

# Non-Expert Construction of Customized Embedded Systems to Enhance STEM Curricula

Anuradha Phalke\*, Miria Biller<sup>†</sup>, Susan Lysecky\*, Christopher Harris<sup>†</sup>

\*Department of Electrical and Computer Engineering  
University Of Arizona  
{aphalke, slysecky}@email.arizona.edu

<sup>†</sup>Department of Teaching and Teacher Education  
University Of Arizona  
{miria, harriscj}@email.arizona.edu

## ABSTRACT

We present an interactive platform to enhance STEM education through project-based activities that complement the existing curriculum, while drawing on problem solving skills and team work. The addition of hands-on projects is intended not only to reinforce the various concepts learnt through normal class readings and discussion, but also make these topics multi-dimensional by enabling students to apply these principles in their everyday surroundings. The types of courses that can take advantage of the eBlock platform range from middle school math and science classes to university level engineering courses. In this paper we present the eBlocks platform which can be utilized to implement a wide variety of projects without requiring users to have programming or electronics experience. Specifically, we summarize the preliminary usability results, discuss possible applications of the eBlock platform for various age groups, and introduce the initial curriculum development for middle school audiences.

## Categories and Subject Descriptors

H5.2 [User Interfaces]: User-centered design; Evaluation/methodology, Interaction styles; Prototyping; H5.m [Miscellaneous]

## General terms

Design, Human Factors

## Keywords

eBlocks, user interfaces, STEM education, interactive embedded systems building blocks, project-based learning

## 1. INTRODUCTION

The math and science knowledge and skills of K-12 students within the United States are continuously evaluated [1][11][12], with less than stellar results. An ongoing concern is that student rankings are dropping with respect to their international counterparts. Beyond secondary schools we see engineering enrollment and retention at the college and university level is adversely effected [22], fewer students are pursuing science and engineering degrees, with only about 6% of undergraduates majoring in engineering [4]. Even further reaching, is the continued debate as to whether there is an “engineering gap” in the United States in that many engineers are retiring but there are not enough engineers to fill these spaces [13][34][36]. The U.S. Bureau of Labor Statistics projects 15 of the 20 fastest growing occupations through 2010 will require significant math and science backgrounds [31]. Given these statistics, we look for

possible methods to enhance STEM (Science, Technology, Engineering and Mathematics) performance, as well as increase interest in pursuing STEM careers.

The benefits of project-based learning, where students actively participate in the learning process rather than passively listening to a lecture, reading a book, or watching a video, have been studied for decades [2][14][15][17][25][28]. Research has shown that students who are able to investigate the world around them and pursue topics that are relevant to their everyday lives not only achieve higher scores than students in traditional classrooms [9][30] but are also more engaged [3][21]. While these learning models have been shown to be effective, traditional STEM courses struggle to provide an active learning environment. Incorporating advanced learning technologies into the curriculum is one promising way to create and sustain project-based learning opportunities. Yet, often times learning technologies are not feasible because test equipment is too expensive and school budgets are already spread thin, while building a custom solution requires a large breadth of programming and engineering expertise. Thus, we are faced with a challenge to develop a platform that is usable by a wide range of teachers and students, cost-effective for schools, and support a variety of STEM learning goals.

We propose to utilize and extend the eBlocks platform [5], a platform composed of numerous fixed-function building blocks, initially developed to enable non-expert users to create customized monitor/control systems. The eBlock platform has the potential to provide an interactive and customizable project-based learning environment that allows students to see how the topics studied in class relate to real-world applications by enabling students to create their own customized systems. In this paper we provide the motivation behind the eBlocks platform, a glimpse of the underlying platform implementation, and initial usability studies to demonstrate the feasibility of the eBlocks platform within STEM education. Lastly, we discuss potential applications of the eBlock platform and future directions.

## 2. RELATED WORK

While a great deal of effort has been put forth to improve STEM education, we only highlight a few such projects that seek to include technology as a mechanism to enhance motivation and learning of participants. Logiblocs [16] are one such technology consisting of small plastic blocks that users snap together to build various systems and consist of light sensors, buttons, AND, OR, NOT, speakers, bleeps, LEDs, etc. Logiblocs are targeted for primary education and educational toys. Children are able to

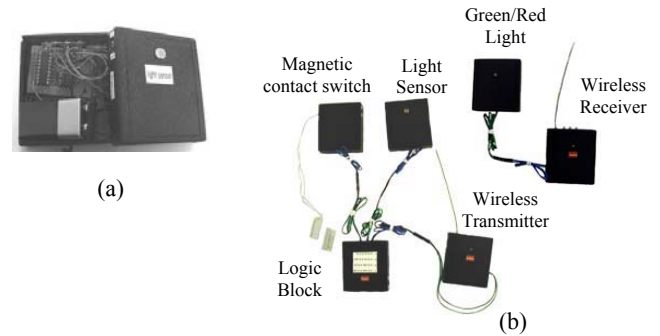


**Figure 1: eBlocks provide a platform which enables non-experts to build a variety of interactive, hands-on projects.**

design various systems and gain insight into the design process by trial and error. While many systems are viable such as flashing warning lights, light breaker alarms, or simplistic robots, the simplistic interfaces limit the possibility to build more advance systems to accompany STEM projects beyond the primary school level. Electronic Blocks [39] also utilize small building blocks consisting of processors incorporated inside of LEGO Duplo Prima blocks. Users simply stack the correct combination of blocks to produce the desired output. Electronic Blocks consist of blocks that detect light, sound, and touch; produce physical output such as illuminating a light, producing sound, or movement; and intermediate blocks alter the expected action by inversion, toggle, or delay. Electronic Blocks are aimed at students between the ages of 3 to 8 years old and are again limited in use for older students due to the simplicity of the blocks.

Perhaps the most closely related project targeting STEM education in primary and middle school is that of MIT Crickets, having evolved from the MIT Programmable Bricks project [18][19]. A Cricket is a tiny computer, powered by a 9-volt battery that can receive information from two sensors and can control two motors. A key principle in Crickets is that people program them to perform a variety of functions. To make programming simple, the Logo language is used [26][27] – a simple, graphical, highly intuitive language. Crickets provided the foundation for the Lego Mindstorm product [37], consisting of numerous sensor and actuator Lego blocks that can be connected to a central microprocessor block to build a variety of small robots, again programmed using a simple graphical language. Crickets provide a general programmable computer, requiring programming by users. While we can learn much from the MIT Cricket approach, we have to consider situations where learning programming languages or having access to computers are not an option. Our goal is to complement the current curriculum without requiring programming or electronics knowledge.

The Infinity Project [10] is an intriguing year-long course available to high schools to expose students to, and excite students about, careers in engineering and technology. The course curriculum is designed for students who have completed Algebra II and at least one science laboratory. A graphical software design environment is provided to run on a PC accompanied by a Technology Kit including a central board built around a digital signal processor (DSP), multimedia peripherals, such as digital web camera, speakers, and microphones, and an accompanying



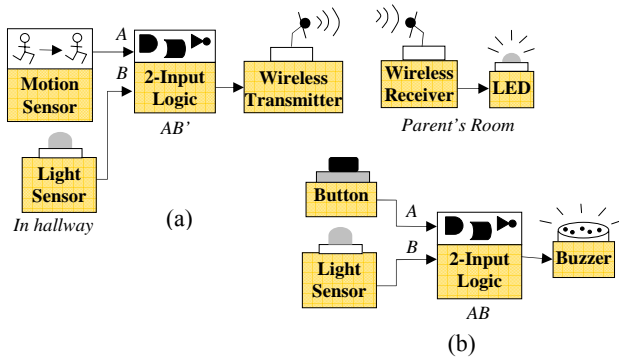
**Figure 2: (a) Internals of an eBlock light sensor, (b) implementation of a Garage-open-at-night system using eBlock prototypes.**

text book. Students learn about engineering through the design and experimentation on topics including the Internet, wireless communications, digital imaging, and music and audio processing. The Infinity Project is a fascinating project but mandates prerequisites in math and science. Additionally, we want to involve students before they reach high school. Our goal is for students of all ages and background to be able to easily utilize the eBlock platform to build and configure a variety of systems, providing an opportunity for hands-on projects relating topics discussed in class. While the goals of each project differ, our proposed project can serve as a complement to the Infinity project. Studies have shown students need to be engaged early. A central purpose of the eBlock project is to give middle school (and eventually elementary school) students the confidence to continue to pursue math and science and participate in programs, such as the Infinity project, throughout their education.

As shown here, numerous platforms are available which expose students to technology, with many focusing on the underlying platform. While an introduction to, and an understanding of, the technologies that makeup the digital age is important, we strive to enable students to utilize technology without having expert knowledge of the underlying platform.

### 3. EBLOCK OVERVIEW

The inspiration for the eBlock platform arose from a simple observation that there exist numerous applications across a variety of domains (residential, commercial, medical, and so on) that can be solved with a handful of sensors and some control logic. For example, a homeowner may want a system to monitor whether a garage door was left open at night. Figure 2(b) illustrates how this system can be implemented using a light sensor, contact switch, a LED (light-emitting diode), and control logic that turns the LED red when no light is sensed (nighttime) combined with an open contact switch (indicating the garage door is open). Additionally, if the homeowner wants to place the red/green warning LED away from the sensors, wireless transmit and receive modules can be utilized. As another example, a homeowner might want to set up a system that detects if their child is sleepwalking in the dark. Figure 3(a) illustrates such a system involving a motion sensor block and a light sensor block, feeding into a logic block detecting the motion sensor outputting true and the light sensor outputting false, wirelessly feeding into an LED or buzzer block. Other examples include, but not limited to, a daytime doorbell, detecting



**Figure 3: Various applications built with eBlocks, (a) Sleepwalking Detector and (b) Daytime Doorbell.**

motion on the property, detecting if a side gate is open before letting pets into the backyard, monitoring the availability of conference rooms within a company, or monitoring nocturnal activity by turning a recording device on when motion is sensed within a defined area.

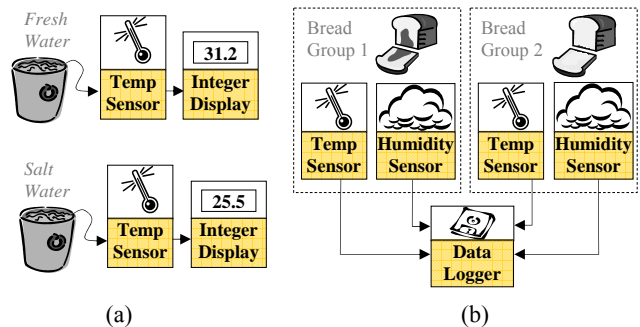
With so many application possibilities, why aren't these systems more prevalent? One reason is cost; most applications are too specialized to be cost effective when one considers the many real costs (packaging, marketing, store placement, etc.) of introducing a new consumer product into the market. Thus, very few off-the-shelf applications appear in the market place. Furthermore, while these applications are seemingly simple, building a custom working system from standard electronic components is beyond the skills of ordinary people (homeowners, teachers, office workers, etc). In fact, building these systems is even a challenge for most engineers who haven't been specifically trained in embedded system design. Lastly, hiring an engineer to build a custom solution is possible, but the cost is seldom justifiable.

The eBlocks platform [5][8] emerged from a desire to empower regular people, having no programming or electronics experience, to build custom electronic systems. Rather than creating specialized systems for each application scenario, the eBlock platform is composed of commonly used building blocks that users connect blocks together like Legos™ to form an application. Because the platform consists of building blocks, the development cost of each block can be amortized over a larger volume, yielding lower costs. Furthermore, through the use of hardware wrappers the underlying platform implementation is made transparent to users, alleviating the need for programming or electronics experience.

### 3.1. EBLOCK PLATFORM

The key to the eBlocks platform is to integrate compute intelligence with previously “dumb” sensors and actuators, as shown in Figure 2(a). In our case, we utilized a PIC16F690 microprocessor [20]. The processor hides low-level interfacing details and communication between nodes so that users simply connect components together to specify the functionality of the application.

A variety of building blocks have been identified, with approximately 20 different blocks physically implemented. These



**Figure 4: Sample eBlock projects. In (a) eBlocks are used to observe the freezing of fresh and salt water. In (b) eBlock temperature sensors are used to observe the effects of temperature and moisture effect mold on a piece of bread.**

building blocks can be classified into one of the following four categories.

- Sensor blocks - monitor the environment, including motion sensors, light sensors, buttons, contact switches, and so on
- Compute blocks - perform basic logic transformations (e.g. AND, OR, NOT) or basic state functions (e.g. prolong, toggle, trip, pulse).
- Output blocks - provide stimuli, and include light-emitting diodes (LEDs), beepers, electric relays, etc.
- Communication blocks - provide wireless point-to-point communication or replicate a signal to send to multiple blocks.

The initial eBlock platform contains both fixed function and programmable blocks. Fixed function blocks have a specific predefined function that may have slight configurability (i.e. which logical operation to perform, the length of time a signal is prolonged). A generic 2-input, 2-output programmable block is also supported, the programmable block can be configured to perform any user specified function. Using the eBlock simulator, the desired functionality of eBlock system is defined. The simulator interface partitions the desired functionality into a programmable block (or multiple blocks depending on the desired functionality) and downloads the new function to the programmable block using a PC interface. The programmable block is easily reprogrammed without requiring compilation or device programming by the user. While we have strived to keep the interface and tool chain accessible to novice users, we note that it is more likely that advanced users or users with some programming background would opt to utilize the programmable block.

The initial version of eBlocks operate on Boolean data, blocks send and operate on “yes” or “no” packets. Applications are built by snapping blocks together and the order in which blocks connected specifies the functionality of the system. Extensive usability testing has been performed on Boolean blocks interacting with close to 500 participants [5][6][7]. Findings indicate that over 50% of participants, with little or no training, are able to successfully create a variety of applications within 10 minutes of being introduced to the platform. Boolean values however limit the types of projects that can be built, in particular

STEM projects require building blocks that operation within the integer domain. Thus, extending the current platform to incorporate integer based eBlocks is essential.

#### 4. EBLOCK EDUCATION KIT

eBlocks have the potential to enhance STEM education by providing students with the opportunity to build customizable and interactive projects that complement the specific topic studied without the burden of first having extensive programming or electronics experience. Course work becomes more interesting, memorable, and other skills such as problem solving and teamwork are utilized and developed. We set out to define an eBlock Education Kit to target STEM education starting with middle school STEM education with future extensions reaching to the high school and university level.

To begin we needed to identify which building blocks (i.e. sensor, intermediate, and output blocks) to include in the eBlock Education Kit so that students would be able to build a wide variety of applications. Yet, at the same time we needed to balance the number of components included so not to overwhelm students with a gigantic catalog of eBlocks. A set of interactive projects for middle school students were identified by studying the topics included in the existing STEM curriculum listed on the local school district’s webpage [35] as well as several other websites that provide descriptions of middle school science fair projects [23][24][29]. Based on these resources, we determined the types of building blocks required in the initial set.

As we discover the project possibilities, we are also compiling a project guide booklet. The booklet is intended to get teachers and students started by illustrating sample systems that can be implemented with the eBlock Education Kit. Each project begins with an overarching question that serves to spark and guide the project. For example, why don’t oceans freeze as quickly as lakes? Next, steps are outlined to help students connect various eBlock components together to build a project that tests the posed question. Figure 4(a) illustrates how temperature sensor and integer display eBlocks can be snapped together to observe the freezing point between containers of fresh and salt water. Students are then asked questions that require interaction with the platform, such as placing both containers into a freezer and recording the temperature of each container every 10 minutes. Additionally a series of follow up questions are provided,

requiring students to reflect on the experiment. Figure 4(b) illustrates the setup for another experiment intended to monitor the effects of temperature and moisture on mold. While the application is different we can see that some of the blocks can be reused to create different projects. Furthermore, while the project guide booklet provides numerous projects it serves as a starting point for students and teachers to create their own projects.

As we identify and develop each building block within the platform, we must also ensure all of the resulting physical prototype have intuitive interfaces such that user can readily understand the functionality of that block without extensive training. Thus, as part of the iterative design process, we have also begun testing to determine the effectiveness of the newly proposed interfaces.

#### 5. USABILITY EXPERIMENTS

##### 5.1. Prototype-based Usability Testing

We conducted a series of informal experiments on our integer-based eBlock prototypes to see what aspects of the eBlock Education Kit were readily understood and what aspects were confusing. While we ultimately plan to perform usability testing with the target audience (middle school students), we started usability testing with small groups of college level students to provide an initial glimpse into the interface design and provide an opportunity for refinement before deploying the eBlock platform into middle schools. If an interface is not usable by university students, the interfaces developed would have no hope in a middle school setting. These experiments provide an initial evaluation of the various interfaces and a second testing phase is planned to target middle school students.

Participants were categorized into two groups, non-expert and expert. Non-expert participants are students at the university who have no engineering background, students in non-engineering majors such as pharmacy, accounting, or political science. Expert participants are students who have had at least one semester in an engineering major such as computer engineering, chemical engineering, or hydrology.

All participants were provided with a handout containing a brief introduction to eBlocks, as shown in Figure 5. The flyer quickly illustrates how blocks are powered on, connect to one another, and communicate with one another. Additionally, participants

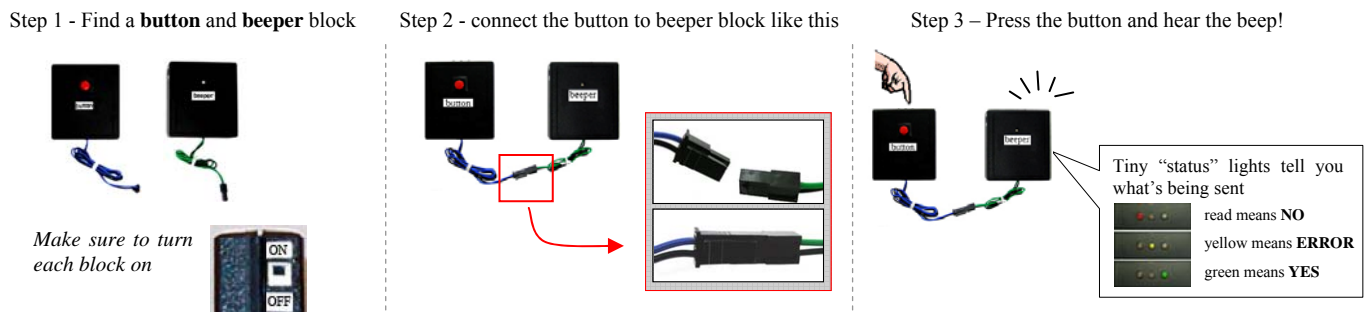
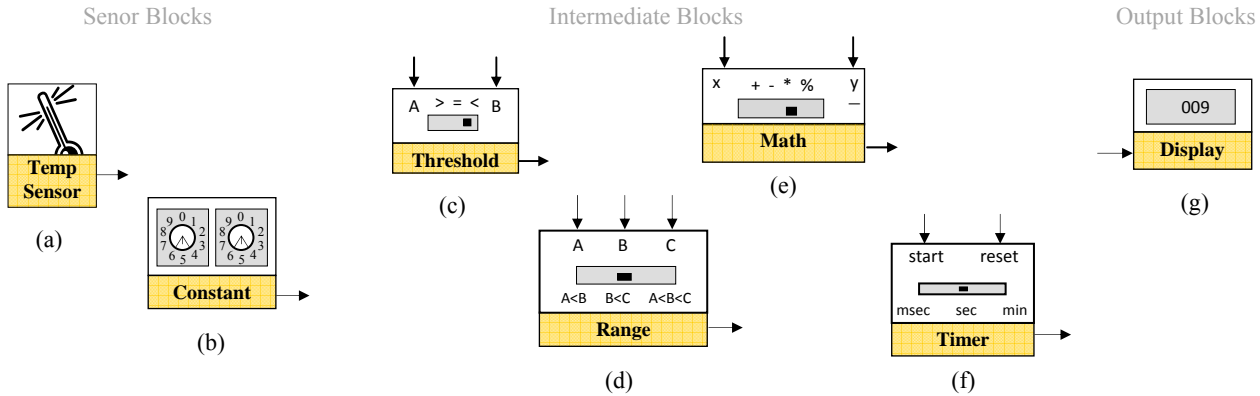


Figure 5: A one page starter guide is provided to participants to quickly illustrate how eBlocks are powered on, connected to one another, and how to determine what the blocks are communicating to one another.







**Figure 8: Newly integrated integer-based eBlocks include (a) a temperature sensor, (b) a constant block, (c) a threshold block, (d) a multi-function range block, (e) a math block, (f) a timer block, and (g) an integer display block.**

switch image, toggling individual switches between “yes” and “no”. eBlocks that sense or interact with their environment, such as light or motion sensors, include accompanying visual representation of their environmental stimuli/interaction to simulate the corresponding environment. For example, each light sensor is accompanied by a light bulb icon that a user clicks to toggle the light on and off. The light bulb indicates the sensor’s external environment (e.g., whether the sensor detects the presence or absence of light). The electric relay is connected to a lamp icon that is on or off depending on the block’s input. In the bottom left corner of the simulator, a gray text box provides context-sensitive help. If a user places the mouse cursor over blocks in the simulator, the block’s description and interface automatically appears in the text box. While the eBlock simulator is behaviorally correct, the simulator does not capture all low-level timing details.

Utilizing the eBlock simulator to gauge usability is expected to yield slightly higher success rates than usability experiments conducted with the corresponding physical prototypes. In the initial testing phases we found that users with little or no engineering background did not know how to use components such as DIP switches or slide switches. Additionally, users frequently forget to power on each block, had trouble connecting blocks and attempted to connect two outputs or two inputs together, or crossed the ground and data wires. While these issues are not fundamentally related to the blocks’ functionality, they nevertheless reduced the resulting success rates. Most of the initial experiments provided no training for users, thus over time we believe the usability obtained from physical testing would increase to match those obtained in the simulator testing environment.

### 5.2.1 Additional Integer eBlocks

In addition to the timer block shown in Figure 6(b), we also expanded the eBlock library to include the six additional integer based eBlocks shown in Figure 8. Two new sensor blocks are introduced, specifically a temperature sensor that detects the ambient temperature and a constant block that a user can configure using rotary switches to output a constant integer value have been included. Four new intermediate blocks have similarly been incorporated into the eBlock platform, with each block

including a slide switch to enable a user to configure the block to perform various operations. The threshold block accepts two inputs A and B, along with a slide switch to configure the block to detect if  $A < B$ ,  $A = B$ , or  $A > B$ . The range block is a multi-function block. If all three inputs are connected, the range block performs the comparison  $x \leq A \leq y$ . If the x and A inputs are connected, the range block performs the comparison  $x \leq A$ . Lastly, if the A and y inputs are connected, the range block performs the comparison  $A \leq y$ . The math block similarly accepts two inputs, A and B, and can be configured to add, subtract, multiply, or divide these two values. The timer block contains a start input that activates an internal counter to increment every millisecond, second, or minute depending on the user configuration. If the start input is zero, the timer block maintains its internal value until the reset input is pulsed. Lastly, the display block receives an integer value and displays the corresponding value on an LCD screen incorporated into the block.

### 5.2.2 Integer Block Testing

To determine the usability of the various integer-based blocks, we performed experiments with 11 participants who were incoming engineering freshmen and 19 undergraduate students who were in non-engineering majors. Participants utilized the eBlock simulator to build various integer-based systems. Students were asked a total of 4 questions, one question from each of four problem classifications listed in Figure 9. Questions classified as “Simple Math” requires construction of a math formulation such as  $5 + 4$ , requiring a single math block. Questions classified “Compare” requires construction of a comparison such as  $4 > 2$ , the comparison can be implemented utilizing either the compare block or by connecting two out of the three inputs of the range block. Questions classified as “Cascaded Math” is a slightly more difficult math formulation requiring two operators such as  $2 * 4 - 5$ , where the output of one math block is used as the input to a second math block. Questions classified as “Range” requires construction of a range formulation such as  $3 < 6 < 8$ . The range operation can be implemented by either using all three inputs of the range block or by utilizing two compare blocks and a 2-input combine block. One comparison block evaluates  $3 < 6$ , the second comparison block evaluates  $6 < 8$ , then both outputs are fed to the 2-input combine block to see if both comparisons are true.

<p><b>Q1. Simple Math (12-3)</b></p> <p>Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates “12 – 3” and displays the result on a Display eBlock.</p>
<p><b>Q2. Simple Math (4*3)</b></p> <p>Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates “4 * 3” and displays the result on a Display eBlock.</p>
<p><b>Q3. Compare (6&gt;=3)</b></p> <p>Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 6 is greater than or equal to 3.</p>
<p><b>Q4. Compare (6&lt;=15)</b></p> <p>Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 6 is less than or equal to 15.</p>
<p><b>Q5. Cascaded Math (5+2*4)</b></p> <p>Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates “5 + 2 * 4” and displays the result on a Display eBlock.</p>
<p><b>Q6. Cascaded Math (7-(6/3))</b></p> <p>Using Constant eBlocks, a Display eBlock, and any other eBlocks you find appropriate, create a system that calculates “7 – (6 / 3)” and displays the result on a Display eBlock.</p>
<p><b>Q7. Range (2&lt;5&lt;9)</b></p> <p>Using Constant eBlocks, a Green/Red Light eBlock, and any other eBlocks you find appropriate, create a system that lights up a green light if the number 5 is greater than 2 and less than 9.</p>

**Figure 9: Questions posed to participants to determine usability of newly added integer-based eBlocks.**

Participants were allotted 25 to 30 minutes to complete all 4 questions.

The majority of participants were able to utilize the math block. As shown in Table 3, 82% of users are able to successfully utilize a single math eBlock, while 72% of users able to cascade two math blocks. Within the current integer based block definitions there are two methods to construct a comparison, the comparison block or the range block. Because the comparison and range block have overlapping functionality, we wanted to determine if a single block could be provided to perform both operations. Participants asked to utilize the comparison block to perform a compare operation yielded an average success rate of 81%, as shown in the “Compare using Compare eBlock” entry. Participants asked to utilize the range block to perform a compare operation experienced a significantly lower success rate of 54%, as denoted in the “Compare using Range eBlock” row. Similarly, the range operation can be implemented utilizing either a range or comparison block. Participants using the compare block to perform a range operation yielded a success rate of 13%, as shown in the “Range using Compare eBlock” entry. Participants provided with a range block yielding an average success rate of 53%, as listed in the “Range using Range eBlock” row.

In both the compare and range experiments the users achieve higher success rates utilizing the block most closely matching the desired functionality. Soloway, et al. performed studies indicating that programming constructs need to “cognitively fit” with the user to be effective [33], thus it is not surprising that we similarly

**Table 3: Success rates for non-expert participants in simulator-based usability experiments given various integer-based interfaces.**

	Percent Success	Number of Participants
Simple Math	82 %	49
Cascaded Math	72 %	60
Compare using Compare eBlock	81 %	26
Compare using Range eBlock	54 %	24
Range using Compare eBlock	13 %	15
Range using Range eBlock	53 %	15

find users achieve higher success rate when a block that closely fits the desired functionality is provided. As part of future work, we plan to conduct additional experiments in which users are provided with both the comparison and range blocks, to determine if higher success rates are actually achieved, or if too many blocks lead to confusion.

## 6. ROLE OF EBLOCKS IN EDUCATION

The flexibility of the eBlock platform enables use across a wide range of age groups and disciplines. A variety of modules can be developed to highlight various aspects of the platform ranging from how to build a custom sensor based system to solve a particular problem, how electronic system communicate with one another, or even expose the underlying physical implementation. In the following sections we discuss the possible scope of the eBlock platform for different age levels, and introduce our initial module development efforts and experiences.

### 6.1. Middle School Audiences

At the middle school level the eBlock platform can complement the existing curriculum by enabling students to build complementary monitor/control systems and test out the ideas studied in class. In addition to improving STEM performance, we are designing the modules to support students in thinking about how technology is integrated into their everyday surroundings and inquiring into how it all works. Because the eBlock platform is intended to complement the existing curriculum, rather than redefine the existing curriculum, students are guided through projects that focus on how to utilize the building blocks rather than understanding details of the underlying implementation.

Before modules can be developed to target a particular subject, teachers and students must first become acquainted with the fundamentals of the eBlock platform. Thus, the initial module development provides a general introduction. Specifically, a six part lesson plan for teachers is included with all of the relevant background materials needed to introduce the eBlock platform, guide discussions with students, as well as troubleshoot problems commonly encountered by new users. Additionally, student activity sheets are included that encourage students to discover how the eBlock platform works for themselves through a series of hands on activities. Completion of this module is intended to provide students with requisite knowledge in how to use the platform to create customized sensor based system, practice problem solving and teamwork, and think about how electronic

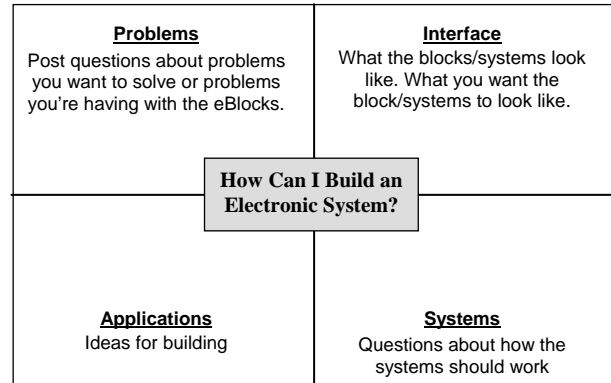
**Table 4: Pre-survey responses from middle school participants defining what an engineer does.**

	Number of Responses
Fix/make Car Engines	18
Build Buildings	11
Build/Design “Things”	8
Miscellaneous	5
Fix “Things”	4
Work on Computer	3
Fix Planes	2
Do Paperwork	1
Make Weapons	1
Design Cell Phones	1
Fix Lights	1

Total Number of Participants 55

systems impact their everyday world. Moreover, the introductory materials provide an ideal opportunity to introduce students to the field of engineering and the everyday tasks of engineers. Depending on intended end application, additional modules can be utilized to further guide students and teachers to target a variety of STEM topics.

The introductory module is comprised of six lessons. In the first lesson students are assumed to have no background in engineering. In fact, in a pre-survey filled out by middle school students we found that most students struggled to come up with a concrete example as to what an engineer is or what engineers do on a daily basis. As shown in Table 4 most students responded that engineers fix engines or construct buildings. A small number of students stated very generally that an engineer fixes or designs “things” or does paperwork. During the initial lesson, students are asked to define what a system is, and draw from their own prior experiences and knowledge to come up with examples such as the circulatory system, weather system, or postal system. As a group they begin to investigate what makes it a system, what are the parts of the system, and what is the relationship between subcomponents within the system. Teachers or group leaders guide students to think about electronic systems in their everyday lives, with examples including cell phones, play stations, or televisions. Again, students begin to make connections that these electronic products are also systems, made up of smaller parts that contribute to a larger electronic system. At the middle school level it is not feasible to go into low-level details, so analysis of a cell phone system is broken down into subsystems such as the screen, antenna, communication, video, storage of numbers, calendar/alarm clock. Additionally, students begin to think about the challenges faced by engineers when building these systems such as how to connect the parts together, how to fix/detect problems in the design, and what the final product should it look like. Throughout this introductory discussion, engineering terms such as testing, prototyping, usability, debugging, and hierarchy are introduced along with a high-level view of the engineering design cycle. The first module strives to show students how



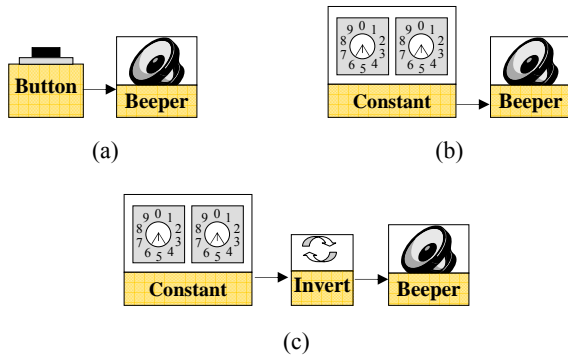
**Figure 10: Driving Question Board.**

electronic systems are relevant to their everyday lives, and serves to create a meaningful context for making sense of the upcoming lessons. We had the opportunity to work with two local middle schools who were asked to describe subsystems for a cell phone or television. These students were able to breakdown these electronic systems into screens, buttons, power source, and so on. Furthermore, when asked to think about the design cycle, though not in technical terms, students were able to describe phases of planning, design, test, and marketing.

One of the main tools used throughout the lessons is a Driving Question Board (DQB) [15][32][38] that is utilized to organize ideas and address problems experienced while working through the lessons. The DQB physically is a bulletin board or white board which houses central questions of the module, as shown in Figure 10. In the introductory module, students address “How can I build electronic systems?” Additionally, sub-questions and/or central ideas are generated branching off the main question. Throughout each lesson, student-generated questions are posted on the DQB. As students investigate the eBlock platform and discover answers themselves, they can also post proposed solutions or responses to the questions on the board. The DQB continually reminds students of the main goal they are working towards, the questions they initially and the evolution of their understanding, as well as what’s left to be figured out.

In addition to group discussions, each lesson provides students with time to interact with the platform and develop their own understanding through hands on investigations. In our experience, students were most excited when they were given a chance to try out the eBlock platform to build their own systems. In the first lesson, for example, students are provided with a quick demonstration of the basic functions, on/off switches, connection blocks together, and status lights located on each block indicating what that block sending to other blocks within their system. Student then work in groups to build their own electronic systems. As shown in Figure 11 students can connect a button and beeper together to make a door bell. A variety of other building blocks are similarly provided, and students are encouraged to connect blocks together and observe what happens. Questions are provided on the student activity sheets to direct students to see if they can classify blocks into groups based on what they do (i.e. sense an event, provide stimuli) or their interfaces (one input one





**Figure 11: Sample eBlock systems constructed by students to investigate the underlying communication protocol of eBlock platform given both Boolean and integer based blocks.**

output, two input one output), as well as how they can utilize these blocks to build larger systems.

In the second lesson, students focus on how blocks communicate with one another. Once a block is powered on, each block begins to send a signal. The signal a block sends is correlated with the block type. Boolean blocks, such as button and motion sensor blocks, send “YES” and “NO” packets. For example, when the button is pressed the block send a “YES” packet and when the button is released the block sends a “NO” packet. Integer blocks, such as temperature sensors or constant blocks, send integer values. In the case of a temperature sensor, a value of 75, 80, or even 90 may be sent depending the ambient room temperature detected. The question arises when both Boolean and Integer blocks are utilized within an electronic system. How does a Boolean block interpret an integer value, or how does an integer block interpret a “YES” or “NO” value. In the case of Boolean blocks, an integer value equal to zero is interpreted as a “NO”, while any integer value greater than zero is interpreted as “YES”. In integer blocks, a “NO” packet is interpreted as a value of 0 and a “YES” packet is interpreted as a value of 1. The student activity sheet guides students through several examples that have students build and investigate Boolean systems, Integer systems, and heterogeneous systems composed of both Boolean and blocks. Through individual and group exploration, students are encouraged to think about the types of connections made (Boolean verse integer) and the resulting system behavior, as well as how to classify blocks into these groups. Additionally, small status lights (green, yellow, and red lights) are integrated into each block indicating that block’s current status. By understanding the underlying communication of blocks, and being able to trace the status of each block within a particular system, students begin to learn how to debug systems. Specifically, students can trace back the behavior of each block to determine if the predicted behavior matches what is actually occurring during execution.

Once students grasp the platform basics, they move to more advanced blocks in lessons 3 and 4. In these lessons, the timer, range, and threshold blocks are introduced to enable composition of increasingly complex systems. Students are challenged to build systems that detect how long a button is pressed, as shown in Figure 12(a), alert a user when the room temperature is greater

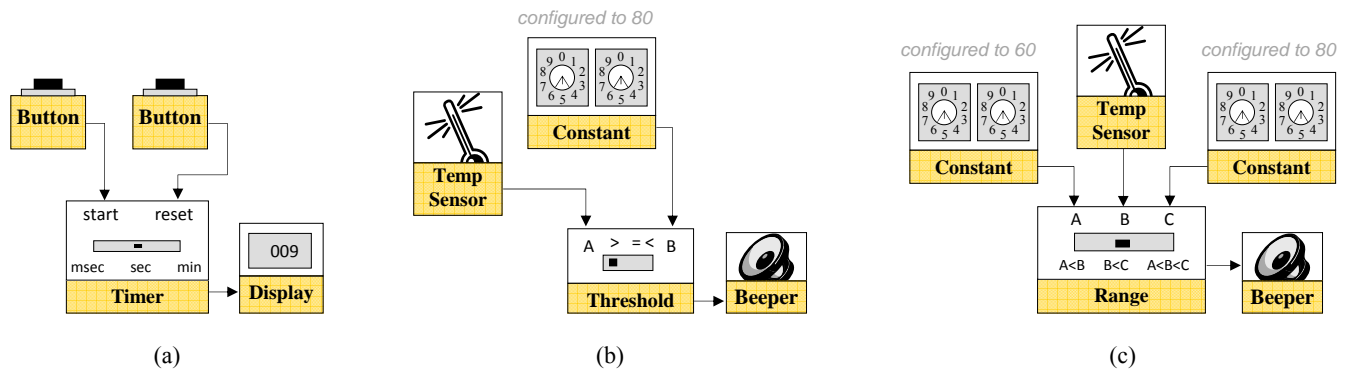
than 80 degrees, as shown in Figure 12(b), or sound an alarm when the room temperature is not between 60 and 80 degrees, as shown in Figure 12(c). Again, students are encouraged to post questions and solutions on the DQB. Additionally, the DQB enables students to see how groups approach these challenges differently and create an assortment of solutions, and help each other discuss and come up with solutions to the posted problems. Additionally, at the end of the fourth lesson students readdress the initial question of they can build their own electronic systems. With their exposure to the eBlock platform, they are again asked to think about how electronic systems are present in their everyday lives, as well as the role these system plays. In the last two modules students will define a problem and design an electronic system to solve that problem. They approach this task as a company selling a product and revisit the engineering design cycle. Also, group members select the roles they plan to play within their companies (i.e. designers, marketing, team leader) and begin the initial planning.

In the final two lessons, students are given an open-ended project that challenges them to relate the eBlock platform to a real life problem. Working within a team, students are tasked with defining a problem that can be solved by an electronic system, creating a solution to that problem using the eBlock platform, and creating a presentation or sales pitch that introduces their system to an audience. Specifically, the presentation is defined to include a poster that discusses the problem they are trying to solve as well as a schematic/drawing of their system, a demonstration, and a question and answer session from invited judges or their peers. These presentations provide an excellent opportunity to get parents, industry, and graduate students involved as guest judges. Furthermore, depending on the judges participating, students have the opportunity to meet working professionals and university students within engineering.

Completion of the introductory module provides students with an understanding of the underlying eBlock platform, as well as an understanding of the individual building blocks within the platform. Students are encouraged to build a variety of systems and develop their own understanding of platform components, as well as practice how to debug a system that does not adhere to the desired specification. Additionally, this module provides students with some insight into engineering and the engineer design process. We note however that students do not need to complete the entire module to effectively utilize the eBlock platform. After the fourth lesson students have sufficient understanding to build their own projects or progress to other modules that target specific math or science courses.

## 6.2. High School Audiences

At the high school level we similarly envision eBlocks complementing the existing curriculum, which would limit discussion of the underlying platform. However, at the high school level courses such as the Infinity Project [10] are intended to expose students to engineering. Depending on the type of courses available, expertise of the instructors, and the availability of resources, the eBlock platform can be similarly be utilized as an introduction to embedded systems. At a higher abstraction, students can practice problem solving by building customized monitor/control systems for various applications, utilize a variety of sensors and actuators, and learn about data collection and analysis. At the lowest level students can utilize eBlocks as a low-



**Figure 12: Sample eBlock systems constructed by students to investigate the usage of intermediate blocks, including systems that (a) detect how long a button is pressed using a timer block, (b) sounds an alarm when the temperature exceeds 80 degrees using a threshold block, and (c) sounds an alarm when the temperature is below 60 or above 80 degrees using a range block.**

cost platform to start investigating communication, interfacing, and usability issues.

### 6.3. University Audiences

At the university level students in engineering disciplines can utilize eBlocks in a variety of programming and hardware courses. Many engineering students spend weeks or months learning theory before ever being able to build a working system. The eBlock platform provides a quick and easy starting point for engineering students to build a variety of computing systems. Then as students learn about various topics, such as interfacing or interrupt service routines, the platform can be dissected to demonstrate the topic studied within a larger working system. By utilizing a top-down methodology students have a better perspective of how each of the topics studied relate to one another.

Similarly, the logic and state blocks can be utilized by students in an introductory programming course as a physical platform or within the simulation environment to setup and test Boolean equations and state-based relationships through a trial and error methodology. Students can relate the functionality of these Boolean and state operators to everyday situations, such as detecting light or button presses to control a doorbell, before applying these operators to programming constructs. Lastly, the eBlock platform has the potential to enhance other university STEM programs. Similar to the middle school application, these students can utilize the eBlock platform to develop complementary monitor/control systems, and illustrate the role of technology within a particular discipline.

## 7. CONCLUSIONS AND FUTURE WORK

To stimulate interest and improve performance in STEM education we have proposed the use of the eBlock platform to enable the construction of interactive projects that complement the existing curriculum at the middle school and high school levels. Initial experiments show that roughly half of the novice participants can readily utilize these building blocks with little or no training. While the initial results are promising, many questions remain unanswered. As future work we must consider a more diverse group of participants, including students at the middle and high school level. We are currently working with

local middle schools and community based organizations to begin testing the usability and feasibility of the eBlocks platform. Furthermore, the eBlock systems constructed by participants contained only a handful of blocks. We are also investigating the usability of the eBlock platform given increasingly complex applications, as well as the ability for users to define and design their own customized system without the aid of a project booklet.

We have additionally begun to develop modules that complement the hardware and introduce students and teachers to the eBlock platform. Students are guided through a series of hands-on projects intended to encourage students to develop their own understanding of how building blocks communicate with one another, the function of individual blocks, as well as how to create custom solutions to a variety of problem scenarios. While the initial module development concentrated on providing students and teachers with platform basics, future modules will target specific STEM topics aligned with important learning goals for both afterschool and in school curriculums.

## 8. REFERENCES

- [1] Arizona Department of Education. Arizona's Instrument to Measure Standards (AIMS) Results, State Report Card, <http://www.ade.state.az.us/srcs/statereportcards/StateReportCard06-07.pdf>, 2007.
- [2] Barron, B., D. Schwartz, N. Vye, A. Moore, A. Petrosino, L. Zech, J. Bransford. Doing with Understanding: Lessons from Research on Problem- and Project-Based Learning. *The Journal of the Learning Sciences*, 7, pp. 271-311, 1998.
- [3] Blumenfeld, P. C., T. Kempler, J. S. Krajcik. Motivation and cognitive engagement in learning environments. In R. K. Sawyer (Ed.), *Cambridge handbook of the learning sciences*, pp. 475-488, 2006.
- [4] Committee Prospering in the Global Economy of the 21st Century: An Agenda for American Science and Technology; Committee on Science, Engineering, and Public Policy. *Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future*. Washington DC: National Academy Press, 2007. <http://www.nap.edu/catalog/11463.html>
- [5] Cotterell, S., K. Downey, F. Vahid. Applications and Experiments with eBlocks - Electronic Blocks for Basic Sensor-Based Systems. *Sensor and Ad Hoc Communications and Networks*, October 2004.

- [6] Cotterell, S., F. Vahid. A Logic Block Enabling Logic Configuration by Non-Experts in Sensor Networks. CHI, April 2005.
- [7] Cotterell, S., F. Vahid. Usability of State Based Boolean eBlocks. HCI, July 2005.
- [8] eBlocks Science Kits. [http://www.ece.arizona.edu/~uecs/eb\\_kits/index.html](http://www.ece.arizona.edu/~uecs/eb_kits/index.html), 2008.
- [9] Geier, R., P., Blumenfeld, R. Marx, J. Krajcik, B. Fishman, E. Soloway. Standardized test outcomes of urban students participating in standards and project based science curricula. Proceedings of the sixth international conference of the learning sciences, 310-317, 2004.
- [10] Infinity Project, The. <http://www.infinity-project.org/>, 2006.
- [11] Institute of Education Sciences, U.S. Department of Education. International Outcomes of Learning in Mathematics Literacy and Problem Solving: PISA 2003 Results from the US Perspective. <http://nces.ed.gov/surveys/PISA/PISA2003highlights.asp>, 2007.
- [12] Institute of Education Sciences, U.S. Department of Education. Trends in International Mathematics and Science Study (TIMSS) 2003, <http://nces.ed.gov/timss/Results03.asp>, 2007.
- [13] Jackson, S. The Quiet Crisis: Falling Short in Producing American Science and Technical Talent. Building Engineering and Science Talent (BEST) Report, 2004.
- [14] Kovac, J. Student active learning methods in general chemistry. Journal of Chemical Education, Vol. 76, No. 1, pp. 120-124, 1999.
- [15] Krajcik, J., C. M. Czerniak. Teaching science in elementary and middle school: A project-based approach (3rd ed.). Mahwah, NJ: Lawrence Erlbaum, 2007.
- [16] Logiblocs, <http://www.logiblocs.com/>, 2007.
- [17] Lunsford, B., Herzog, M. Active learning in anatomy and physiology: Student reactions and outcomes in a nontraditional AP course. The American Biology Teacher, Vol. 59, No. 2, pp. 80-84, 1997.
- [18] Martin, F., et. al. The MIT Programmable Brick. <http://lcs.www.media.mit.edu/groups/el/projects/programmable-brick/>, 2006.
- [19] Martin, F., et. al. Crickets: Tiny Computers for Big Ideas. <http://lcs.www.media.mit.edu/people/fredm/projects/cricket/>, 2006.
- [20] Microchip Technology Inc. PIC16F631/677/685/687/689/690 Data Sheet. <http://ww1.microchip.com/downloads/en/DeviceDoc/41262E.pdf>, 2008.
- [21] Mistler-Jackson, M., N. B. Songer. Student motivation and internet technology: Are students empowered to learn science? Journal of Research in Science Teaching, 37, pp. 459-479, 2000.
- [22] National Science Board. Science and Engineering Indicators 2006, <http://www.nsf.gov/statistics/seind06/>, 2006.
- [23] Newkirk, K. Selah Intermediate School Science Projects, Science Project 2006 Index, <http://www.selah.k12.wa.us/SOAR/SciProj2006/index.htm>, 2006.
- [24] Phillipsburg Christian Academy Science Fair Projects, <http://www.fellowshipch.org/pcsciencefair.html>, 2007.
- [25] Price, S., Y. Rogers. Let's get physical: the learning benefits of interacting in digitally augmented physical spaces. Computers & Education, 43(1-2), August 2004.
- [26] Resnick, M. Behavior Construction Kits. Communications of the ACM 36, No. 7, pg. 64-71, 1993.
- [27] Resnick, M., S. Ocko, S. Papert, LEGO, Logo, and Design, Children's Environments Quarterly 5(4), pg. 14-18, 1988.
- [28] Richards, L., et. al. Promoting active learning with cases and instructional modules. Journal of Engineering Education, Vol. 84, No. 4, pp. 375-381, 1995.
- [29] Rubin, J. Science Fair Projects Resources, <http://www.julianrubin.com/schooldirectory/fairresources.html>, 2007.
- [30] Schneider, R. M., J. Krajcik, R. W. Marx, E. Soloway. Performance of students in project-based science classrooms on a national measure of science achievement. Journal of Research in Science Teaching, 39(5), pp. 410-422, 2002.
- [31] Science, Technology, Engineering, and Mathematics Education (STEMEd) Caucus Steering Committee. K-12 STEM ED Report Card: How Arizona Ranks. <http://www.stemedcaucus.org>, 2006.
- [32] Singer, J., R. Marx, J. Krajcik, J. Chambers. Constructing extended inquiry projects: Curriculum materials for science education reform. Educational Psychologist, 35(3), 165-178, 2000.
- [33] Soloway, E., Bonar, J., Ehrlich, K. Cognitive Strategies and Looping Constructs: An Empirical Study. Communications of the ACM, vol. 26, issue 11, pp. 853-860, 1983.
- [34] Teitelbaum, M. The U.S. Science and Engineering Workforce: An Unconventional Portrait. National Academies' Government University Industry Research Roundtable (GUIRR) Summit, 2002.
- [35] TUSD Curriculum Resource Guide, <http://instech.tusd.k12.az.us/standards/index.asp>, 2006.
- [36] Wallace, K. America's Brain Drain Crisis. Reader's Digest, December 2005.
- [37] Wallich, P. Mindstorms Not Just a Kid's Toy. IEEE Spectrum, Vol. 38, No. 9, September 2001.
- [38] Weizman, A., Y. Schwartz, D. Fortus. The Driving Question Board. The Science Teach, November 2008.
- [39] Wyeth, P., H. Purchase. Tangible Programming Elements for Young Children. Extended Abstract, Conference on Human Factors in Computing Systems, 2002.