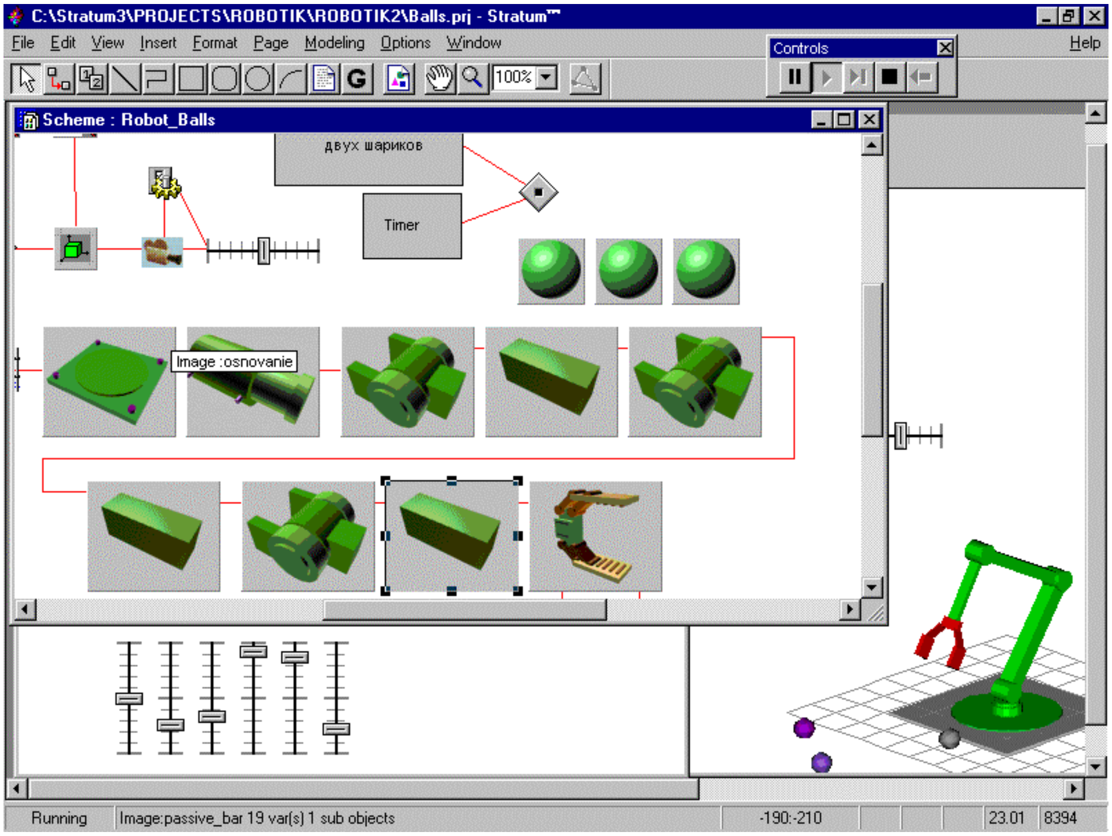

Fig. 3: Stratum's graphical interface

Complex models can be run in a distributed environment (e.g. a local area network) that supports TCP/IP. This dramatically increases the speed of the simulations for large technological processes or for systems with a very detailed model.

- NetSim is an OLE Server application that starts threads as parallel processes for every P-NET virtual node. Each thread is a virtual machine simulating the OSI layers as described in the P-NET protocol (see Fig. 4). Each virtual node includes also an automat for interpretation of pseudo code that is generated by the Pascal compiler included in the visualisation module.

  The main task (which starts the tasks for the nodes) simulates the bus and has functions to analyse the traffic. Packets can be logged to analyse the behaviour of the network, allowing to measure the response time of the slaves and the load of the segment. These functions can be used to optimise the use of hardware in an installation.

  The user does not have to take care of this program, as it is only accessed by the visualisation module and Stratum.

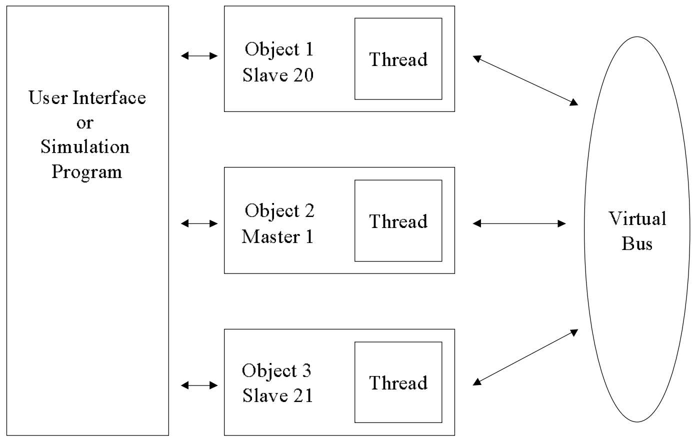

Fig. 4: Architecture of the Network simulation module

- VisualNet is the GUI that allows the user to plan and visualise the network. This is done graphically, by adding segments and nodes from a library. These modules can be modified by drop-down menus and the position of the modules can be changed by dragging and dropping them, making changes in the architecture of the network easy (Fig 5.). In a second step, the user links the I/Os of the virtual nodes to one of the variables listed by the process simulation program.

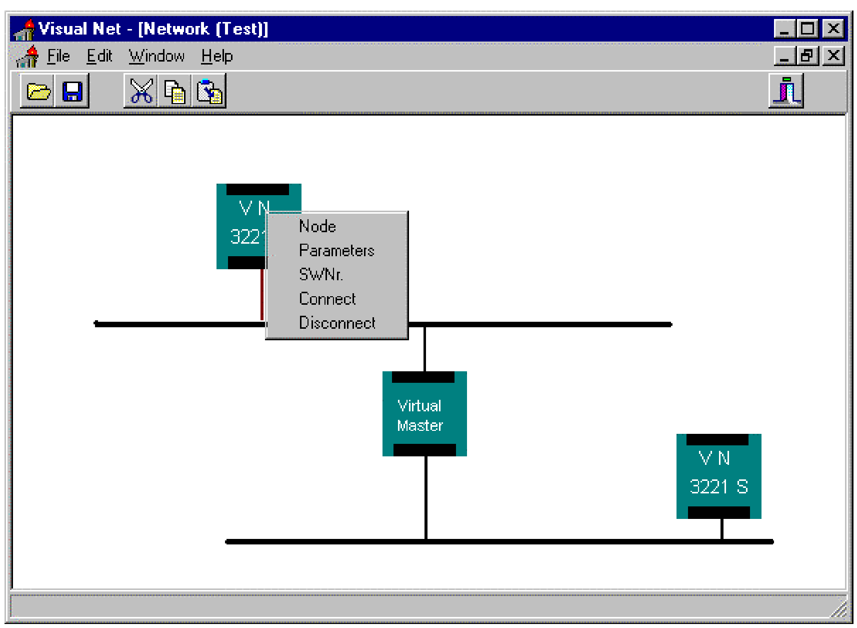

Fig. 5: Network Project

This module includes a Process-Pascal like generator of pseudo code in which the code for the masters can be developed. This code can be "loaded" into the virtual masters which run the program. With this module the user can easily change any parameter on the simulated network (e.g. node address or channel configuration).

3.2 Hardware architecture

To allow a communication with existing P-NET Systems, an ISA card with a microcontroller and a RS-232 and RS-485 interface are used. This card not only works as a master and a router to the virtual network, but is also able to monitor the network and test the segment for packets or log the requests and replies from nodes in the segment. This allows to test if all nodes in the segment work properly and if the load on the P-NET is not to excessive.

4 Conclusion

The described tools allow for a very flexible and complete projecting, testing and installing P-NET systems. One of the main advantages is that work can be reused. The model generated to test the technological process in Stratum can also be used for the visualisation on the operator’s workstation, and the virtual system can be used for training exceptional situations by the operating personnel, too.

These tools should also reduce the time of commissioning, as all the programming and design of the network can be done off-line. Training of the operators can be started as soon as the model is ready and while the system is being installed, making it possible to have trained personnel as soon as the P-NET installation is ready for operation.

The possibility to analyse the load on the bus and on the masters will also allow the designers to optimise the use of modules in the installation, being able to distribute the load of the masters in an optimal way. By analysing the load on the bus, the number of segments for a complex system could also be reduced drastically.