VROBO: A Virtual Robotics Platform for use in Robotics Education and Research

by

Giorgos A. Demetriou

Student Number: 1821

A thesis submitted to the faculty of undergraduate studies of
Frederick University
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Advisor

Dr. Georgios A. Demetriou, *email*: g.demetriou@frederick.ac.cy
School of Engineering and Applied Sciences
Department of Computer Engineering and Computer Science
Frederick University, Lemesos, Cyprus

Signature/Approval Page

This thesis by <insert name> submitted to Frederick University in partial fulfillment of the degree of Bachelor of Science on 28 April 2010 has been examined by the following faculty and it meets or exceeds the standards required for graduation as testified by our signatures below.

____
Dr. Georgios A. Demetriou, Advisor

____
<Committee Member 1 - Type name>

____
<Committee Member 1 - Type name>

____
<Committee Member 1 - Type name>

Biography

Dr. Giorgos A. Demetriou received his Ph.D. in computer science and his M.S. in computer engineering from the Center for Advanced Computer Studies at the University of Louisiana at Lafayette in 1998 and 1994, respectively. Since January of 2006 he has been with the Computer Engineering and Computer Science Department of Frederick University, Lemesos, Cyprus. Before that he was with the Computer Engineering Department of Purdue University, Fort Wayne, Indiana, and with the Computer Science department of the University of Southern Mississippi-Gulf Coast (USM-GC), Long Beach, Mississippi. At Purdue University he was a visiting assistant professor of computer engineering. At USM-GC, he served as an assistant professor, as the director of the Robotics and Graphics Laboratory, and as the coordinator for the computer science graduate and undergraduate programs. Research interests include Intelligence Systems, Robotic Systems, and Robotic Mobile Systems. His teaching interests include, Robotic Systems / Automated systems, Intelligent Systems, Control Systems, and Computer Graphics.

Acknowledgements

The author would like to thank his advisor, Dr. Georgios A. Demetriou, for his guidance, advice, and encouragement toward successful completion of this project. Additional thanks go to…. (thank anyone else you feel that have helped you in this project) ….

Abstract (sample)

Robotics will continue to become intertwined with our daily lives, which will ultimately result in the need for more highly trained individuals to both operate and repair robotics equipment. The ability of academics and researchers to supply individuals capable of performing these tasks will be a substantial challenge in the future. Currently, there are few individuals available to perform these highly skilled tasks; furthermore, institutions and programs for training these individuals are scarce. All of the various sectors of growth point to an increase in the need for robotics technicians in the near future. With this increase will come the need for educational programs to supply the technical skills and training in the various areas of robotics research and development. To keep up with this demand institutions of higher learning will have to adapt and come up with diversified programs for robotics education while overcoming spatial, temporal, and budget limitations. This paper discusses the impediments that face the researcher and academic institutions when trying to implement such training programs and explains the ability of Virtual Modeling and Simulation (VM&S) systems to mitigate such problems. In addition, a solution system, Virtual-Robots (VROBO), is developed to demonstrate the effectiveness of the approach, and its constituent parts are analyzed to show the mapping between the part and the impediment that it tries to eliminate.

Table of Contents

Table of Figures

1. Introduction

The idea of robotic mechanisms has fascinated humans since the first machines were built. Before the first robot was even constructed, the popular view of robotics consisted of human-like machines that could walk, talk, and perform as well as their human counterparts [4]. Despite this popular view of humanoid robots, industrial robotics has been the most dominant area of research and growth in the years that followed. Even today, sophisticated humanoid type robots are still far away from realization. Their industrial type counterparts still constitute the largest percentage of robotics sales and research [23], [13].

Our need for robotics will continue to grow, as we become more emerged in technology and prices for robotic manipulators decrease. The International Federation of Robotics (IFR), a leading authority in the robotics industry, estimated that worldwide robotics sales were up 15% in the year 2000 [23]. Even though the majority of robotics sales will continue to be generated by manufacturing industries such as automotive companies, we are beginning to see robotics spread into other areas including military applications, and aids for home and work use [23]. Some recent examples of the growth of industrial robotics into other areas include the recent use of robotics in packaging the new European currency and the development of a robotic system that de-bones pork loin [18], [11].

Bla bla …

1.1. Problem Domain

Bla, bla, bla….

1.1.1. Next Section

More bla bla …

Bla, bla, bla….

1.2. Physical Robot Problems

More bla bla…

2. Previous Work (Literature Review)

Most existing and future robotic applications are geared towards the military, workspace, and home. Military use promises to be a strong source of growth for the robotics community. Since its formation in 1990 the federally funded Joint Robotics Program (JRP) has received substantial funding averaging around 12 million dollars per year. The main purpose of the program is to develop autonomous and remotely operated robots for use in surveillance and reconnaissance. The military sees benefits that robotics have to offer as remotely operated vehicles for surveillance of hostile areas and remote disarming of explosives [7]. The first area where robots are making our tasks easier is the workplace. One work area that has promising growth is in the aid to medical technicians. Various robots are undergoing trials ……..

2.1. Subsection

……

2.2. Subsection

……

3. Methodology

Before the implementation of VROBO certain criteria were established to be used as guidelines during the design and evaluation phases. The criteria are shown below and they are the same as the criteria used to evaluate virtual modeling and simulation:

- Reduced Cost
- Flexibility
- Complexity
- Portability
- Network/Internet capabilities.

VROBO's architecture is shown in the block diagram in Figure 1, and the systems functionality is as follows:

- The user selects a specific robot to program.
- The programming is done using a generic programming language that was developed specifically for this system and is based on existing robotic programming languages.
- The program is simulated on the robot that is displayed on the GUI.
- The program can be modified and tried again until the user is satisfied with the results.
- Once the program is complete, the user can download the program to the controller of the actual robot being simulated.
- During the download phase, a translation is done from the VROBO programming language to the specific language of the actual robot.
- Finally the program can be executed on the real system.

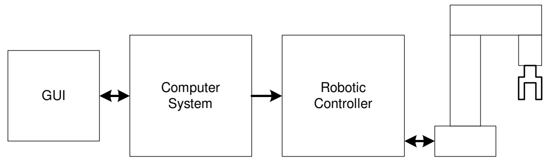

Figure 1. Block diagram of VROBO

The GUI, was built using current Java technologies. The interface consists of four main areas: the Controller Selection List Box (CSLB), the Controller Panel (PL), the Robot Selection List Box (RSLB), and the Robot Panel (RP). When the application is first executed ……….

3.1. Tool Selection and Meeting Cost, Portability and Networking Criteria

In selecting technologies to implement the system, it was necessary to pick tools that would maximize realization of the goals at hand. Some of the choices may actually meet an entire goal, while others just encouraged the success of a compliant system. Nevertheless, by the selection of tools, the system was able to realize large progress for the cost, portability, and networking criteria.

Since the system is based on freely available Java technologies, it was possible to reduce the costs of the developer and the user of the system. The Java components consisted of both core Java technologies and the use of add-on libraries. The Java3D API provides the ability to build customized scene graphs that can be rendered into Java based interfaces using native OpenGL calls on UNIX based and Windows based systems. In addition to the OpenGL binding, support for native DirectX use is available for Windows users [26].

High levels of portability were achieved through the selection of Java technologies. This was possible due to the availability of JREs and Java3D implementations for both UNIX platforms and Microsoft Windows. Furthermore, since OpenGL implementations are provided on most platforms, it is possible for the OpenGL Java3D binding to be used on either UNIX or Windows platforms also [1].

Java itself was developed to take advantage of networking from the beginning. In addition, Java makes it easier to make use of networks and supplies different layers to suit different needs. For example, it provides high-level APIs to the user for HTTP and FTP protocols while still giving access to lower level programming interfaces such as sockets [9]. Not only does the Java environment provide mechanisms for protocol communications, it also provides ways of downloading remote code to be executed either in the Browser or thru the use of Java Web Start technologies.

3.2. System Design and Implementation

In the previous section, three of the criteria were discussed. The entire criterion for portability was realized; however, the criteria of cost and networking were only partly fulfilled by choosing Java based tools. In the case of cost, the only additional gesture that must be performed is the release of the software as open source. The open source paradigm would allow individuals to freely use and modify the code without paying licensing fees or having other types of costs incurred [16]. However, that still leaves the criterion of networking to consider in the design and implementation of the system. This criterion, accompanied by the criteria not directly affected by the tool selection, results in making careful design decisions that will increase the overall flexibility, decrease the technical complexity, and take advantage of the networking capabilities that the Java API has to offer [12].

3.2.1. Flexibility

The system provides a number of controllers and articulated figures via the GUI. These controllers and articulated figures can be mixed and matched as needed which in itself provides a great deal of flexibility. The CSLB currently provides the user with three different controllers.

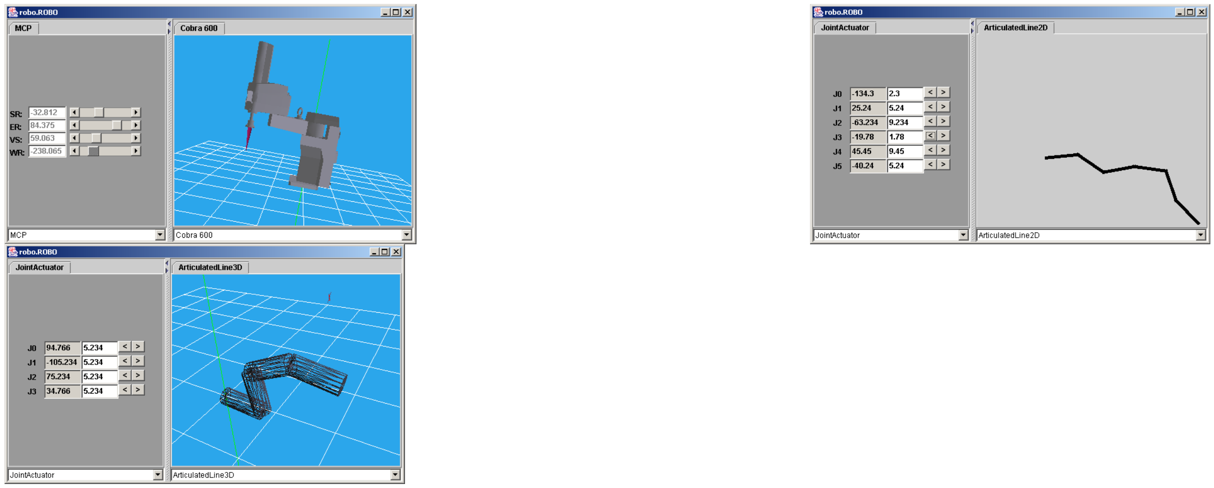

Figure 2 Screen capture of VROBO using the Joint Actuator controller to articulate the Cobra 600 robot, the ArticulatedLine2D, and the ArticulatedLine3D respectively.

Figure 2 shows the MCP controller with the Cobra 600 robot and the ArticulatedLine2D and ArticulatedLine3D. Each of these controllers can be selected at anytime during the duration of the program.

…………

4. Experimental Results

4.1. Case Study 1

……..

5. Conclusion

The VROBO system meets most of the criteria considered under the new system development. Due to the use of freely available JAVA application programming interfaces it was possible to keep the cost of system development to zero. In addition, the system provides the ability to use pre-constructed controllers and articulated figures, create additional controllers and articulated figures via extension of JAVA interfaces, and the ability to do offline programming of the robot with the built in language. These features of the system demonstrate the flexibility of the system. Furthermore, the complexity of the system is provided in a layered approach with the user only needing to manipulate the articulated figures through the supplied controllers. The next layer of complexity is the use of the offline programming capabilities of the system. The user who needs more functionality than these two provide, can extend the system to create new controllers, robots, and work cells. The reliance on Java APIs provides the platform-independent capabilities of the system. This is possible because of the multiple platforms that provide Java Runtime Environments, which the software system developed is capable of utilizing. Finally, increased networking support is demonstrated thru the use of applets and the possibilities that are possible by using the networking packages that are available in the JAVA application-programming interface. Since the system that was developed significantly reduces the barriers that impede the development of robotics programs, it is more likely for these programs to be implemented and utilized to meet the current and future needs of the robotics industry.

6. References

[1] Angel, Edward. Interactive Computer Graphics: A Top Down Approach with OpenGL. Reading: Addison Wesley Longman, Inc., 2000.

[2] "The Availability of Low-Cost Prototyping." Prime Faraday Technology Watch. November 2001. http://www.primetechnologywatch.org.uk/documents/low-cost-prototyping.pdf

[3] Bouvier, Dennis J. "Chapter 2: Getting Started." Getting Started with the Java API. Sun Microsystems, Inc., 2001.

[4] Brooks, Rodney. "Humanoid Robots." Communications of the ACM. Vol. 45 No. 3 (March 2002): 59-63.

[5] Brutzman, Donald P. "Dissertation: A Virtual World for an Autonomous Underwater Vehicle." Naval Post Graduate School. Dec. 1994. 2 Nov. 2002. <http:/www.stl.nps.navy.mil/~brutzman/dissertation/>

[6] Fuller, James L. Robotics: Introduction, Programming, and Projects. New Jersey: Prentice-Hall, Inc., 1999

[7] "Funding for U.S. Joint Robotics Program expected to rise." Aerotech News and Review. 14 Jan. 2002. 23 May 2002. http://www.aerotechnews.com/starc/2002/011402/robotics.html
[8] Gaspari, A. L., and N. Lorenzo. "State of the Art of Robotics in General Surgery." Business Briefing: Global Healthcare. University of Rome. 2002 http://www.wmrc.com/businessbriefing/pdf/healthcare2002/reference/18.pdf
[9] Harold, Elliotte R. Java Network Programming. Cambridge: O'Reilly and Associates, Inc., 1997.
[10] "A History of Educational Robotics." General Robotics Cooperation. 28 May 2002 http://www.edurobot.com/stories/overview.html
[11] "IPA Delivers Automated System to Debone Pork Loin." Robotics Newsletter. No. 44 Dec. 2001 and Jan. 2002. International Federation of Robotics. http://www.ifr.org/newsletter//news/news44.htm
[12] Lambert, Allan and Demetriou, Georgios. "An Extensible Object Oriented Virtual Robotics Development Platform for use in Robotics Education and Research", 2003
[13] "Learn More: History." Robotics Research Group. University of Texas at Austin. 29 May 2002. http://www.robotics.utexas.edu/rrg/learn_more/history/
[14] Merriam, Charles. "Comp.robotics FAQ." Aug. 2001. Dec. 2002. http://www.truegift.com/robots/
[15] Michel, Oliver. "Khepera Simulator version 2.0 User Manual." University of Nice. 1 Mar. 1996. 22 Jan. 2003.
[16] Minansi, Mark. Linux for Windows NT/2000 Administrators. San Francisco: SYBEX, Inc., 2000.
[17] Niku, Saeed B. Introduction to Robotics: Analysis, Systems, Applications. New Jersey: Prentice Hall, 2001.
[18] "Robots Play a Role as Europe Changes to Euro." Robotics Newsletter. No. 44 Dec. 2001 and Jan. 2002. International Federation of Robotics. http://www.ifr.org/newsletter//news/news44.htm
[19] "RoboWorks Frequently Asked Questions." Newtonium. 9 Dec. 2001. 1 Oct. 2002. http://www.newtonium.com/public_html/Products/RoboWorks/RoboWorks_faq.htm
[20] "RoboWorks Support." Newtonium. 9 Dec. 2001. 2 Sep. 2002. http://www.newtonium.com/public_html/Products/RoboWorks/RoboWorks_support.htm
[21] Rosenblatt, M. and Choset, H. "Designing and Implementing Hands-On Robotics Labs." IEEE Intelligent Systems. Vol. 15 No. 6 (November-December 2000): 32-39.
[22] Smith, Nathan, Christopher Egert, Elisabeth Cuddihy, and Deborah Walters. "Implementing Virtual Robots in Java3D using a Subsumption Architecture". State University of New York at Buffalo. Nov. 1999 http://www.cs.buffalo.edu/~egert/papers/webnet99ns.pdf
[23] "Solid Growth for Robotics in the Year 2000." Robotics Newsletter. No. 41 Mar. 2001. International Federation of Robotics. http://www.ifr.org/newsletter//news/news41.htm
[24] van der Smagt, Patrick. "Simderella: a Robot Simulator for Neuro-controller Design." Jan 1994. Department of Computer Systems, University of Amsterdam. http://www.robotic.dlr.de/Smagt/papers/Sma94b.ps.gz
[25] "Simderella 2.1". 6 May 1998. http://www.robotic.dlr.de/Smagt/software/simderella/software/simderella.2.1.tar.gz
[26] "What Can Java Technology Do?" The Java Tutorial. Sun Microsystems. 2003. 2 Feb. 2003. http://java.sun.com/docs/books/tutorial/getStarted/intro/cando.html

Appendix A – Test Data

Note to Students: This section is optional, but if included in the report, the data should be tabulated using the Microsoft Word *Table* construct. A page of *representative* test data (single spaced) usually helps bolster any claims you make about the resutls of your project, and can be useful as a starting point for in-depth discussion.

Appendix B – Technology Transfer Plan

Note to students: A technology transfer plan is an optional one or two paragraph summary of how you plan to introduce the results of your project to business or industry. For example, in the case of the mythical *Astute* grading system, one might list (a) corporate contacts that are interested in receiving more information about the software (list only companies and individuals that you have actually spoken to or contacted); (b) possible future applications that your project results could address, and how you plan to develop such applications; and (c) market potential for your project results, if you have such information. Don't include a lot of wordy nonsense, just a tight summary.

Appendix C – Code

Here you must include your code and other necessary information you thing is relevant.