## 2.2 Some reasons to use Lego Mindstorms NXT

### *2.2.1 Introduction*

The choice of an appropriate platform whereupon to develop educational robotics activities is very important and should be carefully planned by the instructors. In the TERECop project, different robotics platforms have been considered. What was sought was a general platform not tailored on specific activities or educational disciplines. The following requirements for the robotics platform mentioned above have been identified:

- It should be programmable at different complexity levels and should support different programming paradigms;
- It should be exploitable at many levels of complexity in order to be usable at different educational levels (i.e. at different ages);
- It should have simple, but significant possibilities of expansion. This could be achieved by plugging-in additional sensors or by interfacing that robotics platform with other devices to allow remote processing or remote control;

The ultimate choice was the LEGO MINDSTORMS NXT kit (in short NXT) (website: mindstorms.lego.com/). NXT fulfills the above requirements and has several other advantages. In our experience the most important advantages are:

- the start-up time for working with NXT is very short;
- the assembly of the robot is very intuitive and no electrical wiring is necessary. No workshop tools are needed, not even a screwdriver or a solder.
- NXT is very familiar to students. Almost all of them played with LEGO bricks. This motivates them. It reminds them of their toys rather than of their assignments.

NXT complies very well with the constructionist learning approach. First of all, a robot is a "*public entity*" in the Papert's sense (Harel and Papert 1991). NXT comes straight from Papert's experience and works. NXT is the last evolution of the work Papert started with Logo and continued with Dacta. NXT is modular and incremental. It leads to a bottom-up oriented development. Starting from the basic bricks, which define the fundamental standard for all the other elements of the Lego kits, you can build more and more complex architectures by combining simpler, already built parts.

NXT complies very well also with the TERECop philosophy of approaching initially the robot as a "learning object" to be investigated in order to understand how it works and how to control it, but then to use (or better to "**exploit**") the robot as a "learning tool" whereby to study curricular disciplines. From that point of

view, one of the advantages of NXT is that the students achieve very quickly the first step in which the robot is seen as a "learning object". With NXT it is easy to learn how to build and how to program a robot. Thus, the students can move very fast to the second step in which the robot is seen as a "learning tool" to be accordingly utilized.

The modularity of NXT makes it very flexible and expandable as well. From a hardware point of view, beside the sensor and the motors provided by LEGO, several third part sensors and actuators are available whereby to enlarge experimenting possibilities. For instance, companies like Hitechnic (http://www.hitechnic.com/). and Mindsensors (http://www.mindsensors.com/) produce several sensors that can be directly read and logged by the NXT, while other companies, such as Vernier, (www.vernier.com/nxt/) produce NXT adaptors to connect their sensors for the most disparate scientific experiments.

NXT can be controlled and programmed via different programming languages and different programming environments. One can use the graphical programming environment NXT-G, developed by LEGO and National Instruments (www.ni.com/) or the C-like NBC and NXC or the Java-based LeJOS-NXJ. Next Byte Codes (NBC) (http://bricxcc.sourceforge.net/nbc/) is a simple open-source language with an assembly language syntax that can be used to program the NXT brick. Not eXactly C (NXC) (http://bricxcc.sourceforge.net/nxc/).is a high level open-source language, similar to C, built on top of the NBC compiler. It can also be used to program the NXT brick. NXC is basically *Not Quite C* **(**NQC) for the NXT. leJOS NXJ is a high level open-source language based on Java that uses custom firmware developed by the leJOS team (http://lejos.sourceforge.net/)

Moreover, one has the possibility to use several operating systems and/or platforms (URBI, Universal Real-time Behavior Interface (http://www.gostai.com/) for Windows, Mac OS X, Linux or NXT-Symbian running on Symbian 6.0 Java-enabled mobile phones (http://sourceforge.net/projects/nxt-symbian/).

In conclusion, the use of the NXT in the TERECop project was chosen because NXT is a good tradeoff between complexity and expansion possibilities. Last, but not least, cost was taken into consideration as well: the kit's cost allows the students (and the teachers, as well) to buy their own personal kits to continue experimenting at home.

### *2.2.2 Advantages of Lego Mindstorms (from teacher perspective)*

From a teacher perspective, there are other advantages in using NXT in the class. NXT is a widely used platform and there are several resources on the web, ranging from discussion groups to several collections of lab activities at different educational levels. Being adopted worldwide, the teacher is not alone, but part of a large community, which provides technical support and many ready-to-use

examples in the web (even if most of them do not have an appropriate methodological and/or didactical background).

Teachers willing to promote open-source philosophy among their students will find extremely valuable the LEGO choice of going open-source with the NXT firmware. Most of the software projects to fully exploit NXT capabilities are open-source projects and this allows teaching students how complex projects and innovative solutions can be built exploiting the open-source philosophy.

The Mindstorms NXT-G graphical interface is built on National Instruments LabView (www.ni.com/labview/). National Instruments and LEGO are collaborating since 1998 with the development of ROBOLAB, the programming software created for the original RCX LEGO MINDSTORMS. LabVIEW substituted the text-based programming approach for a graphical programming environment. It is used for automated measurement and control systems in many industrial test facilities. LabVIEW is used in several didactical laboratories (especially in technical schools) because of the ease with which it is used and the wide diffusion in the industries. There are many similarities between LEGO MINDSTORMS NXT-G software and LabVIEW. NXT-G retain all of the core of the LabVIEW graphical programming elements, while optimizing the user interface for novice computer users.

There is also a LabVIEW Toolkit for LEGO MINDSTORMS which is used to create and download files to operate and control NXT. Additionally, that toolkit makes it possible to create native blocks for MINDSTORMS NXT software. This allows a step-by-step migration from programming a NXT robot to programming an industrial test facility in LabView. The students will start programming their robots using NXT-G, then they can learn how to program NXT robots in LabVIEW, and, in the end, they will smoothly learn how to program and control real industrial measurement instruments and actuators with the full LabVIEW potentialities.

Similarly, one can use NXT robots to teach MATLAB and Simulink (www.mathworks.com/) programming. MATLAB is a high-level programming language for numerical computing, data acquisition and analysis. It can be used to control LEGO NXT robots over a Bluetooth serial port (serial port communication is part of the base functionality of MATLAB). Simulink is a MATLAB-based environment for modeling and simulating dynamic systems. Using Simulink, a user can design control algorithms, automatically generate C code for those algorithms and download the compiled code onto the LEGO NXT. Several examples of use of NXT programming to introduce MATLAB scientific programming capabilities have been developed, one example being the work developed by Prof. T. Aach at the RWTH Aachen University, which can be found at http://www.mindstorms.rwth-aachen.de/.

However, there is a major disadvantage with NXT: being so general and easy to use, it is somehow limited in its expansion capabilities. Indeed, once the students work with it for a year or more, they can easily reach the hardware limits of the NXT computer brick in terms of both, computational power and hardware expansion capabilities. At this stage, the student is mature to move on to more powerful, more flexible robotics kits. However, when moving to more advanced kits, the teacher should not loose the focus on the didactical and methodological aspects. In order to simplify that transition, examples of TERECop laboratory activities are provided, as implemented on small humanoid robots (see Chapter 5).

## *References*

Official LEGO Mindstorms website: http://mindstorms.lego.com/

HiTechnic website: http://www.hitechnic.com/

Mindsensors website: http://www.mindsensors.com/

Vernier website for NXT sensors: http://www.vernier.com/nxt/

Next Byte Codes (NBC) homepage: http://bricxcc.sourceforge.net/nbc/

Not eXactly C (NXC) homepage: http://bricxcc.sourceforge.net/nxc/

*Not Quite C* **(**NQC) homepage: http://bricxcc.sourceforge.net/nqc/

leJOS homepage: http://lejos.sourceforge.net/

NXT-Symbian homepage: http://sourceforge.net/projects/nxt-symbian/

URBI, Universal Real-time Behavior Interface webpage: http://www.gostai.com/

MATLAB and Simulink webpage: www.mathworks.com/

Harel, I. and Papert S. (1991). *Constructionism.* Norwood, New Jersey: Ablex Publishing Corporation.