PROCEEDINGS

of the 18th International Conference on Process Control
Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011
ISBN 978-80-227-3517-9
http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

Sobota, J., Severa, O., Kocánek, M., Čech, M., Balda, P.: Software Background of the 'Mechatronics-Based Rehabilitation at Home' Concept, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 262–267, 2011.

Software background of the 'Mechatronics-based rehabilitation at home' concept

J. Sobota, O. Severa, M. Kocánek, M. Čech, P. Balda

University of West Bohemia in Pilsen
Univerzitní 8, 306 14 Plzeň, Czech Republic
[jsobota,osevera,mkocanek,mcech,pbalda]@kky.zcu.cz

Abstract: In this paper the concept of 'Mechatronics-based Rehabilitation at Home' is presented. The idea is to transfer as much rehabilitation care as possible from hospitals and rehabilitation centres to patients' homes while keeping the effectiveness and quality of rehabilitation process as we know it today. Such concept naturally heavily relies on technological innovation, especially on control systems and communication technologies. Only then is it possible that the patients perform the rehabilitation exercises remotely and autonomously and the rehabilitation specialists and physiotherapists still have enough information about the progress in the patient to adjust or change the treatment as necessary. Software tools addressing the challenging requirements are presented.

Keywords: rehabilitation, remote visualization, 3D HMI, SFC editor, encryption

1. INTRODUCTION

The largest challenge in rehabilitation in the following years will be to increase (or at least keep) the quality, access and efficacy of care. With respect to the increasing numbers of elderly people and growing importance of chronic welfare diseases, it is obvious that the concept of rehabilitation as we know it today is unsustainable.

The rehabilitation therapy is nowadays offered by therapists in healthcare centres or at the patient's home, usually on one-on-one basis. This approach has several disadvantages, especially when the therapy lasts over years:

- Traveling to/from a healthcare centre is time consuming and expensive.
- Patient's daily program is affected (sometimes even controlled) by the schedule of rehabilitation sessions.
- Although a number of rehabilitation aids and equipment is available, which allows autonomous execution of specific exercises, the major part of the therapy is performed by the physiotherapist. This is physically demanding and expensive due to labor costs.

With the ageing population, all the mentioned problems will intensify, as the number of available therapists will relatively decrease.

There is a clear understanding that better healthcare and quality of rehabilitation can be achieved through technological innovation. Nowadays there are various motorized devices available, ranging from simple continuous passive motion devices to more complex systems aimed at neurological patients. But it is common to all the available devices that they are aimed at the use in rehabilitation centres, where a lot of patients use the same machine supervised by a physiotherapist.

According to the foreseen consequences of the ageing population, especially the growing importance of rehabilitation care and its availability, attempts are made to move a significant part of the rehabilitation treatment from hospitals and rehabilitation centres to the patient's home and to minimize the need of physiotherapist's presence during the treatment. In order to make this happen, the rehabilitation devices must be ready for home use.

2. GENERAL STRUCTURE OF A HOME REHABILITATION DEVICE

The keyword for this transition is mechatronics, the devices have to combine mechanical design with electronic components and IT technologies. A general stucture of the 'Mechatronics-based Rehabilitation at Home' (MRH) concept is depicted in Fig 1. The concept is divided into three main parts, which can be further decomposed into several building blocks (or subsystems).

The essential part is the MRH device itself, which is located at the patient's home and is accessible through the Internet. The patient performs a set of predefined exercises, which are controlled and monitored by a control system through appropriate actuators and sensors. The progress of the exercises is displayed by the Human-Machine-Interface (HMI) and the patient is thus informed about how well he/she is doing, which helps to keep the patient focused and motivated. The individual training sessions are recorded for evaluation of the rehabilitation treatment.

The counterpart to the MRH device is the supervision software used by the rehabilitation specialist or physiotherapist. The parameters of individual exercises can be adjusted, which includes the general type of therapy (con-

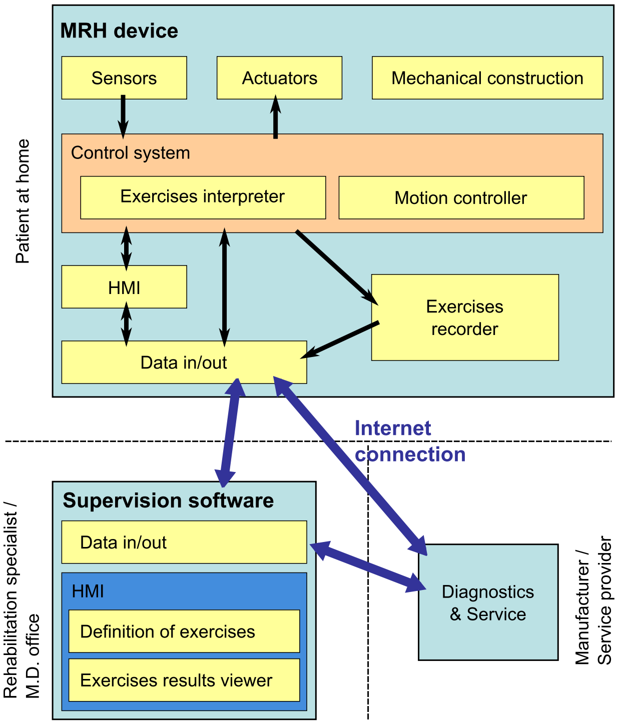


Fig. 1. Building blocks of the 'Mechatronics-based Rehabilitation at Home' concept

tinuous passive motion, active therapy against resistive forces, etc.) as well as more specific parameters like speed of movements, number of repetitions, ranges of motion in individual joints or the resistive forces.

Both these main parts can communicate with the manufacturer or service provider to allow remote diagnostics and service (wear detection, firmware upgrade, etc.)

A specific subsystem is the mechanical design of the MRH device. The construction and mechanical setup is naturally the crucial part of the whole system, it must be easy-to-use for the patient. Even patients with lower motor abilities must be able to position and attach their extremities to the device so that the rehabilitation exercise can be performed safely and efficiently, the risk of performing a non-physiologic movements or even injury due to improper fixation to the device must be minimized. Although this part of MRH device design and development is a demanding and challenging task, it is not addressed in this paper as it is focused on the software part of the MRH concept.

3. SOFTWARE ELEMENTS

As was mentioned above, the software incorporated into the MRH concept plays a significant role. The control system included in the MRH device must not only gather data from sensors and control the actuators, but it must also allow visualization of the data and support communication for remote access over the Internet.

3.1 REX control system

A suitable platform for the MRH device is the REX control system (Balda et al., 2005). The key features of this system include:

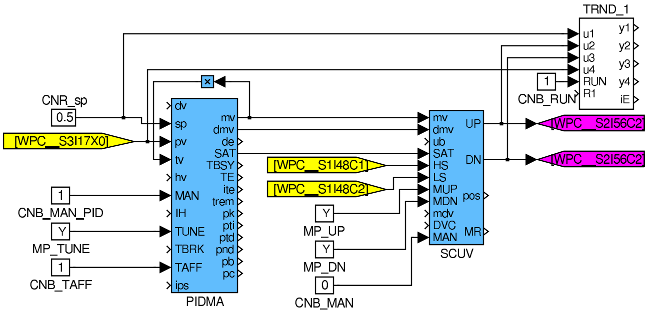


Fig. 2. Example of REX system programming - a PID position control loop for stepper motor

- Control algorithms consist of function blocks from the REX control system library - both standard and advanced control algorithms are included
- Drag and drop programming, simple modification of control algorithms
- Portability to various platforms - various target devices and operating systems ranging from Industrial PCs to microchips
- Compatibility with Matlab/Simulink - the same configuration can be used for simulation in Matlab/Simulink and for the final deployment of the real-time control algorithm (Fig. 2), but no xPC target nor Real-Time Workshop is required
- The control algorithm running on the target platform is accessible through Internet
- Wide range of communication protocols - Modbus, OPC, Ethernet PowerLink, EtherCAT, CAN
- Sophisticated archiving subsystem for storing and retrieving data

All the mentioned properties form a universal platform fulfilling the requirements on software design and development for mechatronics-based rehabilitation devices for home use. The REX control system is well equipped for data acquisition from sensors, processing and storing the data and generating physical motion through the actuators. Majority of the features was described earlier in detail and thus the emphasis is put on new features which were introduced in version 2.0 of the REX control system and have a significant importance for the MRH field.

3.2 HMI development in Java

Human-Machine-Interface is an integral part of any control application. Traditionally, it is composed of several screens displaying current values of important variables from a machine or process. Initially, the screens were quite simple and based on 2-dimensional (2D) graphics. Fifteen years ago, a boom of 3-dimensional (3D) scenes started thanks to the growing power and capabilities of computers (in particular graphics cards and accelerators). Such visualization provides more realistic view of the object existing in real world. The number of elements in the scene is basically not limited, thus one can create a full 3D model of even very complex scene. If the visualization platform together with the data acquisition subsystems satisfy real-time requirements then one gets at each moment an up to date live view of the reality. Let us call it further RT-3D-HMI (real-time). This live view is valuable for remote supervision

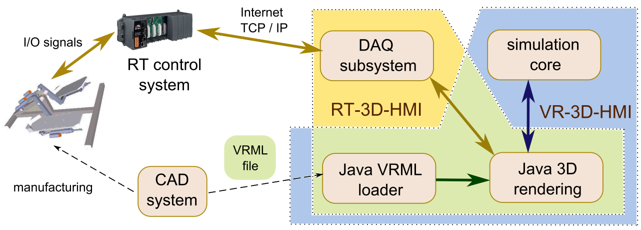


Fig. 3. Scheme of the 3D HMI

of the rehabilitation exercises, the physiotherapist has all information about the exercise in the most natural form as if the patient was performing the exercises in front of him/her.

On the other hand, the 3D-HMI may be supplied by simulation data from a model containing a model of both the MRH device and its control system. Then one gets a fully functioning virtual reality model of the device. Such approach has been proven useful for distance learning (Le Blanc et al., 2005) or training new human operators in industrial practice (Pei et al., 2009). Let us denote such tools as VR-3D-HMI (virtual-reality). The terminology is clearly explained in Fig. 3.

Nowadays, the 3D-modules are available in the majority of leading visualization software (e.g. Indusoft, LabView). Unfortunately, the utilization of mentioned large software packages has several limits. They usually do not allow exporting 3D screens into self-contained Java applets which can be further embedded into a platform-independent web page. Consequently, such HMIs are often executable only on classical desktop computer equipped with Microsoft Windows. Further, the HMI integration with the machine and control system model is quite complicated. Finally, such software is too expensive for simple or embedded applications.

Java platform was already proved suitable for developing interactive web tools [9]. It completely obviates the above mentioned problems. The 3D HMI is based on Java3D package (Oracle, 2011) (rendering package) which is often used to display 3D scenes in Java (Wu and Miao, 2008).

One has always to take two basic parts into account when creating the 3D model. The first part is the 3D rendering engine which takes care of drawing model on the screen. The second part is some SW tool for 3D model creation - usually CAD system (SolidWorks, Autodesk Inventor, CATIA, etc). Note that the CAD system is always used when designing and manufacturing new devices. Hence, using it as a source of model for 3D visualization is quite natural. However, there must exist a 'bridge' between these two parts. In other words, the CAD system must provide export of the 3D model into a format acceptable by the rendering engine.

3D graphics formats Generally, both RT-3D HMI and VR-3D HMI use the same 3D model created by CAD software. For Java platform, it was decided that the VRML (Virtual Reality Modeling Language) will be the final input format for Java 3D renderer. It is well supported and very widely used 3D format. Nowadays it has a successor called X3D which is backward compatible. Although VRML or X3D formats are broadly used on the World Wide Web, only a minority of the 3D CAD software support these formats in default (e.g. SolidWorks). The standard data exchange format supported by almost all CAD systems is STEP. Basically, all of the mentioned formats have a capability to preserve the objects in a natural tree structure which is essential for animation of the individual parts. Let us describe the key features of the formats in more detail.

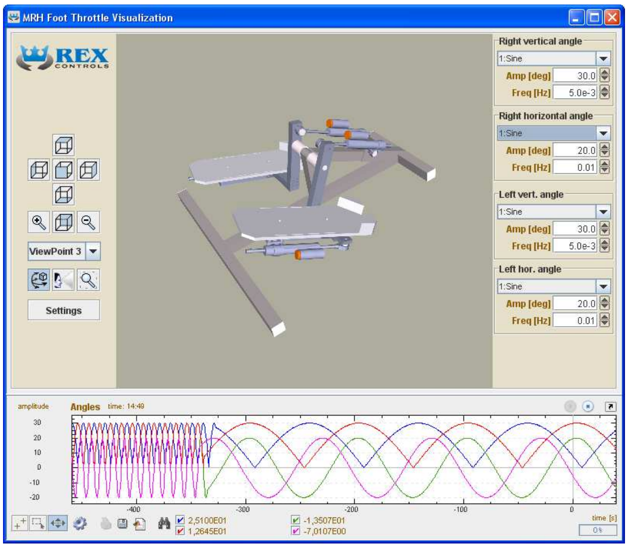
VRML Virtual Reality Modeling (Markup) Language is a standard text file format for representing 3-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It is standardized by ISO (ISO/IEC 14772-1:1997) and maintained by Web3D Consortium. Common extensions are *.wrl or *.wrz for gzip-compressed VRML files.

X3D is a successor of VRML. It is standardized by ISO (ISO/IEC 19775/19776/19777) and also maintained by Web3D Consortium. X3D scene can be stored in a text XML file (*.x3d), Open Inventor-like syntax of VRML97 (*.x3dv) or in binary file (*.x3db). X3D specification support several profiles. These profiles define various levels of capabilities. The big advantage of XML structure is that there exist data parsers (SAX, DOM) in almost all leading programming languages.

Step-file standardized under ISO 10303-21 is the most widely used data exchange form of STEP (Standard for the Exchange of Product model data). This ASCII text file represents data according to a STEP Application Protocols. Step-files are well supported by various 3D CAD software packages. If there is no direct export into VRML available, such format may serve as a bridge between some 3D CAD and VRML. However, one must find a way to convert STEP files into VRML outside the CAD system. One possibility is to use CADExchanger (Lygin, 2010) as a convertor from STEP-File into X3D and then use Flux Studio (Mediamachines, 2007) for further modifications and final export to VRML.

Java 3D package and Java VRML loader Java3D is a set of APIs which is the Java expansion in the three-dimensional field based on OpenGL and D3D. As was mentioned earlier the CAD software provides the 3D model and Java3D can render any 3D scene. Anyways, there is need for a bridge between these two technologies. Java3D VRML Loader is a part of Java3D project. By a few lines of Java code one can parse VRML files and create a tree which represents the whole scene. This tree structure is passed to the Java3D renderer which takes care of the periodical repainting of the scene.

Foot throttle example Fig. 4 shows an example of the resulting RT-3D-HMI for a foot throttle, a rehabilitation machine developed at Fontys University of Applied Sciences for exercises focused on lower extremities, specifically the ankles and knees. The application is divided to four major parts. There is a 3D control panel on the left side followed by the main 3D canvas and the user control panel on the right side. The physiotherapist can control orientation, zoom, view and setting of the 3D canvas through 3D control panel to get a clear overview of the device and position of the individual parts. The user control panel suits remote supervision needs. It is



Right vertical angle
1:Sine
Amp [deg] 30.0
Freq [Hz] 5.0e-3

Right horizontal angle
1:Sine
Amp [deg] 20.0
Freq [Hz] 0.01

Left vert. angle
1:Sine
Amp [deg] 30.0
Freq [Hz] 5.0e-3

Left hor. angle
1:Sine
Amp [deg] 20.0
Freq [Hz] 0.01

☑ 2,5100E01
☑ 1,2645E01
☑ -1,3507E01
☑ -7,0107E00

Fig. 4. HMI for foot throttle rehabilitation device

possible to send commands to the control algorithm, read values and potential error messages. More specifically, the user can control the speed of individual rotations as well as the limit angles based on the patient's abilities. Last noticeable component is a trend panel at the bottom of the application. This trend shows the positions of individual joints on a timescale. A variation of the HMI is prepared for the patient, illustrating the exercise being executed as well as the number of remaining repetitions, estimate of the remaining time, etc. Thus at each moment the patient exactly knows what is happening and what will follow, which greatly improves his/her focus and dedication to the training.

The 3D-HMI entirely depends on data acquired from RT control system, i.e. the visualization exactly reflects the state of reality. Hence, the live connection to the process is necessary (DAQ subsystem in Fig. 3). The connection to the real time device controlling the foot throttle device is ensured by the JavaREX communication protocol (Balda and Čech, 2006). The developed HMI is able to run as an application or applet, so it is possible to embed it into a webserver within the MRH device, which allows remote connection to the device through any web browser with basic Java Runtime Environment.

3.3 Sequential logic programming

Sequential logic is usually a significant part of control algorithms. The MRH applications are no exception. The rehabilitation exercises consist of many individual motion sequences which needs to be switched based on numerous conditions. Plus there is additional logic handling possible error events and states. Even control algorithms of the regulatory type are very often equipped with a sequential logic e.g. for switching of control modes and strategies.

Sequential logic in general can be inserted into control algorithms by several ways. Very simple logic can be programmed in some textual programming language (C, C++, structured text, etc.) directly. As the size of the logic increases, which is the case of MRH applications, the code maintenance becomes more and more complicated and this approach is very error-prone. Hence, several graphical techniques for sequential logic design have been developed. The sequential function chart (SFC) technique (formerly also called Grafcet) is one of them. This formalism is widely accepted by the automation community (vendors, users). Moreover, it is standardized by the IEC 61163-3 standard (IEC, 2003), which describes five languages for programmable logic controllers (PLC) algorithm development.

For MRH applications the need for a graphical sequential function chart editor, which could be integrated into the REX control system development environment RexDraw (Balda et al., 2005) is obvious. The effort to find some open source standalone and smart SFC editor failed. Of course, there are many implementations of SFC editor integrated in commercial PLC development tools but they could not be used for this purpose. This fact initiated the development of an editor which allows the user not only to create various SFC schemes and store them in XML files but also to check their syntax (validate them), compile them for the selected target and to visualize (monitor) the state development of the particular scheme in real-time. The editor is written in C# using Microsoft .NET framework.

Editor architecture Almost overall logic of the SFC Editor can be divided to the three basic categories. The first category contains classes related to the objects which are visible in the graphical user interface (GUI) and should be painted to the canvas. The second category contains classes responsible for handling a chart validation and compilation processes. The last category contains classes implementing the monitoring functionality for algorithm state evolution of the particular chart.

Graphical user interface As mentioned above, the editor is based on Microsoft .NET technology and it is coded in C# language. That is why the user interface consists of familiar windows controls.

Graphical user interface (GUI) of the SFC editor depends on command line arguments passed to the SFC editor. If the editor is called with a parameter determining one of the target devices then it acts as a single document interface (SDI) application shown in Fig. 5. If it is launched as a standalone application (without command line parameters) it acts like multiple document interface (MDI) application.

The standard menus and toolbars provide options for creating new SFC charts, closing, opening, exporting, and printing existing SFC charts. Zooming command (zoom in, zoom out) of the chart view are located in the toolbar at the bottom of the application window.

SFC Editor employs a drag and drop method for creating and editing SFC charts in graphical user interface (inspired by Microsoft Visio or Matlab Simulink editors). Thus, a user can place and order the blocks in the chart arbitrarily.

All the SFC graphical formalism blocks (and some special blocks) are located in the dock panel named ToolBox on the right edge of the editor. A user can easily insert SFC blocks to the chart by simple dragging of the selected block to the designated place in the chart.

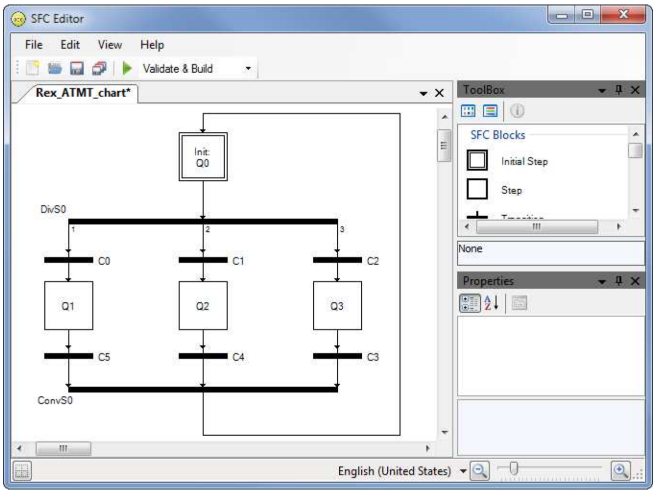


Fig. 5. SFC editor in single document interface mode

SFC block properties can be edited in the Property panel, located by default in the right bottom side of the editor window. The property panel uses smart PropertyGrid component for viewing and editing all public block's properties.

All detected mistakes in the SFC chart and their descriptions are shown in the error list. Similarly, results of the successfully built chart are displayed in an output panel located by default in the left side of the SFC editor.

In addition to such a standard functionality as the printing command of the current chart, which uses a special printing dialog, the SFC editor offers exporting of the chart to the scalable vector graphics (SVG) format.

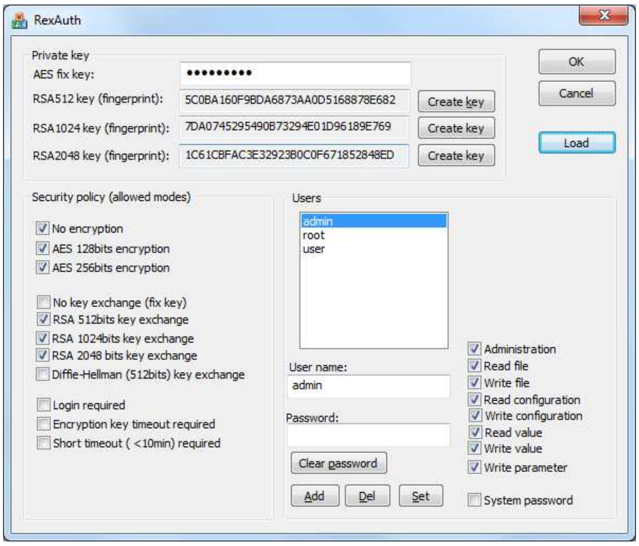
GUI of the SFC editor is mostly inspired by the environment of the Microsoft Visual Studio. The last running mode of the SFC editor, in which the GUI is modified, is the monitoring mode (SFC editor runs as a diagnostic tool). In this case, the additional toolbar with special monitoring controls (Start, Stop, etc.) appears.

3.4 User authentication and encryption

Another important aspect of the MRH device is data security. Because the data stored in the device and transmitted over the Internet are of medical character, it must be considered confidential. Therefore the MRH device must be inaccessible by unauthorized users (hackers) and the transmitted data must be encrypted.

If no security was used, the device would be accessible practically for everybody. The diagnostic program RexView is available for download on the Internet and thus the only necessary information to gain access to the device would be its IP address. The new version of REX control system supports username- and password-based access restriction. Moreover, it is possible to specify rights and privileges for each user.

Further, the communication must be encrypted, otherwise it would be possible to read the data by unauthorized users. To avoid TCP/IP interception, the communication is encrypted using the AES algorithm.



Private key
AES fix key: ••••••••••
RSA512 key (fingerprint): 5C0BA160F9BDA6873AA0D5168878E682
RSA1024 key (fingerprint): 7DA0745295490B73294E01D96189E769
RSA2048 key (fingerprint): 1C61CBFAC3E32923B0C0F671852848ED

Security policy (allowed modes)
☑ No encryption
☑ AES 128bits encryption
☑ AES 256bits encryption
☐ No key exchange (fix key)
☑ RSA 512bits key exchange
☑ RSA 1024bits key exchange
☑ RSA 2048 bits key exchange
☐ Diffie-Hellman (512bits) key exchange
☐ Login required
☐ Encryption key timeout required
☐ Short timeout (<10min) required

Users
admin
root
user

User name: admin
Password:

☑ Administration
☑ Read file
☑ Write file
☑ Read configuration
☑ Write configuration
☑ Read value
☑ Write value
☑ Write parameter
☐ System password

Fig. 6. RexAuth - configuration of security subsystem in the REX control system

Security subsystem in the REX control system The security measures of the REX control system are configured in the RexAuth program as illustrated in Fig. 6. It is possible to allow more encryption levels, the available options are:

- `No encryption` - Data communication is not encrypted.
- `AES 128-bit encryption` - Data is encrypted by AES (Rijndael) algorithm with 128-bit key
- `AES 256-bit encryption` - Data is encrypted by AES (Rijndael) algorithm with 256-bit key

The type of encryption is proposed by the client (supervision software) at the physiotherapist's computer and the server (MRH device) accepts or declines it according to security settings. If encryption is enabled, it is necessary to define how the decryption key will be transferred:

- `No key exchange` - A fixed AES key is used; must be specified in the Private key field.
- `RSA 512-bit key exchange` - The client generates the key for the AES encryption and sends it to the server encrypted by the RSA algorithm (the key for RSA algorithm is 512 bits long). The public key of the server is used for RSA encryption.
- `RSA 1024-bit key exchange` The same as above, only the RSA key is 1024 bits long).
- `RSA 2048-bit key exchange` The same as above, only the RSA key is 2048 bits long).
- `Diffie-Hellman key exchange` A widespread approach where the random key generation is distributed among the client and the server. There is no need to generate the private key in advance.

Configuration of users and their rights is quite straightforward, each user has its own privileges:

- `Administration` - Defines the right to perform special operations like pausing and resuming the control algorithm, performing diagnostic tasks, etc.
- `Read file` - Allows reading of files from the target device.
- `Write file` - Allows writing files to the target device.

- `Read configuration` - Allows reading the whole configuration file of the control system.
- `Write configuration` - Allows writing of the whole configuration file of the control system, i.e. completely change the control algorithm.
- `Read value` - Allows reading of individual signals and parameters in the control algorithm.
- `Write value` - Allows writing of individual signals of the control algorithm (setting them to constant values).
- `Write parameter` - Allows writing of individual parameters of the control algorithm. This right includes the `Write value` right automatically.

4. CONCLUSION

In this paper the concept of 'Mechatronics-based Rehabilitation at Home' was presented. The idea to transfer as much rehabilitation care as possible from hospitals and rehabilitation centres to patients' homes can be put into reality if modern technology is used. A key role in this concept play the communication and control systems. A control system suitable for home-rehabilitation under remote supervision was presented, especially the 3D visualization, graphical programming and communication security aspects were highlighted.

5. ABOUT THE MRH PROJECT

The 'Mechatronics-based Rehabilitation at Home' is a joint initiative of 5 regions of the European Union under the 'Innovation for Welfare' framework (I4W, 2011). A collaborative platform has been created to exchange experience and expertise. The project members are:

- Fontys University of Applied Sciences (Eindhoven, the Netherlands)
- Technical University of Catalonia (Barcelona, Spain)
- University of Brescia (Brescia, Italy)
- Upper Austria University of Applied Sciences (Linz, Austria)
- University of West Bohemia in Pilsen (Pilsen, Czech Republic)

More information can be found at the project website (MRH, 2011).

6. ACKNOWLEDGEMENTS

This paper was supported from the funds of the European Union (ERDF/INTERREG IVC) through the 'Innovation for Welfare' project (subproject I4W-22-MRH). This support is gratefully acknowledged as well as the reviewers' comments.

REFERENCES

Balda, P., Schlegel, M., and Štětina, M. (2005). Advanced control algorithms + Simulink compatibility + Real-time OS = REX. In *Preprints of the 16th IFAC World Congress*. Prague, Czech Republic.

Balda, P. and Čech, M. (2006). Java interface to REX control system. In *Proceedings of Process control 2006*, 1–8. Pardubice.

I4W (2011). Innovation for Welfare, project webpage. www.innovation4welfare.eu.

IEC (2003). IEC 61131-3, Edition 2.0, Programmable Controllers - Part 3: Programming Languages.

Le Blanc, A., Bunt, J., Petch, J., and Kwok, Y. (2005). The virtual learning space - an interactive 3D environment. In *Proceedings of 10th International Conference on 3D Web Technology*, 93–102. Bangor, Wales.

Lygin, R. (2010). CAD Exchanger. http://www.cadexchanger.com/.

Mediamachines (2007). Mediamachines web blog. http://mediamachines.wordpress.com/flux-player-and-flux-studio/.

MRH (2011). Mechatronics-based Rehabilitation at Home, project webpage. www.mrh-project.eu.

Oracle (2011). Project Java3D. https://java3d.dev.java.net/.

Pei, Y., Wu, H., and Zong, R. (2009). Simulated joint robot-hand system based on Java3D. In *Proceedings of the 2009 2nd international congress on image and signal processing*, volume 1-9, 2805–2808. IEEE.

Wu, Y. and Miao, Z. (2008). Construction of the virtual environment based on Java3D. In *Proceedings of IET 2nd International Conference on Wireless, Mobile and Multimedia Networks, ICWMMN 2008*, 512–515. Beijing.