Design of Design Methodology for Autonomous Robots

Eli Kolberg[1], Yoram Reich[2], and Ilya Levin[3]

[1] Bar-Ilan University, School of Engineering, Ramat-Gan, 52900, Israel
`kolbere@eng.biu.ac.il`
[2] Tel-Aviv University, Faculty of Engineering, Tel-Aviv, Israel
`yoram@eng.tau.ac.il`
[3] Tel-Aviv University, School of Education, Tel-Aviv, Israel
`ilia1@post.tau.ac.il`

Abstract. We present a methodology for deriving design methodology for autonomous robots. We designed this methodology in the context of a robotics course in high schools. The motivation for designing this new methodology was improving the robots' robustness and reliability and preparing students for becoming better designers. The new methodology proved to be highly successful in designing top quality robots. In the methodology design, we explored and adapted design methods to the specific designers, the nature of the product, the environment, the product needs, and the design context goals. At the end of this comprehensive design, we selected a synergetic integration of six methods to compose the methodology for this product context: conceptual design, fault tolerant design, atomic requirements, using fuzzy logic for the control of robotics systems, creative thinking method, and microprogramming design.

1 Introduction

In this paper, we deal with the design of robotics systems. Within this research, we developed a new design methodology for robotics systems [1]. In order to conduct this research we needed teams that actually designed. In order to test several design methodologies in a large-scale comparative study to get reliable and valid results, we had to choose an environment that provides such scale. Consequently, the research could not be implemented in industry because it is not possible to interrupt the ongoing work of many engineers in industry. An alternative environment, where a learning process takes place and has a more structured setting than industrial product development, is the education system. We decided to conduct the research among senior students majoring in science from four high schools, who within a robotics course [2], build autonomous mobile robots for participation in an international robotics contest. We discovered after several years of conducting this course with conventional design methodology that consistent problems were manifested [3].

The primary goal of the course was to teach the subject of robotics to high school students. The following were the course overall objectives: 1. Acquiring technical knowledge; 2. Acquiring a system thinking approach; 3. Improving skills of problem solving, decision making, and learning; 4. Developing critical and creative thinking abilities; 5. Experiencing development of a product, with time and budget restrictions; 6. Developing teamwork skills; 7. Improving students' design skills; and 8. Improving students' perception of technology.

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 528–539, 2008.

Until now, we did not take into account the subject of the design course, namely robotics. Fortunately, robotics products are classic examples of contemporary designs; therefore, the subject – robotics – does not change our analysis. It merely fixes the task of acquiring technical knowledge to deal with robotics related subjects.

The main goal of our research was to develop a new context dependent integrative design methodology for robotics systems, and to measure its success in an existing high school robotics course context.

We used the following design methods as a selection tool: Function-means trees, FMEA (Failure Mode and Effect Analysis), failure analysis, QFD (Quality Function Deployment), Pugh's concept selection, and AHP (Analytic Hierarchy Process) [4]. As an outcome we decided to integrate six design methods in the new design methodology [1]: 1. Conceptual design; 2. Fault tolerant design; 3. ASIT creative thinking; 4. ATR design; 5. Microprogramming design; and 6. Fuzzy logic control design. We introduced the new design methodology for robotics systems into the course. We tested the performance of the methodology in the years 2003-2005 [5].

Figure 1 describes the roadmap of this study. It is composed of theory development and course design followed by course implementation. The results of the course feed back into the theory development and the course was redesigned. The theory underlying the course design is a synthesis of ideas, drawn from different disciplines: engineering design, robotics, learning paradigms, engineering education, project base learning, contest oriented design, and learning by design. These disciplines provide the guidance in the course design, its implementation, and testing. The course design starts from requirements that are translated into course goals to be addressed by a design of the design methodology to be integrated into a detailed curriculum, which is implemented and tested. The results lead to reflection that helps improve our understanding and course design.

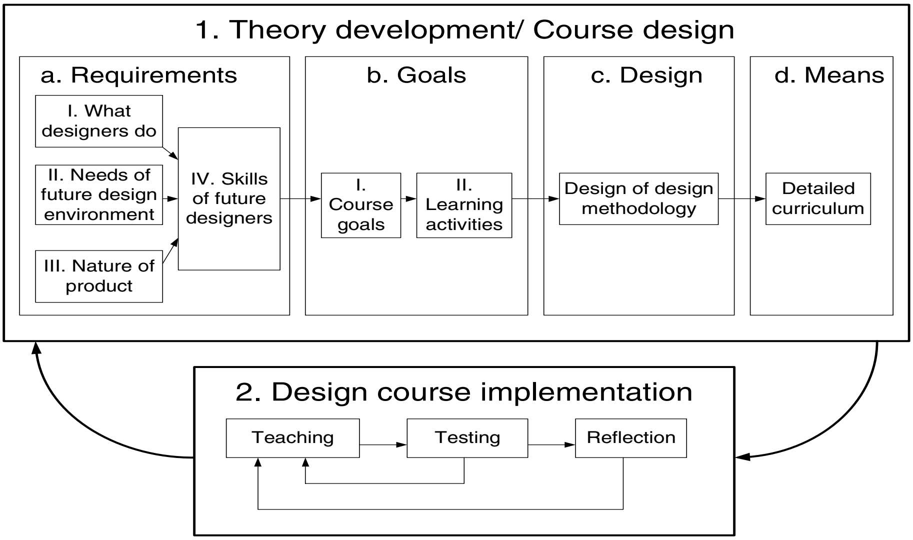

Fig. 1. Roadmap for designing designers

While there have been many studies on the design of curriculum in education and other fields (e.g., [6], [7], [8]), there is no single large-scale study on teaching design that was tested in a controlled experiment and produced conclusive results as the present study.

2 Method

2.1 Design of the Design Methodology (1c in Figure 1)

Since the students had no background knowledge in design, and since they had to complete the course with a quality design in order to compete in the competition, we decided to teach them enough design methods that would allow them to design and build excellent robots. We have also used these methods to teach other general concepts, such that imprecise information could lead to very precise behavior, as in fuzzy logic.

Design environment should allow for a meaningful design experience. The design of a complex product as a mobile robot allows for such experience. The contest supplies both time limits for project completion and environment for testing the results.

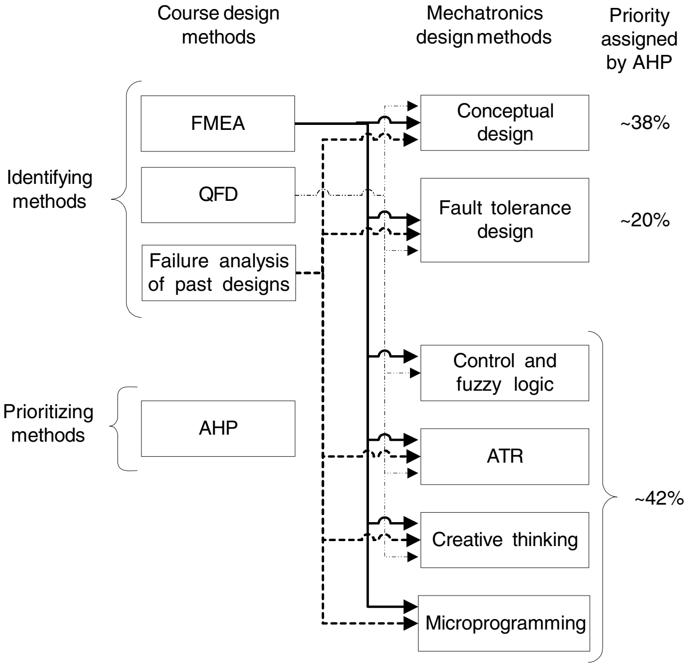

Fig. 2. Course design methods leading to robotics design methodology

Design methods are seldom taught in high schools. Moreover, in spite of their importance, it is even uncommon to teach them in universities. By and large, universities focus on analytical rather than synthesis skills. Our reasons for teaching design methods stem from the course's 2nd to 7th objectives. Moreover, design methods serve as guidelines that help students (as well as designers in real design projects) focus on the critical features when developing any engineering product. In addition, design methods glue technology to science. They help students realize the relations between the different science subjects learned at school and between science and the engineering work in robotics. These are central to understanding robotics as a discipline.

Further support for systematic teaching of design methods arises from feedback obtained from previous courses on the subject. One observation was that ignorance of design methods prevented effective use of expensive equipment purchased to support new technology and science related courses. Another observation was that lack of knowledge about design methods led to numerous occasions in which teams designed robots that violated simple engineering practice, resulting in quick robot failures. While product success is not mandatory for course success, these easily avoidable failures led to students and teacher disappointments, which are undesired. The logic for designing the design methodology and selecting the design methods is presented in figure 2.

2.2 First FMEA Analysis

We did a FMEA analysis for of the common design methods in general ([9], [10]) and specific to the context. We will discuss these methods and present in summary the considerations for disqualifying or accepting each method for further analysis. The design methods were:

1. Selection design – It involves choosing one item from a list of similar items. There is a need to evaluate the potential solutions versus our specific requirements to make the right choice. This is done much more seriously with conceptual design methods. Consequently, we disqualified this design method.
2. Product architecture design deals with the arrangement of the physical elements of the product to carry out its required functions. This is important for any complex system, so this method is kept for further analysis.
3. Configuration design deals with how to assemble all the designed components into the complete product. As the robot has components that have to be assembled, we decided to move this method to next stage.
4. Parametric design identifies the attributes of parts in the design configuration, which become the design variables for parametric design. The objective is setting values for the design variables that will produce the best possible design considering both performance and manufacturability. As the robot's subsystems have different attributes and optimal performances are needed, we decided to move this method to further analysis.
5. Original design – Any time the design requires the development of a process, component, or assembly not previously in existence, it calls for original design. This sort of design being original does not supply tools for doing specific design and thus it is not relevant here. We disqualified it from further analysis.
6. Conceptual design (CD) is one of the two most critical steps in product development. It is the basic design method; it places things into order. It allows to realize the big picture, and to see the important factor out of the large amount of data. It also can be used to divide the work between team members quite effectively and complete the project on time. Hence, we decided to move it to the next stage for further analysis.
7. Concurrent design deals with cross-functional design team, where skills from the functional areas are embedded in the team. This allows for parallel design. This mainly

refers to heavy designed products where for example the manufacturing process development group starts its work as soon as the shape and materials for the product are established, and the tooling development group starts its work once the manufacturing process has been selected. In our case, we talk about small teams where there are no design skills differences among team members, there are no manufacturing or tooling teams, and there is no much meaning to parallel development. That is why we disqualified this method for further analysis.

8. Atomic requirements (ATRs) design – Atomic (which cannot be further divided to two or more requirements) Requirements design is a tool that helps to understand the functionality and debugging requirements; it allows to divide the requirements into very fundamental, thus simple to understand requirements. It also helps to clearly identify unnecessary, overlapping, or conflicting requirements, isolate bug areas, and make clear what is to be done to implement the requirements. In the debugging mode, and problem solving, each requirement can be tested easily and separately. It is an efficient communication tool between people of different backgrounds. It is suitable for making more modular and more convenient to debug and fix robot systems. Hence, we decided to move it to the next stage of further analysis.

9. Ergonomic design deals with interaction between people and the product. As some interaction occurs between the team members and the robot in the testing and operating the robot, we decided to move this method to the next stage of further analysis.

10. Microprogramming (uP) design is common with products that include a microprocessor or a microcontroller. It allows for designing the robot control by moving between two different representations that make it easy for designing, debugging, and coding, simultaneously [11]. Microprogramming design presents the duality between two representations of control schemes and that even though it is more “natural” to use one to describe the robot operation, it is better to use another in order to be more robust and efficient. It also shows a way for being more effective when for example it is possible to combine two or more control schemes and save resources. Generally, it shows duality in two representations and understand that different representations are suited to different needs – a powerful problem-solving principle. The robot's control was based on a microcontroller, so we moved this method to next stage.

11. Industrial design is concerned with the visual appearance of the product and the way it interfaces with the customer. These two are irrelevant to our robot, and thus will not be considered further.

12. Fault tolerance (FT) design is crucial for creating robust products and it is inseparable method of every good design. It brings insight of the difference between products that are designed according to requirements, and robust products that can sustain faults up to a certain degree. It also introduces the possible faults during the design phase which improves the ability to identify and overcome problems. This influences on being more careful when design the robot parts, for example, the sensors array. It also demonstrates that in unstructured environments, no design could survive without making it robust to faults because it is usually impossible to foresee all potential situations. We moved it to the next stage.

13. ASIT (Advanced Systematic Inventive Thinking) creative thinking design is a systematic method for creative thinking, which is designed especially for problem solving. It is important when a solution to a non-trivial problem is needed. By using

this method, it is possible to solve complicated problems. It seems fundamental in all design stages. We moved it to further analysis.

14. Design for serviceability is concerned with the ease with which maintenance can be performed on a product. Products often have parts that are subject to wear and that are expected to be replaced at periodic intervals. That calls for a maintenance service. The robots built by the students are not a product that is intended for an extended use. There is no need for periodic service like oil replacement in cars. The part of design for easy access for parts replacement is covered in conceptual design. Hence, we decided to disqualify this method for further analysis.
15. Fuzzy logic (FL) helps in simplifying things related to motors control. It is more straightforward and can be checked in an easy way, compared to other control methods. It is more intuitive to the students and is faster in implementation than other control methods. Fuzzy logic control design is used successfully in industry and we thought it would be adequate to move it to the next stage.
16. Design for the environment is concerned with issues such as recycling, environmentally friendly materials, product waste minimization, packaging recovery, and noise reduction. Some of the robot parts are reused from previous years' materials, so we moved this method to next stage.
17. Detail design is the way to realize the product. We will move this method for further analysis.
18. Design for manufacturability – As the robots will not be manufactured beside for the project, we disqualified this method from moving to the next stage.
19. Usability design – here the designer fits the product to user's physical attributes and knowledge, simplify user tasks, and make the user controls and their functions obvious. This is irrelevant for the course autonomous mobile robot, so we disqualified this method from further analysis.
20. Design for reliability is quite similar to fault tolerance design (clause 12), which makes it redundant. That is why we disqualified it from further analysis.

After this session, 11 methods remained as candidates: conceptual design, ergonomic design, product architecture design, atomic requirements design, microprogramming design, fault tolerance design, parametric design, configuration design, ASIT creative thinking design, fuzzy logic control design, and detailed design.

2.3 QFD Analysis

We performed QFD analysis for selecting the design methods according to the criteria presented in table 1. The criteria were treated as the requirements and the design methods as the engineering characteristics.

Table 1. Robot's performance evaluation criteria

	Criteria		Criteria
1	Success in the contest	8	Fast navigation to all rooms
2	Driving well in corridor	9	Overcoming uneven floor
3	Making 90 and 180 degrees turns	10	Obstacle avoidance
4	Driving well in reverse mode	11	Non tethered robot operation
5	Finding a white line on a black background	12	Sound activation of the robot
6	Finding a lit candle in a room	13	Navigation from each room back to starting point
7	Fast extinguishing of a lit candle		

Table 2 presents the QFD analysis [10] for choosing the appropriate design methods. Based on the criteria, the “whats” are listed in room 1. Room 4 lists the various design methods that should be checked against the criteria. Next, we turn to room 2. The *criteria importance* was established by interviewing teachers and mentors, and allocating the views along a 1-5 scale, where 5 is the highest. The previous years' robots were ranked according to the way in which they satisfied requirements, on a 1-5 scale, and subsequently, the planned robots were rated against the requirements. In room 3, the ratio between the planned to previous robots is called the *improvement ratio*. The product of *criteria importance* x *improvement ratio* gives the *total improvement ratio.* The *relative weight* is each value of *total improvement ratio* weight divided by the sum of all values of importance weight. The relationship matrix, room 6, tells us how each design methods help attain the criteria list. Here a strong impact is worth 9, a medium high impact 5, a medium low impact 3, and a weak impact 1. The importance of the design methods in room 7 is determined by multiplying each of the cells in the matrix by its *relative weight* and summing each column to give the *absolute importance*. The *relative importance* in room 8 is obtained by dividing the *absolute importance* by the sum of all absolute importance values. Six methods rank highest and almost twice as high as the next in line: conceptual design, fault tolerance design, atomic requirements design, ASIT creative thinking method, use of fuzzy logic in robot control, and microprogramming design.

Table 2. QFD analysis of design methods

		Conceptual design	Ergonomic design	Product architecture design	Atomic requirements design	Microprogramming design	Fault tolerance design	Parametric design	Configuration design	ASIT creative thinking method	Use of fuzzy logic in robot control	Detail design	Criteria importance	Previous years robots	Planned robot	Improvement ratio over previous robots	Total Improvement ratio	Relative weight
1.	Performance	9	1	9	9	9	9	9	9	9	9	9	5	4	5	1.3	6.5	0.058
2.	Real time hardware failure resistance	9	1	1	5	5	9	1	3	5	9	1	5	2	5	2.5	12.5	0.111
3.	System simplicity	9	1	3	9	9	5	1	1	9	9	1	4	3	4	1.3	5.2	0.046
4.	Flexibility	9	1	9	5	9	9	5	1	9	9	1	4	2	5	2.5	10.0	0.089
5.	Robot reliability	9	1	3	9	5	9	5	5	5	9	3	5	3	5	1.7	8.5	0.076
6.	Software modularity	5	1	1	9	9	9	5	1	5	3	1	3	2	5	2.5	7.5	0.067
7.	Robot testing ability	9	1	9	9	9	9	1	1	9	9	3	4	2	5	2.5	10.0	0.089
8.	Fast hardware fixing	9	3	9	9	1	5	1	3	9	1	1	4	2	4	2.0	8.0	0.072
9.	Ability of upgrading	9	3	3	1	9	9	9	1	9	9	9	2	2	5	2.5	5.0	0.045
10.	Cost saving	9	1	1	9	9	9	1	1	9	9	3	3	1	5	5.0	15.0	0.134
11.	Ease of transferring the subject matter	9	1	1	9	5	5	1	3	9	9	1	5	3	5	1.7	8.5	0.076
12.	Short learning time	5	3	3	9	5	5	1	1	9	9	1	5	5	5	1.0	5.0	0.045
13.	Ease of use	9	1	5	9	9	9	1	1	9	9	3	5	5	5	1.0	5.0	0.045
14.	Can be modified to high school students	9	1	3	9	9	9	3	3	9	9	3	5	5	5	1.0	5.0	0.045
	Absolute importance	8.53	1.32	4.16	7.82	7.17	8.03	2.84	2.37	7.97	8.00	2.6	60.8				111.7	0.998
	Relative importance	**0.14**	0.02	0.07	**0.13**	**0.12**	**0.13**	0.05	0.04	**0.13**	**0.13**	0.04						

2.4 Failure Analysis and Maim Problems Encountered with Previous Robots and Possible Solutions

Another method used for selecting the design methods was failure analysis where poor design practice was analyzed. We reviewed many of the previous robots available description and data, including interviews with teams, reading project reports, observing failures of robots from previous competitions, and getting the robots performance in local and international competitions. Upon organizing and sorting the data, we found the following as the main problem issues.

1. Need for several hardware and software changes and modifications. We found that it was common among many teams to totally redesign their robot more than once. The most appropriate solution to this kind of problem would be implementing conceptual design methods.
2. Malfunction equipment. Sometimes robots are not qualified in their trial runs due to malfunctioning equipment. A solution to this could be to introduce checkers that identify sensor failure and upon identifying the above failure, change the position control.
3. We observed that high school students in general had difficulties in designing reliable robot speed and position control. The students had difficulties to calculate or experimentally find the proper gains of the PID control loop and were not aware what was happening with the robot control. They knew the formality needed for implementing the control but they knew neither the essence of it nor how to decide on proper gains. In some cases, the improper gain values caused the robot to be too slow or too fast, and as a consequence the robot hit the wall. The use of fuzzy logic control could remedy these difficulties.
4. When students reached the design stage, they stated the robot requirements among their team members in an ambiguous way. There was also lack of ability to test and debug the robot, because of contradicting or unclear requirement definitions. The ATR method would address these problems.
5. Occasionally, the teams did not overcome the problems properly. Solving these problems was possible if the students would apply a creative thinking method like ASIT.
6. The last noticeable group of problems was the difficulty to follow and debug an ASM algorithm; in many occasions the students did not cover all possible situations. Using an FSM algorithm might inherently prevent these situations. In FSM, all states and transitions must be declared and taken care off, and it is easy to find an uncovered situation. Hence, translating the ASM to FSM is important, and was done by microprogramming techniques, which also allow for integrating several sub algorithms and saving code.

2.5 Another FMEA Analysis of Adapting the Design Methods for High School Students

In order to reduce the chances of failing with this methodology in the particular context in which it was implemented, we exercised FMEA and tried to think of the issues that could make it fail and to produce some counter measures. As the students were inexperienced, we had to adapt the methodology to their lack of engineering mathematics skills and experience. This led to preparing an appropriate training course for these students and designing the curriculum to enable logical teaching to the students. Another critical issue that we faced was the modification of industry methods to suit the teaching environment of a high school where students lack prerequisite knowledge. Next, we will describe the modifications made to each of the design methods for their inclusion in the course material.

Conceptual design**:** The teaching of conceptual design requires no prerequisite knowledge; however, the time constraint forced a short version that was modified to suit the needs of the students. The stages of problem definition, and identifying customer needs with subjects, such as how to interview the customer, making focus groups, preparing customer surveys and handling customer complaints, were not taught because contest rules can be regarded as stating the problem and covering the customer needs. Only a small part of benchmarking was taught, as there was no identical commercial product to test against. There were robots from the year before, which were analyzed by the teams in comparison to their robots.

Creative/inventive thinking**:** ASIT was taught completely as it requires no special background and could easily be taught to the students in a short time. Another assisting factor in using ASIT was that we had an accessible simple training material that could be distributed to students for home practice.

Fuzzy logic**:** As the designers were high school students, no intensive mathematics background was introduced. The fuzzy logic (FL) control subject was introduced to the students as a technical straightforward procedure. The students learned to create the variable membership functions, adapted to the capabilities of the microcontroller they used; derive the fuzzy rules; and receive the output variable for further processing.

Robot control**:** Robot control was taught using an innovative teaching method built upon the use of dual representations. The method was taught without the intensive mathematical manipulations. It is further explained through the microprogramming subject.

Atomic Requirements**:** This method was taught completely. It requires no special background and could easily be taught to the students in a short time.

Microprogramming**:** Microprogramming is an approach to teaching a number of subjects related to computer hardware. We adapted microprogramming for designing robotics systems. The main idea of this adaptation is based on considering a robotics system to be a composition of two units: *a control unit* and an *operational unit* [12]. The operational unit of the system includes such building blocks as motors, sensors, lamps, manipulators, etc. The control unit receives information from the operational unit and produces a sequence of control signals that results in executing desired operations by the operational unit. Usually microprogramming is a subject that is studied at the undergraduate level. It is built on a number of strong prerequisites including introductory logic design and programming. For introducing the subject into the high school robotics course, we have developed a specific “microprogramming curriculum” including a number of well formulated formal notations and definitions. The curriculum skips some technical details connected to specific computer architecture. Further, the presented microprogramming concept includes only a Finite

State Machine (FSM) based microprogrammed controller, and not the classical Wilks architecture. It allows presenting the concept of microprogramming in a simpler manner and makes it practically productive for the process of robotics design.

***Fault Tolerant Design*:** All fault tolerant (FT) components are useful in robotics systems design. Particularly, the robotics design described in this study includes one necessary component for fault tolerant design, which is the self-checking design. Within high school curriculum, the self-checking design was based on the development of specific redundant units, so-called checkers. The main goal of the checker is to prevent entering incorrect data to the control and operation parts of the robotics system. Students are able to construct checkers for robotics systems by using a number of standard solutions for the checkers design. These solutions are based on fundamental principles of self-checking design: fault secure and self-testing property. Students had to develop an appropriate checker and also prove it's self-checking.

Within the current course design, the mathematics involved with FL, microprogramming and FT was too complicated. Yet, even by eliminating the mathematical details, there was sufficient benefit to teach these methods and use them. We considered teaching neural networks control but found it too complicated and of little importance. We also considered teaching 3D modeling and schematic software but the teaching overhead and the software cost would not justify their inclusion.

2.6 The Chosen Methods for the Design Methodology and AHP Analysis of Design Methods Selection

To conclude, besides the general confidence about introducing design methods into the classroom, we used three guidelines to design the design methodology to teach: (1) addressing poor design practice by previous years' teams; (2) introducing methods that had high impact on attaining course's goals; and (3) avoiding complex methods. The six methods selected are complementary and cover the complete design process; they include: Conceptual design, ASIT creative thinking method, ATR design, Fault tolerance design, Microprogramming design, and Fuzzy logic control design.

Within the scope of the possible robotics design methods, these have an important role or influence over the product quality and its performance in field conditions. Moreover, these methods allow appreciating issues beyond the original goals. For example, fuzzy logic allows appreciating that mathematics is not always about precise numbers. In fact, a great deal of engineering reasoning is qualitative and imprecise [13]. Fuzzy control demonstrates that imprecise concepts lead to very robust behavior that is relatively easy to attain.

Subsequent to identifying the design methods, two experts used AHP [4] to prioritize the methods in order to allocate them the necessary teaching resources. It was agreed that conceptual design is the most important method (importance 42% out of 100% for one expert and 34% for the second)**.** The method that was secondly important was fault tolerance and testability (19% and 22%, respectively). The expert agreed on the following four methods but differed in the order of importance that they assigned to each method. Nevertheless, the expert assessment and our own judgment were quite consistent. After the relative importance evaluation, and given the stringent teaching hours limit, we decided to teach subsets of these design methods that deemed critical to the robot design or that would contribute significantly to other course goals. The findings and the experience from the first year of conducting this research approve the effectiveness of these methods.

3 Implementation, Evaluation and Validation of the Design Methodology

The resulted methodology was taught fully or partially in three schools and traditional design study was taught in a fourth school. The teaching in all of the four schools was conducted in parallel. The parallel tracks ended with measurement, evaluation, and comparison between the four schools.

In order to evaluate and validate the methodology we followed the principles presented by Nevo [14], and Marshall [15]. The following points support the design methodology validation.

1. Nevo's structure for evaluation recommendation items like evaluation background, the conceptual frame, the questions which the evaluation tried to answer on, and methods, and outcomes, are covered and described in this paper.
2. All the design methods included in the methodology are known and proved to be efficient and effective.
3. The criteria for testing the design methodology are clear and can easily be tested.
4. The methods within the design methodology are complete and orthogonal.
5. The testing and validation process was done in large scale within three years: pilot study in the year 2003, full implementation in the year 2004, and transfer implementation in the year 2005. In each phase, a careful research work was done. The implementation in four schools among 127 students and 7 teachers further supports the validation of the methodology.
6. All results are measurable.
7. The methods in the methodology were chosen from a larger list. Moreover, as we state that the methodology is context dependent; it might and probably will change in the case of different contexts.

4 Discussion

Through careful design, implementation, and testing, we developed a design methodology for robotics systems. We described the process of deriving the design methodology and the importance of context dependent design. In this case, we made adaptations related to the context of the design, namely: high school students, high school environment, the product, and the contest. Each of the six methods had its own special contribution to the design, to the product of the design – the mobile robot- and to the students. The students were aware of the design methods they learned and we observed that the students developed abilities to apply the proper design methods to specific problems they encountered in the project.

We argue that design is context dependent. We showed that taking into account the specific design conditions leads to a tailored design methodology. We believe that with the same design methodology approach, technology courses can be taught in universities and industry, yielding even more profound benefits to designers.

References

1. Reich, Y., Kolberg, E., Levin, I.: Designing designers. In: Proceedings of International conference on engineering design, ICED 2005, Melborne (2005)
2. Verner, I., Waks, S., Kolberg, E.: Upgrading Technology Towards the Status of High School Matriculation Subject: A Case Study. Journal of Technology Education 9(1), 64–75 (1997)
3. Kolberg, E., Reich, Y., Levin, I.: Project-Based High School Robotics Course. International journal of Engineering Education 19(4), 557–562 (2003)
4. Saaty, T.L.: The Analytic Hierarchy Process. McGraw-Hill, New York (1980)
5. Reich, Y., Kolberg, E., Levin, I.: Designing contexts for learning design. International Journal of Engineering Education 22(3), 489–495 (2006)
6. Barrows, H.: How to Design a Problem Based Curriculum for the Pre-Clinical Years. Springer, N.Y (1985)
7. Clark Jr., E.T.: Designing and Implementing an Integrated Curriculum: A Student-Centered Approach. Holistic Education Press, Brandon (1997)
8. Diamond, R.M.: Designing and Assessing Courses and Curricula: A Practical Guide, Jossey-Bass, San-Francisco (1998)
9. Ullman, D.G.: The Mechanical Design Process. McGraw-Hill, N.Y (1992)
10. Dieter, G.E.: Engineering Design: A Materials and Processing Approach. McGraw-Hill, Singapore (2000)
11. Levin, I., Kolberg, E., Reich, Y.: Robot Control Teaching with a State Machine Based Design Method. International Journal of Engineering Education 20(2), 202–212 (2004)
12. Baranov, S.: Logic Synthesis for Control Automata. Kluwer Academic Publishers, Netherlands (1994)
13. Subrahmanian, E., Konda, S.L., Levy, S.N., Reich, Y., Westerberg, A.W., Monarch, I.: Equations aren't enough: Informal modeling in design. AI EDAM 7(4), 257–274 (1993)
14. Nevo, D.: Useful Evaluation: Evaluating Educational and Social Projects, Massada, Israel (Hebrew) (1989)
15. Marshall, C.: Goodness criteria: are they objective or judgment calls? In: Guba, E.G. (ed.) The paradigm dialog, Goodness Criteria, Sage publications, California (1990)