

# An introduction to the hybrid simulation – the conception of the simulation system

**K. Foit\***

Institute of Engineering Processes Automation and Integrated Manufacturing Systems, Silesian University of Technology, ul. Konarskiego 18a, 44-100 Gliwice, Poland

\* Corresponding author: E-mail address: krzysztof.foit@polsl.pl

Received 19.02.2010; published in revised form 01.04.2010

## Analysis and modelling

### ABSTRACT

**Purpose:** of this paper: The aim of this paper is to present a simple hybrid simulation system, which is composed of virtual reality software and a mathematically oriented application. From the engineer's point of view, an important possibility is to link together the mathematical software programs with these for creating presentation graphics or virtual reality, in order to create a simulation system with the large customisability.

**Design/methodology/approach:** The coupling of mathematical and virtual reality system can be done in several different ways, using mechanisms for the Interprocess Communication.

**Findings:** It has been assumed that EonX control will be used in a program conceptually similar to the EON Viewer, but equipped with support of basic interprocess communication interfaces in order to maximize flexibility and possibility of data exchange with different applications.

**Research limitations/implications:** The one of specific requirements is to create an appropriate virtual world in EON Studio with use of external events nodes.

**Practical implications:** The simple application named SockED has been created for testing purposes. The SockED application, which is hosting EonX control, acts as DDE server. As an external application any mathematical program that supports the DDE communication can be used (eg. Microsoft Excel, Matlab). There is also theoretical possibility of use the Programmable Logic Controller in similar manner like in case of SCADA application, but this case was not tested yet.

**Originality/value:** There is no program on the market that allows complicated mathematical computation along with high resolution, 3D presentation graphics. The creation of SockED application allowed combining a mathematical program with the powerful graphics engine from EON Reality.

**Keywords:** Numerical techniques; Engineering design; Virtual reality; Hybrid simulation

#### Reference to this paper should be given in the following way:

K. Foit, An introduction to the hybrid simulation – the conception of the simulation system, Journal of Achievements in Materials and Manufacturing Engineering 39/2 (2010) 176-183.

## 1. Introduction

Modern engineers are increasingly turning to new opportunities to present their projects. Mathematical formulas and

charts are still the basis for discussion among professionals, but are incomprehensible to people not associated with a particular industry. It is true that some programs for computer-aided design have a possibility to build three-dimensional presentation based

on the built-in modules, but they are not as attractive as the presentations made in special programs for three-dimensional visualization. On the other hand, specialized programs for three-dimensional presentation rarely have advanced modules that are responsible for mathematical calculations. Most of them have simple physical phenomena interpreters, which do not support advanced computing capabilities. On the other hand, specialized programs for mathematical computation sometimes have the appropriate modules which allow creating three-dimensional simulation, but they are not as advanced as in the case of an application for creating virtual reality.

Faced with the problem of selection of computational and simulation tools, the engineer has a choice of many mathematical and 3D graphics software [1-8]. The first group may include packages such as Mathematica, Matlab, MathCAD, Dymola, etc., while the second group can contain such software as 3D Studio, EON Reality, Octaga, Autodesk Maya, etc. There are also counterparts of these programs in the area of free software although there is not as large choice as among commercial software. For mathematics computation can be used programs such as Scilab and Octave. Blender, GIMP, or Wings 3D could be proper for the creation of presentation graphics.

From the engineer's point of view, an important possibility is to link together the mathematical software programs with these for creating presentation graphics or virtual reality in order to create a simulation system with the large customisability [1-7,9-13,15]. In this case, it is important to connect the programs in a manner that allows real time data exchange. This can be done in several different ways, including mechanisms for the Interprocess Communication (IPC) and scripting system [15].

In the further parts of the paper, some simple examples of such hybrid simulation system will be shown.

## 2. The short description of the Interprocess Communication and scripting capabilities in Microsoft Windows

As it has been mentioned, applications in the MS Windows environment can communicate to each other using so-called Interprocess Communication (IPC) [14] or scripting system. The IPC is a generic term for a set of methods through which applications can communicate within the operating system or via a computer network. The system scripting interface is a special property that allows running an interpreted program (script), which is executed and managed by an operating system.

### 2.1. The Interprocess Communication

The two, most popular IPC methods are Dynamic Data Exchange (DDE) with its network variant NetDDE [14], and Component Object Model (COM) with ActiveX technology.

DDE is a simple protocol for exchanging data between applications. Communication takes place via the so-called channels. Providing of the appropriate message to the application

requires three parameters: application name, topic and subject. Information exchange is done in a client-server manner, what means that one of the applications must be the DDE server, while the other is the client. Client, wishing to establish a connection, opens a DDE channel and gives three parameters: the name of the server application, topic and item. The topic can be generally understood as a group data, while the item is a specific, desired value. Data flow is possible in two directions: the client can retrieve and write data to the server application. In this way, it is possible to control the given application. The principle of DDE is shown in Fig. 1.

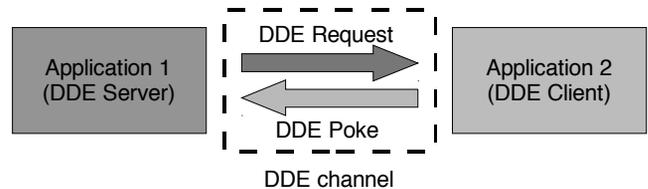


Fig. 1. The communication via DDE channel

The network variant of the DDE protocol is called NetDDE. Today, it is often used for gathering and presenting of data in SCADA systems. The Microsoft Windows comes with basic version of NetDDE - now it is no longer supported and has been abandoned in favour of other solutions.

Running the NetDDE in Windows requires the appropriate drivers and libraries. Because the NetDDE is dependent on NetBIOS protocol, it is intended to use in local networks [14]. The other versions, intended for use with SCADA and often supplied with them, have support for TCP / IP protocol.

Prior to enabling the communication, the relevant shares must be configured on both computers. This operation can be done by using an appropriate program, or by direct writing to the registry.

The access of data via the NetDDE requires similar parameters as in the case of regular DDE. The application name in this case is replaced with the appropriate network path and the name "NDDE\$", such as \\COMPUTER\NDDE\$. The topic is the name of the share and the value identifies a specific item. The NetDDE communication passes through the one channel, while a proper application can be selected by giving relevant topic. The schematic diagram of communication through the NetDDE is shown in Fig. 2.

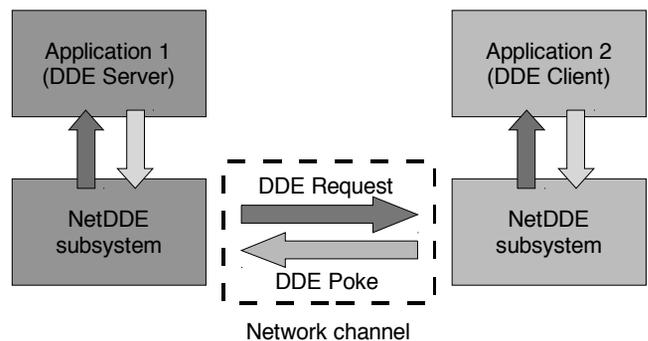


Fig. 2. Schematic diagram of NetDDE communication

Another possibility to communicate between applications is the use of special objects called ActiveX. Their other name is an ActiveX controls, because they often serve as extension of the interface of an existing application. They are supported by most software, in particular by Microsoft products. The ActiveX control is embedded in the application and it works as stand-alone program, but with the possibility of exchanging information with parent. In this way it allows to extend the capability of the parent program. The major inconvenience of the ActiveX is the need to register the controls in the operating system. Without this, the extension may not work. Fig. 3 shows two examples of the ActiveX controls running within Windows Internet Explorer. The first one is a virus scanner and the second is the Visual Basic program, converted to web control.

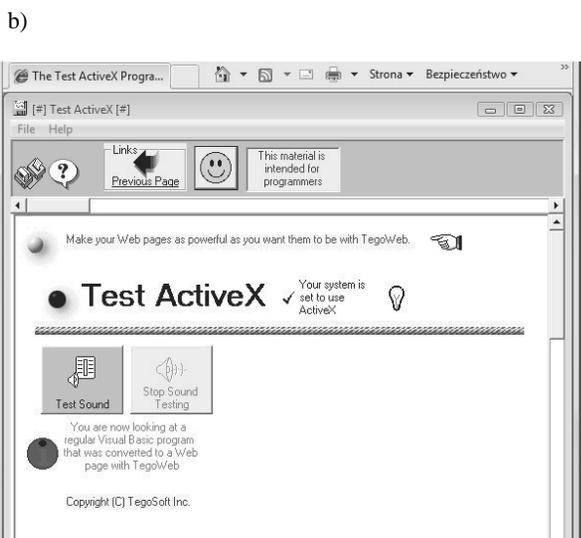
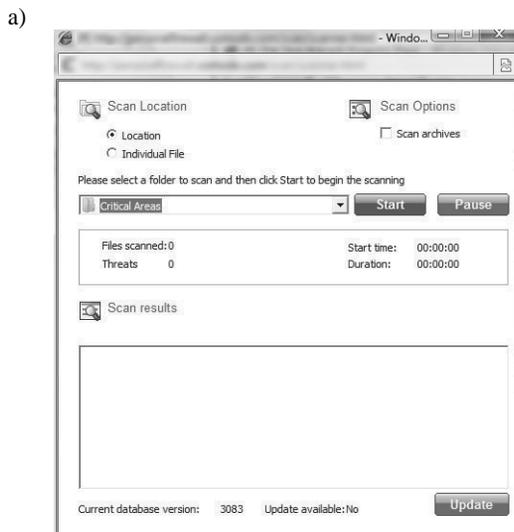


Fig. 3. Microsoft Windows Internet Explorer running ActiveX controls: a virus scanner a) and the Visual Basic program b) [16-17]

## 2.2. The scripting engines

The Microsoft Windows contains a component called Windows Scripting Host, which is the successor to DOS batch files. By default, WSH supports two scripting languages: JScript (JavaScript variant, created by Microsoft) and VBScript (the language based on Visual Basic for Applications) [18]. It is also possible to install other engines, scripting languages, such as PerlScript, KiXtart, OpenObject REXX, Python.

Scripting engines are not directly related to the exchange of data between applications. The script runs on the operating system level, so it is possible to use all the available OLE/COM components and control running applications. In some cases, the application can use the WSH to run scripts inside its own environment [18]. Then it is possible to use all the scripting engines installed in the operating system. The Microsoft Windows supports yet another type of script files - WSF (Windows Script File). In this case, a special syntax allows mixing multiple scripting languages within one file.

## 3. An attempt to build simple hybrid simulation system

In this section the attempt to build a hybrid simulation system using a virtual reality environment will be discussed. The graphical part of the system will be done in the EON Reality Studio application, while the computational part will be implemented as an external application.

### 3.1. The EON Reality software

EON Studio is commercial software based on VRML/X3D standards, but offering the ability to support special input devices and displays. Furthermore, it can work in a software cluster, where one application is the master and others are slave. The philosophy of creating a virtual world does not differ too much from the classical method based on semi-automatic generation of X3D code – like in any other applications of this type. EON Studio uses a visual GUI (like contemporary, high level, visual oriented programming languages) to create a code. It is based on inserting the appropriate objects to simulation tree with a subsequent determination of their parameters. The program does not have a graphical editor, so graphics must be made in the application that allows the creation of 3D objects and imported into the EON Studio. The EON Studio allows to import some popular 3D graphics file formats. More formats can be managed by using the EON CAD application, which is based on the Deep Exploration software from Right Hemisphere. In addition, the EON CAD allows the preliminary preparation of 3D models, like reducing the complexity of the model, applying textures etc.

The main window EON Studio, as shown in Fig. 4, is divided into three parts. The first one contains a set of nodes, simulation prototypes, and routes. In the second part, the Simulation Tree is placed, which contains hierarchically arranged simulation objects. The third part is the Property Bar, where the parameters of individual objects can be entered.



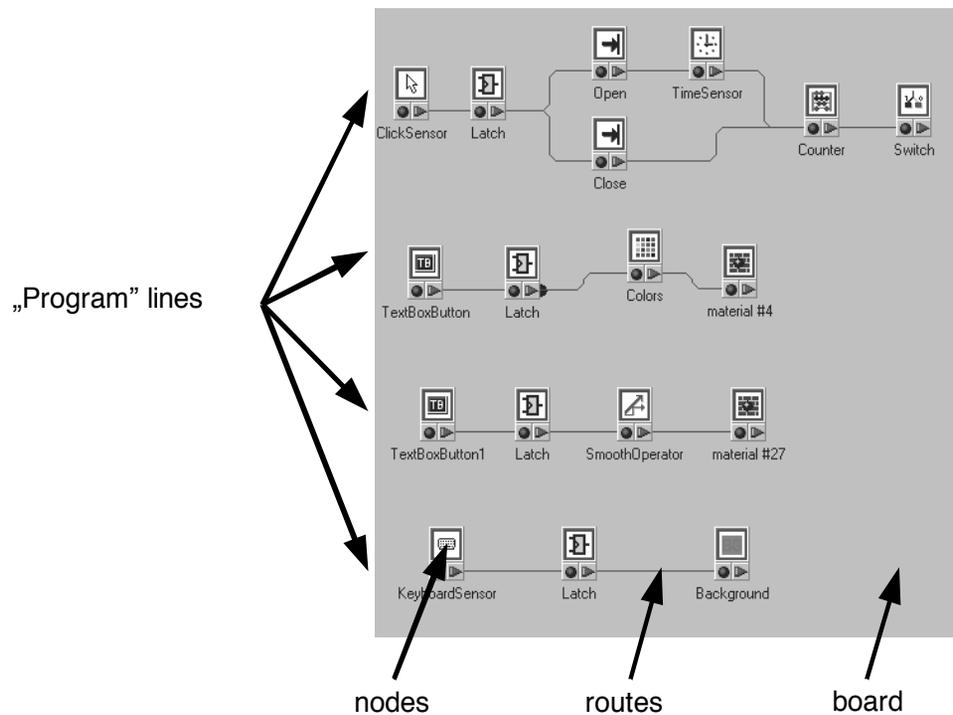


Fig. 6. An example "program" for interactive simulation

The terms "nodes", "prototypes" and "routes" are directly taken from the VRML/X3D language specification [15].

Broadly speaking, the node corresponds to an object, while the prototype is a group of nodes. In this way, it is possible to create new prototypes based on the existing nodes and prototypes, as in the case of high level programming languages, where is possibility to create new functions on a base of existing functions and keywords.

Each node has fields that define its properties. In order to introduce interactions nodes also have so-called events, which are divided into input events (EventIn) and output ones (EventOut). Output event of the one node can be linked with the corresponding input event of another node. Such a link is called "route". A graphical representation of exemplary node in EON Studio, together with a list of fields is shown in the Fig. 5.

By combining the appropriate node in EON Studio an interactive simulation is created. A "program" is created in a graphical way, by positioning the appropriate nodes on the board and connecting them with the routes. Fig. 6 shows an example of "program".

The special type of node is "Script" (Fig. 7). It can be used for nesting a script code in the simulation. The script can be written in any language, which is registered as a script engine in the Windows Scripting Host. The JScript and VBScript are supported as a standard, but it is possible to install other extensions, as described in Section 2.2.

Using the node "Script" is not synonymous with the enabling communication between the EON Studio and other processes [15]. It is however possible to use all the opportunities offered by the WSH, in particular file and database access. The "Script" node is treated as an internal scripting, like the VBA in the Microsoft Office applications.

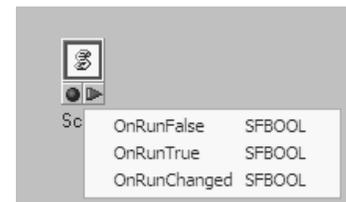


Fig. 7. The "Script" node and its outgoing events

### 3.2. The EonX extension

The EonX is a typical ActiveX control, designed to display and control the EON Studio simulations in other programs. The EonX control allows running a simulation in the Internet Explorer web browser, PowerPoint presentations, as well as using it in the programs written in high level programming languages [15]. The main tasks of the control EonX (apart from showing the simulation) include:

- downloading and installation of the other components of EON, depending on the requirements of the simulation,
- a communication between the simulation and the host application,
- providing information about the software version,
- viewing simulation in full screen mode,
- displaying EON's configuration dialog,
- creating an snapshots images of the EON simulation,
- a verification of the appropriate license.

The EonX control provides a number of properties, methods and events, which are available from the host application. There is one event, provided by EonX, called "OnEvent". This event provides the host application, all the events of "OutEvent" in the EON simulation. Communication with the host application will be possible after insertion of special fields into the simulation (so-called out-event fields and in-event fields) and conversion of ordinary events to external out-event fields. After making these changes, the control of the simulation from the host application will be possible using two methods: SendEvent and OnEvent. The first one triggers events within the running EON simulation, while the other receives the notifications of events from the simulation. For example to send Boolean value for in-event "InEvent1" of SFBool type, to the simulation embedded as "EonX1", the following code could be used:

```
EonX.SendEvent("InEvent1", true)
```

The example function that derives the event from a simulation can take the form EON\_OnEvent(e,v), where e is the event's name and v is the value.

As it was mentioned earlier, in order to ensure communication, the "Route" board must contain in-event and out-event fields (Fig. 8).

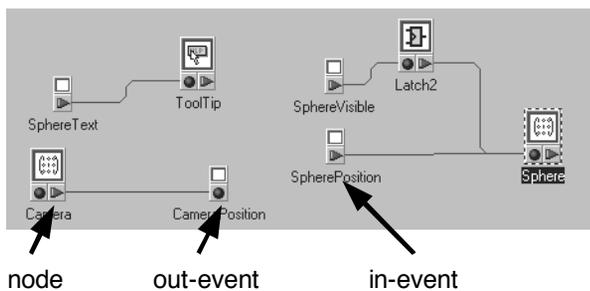


Fig. 8. The "Route" board in EON Studio with the in-event and out-event elements

One of the applications, using the EonX control is EON Viewer - the program used to run simulations without need to install the complete EON Studio environment.

### 3.3. The simple, hybrid simulation environment

The project of a hybrid simulation environment is based on the use of the EonX control. It has been assumed that this control will be used in a program conceptually similar to the EON Viewer, but equipped with support of basic interprocess communication interfaces, with particular emphasis on the Dynamic Data Exchange (DDE). The aim of such approach is to maximize flexibility and possibility to exchange data with different applications. The one of specific requirements is to create an appropriate virtual world in EON Studio with use of external events nodes. The application, which is hosting EonX control, acts as DDE server. The simplified communication scenario is shown in Fig. 9.

As an external application any mathematical program that supports the DDE communication can be used (eg. Microsoft Excel, Matlab). There is also theoretical possibility of use the

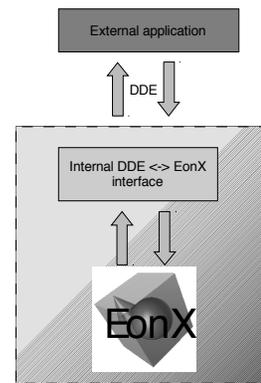


Fig. 9. Schematic diagram of communication between EON simulation and an external application

Programmable Logic Controller in similar manner like in case of SCADA application, but this case was not tested yet. It is obvious, that for some special purposes, where a dedicated program should be created, it is better to use EonX control directly in newly designed application than use DDE engine, which is relatively slow in comparison with direct access of the control.

The simple application named "SockED" has been created for testing purposes. The name is the combination of the word "socket" with upper case "E" meaning "EON" and "D" meaning "DDE". The main requirement for use of the program is a presence of registered EonX control in the operating system. The main interface of the application is very simple and contains only two menu entries. The one of them is standard "About" entry, which contains a short note about the program. The "heart" of application is the "File" menu. It consists of the following entries:

- Open - it is used for loading EON simulations,
- Run - runs the loaded simulation,
- Monitor - runs simple data monitor, used for communication debugging purposes; the use of the monitor could slow down the simulation,
- Close - closes active simulation file and shuts down opened communication channels,
- Exit - exits the program, closes file and DDE channel.

After the startup, the application shows the main window (as in the Fig. 10) with the EonX logo. This can be treated as a test of the correct startup - it informs that the EonX control has been properly initialized. The next step is to load an EON simulation, which should be controlled from the other application. The test version of SockED does not have full error support, so in the case that simulation without external event nodes will be loaded, the application does not inform about it. Another important thing is that test version does not run the simulation automatically - it can be started by selecting the Run command from the File menu or by sending a proper command from external application by the DDE.

The communication with SockED is very simple. It requires establishing the DDE connection from an external program, using "socked" as the "application" parameter. The "topic" should contain the simulation file name. If the simulation is not opened, it will be loaded after establishing DDE communication between applications. The "item" parameter should be set to the name of the



Fig. 10. The main window of the SockED application

event, which value should be written or read. There is no “SendEvent” and “OnEvent” methods equivalent. The proper activity (i.e. read or write of the value) is selected by the DDE commands. Sending a value to SockED means “SendEvent”, reading - means “OnEvent”. An example of simple simulation is

shown in Fig. 11. The application has been tested with the old Matlab version, Microsoft Excel and Working Model. In the case of Matlab a simple function was created, which allows to carry out the inverse kinematics computation, and simulate the movement of the wrist along straight path between two points. In the Excel and Working Model applications, a simple interface has been created, which allows to moving the robot arms using sliders.

## 4. Conclusions

This article discusses the sample project, a hybrid simulation system, consisting of the program to display virtual reality and mathematical applications. As an example, an application called SockED has been created, which is based on ActiveX control called EonX, originating from the EON Studio. The main goal of SockED is to provide the DDE communication between EonX and the external application. The program has been tested with Matlab, Microsoft Excel, and Working Model applications. The next stage of development will try to perform a simulation in virtual reality environment, intended to cooperate with the PLC via a link data acquisition.

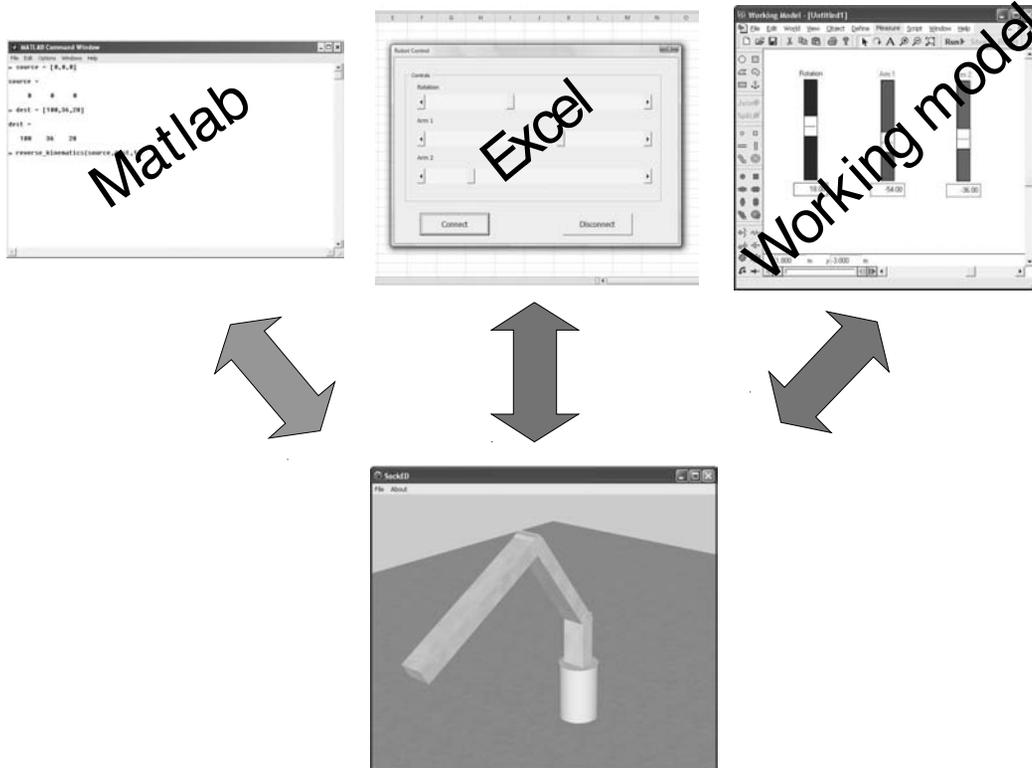


Fig. 11. An example of simple simulation

## Acknowledgements

This work has been conducted as a part of the research project No. N N502 337636 in the years 2009-2011.

## References

- [1] T.S. Mujber, T. Szecsi, M.S.J. Hashmi, Virtual reality applications in manufacturing process simulation, *Journal of Materials Processing Technology* 155-156 (2004) 1834-1838.
- [2] Q. Peng, F.R. Hall, P.M. Lister, Application and evaluation of VR-based CAPP system, *Journal of Materials Processing Technology* 107 (2000) 153-159.
- [3] K. Foit, J. Świder, The project of a platform-independent, off-line programming system for industrial robots, *Proceedings of the 7<sup>th</sup> International Scientific Conference "Computer Integrated Manufacturing - Intelligent Manufacturing Systems" CIM'2005, Gliwice-Wisla, 2005*, 62-65.
- [4] J. Świder, K. Foit, G. Wszolek, D. Mastrowski, The off-line programming and simulation software for the Mitsubishi Movemaster RV-M1 robot, *Journal of Achievements in Materials and Manufacturing Engineering* 20 (2007) 499-502.
- [5] J. Świder, K. Foit, G. Wszolek, D. Mastrowski, The system for simulation and offline, remote programming of the Mitsubishi Movemaster RV-M1 robot, *Journal of Achievements in Materials and Manufacturing Engineering* 25/1 (2007) 7-14.
- [6] W.B. Lee, C.F. Cheung, J.G. Li, Applications of virtual manufacturing in materials processing, *Journal of Materials Processing Technology* 113 (2001) 416-423.
- [7] Y.D. Gong, B. Wang, W.S. Wang, The simulation of grinding wheels and ground surface roughness based on virtual reality technology, *Journal of Materials Processing Technology* 129/1-3 (2002) 123-126.
- [8] M.A. Bossak, Simulation based design, *Journal of Materials Processing Technology* 76/1-3 (1998) 8-11.
- [9] L.A. Dobrzański, R. Honysz, Development of virtual light microscope for a material science virtual laboratory, *Journal of Achievements in Materials and Manufacturing Engineering* 20 (2007) 571-574.
- [10] B. Korves, M. Loftus, Designing an immersive virtual reality interface for layout planning, *Journal of Materials Processing Technology* 107/1-3 (2000) 425-430.
- [11] T.P. Grantcharov, Virtual reality simulation in training and assessment of laparoscopic skills, *European Clinics in Obstetrics and Gynaecology* 2/4 (2006) 197-200.
- [12] V. M. Tronnier, A. Staubert, M.M. Bonsanto, C.R. Wirtz, S. Kunze, Virtual reality in neurosurgery, *Der Radiologe* 40/3 (2000) 211-217.
- [13] L. Zhang, B.C. Abreu, G.S. Seale, B. Masel, C.H. Christiansen, K.J. Ottenbacher, A virtual reality environment for evaluation of a daily living skill in brain injury rehabilitation: reliability and validity, *Archives of Physical Medicine and Rehabilitation* 84/8 (2003) 1118-1124.
- [14] K. Foit, J. Świder, The use of networked IPC techniques in hybrid description of a simulation model, *Achievements in mechanical and materials engineering. Proceedings of the 13<sup>th</sup> International Scientific Conference „Achievements in Mechanical and Materials Engineering” AMME'2005, Gliwice-Wisla, 2005*, 225-228.
- [15] K. Foit, Introduction to use virtual reality visualisations in the exploitation and virtual testing of machines, *Journal of Achievements in Materials and Manufacturing Engineering* 25/2 (2007) 57-60.
- [16] The Test ActiveX Program Page, <http://www.tegosoft.com/WebPrograms/TestActiveX/TestActiveX.htm>, accessed 1.03.2010.
- [17] Comodo AV Scanner, <http://personalfirewall.comodo.com/scan/scanner.html>, accessed 1.03.2010.
- [18] Scripting, <http://msdn.microsoft.com/en-us/library/ms950396.aspx>, accessed 1.03.2010.