A more structured way to teach robotics with robotics

Kostas Dimitriou
Hellenic American Educational Foundation,
Athens College-Psychico College,
Psychico College, High School.
Psychico: 15 Stefanou Delta, Psychico 154 52, +30 210 6798100
kdim@haef.gr

Abstract. The scope of this paper is to investigate different ways to teach robotics in high school. In most cases during problem based learning activities, students start with an initial solution of the problem given and then try to develop a robotic construction and a program that serve the initial solution. This unstructured process involves a lot of hidden steps and decision loops where the initial solution is heavily modified. An alternative teaching way is to oblige students to follow certain procedures and perform specific tasks during the analysis, design and implementation phase. The second approach follows the basic principles of information systems development, is more structured and simulates industrial and scientific procedures.

Keywords: Teaching robotics, dataflow diagrams, Lego digital designer, NXT, MRDS

1 Introduction

This section describes the theoretical framework and materials used during the educational robotics activities discussed in this paper. It is important to mention that the theoretical framework and the didactic procedures followed in a great extent dictated the choice of materials. All activities took place in school settings during elective computer science courses and artificial intelligence and robotics club extra curriculum actions.

1.1 Theoretical framework

The educational activities described in this paper aimed at designing a learning niche that actively involves students in the solution of authentic problems. The overall framework targets skills such as artificial intelligence, robot programming, robot construction, robot design and autonomous robots.

Problem-based learning (PBL) is an effort to challenge students to face real-world problems. Robotic activities that challenge students to address authentic situations create opportunities for meaningful activities, higher-ordered thinking and critical thinking. PBL helps students to gain experience and expertise and promotes communication and cooperation. When a real life situation is defined as a problem to

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

be solved a definitive or unique solution does not exist independently of the student's experience and knowledge (Glazer 2001).

According to Osteen, PBL has some unique attributes:

1. A PBL starts with a problem to be solved
2. Problems try to emulate the complexity of real life
3. Problem-solving process and final product are important outputs
4. Defined problem is the major driving force

The Systems development life cycle (SDLC) in systems engineering, information systems and software engineering, is a method of building information systems. During educational robotics activities this structured framework can be used to enhance PBL methodology. The major steps of SDLC are:

1. Analysis. Definition of project goals, criteria of success, processes, inputs and outputs
2. Design. Description of anticipated features and operations in detail, algorithms, flowcharts, process diagrams, pseudocode etc
3. Implementation. The real code and robot are developed during this phase.
4. Testing. Multiple checks for errors, see how the hardware collaborates with software.
5. Evaluation. Evaluation of the final construct

1.2 Materials

During the design phase the Lego Digital Designer software was used to familiarize students with Computer-aided design (CAD). This software allows users to build models using virtual bricks. Students are able to develop a robot in a virtual environment, see their final model and print building instructions.

Data-Flow Visual Programming Language version 3.021 was used for the development of flow charts. This application allows students to run and test the algorithms developed with flow chart software. This application was developed by Yuri Margolin and later modified and translated in Greek by Nikolaos Tselios (University of Patras).

The only robotic kit used during all activities was Lego Mindstorms NXT. NXT has some obvious advantages such as modularity, flexibility, expandability, ease of use and low cost. While the robot construction follows a bottom up development starting with the central brick and gradually leading to complex architectures the programming could follow a top down programming approach. NXT can be programmed with different programming languages and at different complexity levels. LEGO provides a lot of sensors while third party sensors are easy to be obtained. One strong advantage of NXT is that the underlying concept follows the constructionism learning approach (Papert 1980, 1986) and has its foundation on Papert's didactic recommendations and experience (Harel & Papert 1991).

The main programming language used was NXT-G. This graphical programming environment is easy to learn and use, intuitive and icon-based. It supports drag-and-drop and its main purpose is to introduce students to programming. Any Input Process Output (IPO) model could be easily developed by choosing program blocks. Programming blocks represent motors, sensors, program loops, random numbers,

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

Boolean and logic expressions and many more programming components. The block by block development could be used to develop programs that range from simple to complex. Procedures that encapsulate complexity could be constructed with the use of MyBlocks while Parallel sequence beams are actually parallel threads. It is important to mention that students that have past experience with StarLogo TNG get familiar with NXT G in minutes.

Unfortunately, the NXT G -Lego Mindstorms NXT ideal combination described above lacks significant computational and storage capabilities. Microsoft Robotics Developer Studio (MRDS R4 Beta) provides a wide range of support to develop and test complex robot applications. RDS4 Beta 2 programming model supports Lego Mindstorms NXT through various services. In MRDS almost every action is performed by a service, which controls a particular software or hardware entity. All services interact with each other much like threads. Services are the basic building blocks of RDS. The use of services is compatible with the idea of behaviors since each service expresses an isolated behavior that is defined independently. Services can be combined (a process called orchestration) so that they can achieve higher-level behaviors by working together.The lightweight state-oriented Decentralized Software Services (DSS) framework organizes, creates, manipulates the services and enables programmers to develop modules that can interoperate on a robot and connected PCs by using a relatively simple, open protocol. A new service can be linked to other services to develop complex services. Visual Programming Concurrency and Coordination Runtime (CCR) helps to handle asynchronous input and output by managing the execution of services and interactions between them. DSS Manifest Editor (DSSME) provides a quite simple development of application configuration and distribution scenarios. DSS Manifests are XML files that can be read by the Manifest Loader Service indicating a set of services that are to be started. Each manifest provides the link between the software and the hardware and can simply be changed to link to other hardware platforms. In addition to these features MRDS also includes a Visual Programming Language (VPL) that provides a relatively simple drag-and-drop environment to develop robotics applications. VPL also provides the ability to merge a collection of connected blocks in a single block. VPL is also capable of generating human-readable code. Moreover MRDS provides a Visual Simulation Environment (VSE) that allows programmers to develop robotics applications without the hardware. In VSE users can simulate a variety of complicated scenes. Any robot compatible with MRDS can interact with the simulated environment. Sample 3D virtual environments enable developers to test high cost robots in a variety of simulation modes. A physics engine accurately handles all parameters of the simulated 3D environment so robots react in real time the way they would in the real world (Workman and Elzer).

2 Proposed Methodology

All activities involved high school students with great interest in robotics and computer science. Most students had a strong theoretical background on computer science and mathematics. They were familiar with Logo programming, flow charts,

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

variables and procedures. A total of six teams with 22 students were involved in the project. A computer science lab was used with 18 computers plus a smart board. All computers were located on surface 1 facing the center of the classroom. Surface 2 served as the testing place of robots and as collaboration table (figure 1).

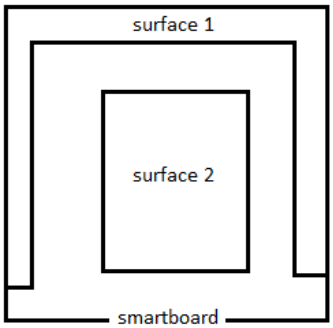

Fig. 1. Class arrangement

The proposed methodology consists of seven different steps (phases). During the first two phases teacher follows a predetermined pattern where the main objective is to explain theoretical concepts and train students in software. All other phases require facilitation of the learning process, so teacher acts as a coach and cognitive modeler. The teacher should give students control over how they acquire knowledge and support the overall process.

Phase 1: Teaching theory

The main objective of this phase was to teach students basic terms, principles and concepts related to computer science and robotics. Table 1 depicts some of the major components of the curriculum taught. Lectures and class discussions were the teaching strategies used during this phase. The total duration of phase 1 was 6 hours.

Table 1. Curriculum components.

Component	Emphasis
Algorithm	Set of instructions for carrying out a procedure or solving a problem.
Behaviors	Behavior based robotics, intelligent robotics and autonomous agents
Bottom up	Components are linked together to form larger systems
Conditional programming	Conditional statements, conditional expressions and conditional constructs.
Data	Data in computing, data processing, data as results of measurements, data as the lowest level of abstraction from which information and knowledge are derived.
Data flow diagram	Data flow diagrams as graphical representation of the "flow" of data through any information system.
Feedback	Feedback loops in software engineering and computing systems.
Flowcharts	Visual representation of algorithms.
Hardware	Physical elements.
Input	Input devices, data and sensors.
Loop	Control flow, repetition, satisfaction of a condition and forever.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

Output	Output devices, motors and lamps.
Procedure	Problem decomposition and operations or calculations that accomplish a specific goal.
Processing	Processing that a robot does to data gathered by sensors and proper manipulation of data.
Sense Plan Act	Gather information, create a world, model and then act.
Sensors	Measurable attributes, analog to digital conversion and calibration.
Software	Various forms and roles of digitally stored data.
Threshold	In relation to sensors.
Top down approach	Breaking down of a problem (situation) to gain insight into its compositional sub-problems.
Variable	Symbolic name and storage location.

Phase 2: Teaching tools

The main objective of this phase was to familiarize students with Lego NXT software (v 2), Lego mindstorms NXT robotic kit, flow chart software, Microsoft Robotics Developer Studio 4 Beta and Lego Digital Designer 4.2. During this phase students were taught the key characteristics and functionality of software used. The total duration of phase 2 was 6 hours.

Phase 3: Problem selection

At the beginning of this phase students were asked to form groups of three or four. Each group was asked to write on a piece of paper a possible situation where a robotic construction could carry out a task. Students were advised to describe problems from real life that have a simple solution. All pieces of papers that described different problems were folded and collected by the teacher. The teacher randomly gave to each team a folded paper. The result of this process was that all teams had a problem that was previously described by another team. Some examples of proposed problems are:

1. A car that moves and tries to find a parking space
2. A crane that finds an object and moves it
3. A vehicle that measures the dimensions of a room

The total duration of phase 3 was 1 hour.

Phase 4: Analysis.

During this phase teams had to define the problem in a more scientific way. Project goals were described in great detail. Teams were asked to describe and construct a simple model that would represent the surrounding environment of the problem. Empty paper and plastic boxes were used to create walls. Inputs, sensors, outputs and motors that had to be used were identified and connections between them were explained. Teams also described specific criteria of success.

Criteria of success included:

1. Detailed description regarding the anticipated behavior of the robot to solve the problem.
2. Maximum time needed by the robotic construction to carry out the task.

The total duration of phase 4 was 2 hours.

Phase 5: Design.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

During this phase students were asked to

1. Write an algorithm to solve the problem.
2. Use the flow chart software to test their algorithm.
3. Use the LDD to virtually develop their robot.

The total duration of phase 5 was 4 hours.

Phase 6: Implementation.

Teams were asked to

1. Build the robot using Lego kit. All teams had to follow exactly the instructions of the building guide automatically developed by LDD.
2. Program the robot using NXT G and then using MRDS. All teams had to follow exactly the algorithms developed during design phase.
3. Test their robots. During testing a team may need to return to the design phase one or more times. The total duration of phase 6 was 3 hours.

Phase 7: Evaluation.

During this phase teams used the specific criteria of success set during Analysis phase to evaluate the performance of their final products. All teams used checklists to evaluate the performance of their final products. The total duration of phase 7 was 2 hours

3 Conclusions and recommendations

At the very beginning when students were informed that they had to follow a specific set of steps they argued that they don't like this process and that they prefer to develop and follow their own methodology. In a short period of time they understood that major constructions, cars, sophisticated products etc are built using a similar approach. They understood that it is very difficult to develop a complex product without good planning.

During the Analysis phase most students understood the importance of problem definition. The definition of inputs and sensors was easy for most students while some teams had trouble in finding the best number of motors needed. All team members identified the connections between inputs and outputs.

During the Design phase it came apparent that team members had to undertake different tasks. Some students cope with LDD while others with the development of algorithms. The collaboration between them was excellent and finally the outputs of different working groups were compatible. Few students didn't find the LDD environment helpful and argued that it would be much better for them to start building the robot directly.

During the Implementation phase the actual program was developed in NXT G and MRDS. Students realized that it is much better to have a clear solution (algorithm) when actually start to write the program. It was very easy for them to use both programming environments. The use of LDD building guide gave the opportunity to students to finish their robot in a short period of time. Only one team had to return to the design phase and correct the algorithm.

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169

During the evaluation phase all teams met their predefined criteria of success.

Most students realized that the proposed methodology serves their needs and answered that they clearly connected computer science concepts to robotics activities. They realized that it is easy to face simple problems but when it comes to complex situations a step by step approach is preferable.

It is very difficult to implement such an approach with younger students without prior computer science knowledge.

In the near future the Visual Simulation Environment of MRDS will be used to test how students will benefit from the development of robotics applications without the use of the actual hardware.

References

1. Glazer, E.: Problem Based Instruction. In: M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology. Retrieved 1/3/2012, from http://projects.coe.uga.edu/epltt (2001)
2. Harel, I., Papert S.: Constructionism. Norwood, Ablex Publishing Corporation, New Jersey (1991)
3. Lego Digital Designer. Retrieved 1/3/2012, from http://ldd.lego.com/
4. Mindsensors. Retrieved 1/3/2012, from http://www.mindsensors.com/
5. Mindstorms. Retrieved 1/3/2012, from http://mindstorms.lego.com/en-us/Default.aspx
6. MRDS. Retrieved 1/3/2012, from Microsoft Robotics Developer Studio, help and support files R4 Beta 2011, Microsoft Corporation
7. Osteen, J.: Comparing/Contrasting Problem and Project-Based Learning. Retrieved 1/3/2012, from http://www.coe.uga.edu/epltt/images/pbl.gif
8. Papert, S.:"Constructionism: A New Opportunity for Elementary Science Education, a proposal to the National Science Foundation (1986)
9. Papert, S: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books. New York (1980)
10. Robotics Academy, The National Robotics Engineering Center, Carnegie Mellon University. Retrieved 1/3/2012, from http://www.education.rec.ri.cmu.edu/previews/robot_c_products/teaching_rc_lego_v2_preview/variablesfunctions/behaviors_functions/documents/NXT_VF_PatternsOfBehavior.pdf.
11. Starlogo TNG. Retrieved 1/3/2012 from http://education.mit.edu/projects/starlogo-tng
12. Tselios, N: Data-Flow Visual Programming Language 3.020. Retrieved 1/3/2012, from http://www.ecedu.upatras.gr/flowchart/Overview.htm
13. Workman, K., Elzer, S.: Creating An Upper-Level Undergraduate Robotics Elective In Computer Science. Retrieved 1/3/2012 from cs.millersville.edu/~csweb/lib/userfiles/selzer/ publications/Workman-final.doc

Proceedings of 3rd International Workshop
Teaching Robotics, Teaching with Robotics
Integrating Robotics in School Curriculum
Riva del Garda (Trento, Italy) April 20, 2012
ISBN 978-88-95872-05-6
pp. 163-169