RobotStadium: Online Humanoid Robot Soccer Simulation Competition

Olivier Michel[1], Yvan Bourquin[1], and Jean-Christophe Baillie[2]

[1] Cyberbotics Ltd., PSE C - EPFL, 1015 Lausanne, Switzerland
Olivier.Michel@cyberbotics.com, Yvan.Bourquin@cyberbotics.com - http://www.cyberbotics.com
[2] Gostai SAS, 15 rue Vergniaud 75013 Paris, France
baillie@gostai.com - http://www.gostai.com

Abstract. This paper describes robotstadium: an online simulation contest based on the new RoboCup Nao Standard League. The simulation features two teams with four Nao robots each team, a ball and a soccer field corresponding the specifications of the real setup used for the new RoboCup Standard League using the Nao robot. Participation to the contest is free of charge and open to anyone. Competitors can simply register on the web site and download a free software package to start programming their team of soccer-playing Nao robots. This package is based on the Webots simulation software, the URBI middleware and the Java programming language. Once they have programmed their team of robots, competitors can upload their program on the web site and see how their team behaves in the competition. Matches are run every day and the ranking is updated accordingly in the "hall of fame". New simulation movies are made available on a daily basis so that anyone can watch them and enjoy the competition on the web. The contest is running online for a given period of time after which the best ranked competitors will be selected for a on-site final during the next RoboCup event. This contest is sponsored by The RoboCup federation, Aldebaran Robotics, Cyberbotics and Gostai. Prizes includes a Nao robot, a Webots package and a URBI package.

1 Why We Need Another Simulation Competition

1.1 Introduction

The RoboCup is an excellent robotics benchmark [4] allowing different researchers to address a common challenge and to compare their results with each other. Moreover, RoboCup allows for a measurement of the progress of research over years as the benchmark specifications are fairly stable over time. This turns out to be a very valuable tool to evaluate the performance of mobile robots in the real world.

The RoboCup is divided into a number of robot soccer leagues, including the small size league, the middle size league, the standard platform league, the humanoid league and the simulation league [5]. There are also other leagues (rescue, at home, junior, etc.) which are out of the scope of this paper.

The Nao robot was recently chosen to become the official platform for the standard league competition of the RoboCup, replacing the discontinued Aibo robot from Sony. During the 2008 edition of the RoboCup, the first Nao-based standard league competition will be held in parallel with the last Aibo-based standard league.

However, it is very costly for a research lab to invest in RoboCup research as it requires to purchase, build and maintain a large number of mobile robots as well as a soccer field which occupies a large space with a good control on the lighting conditions. An alternative to these investments is to compete in the simulation league. However, although there are currently a number of simulation league, including 2D and 3D simulation, none of these simulations correspond to a real RoboCup setup. For example the 2D simulation is not a realistic simulation software in the sense that it doesn't try to simulate the physics of objects, sensors and actuators of the robots. Hence it makes it difficult to reuse research developed within a simulation league and transfer it to real robots. Moreover, the 3D simulation is currently in a very early stage as a new simulator was introduced only recently (2007).

This matter of fact is very disappointing because it has been observed that the strategies and level of intelligence developed in the simulation league were very interesting from a scientific point of view (very advanced AI, learning, multi-agent coordination, etc.). This can probably be explained by the fact that the researcher working in simulation can spend more time focusing on real control and AI problems rather than on mechanical, electronics or low level software problems. Moreovoer, simulations are often easier to reproduce, analyse, compare, etc. It would be really nice if such simulations could be easily transfered to the real world. But this is unfortunately not the case so far.

Making simulations closer to real RoboCup setups would help a lot researchers to develop new ideas in simulation and transfer them more easily onto the real robots. That would also allow for interesting comparisons between simulations and real setups. Finally, it would benefit in a straightforward way to the overall RoboCup competition.

2 Going Further with RoboCup simulation

2.1 Realistic simulation

Accuracy is a very important aspect of a robot simulation. The environment, robots and rules should be modelled very carefully to match the features of the real robots. This requires a calibration phase of the simulation in which a series of control algorithms are tested on both the simulation and the real setup. The results are then compared and the simulation models are tuned to better fit the real setup. This operation is repeated several times until a satisfactory level of reliability is achieved. Such calibrations methods have been conducted several times on the Webots simulation software [6] to refine complex robot model and allow an easy transfer from simulation to the real robot. They involve several levels of calibration, including sensor calibration (i.e, lookup tables), actuator calibration (physical parameters, like the maximum torque of a servo or the precise range of movement), physics calibration (mass distribution, friction parameters, etc.). A Fujitsu HOAP-2 humanoid robot was successfully calibrated in Webots using such a method [3]. Moreover, the utilization of the URBI middleware [2] for programming both the simulation and the real robots can dramatically facilitate the transfer onto the real robots.

This way, the simulation can be transfered into a real robot fairly accurately, hence making the use of a simulation software even more interesting.

2.2 Free and easy to use

Unlike real robots, simulation software can be made available very easily to a large community of users. Although our Webots simulation environment is a commercial package, including the URBI middleware, a special simulation of the RoboCup Nao Standard League was released free of charge, so that anyone can download it, install it and use it to develop robot controller programs within the framework of the RoboCup Nao Standard League. Our Webots-based simulation environment runs on all major platform, including Windows, Linux and Mac OS X. The utilization of the URBI middleware in combination with Java makes it very simple and powerful to program the Nao robots. Examples provided in the package help users to get started with programming their simulated Nao robots within minutes. Moreover, a number of graphical tools facilitating the generation of servo trajectories are included in the package (see figure 1).

2.3 Online competition

In order to be available to a wider audience, a simulation contest can be held on the Internet. This allow everyone to enter the competition regardless of his/her geographic location or ability to travel abroad to an international conference. Having the competition available on the web is also very attrative for potential competitors who can learn about the competition, register and get started within minutes.

Another very attractive aspect is the fact that competition match are held on a regular basis very often (like once a day). This way, competitors can view the results in real time, improve their robot controllers and submit new versions until the end of the online competition. The main consequence of this is that the developed robot controllers are continuously tested in real condition during the

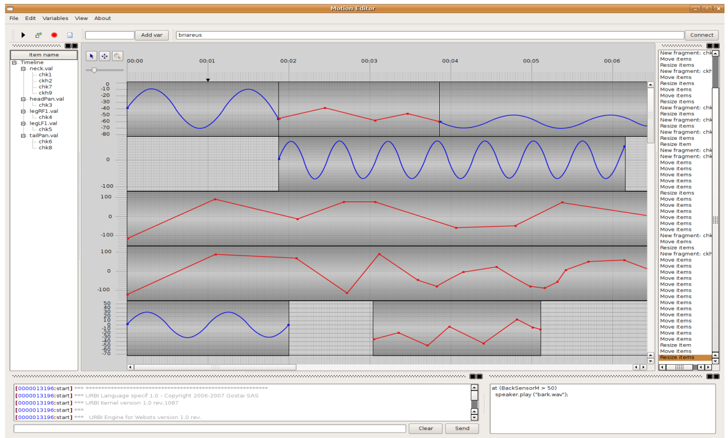

Fig. 1. The URBI motion editor

development phase. This undoubtably yields to better results at the end. The Rat's Life contest is an example of successful online competition [7, 8].

2.4 Durable benchmarking

Since the RoboCup is recognized as a reference benchmark in robotics, our simulation environment should also be used a useful benchmark, contributing to scientific progress in robotics research. A benchmark is useful if users can compare their own results to others and thus try to improve the state of the art. Hence a benchmark should keep a data base of the solutions contributed by different researchers, including binary and source code of the robot controller programs. These different solutions should be ranked using a common performance metrics, so that we can compare them to each other and evaluate the general progress over time. Keeping an archive of the contributed robot controllers can help the RoboCup community to measure the quality of their results.

3 Competition organization

This first edition of the simulated Nao RoboCup online contest is sponsored by the RoboCup federation along with three private companies: Aldebaran robotics, the makers of the Nao robot, Gostai, the developers of URBI and Cyberbotics, the developers of Webots.

3.1 Nao

Bruno Maisonnier, founder of Aldebaran Robotics [10], has been convinced for 25 years that the era of personal robotics is coming. During these years, he has developed prototypes, evaluated technologies, met with research teams and analyzed the markets, in addition to serving as the CEO of several companies in multicultural contexts. With the rise of mobile technologies and the coming together of key collaborators, the potentials have now become the possibilities: In 2005 he launched Aldebaran Robotics, the first French company dealing with humanoid robotics. Aldebaran Robotics team, which currently consists of 44 members, entirely dedicated to the development of its first robots, will continue to grow throughout 2008.

The Nao project, launched in early 2005, aims to make available to the public, at an affordable price, a humanoid robot with mechanical, electronic, and cognitive features (see figure 2).

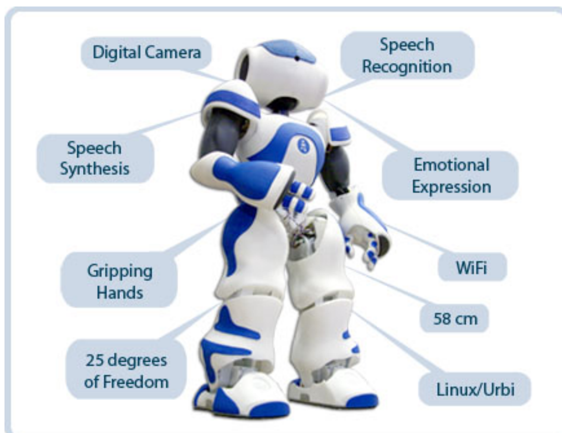

Fig. 2. The Nao robot

Initially, delivered with basic behaviors, the robot will be, at its market introduction, the ideal introduction to robots. Eventually, with many improved behaviors, it will become an autonomous family companion. Finally, with more sophisticated functions, it will adopt a new role, assisting with daily tasks (monitoring, etc.).

Featured with an intuitive programming interface, the entire family will be able to enjoy the robot experience. Yet, full of new technologies, the robot will also satisfy the demanding techno-addicts expectations.

Designed for entertainment purposes, the robot will be able to interact with its owner, with evolving behaviors and functionalities. Additionally, the user will be able to teach the robot new behaviors using a computer with Wi-Fi connectivity. The behaviours creating software is designed to fit with any users levels:

from graphical blocs editing for beginners to code for more skilled users. The possible behaviors are limited only by the imagination of the users!

With a conviction that design is key to successful adoption in a home environment, Aldebaran Robotics has partnered with a Parisian design school, particularly with designers Thomas Knoll and Erik Arlen. The robots hull will include customizable features, allowing each to have a unique appearance.

The robot is based on a Linux platform and scripted with URBI, an easy-to-learn programming language, with the option of a graphic interface for beginners or code commands for experts.

Currently in the final phase of development, the first Nao has been presented early 2007. The first units, dedicated to laboratories and universities were sold early 2008 for the RoboCup competition, and the general public release is planned for the end of 2008.

3.2 URBI

Created in 2006 after 4 years of R&D in an academic research lab (CogRob, ENSTA), Gostai [12] has a 6 years of experience in complex AI-driven robotics applications and development tools. The core of its technology is the Urbi middleware for robotics, initially developed for the Aibo by J.-C. Baillie, a former researcher at Sony Computer Science Lab.

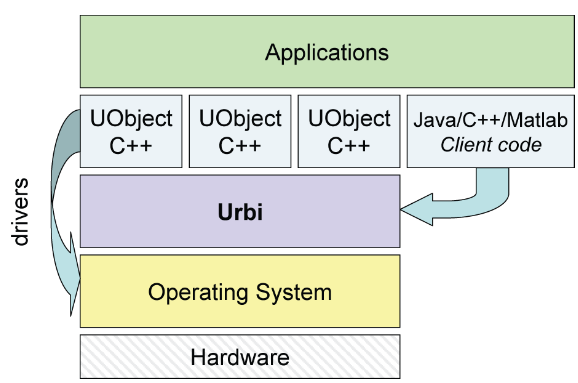

Fig. 3. URBI

Urbi [1, 2] is a middleware for robotics, which includes dedicated abstractions to handle parallelism and event-based programming from within C++, Java or Matlab, together with a distributed component architecture called UObject which can be interfaced with Microsoft Robotics Studio and CORBA. The main focus is on simplicity, flexibility and code re-use, while providing convenient abstractions needed in the development of complex robotics applications.

Urbi is based on a client/server architecture and a dynamic language called Urbiscript that can be used to coordinate the UObject components. The novelty of Urbiscript as a programming language is that it brings new abstractions to handle parallelism and event-based programming, directly integrated into the language semantics. Urbiscript programs routinely run hundreds of parallel threads and react to several events at the same time.

The idea is to separate the logic of the program on one side (Urbiscript) and the fast algorithms on the other side (UObjects in C++), and use the dynamic capabilities of script languages to build the glue (see figure 3). This is already a widely used approach in videogames, with script languages like LUA or Python. Urbi brings this same approach to robotics and improves the script language side with parallelism and event-based programming.

UObject is a C++ based component architecture. Any C++ class can inherit UObject and become visible inside Urbiscript as a regular object that can interact with other objects already plugged. UObjects can be either linked to the Urbi Engine (the interpretor of the Urbiscript language) and share its memory and other UObjects memory, or it can also be run remotely as an autonomous separate process. In that case, the interface between Urbi and the UObject is done transparently through the network via TCP/IP and the C++ object is reflected inside the Urbiscript language in exactly the same way as in the "linked" mode. This allows to have large distributed network of objects interconnected through scripts coordinating them in a parallel and event-driven way, which contrasts with the traditional one-to-one component interactions of CORBA or similar component architectures.

CORBA and other component architectures can be bridged through UObject using the dynamic feature of the Urbiscript language, which can dynamically create a proxy object within the language to access the remote component. Using this feature, Urbi aims at providing a universal bridge between heterogeneous component architectures.

Urbi also includes a set of graphical programming tools called Urbi Studio, which comprises a hierarchical finite state machine editor, an animation editor and a universal remote controller for Urbi equipped applications. These tools are graphical front-ends which generate Urbiscript code that can be reused in various contexts.

Urbi is now compatible with 12 different robots and simulators, including Webots, and is used by industrial companies and more than 30 universities in the world.

3.3 Webots

Webots [6] is a commercial software for fast prototyping and simulation of mobile robots. It was originally developed by Olivier Michel at the Swiss Federal Institute of Technology in Lausanne (EPFL) from 1996 and has been continuously developed, documented and supported since 1998 by Cyberbotics Ltd. [11]. Over 500 universities and industrial research centers worldwide are using this software for research and educational purposes. Webots has already been used to organize robot programming contests (Rat's Life, ALife contest and Roboka contest).

Fig. 4. Webots

Webots offers a rapid prototyping environment, that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc. based on the ODE physics engine [13] and the OpenGL rendering engine. The user can add simple passive objects or active objects called mobile robots. These robots can have different locomotion schemes (wheeled robots, legged robots, or flying robots). Moreover, they may be equipped with a number of sensor and actuator devices, such as distance sensors, drive wheels, cameras, servos, touch sensors, grippers, emitters, receivers, etc. Finally, the user can program each robot individually to exhibit the desired behavior. Webots contains a large number of robot models and controller program examples to help users get started.

Webots also contains a number of interfaces to real mobile robots, so that once your simulated robot behaves as expected, you can transfer its control program to a real robot like Khepera, Hemisson, LEGO Mindstorms, Aibo, etc.

Although Webots is a commercial software, a demo version is freely available from Cyberbotics's web site. This demo version includes the complete Nao RoboCup simulation with a URBI programming interface. So, anyone can download, install and practice the simulation of the Nao RoboCup competition at no cost (see figure 4).

4 Competition Description

This paper doesn't claim to be a technical reference for the simulated Nao RoboCup competition. Such a technical reference is available on the Robotstadium web site [9].

4.1 Rules

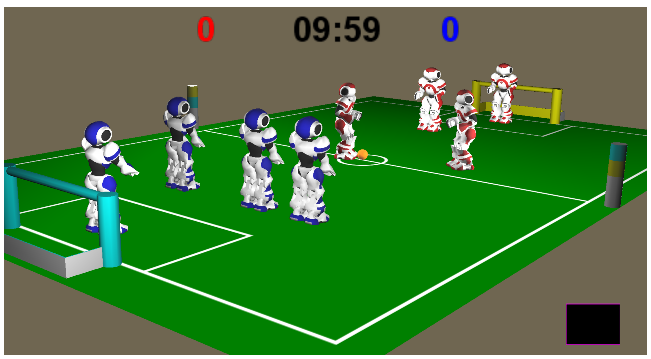

Fig. 5. Screenshot of the robotstadium simulation model

In order to get started rapidly and to take advantage of the feedback provided by early competitors, no strict rule will defined from the beginning. The idea is to invite constants to respect the standard basic soccer rules (like do not catch the ball with the hands) and to develop fair robot controllers (like robots should not attempt to block the game by locking the ball, or they should not try to attack other players, etc.).

A standard Nao RoboCup soccer environment is provided to the competitors (see figure 5) which includes a supervisor process responsible for counting the time (a match has two half time of 10 minutes each), counting the score, resetting the ball and the player at initial position once a goal is scored, etc.

The supervisor process is actually an automatic referee which saves the contest organization from the human supervisor. Hence matches can be run automatically by a script program and results can be published automatically as well.

This supervisor process will be continuously developed to refine the rules (to possibly introduce some penalties if some illegal actions are taken by some robots) during the competition. This will allow the organizers to detect problems in the rules and correct them while the contest is running. However, we will try to introduce only slight modifications of the rules in respect of the principles of the standard basic soccer rules and fair play. Moreover, the rules will be fixed about one month before the end of the contest. This will allow contestant to rely on a fair and stable version of the rules for the last mile of competition.

4.2 Online Contest System

Fig. 6. Banner of the robotstadium web site

Web site: The Robotstadium online contest (see figure 6) will allow contestant to learn about the contest, view simulation movies of contest matches, register, download the necessary software, learn how to develop their own robot controllers, upload their robot controller, see their ranking in the "hall of fame" updated every day. In order to provide all these services, the web site will contains a number of standard web tools, including a registration system, a forum, a file management system, a shout box, and a number of sections: movies, documentation, rules, FAQ, etc.

Participation to the Contest: In order to participate in the online contest, the competitors can download the free version of Webots from Cyberbotics' web site [11]. They can program the simulated Nao robots to play soccer on the simulated soccer field. Then, they have to register a contestant account on the contest's web site [9]. Once open, this account allows the competitors to upload the controller programs they developed with the free version of Webots. Participation to the contest is totally free of charge.

Ranking System: Every day at 12 PM (GMT) a competition round is started in simulation and can be watched online from the Robotstadium web site [9]. A hall of fame displays a table of all the competitors registered in the data base and who submitted a robot controller program. If there are N competitors in the hall of fame, then $N-1$ matches are played. The first match of a round opposes

the last entry, i.e., number N at the bottom of the hall of fame, to the last but one entry, i.e., number $N-1$. If the robot number N wins, then the position of these two robots in the hall of fame are switched. Otherwise no change occurs in the hall of fame. This procedure is repeated with the new robot number $N-1$ (which may have recently changed due to the result of the match) and robot number $N-2$. If robot number $N-1$ wins, then it switches its position with robot number $N-2$, otherwise nothing occurs. This is repeated with robots number $N-3$, $N-4$, etc. until robots number 2 and 1, thus totaling a number of $N-1$ matches.
This ranking algorithm is similar to the bubble sort. It makes it possible for a newcomer appearing initially at the bottom of the ranking, to progress until the top of the ranking in one round. However, any existing entry cannot loose more than one position in the ranking during one round. This prevents a rapid elimination of a good competitor (which could have been caused by a buggy update of the controller program for example).

And The Winner Is... The contest is open for a fixed period of time. During this period of time, new contestants can register and enter the contest. The contestants can submit new versions of their controller program any time until the closing date. Once the closing date is reached. The top five entries of the hall of fame will be selected for the finals to be held during the next RoboCup event.
The finals will take place at RoboCup event to ensure that a large number of people, including a scientific committee, attends the event and can check that nobody is cheating the contest.
The contest will run continuously over years so that we can measure the progress and performances of the robot controllers over a fairly long period of robotics and AI research.

5 Expected Outcomes and Conclusions

Thanks to the Robotstadium contest, it will become possible to evaluate the performance of various approaches to the RoboCup benchmark with humanoid robots. The performance evaluation will allow us to make a ranking between the different control programs submitted, but also to compare the progresses achieved over several years of research on this problem. For example, we could compare the top 5 controller programs developed in 2008 to the top 5 controller programs developed in 2012 and evaluate how much the state of the art progressed.
The control programs resulting from the best robot controllers could be transfered to the real world robotics setup of the RoboCup, especially the new Nao-based standard league. This will help the teams involved in the standard league to accelerate their developments. Hopefully, this will allow researchers to go further in the development of advanced intelligent control architectures.

Moreover, because of the easy availability of the robotstadium simulation setup, a large number of researchers is expected to start investing their time in developments for the simulated Nao standard league. In turn these people will be keen to try to transfer their results in the real Nao standard league, thus bringing even more success to the real league.

Finally, by making available a large number of simulation movies on the Internet, this simulation competition may attract the interest of a wider audience, including general public and the media.

We hope that this initiative is a step towards a more general usage of realistic simulation and advanced programming languages in robotics research, as we are convinced that roboticists need more high quality software tools in order to focus more efficiently on their research and achieve significant breakthoughts in robotics research.

References

1. J.C. Baillie, "Design Principles for a Universal Robotic Software Platform and Application to Urbi", *in Proceedings of ICRA'07 Workshop on Software Development and Integration in Robotics*, 2007.
2. J.C. Baillie, "Urbi: Towards a Universal Robotic Low-Level Programming Language", *in Proceedings of IROS'05*, 2005, pp 820-825.
3. Cominoli, P.: "Development of a physical simulation of a real humanoid robot", Master's thesis, EPFL, Swiss Federal Institute of Technology in Lausanne, 2005.
4. Dillmann, R.: "Benchmarks for Robotics Research", EURON, April 2004. http://www.cas.kth.se/euron/eurondeliverables/ka1-10-benchmarking.pdf
5. Kitano, H., Asada, M., Kuniyoshi, Y., et al: "RoboCup: the robot world cup initiative", IJCAI-95 workshop on entertainment and AI/ALife (1995)
6. Michel, O. "Webots: Professional Mobile Robot Simulation", Journal of Advanced Robotics Systems, 39-42 Vol.1 number 1 / 2004 http://www.ars-journal.com/International-Journal-of-Advanced-Robotic-Systems/Volume-1/39-42.pdf
7. Michel, O., Rohrer, F. and Bourquin, Y.: "Rats Life: A Cognitive Robotics Benchmark", European Robotics Symposium 2008 (EUROS'08), Springer 223-232, Volume 44/2008
8. Rat's Life contest. http://www.ratslife.org
9. Robotstadium contest. http://www.robotstadium.org
10. Aldebaran Robotics SAS. http://www.aldebaran-robotics.com
11. Cyberbotics Ltd. http://www.cyberbotics.com
12. Gostai SAS. http://www.gostai.com
13. ODE: Open Dynamics Engine. http://www.ode.org