Software Review

R. Téllez* and C. Angulo

rtellez@lsi.upc.edu, cecilio.angulo@upc.edu

Technical University of Catalonia

Rambla del'Exposició s/n E-08800 Vilanova i la Geltrú

Barcelona, Spain

30th August 2006

Webots simulator 5.1.7

Cyberbotics Ltd., (2006).

Available at different versions and prices, *3600 CHF* ($2940) for the PRO version, normal price.

* Corresponding author

1 Description

Webots^{TM} [1, 14] is a commercial software for the simulation and prototyping of mobile robots. It is developed and supported by Cyberbotics Ltd. [1], a leading company in simulation software founded in 1998 as a spin-off from the Micro-Computing and Interface Lab (LAMI) of the Swiss Federal Institute of Technology in Lausanne (EPFL).

[INSERT FIGURE 1 HERE]

The Webots software allows the user to create 3D virtual worlds for the simulation of robots and their environments, including physics properties like mass distribution, friction coefficients, inertias, etc. Simulated robots can have different locomotion schemes, including wheeled, legged, swimming or flying robots, and can be equipped with several sensor and actuator devices like IR distance sensors, motor wheels, cameras, servos, touch sensors, grippers, emitters and receivers, and many more. Webots is provided with a number of models and transfer libraries for some commercial robot, like the Sony Aibo, LEGO Mindstorms, Khepera, Hemisson and e-puck robots.

Webots can be used for different types of research, like mobile robot prototyping for the automotive industry, multi-agent research in collaborative mobile robots groups, or adaptive behavior research with genetic evolution, neural networks and artificial intelligence techniques. It also generates impressive looking result presentations by using a built-in module which obtains screenshots and movies from the simulations.

Developed since 1996 and available for Windows, Linux and Mac OS X operating systems, Webots has become a reference software used by over 300 universities and research centres worldwide. The first version was released in 1997 (v0.4.alpha) [13] and has been yearly updated – Webots v1.0 (1998), v2.0 (1999) v3.0 (2000) –. Version 4.0.beta1 was released on February 2003 and included a working cross-compilation for the LEGO Minstorms RCX. Released on January 2005, version 5.0.0 has been updated integrating a multi-platform binary server for Aibo ERS7/ERS210 and the Aibo ERS7 cross-compilation system [12]. Current version is Webots 5.1.7 (as of June 29th, 2006). Since Webots was originally developed as a research tool, a full upward compatibility is not always ensured between versions, however continuous users support for technical questions concerning the simulator is offered by Cyberbotics, and an active *webots-users* mailing list [10] for discussion among Webots users proves to be a useful tool.

Cyberbotics promotes Webots based *Roboka* [7] robot programming contest, which is a competition held on the Internet where a free version of the mobile robot simulation software is used. This allows the contestants to program the robots in Java for a Judo competition between two humanoid robots, that are models [15] of the Sony QRIO robot.

2 The simulation engine

[INSERT FIGURE 2 HERE]

Webots simulations are specified by two main types of files: the *world* files and the *controller* files. A *world* file contains a description of the scene, including the objects and the robots that will evolve into the simulation. *Controller* files are C/C++ or Java programs provide the control programs for the robots in the scene. Webots is distributed with about 70 sample world files and associated controller files including a large number of simulated objects and robot models, like the Hoap-2 robot, Aibo, Hemisson, Khepera, Pioneer or Sojourner among others (see figure 1). New robots and environments can be designed by the user from a graphical user interface (called the *Scene Tree*, see figure 2) and a large library of sensors and actuators. The interface is not intuitive and requires some time to be mastered. However it is still a better tool to use than directly text editing files, like other simulators do (for example the Player/Stage [6] or the 3D Simulator BOB [8]), since it provides immediate visual feedback.

The division of the simulation into world files and controller files allows the simulation of multiagent systems, where several robots interact at the same time, each one running its own controller program (see figure 3). This capability is well suited for swarm robotics or robot soccer games. However, since all controller programs are executed on the same computer as different processes, the simulation speed quickly decreases as the number the robots increases. This solution obtains lower performance results than the client/server solution implemented by other simulators (for example the Player/Stage simulator [6] or the Java Mobile Robots simulator [4]), where the server runs the world simulation and the clients, which may be at different computers, run the robots controllers and connect to the server for interaction with the world.

The simulation is displayed on a 3D OpenGL window on which the observation point can be moved around with the mouse. This kind of representation results in a nice look and an easy understanding of the simulation. As a drawback, the simulation runs slower than in other simulators like Mobotsim [5] and Yaks [11] (compared when running Khepera robots), even when the simulation is switched to the *fast-simulation* mode, which avoids the graphics rendering in order to run faster.

A very convenient capability is the drag-and-drop feature using the mouse over the pre-defined objects. Once the objects of the scene have been defined in the *Scene Tree*, they can be manipulated with the mouse, allowing for manual modification of the simulation scene by rotating and changing the position of any object, even during running time.

The Webots simulation engine can perform both, kinematic and dynamic simulations, but it is up to the user to decide if the simulation should or should not include physics simulation. Kinematic simulations do not take into account the physical properties of the simulated elements (such as weight, inertia or friction), and therefore they run faster. For the simulation of physical interactions, Webots relies on the ODE (Open Dynamics Engine) library. Since it is a very powerful library for physics simulation, Webots is capable of accurate physic interactions between robots and their environments. Furthermore, Webots allows the user to introduce custom physics libraries adding ODE forces as well as bodies and joints that are not simulated in a standard world, like the modelling of hydrodynamic forces, wind,etc.

[INSERT FIGURE 3 HERE]

In our experiments, simulation on dynamic and kinematic situations have been used, and the difference of perceived speed can reach several orders of magnitude. As a conclusion, physics simulation should be avoided when it is not necessary. Physics simulations are accurate enough for most of the experiments and the limitations are set only by ODE's own capabilities.

3 Robot controller programming environment

Webots can execute a controller for each robot, written in C/C++ or Java, to control the actions of its associated robot. This means that a world can contain several robots, each one having its own controller. In addition, a special controller called the *supervisor* can be implemented by the user. The *supervisor* is a special controller that plays the role of a super-user which monitors the whole simulation. It can act on any of the simulation elements and modify their attributes, include new elements into the world or change the simulation setup. This controller is particularly useful on evolutionary simulations, where the elements of the simulation must be returned to their original (or random) positions after each evaluation. The supervisor can communicate with any robot controller via a simple messaging system.

Webots includes an embedded controller editor and compiler that allows the creation of controllers from within Webots itself. Our experience shows that it is less confusing to use an external editor and then compile using the provided *Makefile*, rather than using the built-in editor, especially if you have several controllers being handled at the same time.

4 Real robot control

Once the user is happy with the controller and its result in the simulation, there is the option to directly transfer it to the real robot and test it on a real environment. Webots supports the direct transfer of controller code to real hardware for several commercial brands of robots, which include Lego Mindstorms, Khepera, Koala, Aibo, e-puck and Hemisson. It also allows the user to create his own transfer system for any custom robot, even though this process is not extensively documented.

Real robot control can be performed in two different modes: cross-compilation, where the Webots controller is cross-compiled and transferred to the real robot for execution onboard; and remote-control, where the controller is executed on the Webots simulator but the commands are sent to the real robot through a connection from the computer to the robot (wireless or direct cable). Not all the simulated robots support these features, though. For example, the Khepera robot supports both modes by using a serial cable attached to the robot. The Aibo robot also supports both modes by using a wireless connection. The Lego Mindstorms robots only support the controller transfer. The *User Guide* documentation provides instructions on how to implement your own remote-control or cross-compiler system for any extra robot that you may have designed.

In our case, we have extensively used the Aibo transfer feature and found that the controllers behave almost exactly in the simulation and in the real Aibo [16]. Direct control of the robot from the computer is also possible, but at present the wireless communication is not fast enough to be considered real time control.

Specially for the Aibo robot, there exists a third way of controlling the real robot from the simulator, which consists of using the URBI scripting language. URBI [9] is a scripting language developed by Gostai SAS [2], which has been integrated into the Webots package for the control of simulated and real Aibo robots. A URBI script for Aibo can be tested and executed on the simulator and then directly executed on an Aibo robot equipped with a URBI server, with no change on the code.

5 Documentation

The documentation is composed of two books: the *User Guide* explaining how to get started on using the package, and the *Reference Manual* describing in detail all the nodes for designing the *world* files and the API for controller programming. They are provided in both, electronic and paper format.

Documentation is the weakest point of this software package. Although the content is huge, it is sometimes a little bit confusing on some of the aspects of the software. Multiple options in this software should be better explained. Deeper explanations on how to create custom cross-compilation libraries and remote control of robots, or how to use the switching feature to fast 2D mode, would be very useful, as well as clearer descriptions on the use and modification of some simulation parameters and elements. This lack of information implies an extra job to the user which has to discover what the manual means. However, this drawback is compensated to some extent by a huge number of examples included in the package, and a good customer support provided through email or a public forum.

6 Some research applications

A permanent academic collaboration is maintained by Cyberbotics with the Biologically Inspired Robotics Group (BIRG) and the Swarm Intelligent System Research Group (SWIS) of the EPFL through the Swiss CTI technology transfer program. A list of related research projects include topics like,

Robotics. An amphibious snake-like robot; a frequency-adaptive hopper robot; bio-inspired vision-based flying robots; humanoid robot simulation as EPFL robotics research

Swarm intelligence. A three lane motorway inspection system; Particle Swarm Optimisation (PSO) for unsupervised robotic learning; distributed odor source localisation as EPFL-CalTech collaborations.

Others. Multiagent system's research at University of Southern California or University of Edinburgh; adaptive behavior based robots at University of Bristol; vision-Centric Robots at Carnegie Mellon University; Emergent robotics at Sony CSL; resource allocation at CalTech

Links to several of these research projects and other are available from [1].

Since January 2006, Cyberbotics takes part in the ICEA project [3] – Integrating Cognition, Emotion and Autonomy –, a four-year european project aiming at bio-inspired cognitive robotics and embodied cognition. Cyberbotics is in charge of developing the ICEAsim, a rat-like simulation platform based on the Webots robot simulation toolkit that will be available for free as a standard tool for research. Some interesting features will be simulated like active whiskers or internal metabolism.

In our case, we have used the software in a complex setup, for the validation of our research. The successful simulation involved the evolution of a distributed controller composed of 21 Continual Time Recurrent Neural Networks that should coordinate to control a dynamic walking behavior on an Aibo robot. Once the simulation evolved the correct behavior on the robot, the controller was directly transfered to the real robot with almost no loss of performance [16].

7 Conclusions

Webots is a good simulator providing all the required tools for the generation of professional scientific and industrial results. It can save a lot of time in prototyping and evaluating robot behaviors. Moreover, it allows researchers to present their results in an impressive-looking and professional way.

In very specific cases, Webots is outperformed by other simulators which are more adapted to those particular situations. However, Webots is the only simulator available that is capable of simulating any robot, on any situation, with easiness of use and quick start-up. Eventhough you may encounter some minor bugs or annoyances occasionally, the Cyberbotics team usually solves them within a couple of days.

To remain on the bleeding edge, Webots is continually evolving and upgrades are released frequently, sometimes leading to instabilities. However, our experience is that the support team provides an excellent e-mail support to the customers and could help them finalising their experiments very rapidly.

References

[1] Cyberbotics Ltd. http://www.cyberbotics.com/.

[2] Gostai sas. http://www.gostai.com.

[3] ICEA project. http://www2.his.se/icea/default.htm.

[4] Java mobile robots. http://www.rvg.ua.es/jmr/eng/.

[5] Mobotsim, mobile robot simulator. http://www.mobotsoft.com/mobotsim.htm.

[6] Player/stage simulator project. http://playerstage.sourceforge.net.

[7] Roboka contest. http://roboka.org/.

[8] Simulator bob. http://simbob.sourceforge.net.

[9] Urbi scripting language. http://www.urbiforge.com.

[10] Webots users. http://groups.yahoo.com/group/webots-users/.

[11] Yaks, yet another khepera simulator. http://www.his.se/iki/yaks.

[12] Holh, L., Téllez, R. A., Michel, O., and Ijspeert, A. J. (2006) Aibo and webots: simulation, wireless remote control and controller transfer. *Robotics and Autonomous Systems*, in Press.

[13] Michel, O. (1998) Webots: Symbiosis between virtual and real mobile robots. *VW '98: Proceedings of the First International Conference on Virtual Worlds*, London, UK, pp. 254–263, Springer-Verlag.

[14] Michel, O. (2004) Cyberbotics Ltd - Webots^{TM}: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, **1**, 39–42, ISSN 1729-8806.

[15] Mojon, S. (2003) Realization of a physic simulation for a biped robot. Semester project report, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne.

[16] Tellez, R., Angulo, C., and Pardo, D. (2006) Evolving the walking behaviour of a 12 dof quadruped using a distributed neural architecture. *2nd*

International Workshop on Biologically Inspired Approaches to Advanced Information Technology, Springer-Verlag.

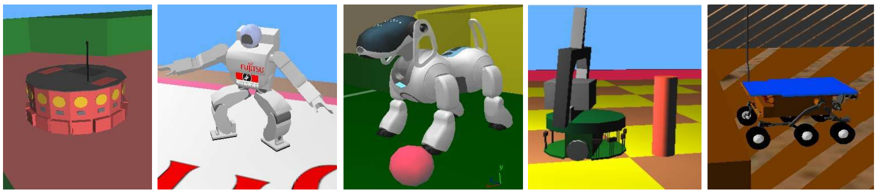

Figure 1: Several simulated robots provided off-the-shelf by Webots

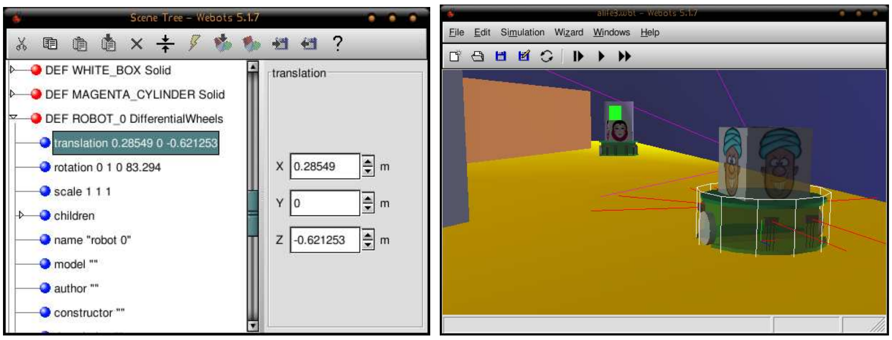

Figure 2: Scene tree and simulation windows

Figure 3: Two examples of multi-agent simulations included in Webots