Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions

Luc Guyot
PSE C - EPFL
Cyberbotics Ltd.
Lausanne 1015, Switzerland
Email: Luc.Guyot@Cyberbotics.com

Nicolas Heiniger
Cyberbotics Ltd.
Lausanne 1015, Switzerland
Email: nicolas@photochrome.ch

Olivier Michel
PSE C - EPFL
Cyberbotics Ltd.
Lausanne, Switzerland
Email: Olivier.Michel@Cyberbotics.com

Fabien Rohrer
PSE C - EPFL
Cyberbotics Ltd.
Lausanne, Switzerland
Email: Fabien.Rohrer@Cyberbotics.com

***Abstract*—We introduce a robotics curriculum intended for all levels of learning and discuss results of related in-class experiments and competitions. The curriculum is an open document with a collaborative format, hence freely accessible and extendable. Based on the e-puck mobile robot and the Webots simulator, it addresses a dozen of topics ranging from finite state automata to particle swarm optimization. While beginners familiarize with a user-friendly graphical programming interface, most advanced readers benefit of apt exercises to tackle robotics contests.**

***Index Terms*—education, robotics, e-puck, competitions**

I. Introduction

Robotics is widely considered as an excellent tool for teaching science and engineering [14], [17] and such a belief is reinforced by the ever growing number of successful educational experiments, up-to-date curricula and new didactic approaches [10], [25], [28]. Nevertheless Mataric bemoans the lack of age-appropriate teaching materials in her workshop report [17] and urges the robotics community to broaden both scope and audience of educational supports. Our curriculum is an attempt to fulfill this twofold expectation.

Our curriculum aims at teaching hands-on robotics through the cost-friendly and widely used e-puck robot, benefiting further of its rich interplays with the Webots simulation software. Grounded on a former document involving the Hemisson robot [15], the curriculum presented here was originally written by the last three authors [16], [24]. Distributed for the first time in 2008, it has been used ever since as a support for master courses at the École Polytechnique Fédérale de Lausanne (Switzerland). The document stems from privileged interactions between two educational and research supports, namely e-puck and Webots. Aimed at the broadest possible audience, it defines five levels of learning where beginners can acquire the basics of robotics without any prior programming knowledge (Section II-A) while skilled users are prepared to compete in Rat's life [22] and RobotStadium [21] (Section II-E).

The curriculum is released as a wikibook [4] under the terms of the GNU Free Documentation License and the Creative Commons Attribution-ShareAlike 3.0 Unported License, so it benefits from robotics community contributions. (A PDF version can also be downloaded from Cyberbotics website [3].)

The curriculum begins with a general introduction on robotics and then describes the e-puck robot and Webots. The remaining part divides into five sections, each dedicated to a specific level of learning. In keeping with this framework, our article gives a description of every individual section (except the general introduction) so that interested teachers can quickly grasp the bulk of it. The last section is devoted to the analysis of in-class experiments.

A. The e-puck robot and Webots simulation software

a) e-puck: The e-puck mini mobile robot was originally developed at the EPFL for teaching purposes by the designers of the Khepera robot. The e-puck hardware and software is fully open source, providing low level access to every electronic device and offering unlimited extension possibilities. The robot is already equipped with a large number of sensors and actuators (Figure 1) and possesses a Microchip dsPIC 30F6014A with a frequency of 60MHz. It is well supported by the Webots simulation software which provides simulation models, remote control and cross-compilation facilities. The Webots-oriented programming toolchain comprises a multi-platform MPLAB cross-compiler and a firmware dedicated to the software-hardware interplay. The firmware allows programming in C, C++, Java, Python and matlab.

The official e-puck web site [9] gathers a large quantity of information about the robot, extension modules, software libraries, users mailing lists, etc.

b) Webots: The Webots robot simulation software is a commercial software for fast prototyping and simulation of mobile robots which has developed tight links with the e-puck robot. It was originally developed at the Swiss Federal

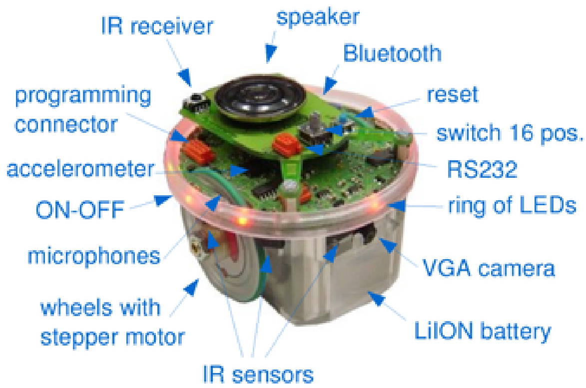

Fig. 1. e-puck devices

Institute of Technology in Lausanne (EPFL) since 1996 and has been continuously developed, documented and supported since 1998 by Cyberbotics Ltd. Over 500 universities and industrial research centers worldwide are using this software for research and educational purposes. Although Webots is a commercial software, a demo version is freely available from Cyberbotics web site [5]. This demo version includes the complete Rat's Life simulation, a cognitive benchmark described at the end of the curriculum (see Section II-E)

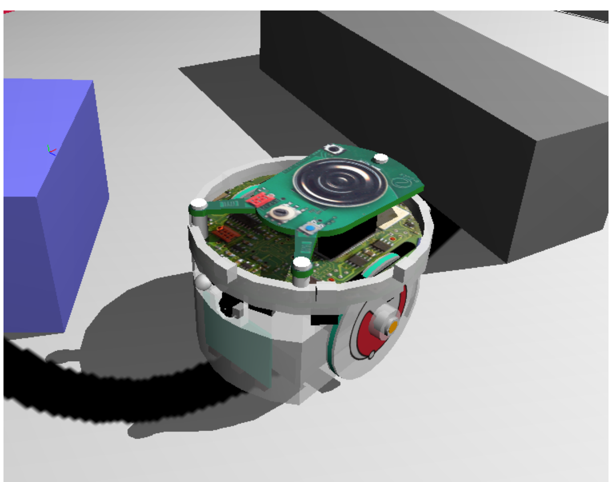

Fig. 2. Simulated e-puck in Webots

c) e-puck interface and simulation model: On hardware side, Webots allows remote control sessions of the real robot through Bluetooth connection and supports cross-compilation. On simulation side, it provides an accurate e-puck model including the differential wheel motors and their encoders, the infra-red sensors for proximity and light measurements, the accelerometer, the camera and the 8 surrounding LEDs (see Figure 2). Webots has moreover a graphical dedicated interface which displays sensor and actuator values (Figure 3) and enables beginners to step into robotics without a single line of code (see BotStudio in Section II-A).

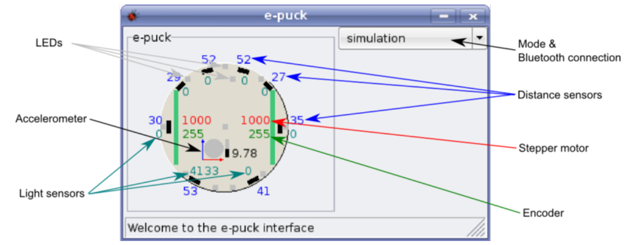

Fig. 3. e-puck window in Webots

B. *Five levels of learning*

The curriculum is designed to be used by pupils, under- and postgraduate students, autodidact hobbyists and researchers as well. The target audiences read as follows.

1. Beginner: The user is totally new to robotics and computer science. A typical beginner would be a high school pupil using a programming environment and a robot simulator for the very first time.

2. Novice: The user has a basic knowledge of programming and can decipher a simple C code but does not necessarily have a background in robotics. A typical novice would be an undergraduate student attending an introduction to robotics.

3. Intermediate: The user has an experience in both programming and robotics and can write his own C code. A typical intermediate user would be a student in science.

4. Advanced: The user has an important experience in both programming and robotics and can program complex controllers. A typical advanced user would be a post-graduate student or a researcher looking for working examples and references.

5. Expert: The user is a professional in robotics. A typical expert would be a participant of a robotics contest.

II. Curriculum's content

Curriculum's exercises are ordered with an increasing level of difficulty; sections break down accordingly.

A. *Beginner*

The beginner section consists of eight exercises culminating with a line following algorithm and a challenge named *Rally*: create a finite state automaton that makes the e-puck robot follow a winding road without leaving it. The user manipulates either a simulated e-puck robot on Webots interface, or the real robot by uploading controllers via a Bluetooth connection. S/he will discover most of the e-puck devices along her/his progression.

BotStudio: BotStudio is a graphical programming interface specially designed for educational purposes. It makes it possible to create a finite state automaton by simple drags and drops (see Figure 4). The automaton is then turned into a controller file that can be run on both virtual and real e-pucks. In the latter case, BotStudio uploads controllers, for the ease of the user.

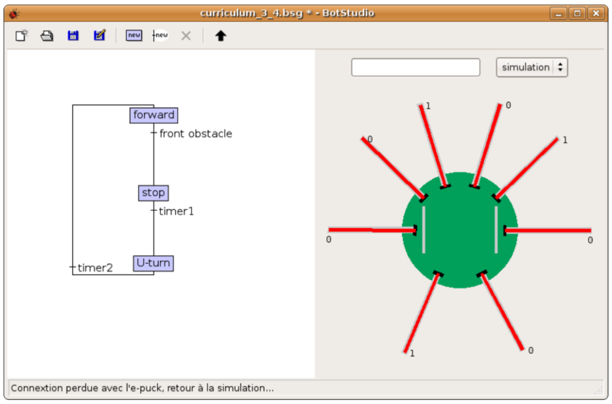

Fig. 4. An automaton in BotStudio to make a U-turn with e-puck

Key features:

- Graphical programming
- Sensors and actuators: infra red sensors, stepper motor, linear camera, LEDs
- Finite state automata
- Collision avoidance algorithm
- Line following algorithm

B. Novice

The novice section consists of eight exercises exploring BotStudio and the e-puck devices in greater depth. C programming is progressively introduced to code old and new behaviours of the e-puck robot. The set up of a remote control session and the cross-compilation process are thorougly described.

Key features:

- Basic C programming
- Elaborated finite state automata
- Simulation of robots interacting with each others
- Wall following algorithm
- Calibration of infra red sensors, e-puck's camera and accelerometer

C. Intermediate

The intermediate section consists of an introduction to behavior-based robotics made of five examples of behavioral modules. It culminates with a program making the e-puck robot avoid obstacles while following a line.

Key features:

- Behavior-based artificial intelligence
- C programming of behavioral modules: Line following, wall following, obstacle avoidance and scanning

D. Advanced

The advanced section consists of seven exercises culminating with a probabilistic approach to the simultaneous localization and mapping problem (SLAM). These exercises aim at giving an insight of today's robotics while laying the foundations for expert robotics competitions. Indeed, exercises on odometry, pattern recognition and SLAM teach how to use the e-puck visual system efficiently and how to navigate in an unknown environment. These are two of the main challenges of curriculum's expert benchmarks.

Key features:

- Odometry and calibration
- Path planning: potential field and NF1 algorithm [27]
- Supervised learning: artificial neural network for pattern recognition [11]
- Unsupervised learning: particle swarm optimization [12], [20], [26]
- Genetic algorithm [13], [18], [19] using Braitenberg vehicles [2]
- SLAM [6]–[8]

E. Expert

At this stage, the user is fit for robotics contests. This last section reads as an invitation for challenges and offers several clues as how to tackle the two following benchmarks.

1) Rat's life:

a) The competition: Rat's Life is a cognitive robotics benchmark particularly suited for research in SLAM, autonomy and vision. Easily reproducible in a lab with limited resources, it relies on two e-puck robots, some LEGO bricks and the free version of the Webots software. It is a survival game where two robots compete against each other for resources in an unknown maze (see Figure 5). Like the rats in cognitive animal experimentation, the e-puck robots look for feeders which allow them to live longer than their opponent. Once a feeder is reached by a robot, the robot draws energy from it and the feeder becomes unavailable for a while. Hence, the robot has to further explore the maze, searching for other feeders while remembering the way back to the first ones. Rat's life online simulation contest runs its third edition in 2011.

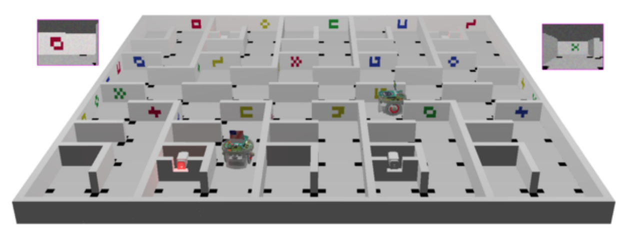

Fig. 5. Two e-puck in Rat's life maze and their camera displays

b) Evolution:

Year	Number of competitors
2008	41
2010	16
2011	15

Simulation movies of the competitions are stored in a data base in order to analyze the evolution of the competition over the time. The movie database contains more than 2500 movies (totaling more than 50 GB of data) and is freely available online at http://www.cyberbotics.com/ratslife/movies/.

Intensive observations revealed a co-evolution dynamic similar to the ones encountered in genetic algorithms. Once a performance breakthrough is made by one competitor, it is immediately analyzed by the other competitors seeing the simulation movies. They take inspiration from it to improve their own robot controller and submit their new improved version to the contest. Rapidly, a large number of competitors replicate this winning behavior on their robot and most of the robots in the contest adopt this new efficient behavior. We list the most significant breakthroughs made by challengers in chronological order.

- *Random Walkers:* The random walkers came actually from the very first version of the sample source code included with the contest software development kit, made available to all competitors. This simple control algorithm similar to Braitenberg vehicles [2] lets the robots move randomly while avoiding the obstacles.
- *Vision-Enabled Random Walkers:* They are an improved version of the original random walker making an extensive use of vision to recognize the feeders and adjust the trajectory of the robot to reach the feeder instead of simply moving randomly. This results in slightly more efficiency as robots will not pass in front of a feeder without getting energy from it.
- *Right Hand Explorers:* A Braitenberg vehicle behavior is not very efficient at exploring extensively a maze and hence at finding the feeders. Maze exploration algorithms exist and are much more efficient. The right hand algorithm is one of the simplest and best known maze exploration algorithms. It consists in simply following the first wall found on the right hand side of the robot. Using this algorithm combined with some vision to reach efficiently the feeders, a significant performance breakthrough was reached.
- *Energy-aware robots:* Getting the energy from the feeder as soon as you find the feeder is nice, but there is an even better strategy: Once a robot finds a feeder, it can simply stop and sit in front of the feeder, thus preventing the other robot from reaching this feeder. In the meanwhile the robot sitting in front of the feeder should watch its energy level and decide to move to the feeder once its energy level reached a very low value, just enough to make that move to the feeder and refuel. During this waiting time, the other robot may be struggling to find a feeder and possibly loose the game if it runs out of energy.
- *SLAMers:* Compared to other techniques mentioned above, it involves a much more complicated algorithm and requires an efficient image processing. SLAMer robots actually seems to use the right hand algorithm on a first stage to explore extensively the maze, but they dynamically build a map of this maze while exploring it and eventually don't use the right hand algorithm at all. Their internal representation of the environment contains the walls, the feeders and likely the landmarks. This map is then used by the robot to get back to previously found feeders. It turned out to be very efficient and clearly outperformed the simpler reactive controllers.
- *Super-SLAMers:* A major improvement of SLAM-based robot controller was probably the estimation of the status of the feeders, combined with an estimation of the time needed to travel through the maze to reach the feeder. A Super-SLAMer seems to be able to anticipate that a mapped feeder will become available again: when the feeder is still red, it starts to navigate towards this feeder and about one second before it reaches the feeder, the feeder becomes green again.

2) RobotStadium:

a) The competition: Robotstadium is an online simulation contest based on the new RoboCup Nao Standard Platform League (SPL) [23] and relying on a free version of Webots. Running every year since 2008, this 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 Nao SPL (see Figures 6 and 7). Competitors simply register on the web site and download a free software package to start programming their team of soccer-playing Nao robots. 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 an on-site final during the next RoboCup event.

Fig. 6. A match in RobotStadium

b) Evolution:

Year	Number of participating teams
2008	15
2009	14
2010	7
2011	14

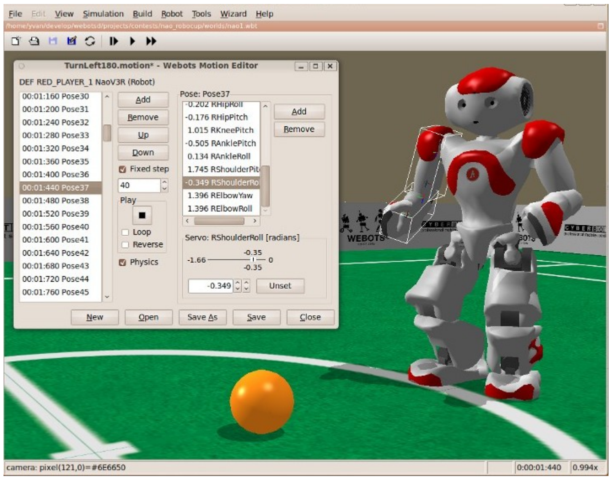

Fig. 7. Simulated Nao in Webots

Every year, participants to RobotStadium also take part into RoboCup Nao SPL. In 2008, CMRobokids (Carnegie Mellon University, USA) won RobotStadium and performed as finalist of the RoboCup Nao SPL. They notably used the same team strategy and representation algorithms in both competitions and had only to adapt movements and image processing for the real robots. The same year, Kouretes (Technical University of Greece) performed at the fifth and the third place in RobotStadium and the RoboCup Nao SPL respectively.

III. Analysis of in-class experiments

The curriculum has been tested in high-school and university. Teachers provided us with assessment results and comments.

A. High school

For his master degree at EPFL, Nicolas Heiniger led in-class experiments from November 2008 to February 2009 in seven groups coming from different Swiss high schools. Two sets of exercises were designed to assess the part of the curriculum that does not need C programming (eight in the beginner section, one in the novice section). A group consisted on average of ten pupils and each group was given one of the two sets. All in all 64 pupils participated in the experiment and answered Heiniger's survey. Statistics and a detailed analysis are available on Cyberbotics website [16]. The results of the survey led to improvements of the shape, the content and the usability of the curriculum, especially of the beginner section. This educational project led to the first publication of the curriculum on wikibooks. We only give here a short account:

- The exercises were considered as easy by the pupils, the average mark being 3.32 in a range from 0 (too easy) to 6 (too hard).
- More than 80 % agreed that they learned something about robotics through the exercises.
- More than 90 % agreed that using the real robot was more pleasurable than simulation alone.
- More than 70 % agreed that Webots was easy to use.

Nicolas Heiniger observed that the curriculum cannot substitute the guidance of a teacher and that using the text alone requires a reasonable level of autonomy. Indeed, the pupils largely preferred asking him directly every time they faced problems although the provided text contained answers to most of their questions. The feedback sessions also revealed that they expected a longer oral introductory presentation.

B. University

The intermediate section of the curriculum is used at the EPFL as a support for master courses on robots navigation every year since 2009. Here we collect observations of assistants having taught with it:

- The exercises are easily completed by a large majority of students.
- The simulation part takes most of the time and is crucial to experiment intensively with algorithms.
- The set of exercises could be extended with an example of navigation in a maze needing more intricate behavioral modules.

IV. Conclusion

In accordance with our main goal, the curriculum is being used both in high school and university and evolves on the impulse of teachers and wikibook contributors. Regarding the expert level, we observed that worldwide competitors renew their participation every year to Rat's life and RobotStadium. By continuously rising the level of these competitions, they keep them attractive to robotics experts and researchers looking for benchmarks. However intermediate levels of learning should be assessed to validate the entire document. We expect that its open format and its frequent improvements will help to reach every target audience in a near future.

References

[1] Bailey, T.; Durrant-Whyte, H., "Simultaneous localization and mapping (slam): part II". Robotics & Automation Magazine, IEEE, 13(3):108-117, Sept. 2006.

[2] Braitenberg, V., "Vehicles: Experiments in Synthetic Psychology", Cambridge, MA: MIT Press, 1984.

[3] Curriculum PDF. http://www.cyberbotics.com/curriculum

[4] Curriculum wikibook. http://en.wikibooks.org/wiki/Cyberbotics\%27_Robot_Curriculum

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

[6] Dissanayake, M.; Newman, P.; Clark, S.; Durrant-Whyte, H; Csorba, M., "A solution to the si- multaneous localization and map building (SLAM) problem. Robotics and Automation, IEEE Transactions on, 17(3):229-241, Jun 2001.

[7] Dodds, Z.; Greenwald, L.; Howard A., Tejada, S.; Weinberg, J., "Components, Curriculum, and Community: Robots and Robotics in Undergraduate AI Education". In AI Magazine 27(1):Spring 2006.

[8] Durrant-Whyte, H,; Bailey, T., "Simultaneous localization and mapping: part I". Robotics & Automation Magazine, IEEE, 13(2):99-110, June 2006.

[9] e-puck web site. http://www.e-puck.org

[10] Gaudiello, I.; Zibetti, E.; Carrignon, S., "Representations to go: learning robotics, learning by robotics". Proceedings of SIMPAR 2010 Workshops Intl. Conf. on Simulation, Modeling and Programming for Autonomous Robots, Darmstadt (Germany) November 15-16, p. 484-493, 2010.

[11] Gerstner, W., "Supervised Learning for Neural Networks: A Tutorial with JAVA exercises". To appear in: Intelligent Systems, An EPFL graduate course. D. Mlynek and H.-N. Teodorescu (Eds.).

[12] Eberhart, R.; Kennedy J., "A new optimizer using particle swarm theory" in Proceedings of IEEE International Conference on Neural Networks (ICNN '95), vol. 4, pp. 19421948, 1995.

[13] Harvey, I. ; Husbands, P. ; Cliff, D. ; Thompson, A.; Jakobi, N., "Evolutionary Robotics: the Sussex Approach. In Robotics and Autonomous Systems", v. 20, p. 205224, 1997.

[14] Johnson, J., "Children, robotics, and education, Artificial Life and Robotics, 7 (1-2), p. 16-21, 2003.

[15] K-Team, Hemisson. http://www.k-team.com/kteam/index.php?site=1&rub=22&page=16&version=EN

[16] Heiniger, N., "Design and Assessment of a Robot Curriculum based on the E-puck Robot and Webots", master project EPFL-Cyberbotics, 2009. Available at http://www.cyberbotics.com/files/papers/

[17] Mataric, M. J., "Robotics Education for All Ages", Proceedings, AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education, Palo Alto, CA Spring 2004.

[18] Mondada, F.; Floreano, D, "Evolution of neural control structures: Some experiments on mobile robots". Robotics and Autonomous Systems, 16, p. 183-195, 1995.

[19] Nolfi, S.; Floreano, D., "Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines", MIT Press, 2000, ISBN 0-262-14070-5.

[20] Poli, R.; Kennedy, J.; Blackwell, T., "Particle swarm optimization". Swarm Intelligence, 1(1):33-57, August 2007.

[21] Michel, O.; Bourquin, Y.; Baillie, J.-C.,"RobotStadium: Online Humanoid Robot Soccer Simulation Competition". In Robocup 2008: Robot Soccer World Cup Xii, vol. 5399, 2009, p. 580-590 Springer-Verlag New York, 2009.

[22] Michel, O.; Rohrer, F.; Bourquin, Y.," Rat's life: A cognitive robotics benchmark". In European Robotics Symposium 2008, vol. 44, p. 223-232, 2008. Springer-Verlag New York, 2008.

[23] RoboCup website. http://www.tzi.de/spl/bin/view/Website/WebHome

[24] Rohrer, F.,"Curriculum of exercises for e-puck and Webots", master project EPFL-Cyberbotics, 2008. Available at http://www.cyberbotics.com/files/papers/

[25] Sergeyev, A. and Alaraje., N., "Promoting Robotics Education: Curriculum and State-of-the-Art Robotics Laboratory Development", The Technology Interface Journal, Volume 10 Number 3, 2010.

[26] Shi, Y.; Eberhart, R., "A modified particle swarm optimizer". Proceedings of the IEEE International Conference on Evolutionary Computation, p. 69-73, 1998.

[27] Siegwart, R. ; Nourbakhsh; I. R. , "Introduction to Autonomous Mobile Robots". MIT Press, 2004.

[28] Tejada, S.,"AI, AIBO and ART: Inspiring Interaction with Computer Science", AAAI Spring Symposium on Using AI to motivate greater participation in Computer Science, Stanford, CA 2008.