

# **AC 2008-1550: THE VIRTUAL CLASSROOM ENVIRONMENT OF A WWW-BASED AUTONOMOUS ROBOTICS LABORATORY: FACTORS AFFECTING STUDENT PARTICIPATION, COMMUNICATION, AND PERFORMANCE**

## **Richard Drushel, Case Western Reserve University**

Dr. Richard F. Drushel is Instructor and Executive Officer in the Department of Biology, Case Western Reserve University, Cleveland, Ohio. He co-invented and co-taught for 19 semesters a highly-successful LEGO- and microcontroller-based autonomous robotics course for undergraduates, as well as several summer courses for educators and secondary-school students. He has also taught lecture/lab courses in human anatomy and comparative anatomy, and lectured in physiology, chemistry, and biochemistry. His research interests include 3-D kinematic modelling of soft-tissue structures in the feeding of marine molluscs, and the use of computers and robotics in education.

## **John Gallagher, Wright State University**

Dr. John C. Gallagher is an Associate Professor with dual appointments in both the Department of Computer Science and Engineering and the Department of Electrical Engineering at Wright State University, Dayton, Ohio. His research interests include analog neuromorphic computation, evolutionary algorithms, and engineering education.

# **The Virtual Classroom Environment of a WWW-Based Autonomous Robotics Laboratory: Factors Affecting Student Participation, Communication, and Performance**

## **Abstract**

Virtual classrooms made possible by WWW-based course delivery have the potential to improve student access to scarce resources (faculty expertise, unique laboratory environments) as well as to unite students from geographically-separated backgrounds in shared educational activities. The long reach of WWW-based courses, however, is through a narrow window whose ultimate size may face critical limits (available client or host computing power, network bandwidth, display screen size) that might impair student learning. In order to translate an existing traditional course into a distance-learning version, not only must the key features of “success” (curriculum, course design, classroom environment) for the traditional format be identified, but these features must actually be deliverable intact via the computing and network infrastructure available. Laboratory and group-based practicum courses are particularly sensitive to the communication environment. We report here some of the difficulties we have encountered over several iterations of a WWW-based autonomous robotics laboratory course for engineering undergraduates, as well as during the gradual migration of a two-week summer robotics practicum for secondary-school students and STEM educators from a traditional in-person laboratory to a virtual distance-learning classroom. Both courses have highly successful traditional versions (20 and 5 offerings, respectively) which serve as reference controls. While some identified problems may eventually be overcome by improved delivery technology, there remain fundamental differences between our physical and virtual classrooms that, based on observations of student performance and feedback, may require significant how-to-use-the-interface training and familiarity before attempting to deliver any formal curriculum content. For some students, the effort required in learning to use the virtual classroom and transcend its limitations significantly reduces their participation and performance.

## **1. Introduction**

Various institutions are moving to put entire technical curricula online. Project and practica-based courses are integral parts of such curricula and are identified as such both by common assessments of good practice and formal accreditation requirements. Synchronous, lecture-style courses can be translated into online environments with little difficulty. Translating practica courses, in which instruction tends to be more individualized and rich in interpersonal contact, can be somewhat problematic. In this paper, we will report on some of the difficulties we have encountered over several iterations of a WWW-based autonomous robotics laboratory course for engineering undergraduates, as well as during the gradual migration of a two-week summer robotics practicum for secondary-school students and STEM educators from a traditional in-person laboratory to a virtual distance-learning classroom. We will begin with descriptions of the courses and their mechanics. Following, we will provide both quantitative and qualitative comparative data, indications of potential problems, and preliminary suggestions on how to fix them.

## 2. Mechanics of Traditional Robotics Courses

There are three outstanding features of the traditional autonomous robotics courses discussed in this paper: (1) group-based work in a shared physical classroom environment; (2) equal emphasis upon mechanical design and software control of robots; and (3) a wide range of acceptable lab notebook documentation media and styles.

First, in any traditional laboratory, all students and instructors are working in the same physical room. Even if students are focussed upon their own tasks (*e.g.*, a small group working at its own workstation on its own robot), they are not insulated from the background conversation and activities of other students, or of the instructor. A chance observation of what another group is doing, or what problems they are having, or overhearing a discussion between the instructor and other students, can be an important influence — a source of new information or ways of thinking about a problem. Since the curriculum is group-based, inter-student cooperation is essential, and discussions arise naturally. Normal conversation includes both verbal cues (tone of voice) and non-verbal cues (facial expression, body language). Moreover, the speed of transmission of information can be very rapid, especially when communicated by the instructor. For instance, a student may identify a bug in a software library, demonstrate it to the instructor, who then immediately can make a public announcement from the center of the room; another student may then find a work-around, which also can be communicated to the instructor, who then is able to update the library code from his own workstation and distribute it electronically to all the students. The entire situation may be resolved within a single class period. Both formally and informally, every student is exposed to the success or failure of every other student in the common laboratory. This is a powerful learning asset: it increases the net experience of every student, because students are not limited to personal experiences with their own tasks, but can also absorb the experiences of others around them.

Second, the curricula of both traditional robotics courses included equal amounts of hands-on robot mechanical design (using LEGOs as primary structural elements) and principles of software control (including elementary programming constructs and sensory feedback). The final contest robots in each course could be successful only if they had efficient geartrains, robust frames, and effective behavioral software. The balance between modularity and integration was key. Multi-student groups allowed for individual specialization on subtasks as suited to individual preferences and abilities: the best groups had a mechanical-oriented member, a software-oriented member, and an uncommitted generalist, each acting as a sanity check for the other. This also tended to foster wider exploration of problem solutions, as each student approached problems from a unique background. Design logics also needed to be articulated more explicitly, and sometimes more convincingly, in order to be understandable to a wide audience.

Third, the traditional robotics courses lent themselves well to handwritten documentation of student, especially for mechanical design sketches and other diagrams (such as software flowcharts or state diagrams), and for quick exploration of ideas. Moreover, a physical paper notebook was flexible enough to incorporate printouts of code (already typed for the compiler anyways) and digital photographs (after the lab obtained a digital camera) by taping or stapling them in, or else keeping the notebook as a 3-ring binder, to allow easy insertion of pages. Some

students made rough handwritten notes in class and then typed them into a word processor on their own computers, especially those who had poor handwriting but were efficient typists. Overall, the immediacy afforded by just writing down observations in real time during class allowed high-quality documentation to be produced by the greatest number of students. During the long history of the courses, as electronic authoring tools improved, more notebooks moved toward electronic formats (*e.g.*, the entire notebook as a word processing document, complete with observations, photographs, and code excerpts). Since the instructors were local to the class, grading of physical notebooks was not a serious issue, even if it was a large pile — students just left the notebooks with the instructor at the end of class.

## 2.1 Autonomous Robotics (LEGO 375)

Over 20 semesters and 10 years, 538 students participated in Autonomous Robotics (colloquially known as *LEGO 375*), one of the most popular and successful laboratory courses at Case Western Reserve University (CWRU)<sup>2,5,4</sup>. Table 1 summarizes the detailed history and evolution of the course, which was directly inspired by the early 1990s versions of the MIT 6.270 Autonomous Robot Design course<sup>16</sup>. Students (working in groups of 3) constructed robots from LEGO beams, plates, gears, and motors, and controlled them with an 8-bit microprocessor system (the MIT 6.270 board<sup>13</sup>, containing a Motorola 68HC11 CPU), which was programmed in Interactive C (IC). Various light and contact sensors provided for sensory feedback. Such robots were model systems for the study of animal behavior, as well as for general principles of robotics and embedded control systems. Because they had self-contained sensory and motor systems and control software, and no human intervention or control (*e.g.*, via joysticks), these robots were fully autonomous. During the first half of the course, students were taught practical skills in C programming, mechanical design with LEGOs, principles of sensors, software design, and animal behavior strategies. The second half of the course consisted of a major design project: an autonomous robot capable of participating in a public capstone contest: the Egg Hunt. In a walled arena filled with colored plastic eggs, teams of two robots each were required to collect eggs, sort them by color, and bring eggs of desired color to a home nest (distinguished from the opponent’s nest by polarized light beacons of opposite polarity). Since some egg colors were worth negative points, robots could have different behavioral strategies, such as ignoring “bad” eggs, or attempting to put them into the opponent’s nest. Course grades, however, were assigned on the basis of class participation and written design notebooks, which the students were required to keep throughout the semester. All in all, *LEGO 375* taught competence in many disciplines, real-world problem solving, group-based project management, and critical thinking. The exercises and Egg Hunt contest were open-ended enough that, even though certain strategies or design ideas recurred throughout the course’s decade of existence, every group’s implementations were nonetheless unique.

Table 1. Overview of Autonomous Robotics (*LEGO 375*) at CWRU.

Class	Semester	Participants	Course Structure	Egg Hunt Contest Structure	Contest Site	Course Grading	Design Notebook Format	Extant Notebooks
1	Spr. 1995	19	A	D	campus	K	hand	3
2	Spr. 1996	21	A	E	campus	K	hand	3
3	Fall 1996	22	B	F	campus	L	hand	9
4	Spr. 1997	21/24	C	G	campus	M	hand	12/13

5	Fall 1997	26/29	C	H	campus	M	hand	0/3
6	Spr. 1998	27/30	C	H	campus	N	hand	2/3
7	Fall 1998	27/30	C	H	campus	N	hand	3/3
8	Spr. 1999	27/30	C	H	campus	N	hand	0/0
9	Fall 1999	24/30	C	H	campus	N	hand	5/6
10	Spr. 2000	27/30	C	H	campus	N	P	5/6
11	Fall 2000	27/30	C	H	GLSC	N	P	9/9
12	Spr. 2001	28/29	C	H	GLSC	N	P	4/4
13	Fall 2001	26/30	C	H	GLSC	N	P	7/8
14	Spr. 2002	25/27	C	H	GLSC	N	Q	1/1
15	Fall 2002	27/30	C	H	GLSC	N	Q	15/15
16	Spr. 2003	27/28	C	H	GLSC	N	R	12/12
17	Fall 2003	27/30	C	H	GLSC	N	R	13/13
18	Spr. 2004	24/29	C	H	GLSC	O	R	7/9
19	Fall 2004	21/24	C	I	GLSC	O	R	19/22
20	Fall 2005	9/16	C	J	class lab	O	R	16/16

Note: *LEGO 375* is the umbrella term (a student nickname) for all the different crosslists (BIOL, CMPS, ECMP, ECES, EECS, NEUR, EBME), numbers (375, 475, 479), and names (originally *Seminar in Computational Neuroscience*, became *Autonomous Robotics* in Class 4) used at various points during the course's existence.

A more detailed version of this table has appeared previously<sup>5</sup>.

Key:

Participants:

Numbers are given as undergraduates/total students, except for Classes 1–3, which had a graduate-only listing, though both undergraduates and graduate students were enrolled.

Course Structure:

- A All students: 6 first-half exercises with checkouts or brief lab reports; second-half project (did not have to be Egg Hunt robot), design notebook for project only.
- B All students: 7 first-half exercises, no lab reports; second-half project is Egg Hunt robot; design notebooks kept throughout the semester.
- C Like B except, for graduate students only, 7 first-half lab reports and a 10-page design paper about the final Egg Hunt robot were required, in addition to the semester-long design notebook.

Egg Hunt Tournament Structure:

- D 4-team round-robin, 1 trial for each pair, 2 highest total collectors met in final round, pastel/black egg ratio 42/6, egg values +1/–3.
- E 4-team double-elimination, seeds by first-half performance benchmarks, pastel/black egg ratio 36/10, egg values +1/–4.
- F 4-team double-elimination, seeds by results of 2 mock competitions, pastel/black egg ratio 40/10, +1/–4.
- G 4-team double-elimination, seeds by staff to balance teams, pastel/black/gold egg ratios 40/10/1, egg values +1/–4/+2 (gold egg motorized and erratically mobile).
- H 5-team modified double-elimination (all teams guaranteed to play twice, but one part of bracket faced elimination with only one loss), seeds by staff to balance teams, egg colors/ratios/values as in F.
- I 4 team double-elimination, seeds and egg colors/ratios/values as in H.
- J 3-team round-robin, 3 trials each, 2 highest total scores advanced to final, winner was best out of 3 trials, egg colors/ratios/ values as in F.

Contest Site:

- campus Large auditorium on the campus of Case Western Reserve University; publicly-advertised event.
- GLSC Large auditorium at the Great Lakes Science Center, Cleveland, Ohio; publicly-advertised event.
- class lab The class laboratory space at Case Western Reserve University; private; held over 2 regular class periods.

### Course Grading:

- K First-half assignments checked out by demonstration to instructor, sometimes with 1- or 2-page lab report, graded satisfactory/unsatisfactory; same for design notebook, with the threat of lowering the course grade by one letter if it were deemed unsatisfactory (never applied).
- L First-half assignment checkouts as in K, but no lab reports; all results, analysis, and final Egg Hunt robot were to be recorded in the design notebook; grading still overall satisfactory/unsatisfactory.
- M Design notebook usage like L, but quantitative grading scheme (100 points, first and second half, course grade 33% midterm + 67% final): attendance (15), subjective contribution to group (10), regularity of design notebook entries (10), neatness (5). mechanical design (25), software design (25), performance evaluation (10).
- N Like M except weighted 50% midterm + 50% final, and components revalued: attendance (10), subjective contribution to group (20), regularity of design notebook entries (9), neatness (4), mechanical design (24), software design (24), performance evaluation (9).
- O Rubric-matrix system, including informational-but-does-not-count spot-grading of first 8 notebook entries, peer review, and instructor-assigned group dynamics grades (100 points, first and second half; 35% midterm + 65% final).

### Design Notebook Format:

- hand handwritten notebooks assumed, no mention of typed or electronic content.
- P Electronic format notebooks strongly discouraged, based on prior webpage and other attempts
- Q Electronic design notebook experiment: +5 per half-semester for participants as a “hardship reward” to offset any extra trouble encountered in using electronic content-generation tools; no requirement that they be 100% electronic.
- R Electronic design notebooks permitted, encouraged for typed text; no “hardship reward”.

### Extant Notebooks:

Given as number of undergraduate notebooks/total notebooks extant.

A significant amount of student work is extant, including code, digital photographs, videotape (class activities, Egg Hunts, exit interviews), copies of grade sheets, and 158 unclaimed lab notebooks<sup>5</sup>. These are invaluable records of the operation of the course, and provide concrete references for comparison with other robotics courses, both traditional and online. Moreover, one of the current authors (RFD) was a co-instructor for all but the 20th semester of the course, affording a perhaps unprecedented degree of continuity and perspective.

## **2.2. Autonomous Robotics for High School Science Teachers (BIOL 803a)**

After only a few offerings of LEGO 375, it became clear that the group-based curriculum design enabled students to become competent in many subject areas, despite lack of formal prerequisites. By course’s end, a typical student with no prior programming experience could write simple, robust, and well-commented code, of good quality, without much formal instruction in C programming *per se*. Similarly, students with little mechanical design experience, or who had not previously played with LEGOs, could construct simple machines, design geartrains capable of trading speed for power, and build sturdy structures, simply by participating in the course and interacting with their groupmates.

These observations suggested that the LEGO 375 curriculum and laboratory design could help STEM educators to teach computer or robotics laboratories at the secondary school level. In July 1997, a group of high school science teachers and their students participated in a 5-week pilot autonomous robotics course at CWRU that was loosely based upon the current LEGO 375

curriculum, just to see what would happen (Table 2, Class 1). Participants borrowed the LEGO kits from LEGO 375, but used the then-new MIT Handy Board<sup>14</sup> robot system, a more durable descendant of the 6.270 board that was compatible with LEGO's new 9-volt DC motors (the 6.270 board was a 5-volt system and LEGO discontinued their original line of 4.5-volt motors). Popular interest in the Mars Pathfinder mission<sup>11</sup> (which landed just prior to the start of the pilot course) and the Sojourner semiautonomous robotic rover led to the creation of a novel exercise to replicate the mechanical design of the 6-wheeled Sojourner in LEGO and the software architecture with the Handy Board, guided by published detailed engineering descriptions on NASA websites<sup>12</sup>. Response from both teachers and students was enthusiastic: the teachers borrowed a complete Handy Board kit for further testing, and by Spring 1998 had made a major equipment purchase to equip a lab at their school with Handy Boards and LEGOs, using their own variants of the Summer 1997 robotics curriculum. This encouraged the formal development of a new 10-day summer course, Autonomous Robotics for High School Science Teachers (BIOL 803), that began with 10 participants in Summer 1999. To distinguish this initial traditional in-person robotics laboratory from the later distance-learning version, which had a different name but the same course number, we will use the names *BIOL 803a* and *BIOL 803b*, respectively.

Table 2. Overview of Summer Autonomous Robotics Courses (BIOL 803) at CWRU.

Class	Year	Participants	Course Structure	Contest Structure	Classroom Type	Course Grading	Design Notebook Format	Extant Notebooks
1	1997	3/4/0/0	A	none	lab	none	hand	0/7
2	1999	6/4/0/0	B	F	lab	K	hand	5/10
3	2000	8/1/0/1	B	F	lab	K	hand	7/10
4	2001	3/3/0/0	B	F	lab	K	hand	4/6
5	2003	4/4/2/0	B	F	lab	K	hand	0/10
6	2004	3/4/3/0	C	F	H	K	hand	8/10
7	2005	3/1/1/1	D	G	I	K	electronic	5/6
8	2006	4/2/2/1	E	G	J	K	electronic	9/9

Note: The text refers to classes 1–5 (traditional lab) as *BIOL 803a*, and to classes 6–8 (transition to online) as *BIOL 803b*, in order to distinguish the two course designs. While all classes were actually listed as *BIOL 803*, two different titles were used: *Autonomous Robotics for High School Science Teachers* (classes 1–4), and *Inquiry-Based Approaches to Autonomous Robotics* (classes 5–8), after participation was extended to include K–12 teachers and grade 7–12 students.

Key:

Participants:

Numbers given as male teachers/female teachers/male students/female students.

Course Structure:

A Pilot project. Course met 10 days over 5 weeks, 3 hours per day. 7 sessions on LEGO mechanical design and construction of a wheeled robot chassis, 3 sessions on sensor properties and obstacle avoidance using instructor-written control software. Exercises were informally adapted from those in use at the time in the undergraduate Autonomous Robotics course. Participants were also allowed to explore an instructor-provided LEGO implementation of the NASA Mars Sojourner robot, whose mechanical design and control software architecture were modelled after descriptions published on NASA websites. Participants worked in teacher-student pairs. No formal lab notebooks, though some participants did record observations on an irregular basis.

- B Formal adaptation of the undergraduate Autonomous Robotics course. Course met 10 consecutive weekdays, 6 hours per day. 6 days of structured exercises (LEGO mechanical design, novel locomotion, IC programming, obstacle avoidance, light detection, and sensory integration), 1.5 days of contest robot design, 1.5 days of testing and contest, 1 day of presentations, exit interviews, disassembly, and inventory. Participants worked in pairs; students were always paired with a teacher. Formal design notebooks required.
- C Same as B, except part of the mechanical design exercise and all of the novel locomotion exercise were removed to make room for 1.5 days of teleoperated robotics, using the Khepera simulator and robot at WSU. The remainder of the course utilized the LEGO/Handy Board hardware platform as before. This was the first transition towards a distance-learning version of the summer robot course.
- D New teleoperated robotics curriculum, using the Khepera simulator and robot at WSU for all but 2 days of obstacle avoidance from the LEGO/Handy Board platform. New Khepera-based maze navigation contest. New curriculum included structured exercises in teleoperated robotics, Java programming, sensory integration, contest controller design and testing, the maze contest, and a final day of presentations and exit interviews. Participants were encouraged to work on their own, not in groups. Electronic design notebooks required.
- E Same as D, except the last of the original LEGO/Handy Board curriculum was phased out; the class used the Khepera/distance-learning curriculum exclusively. Course met 8 days over 2 weeks, 6 hours per day.

#### Contest Structure:

- none No contest, just an obstacle-avoidance free-for-all using all robots in a cluttered, walled arena.
- F Light Tag: in a cluttered, walled arena, participant robots had to avoid an instructor-supplied autonomous robot (*IT*), which carried bright incandescent lights and is detectable by various light sensors. If *IT* touched a robot, the robot was moved to the opposite end of the arena and restarted; 3 tags and the robot was eliminated; last robot left moving that had not been tagged 3 times (or had the fewest tags if *IT*'s motor batteries were depleted first) was the winner.
- G Maze navigation: each participant could either (1) write a controller in Java for the Khepera robot (either the physical robot or the simulator) to drive through a maze autonomously; or (2) use an instructor-provided direct-drive controller that accepted input of direction and speed from the user via keyboard, while progress was monitored via video feed or the simulator. The fastest time through the maze was the winner, for each method (1) or (2).

#### Classroom Type:

- lab Large open laboratory space: 10 6-foot work tables, each with a desktop computer, arranged around the perimeter facing the walls, with open floorspace in the middle. Instructor and students communicated freely by voice and personal presence. This is the same laboratory facility as used by the undergraduate Autonomous Robotics course.
- H Same physical lab space as above. The teleoperated robotics segment was done with pairs of participants working at single desktop workstations. The instructor circulated freely about the room to offer assistance as needed. Participants were in chat-client-only contact with the remote instructors at WSU, but in-class communication between students and the local instructor was by voice and personal presence.
- I Class was mainly held in a dedicated computer lab consisting of parallel rows of desktop workstations, all facing one direction, using the teleoperated robotics and simulator environment. Participants each had their own workstations, and were encouraged to communicate with others (both local and remote) only through the chat client, to simulate the entire class being remote. A video projector was used to display instructor lecture materials and live robot video from WSU on a screen at the front of the room. During the 2 days of obstacle avoidance using the LEGO/Handy Board platform, the class relocated to the previous undergraduate robotics lab, and worked in teacher-student pairs, as in the traditional versions of the course.
- J Similar to I, except a different computer lab consisting of laptop computers with better screen resolution was used, and the class was held entirely in the computer lab.

#### Course Grading:

- none There was no graded work in the pilot course. The object was participation-only.
- K 100 points total, lab notebook (70), class participation (30). Lab notebook: regularity of design notebook entries (9), neatness (4), mechanical design (22), software design (24), performance evaluation (9). Class participation: attendance (10), subjective contribution to the group (20). This is the same as the

undergraduate Autonomous Robotics course (pre-rubric-matrix grading scheme, which was never introduced in BIOL 803).

Design Notebook Format:

hand	Handwritten notebooks assumed. Printouts of code listings and digital photographs could be manually inserted (taped or stapled).
electronic	All notebook observations typed into a word processor. Code listings, digital photographs, and simulator screen captures could be inserted into the notebook file.

Extant Notebooks:

Numbers given as extant/total students.

BIOL 803a followed the same educational paradigm as the parent LEGO 375 course: multi-student groups (pairs), structured exercises in LEGO mechanical design, IC programming, sensor properties, and feedback control; a capstone contest (Light Tag<sup>1</sup>), and handwritten design notebooks (Table 2, Classes 2–5). The object of Light Tag was to create an autonomous wheeled robot capable of both obstacle avoidance in a cluttered arena, and light-guided avoidance of an instructor-provided autonomous robot, *IT*, which carried bright incandescent lights to announce its presence. *IT* was modified from *Cambot*, an autonomous obstacle-avoiding robot used during LEGO 375 Egg Hunt contests to broadcast a robot’s-eye view of the competition to spectators via color video camera and short-range TV transmitter. The structured exercises were simplified versions of the LEGO 375 curriculum; however, an IC tutorial developed for BIOL 803a proved so effective at teaching elementary concepts to non-programmers that it was introduced into the LEGO 375 curriculum as an addendum to the standard IC programming exercise. Indeed, BIOL 803a was frequently used as a testbed for possible improvements to the LEGO 375 curriculum.

While the pilot course (Course 1) included both teachers and secondary-school students, the production version was intended for teachers only, mostly due to perceived difficulty with the original IC programming exercises. After the outstanding experience of one female high-school student in Class 3 (who, after taking the class paired with a teacher from her school, built an elaborate and national prize-winning science fair project to study the kinematics and control of dog walking, modelled with Handy Board-controlled LEGO structures), this limitation was reconsidered and, after a 1-year hiatus for curriculum redesign, removed. Class 5 expressly recruited teacher-student pairs from the same school, and was renamed *Inquiry-Based Approaches to Autonomous Robotics*.

### **3. Mechanics of Online Robotics Courses**

The two online robotics courses discussed in this paper represent attempts to deliver the rich and successful educational experience of the in-person traditional courses via an electronic virtual classroom. Three significant changes from the traditional course were made in order to accommodate an online format: (1) replacement of group-based work in a shared physical classroom with single-student tasks in remote environments having various network-based “windows” for communication and access to shared robot hardware; (2) elimination of the mechanical design curriculum, focussing instead upon software control of a single, fixed robotic architecture, available in both local simulator and remote-access physical versions; and (3) electronic-only lab notebook format requirements.

First, while virtual classrooms offer significant advantages (improved student access not limited by geography, reduced cost due to sharing of expensive resources or availability of inexpensive simulators), there are intrinsic challenges that must be overcome, which are not part of a traditional in-person classroom environment. Novel communications channels must be provided for all the kinds of information that must be exchanged, both technical and personal. While some data can be received passively (*e.g.*, a streaming video feed from the remote laboratory which houses the physical robot), active communication among students and instructors is through a text-based chat client, rather than spoken conversation. This may not be as natural or effective for some students, since traditional verbal and non-verbal cues are absent, or imperfectly communicated (*e.g.*, emoticons, text-messaging shorthand or abbreviations). In addition, the lack of group-based tasks does not provide a natural impetus for any students to communicate with each other, only with the instructor, so opportunities for students to receive peer assistance are reduced. All the communication channels and local work areas (simulator, code editor, word processor) are confined to a single computer screen, rather than being diffused throughout a physical classroom. Finally, the emphasis on individual work requires students to be more independent and self-reliant — students bear the entire responsibility for the success of their own tasks; there is no group safety net.

Second, short of providing some kind of 3-dimensional robot-building simulator, there was no practical way to include the mechanical design exercises of the traditional robotics courses in the online versions. Providing each student a physical robot kit would not allow full exploration of the shared-hardware concept, and without high-resolution 2-way video conferencing capabilities among students and instructors, it would have been very difficult for participants to manage design complexities and communicate spatial information effectively. The online courses, therefore, were designed to utilize a fixed hardware platform (the Khepera<sup>17</sup>), guaranteeing the same level of functionality of the physical robot for all students, as well as simplifying the task of developing a robust simulator<sup>15</sup> for individual, local use. The focus became exploring principles of software control for the fixed platform, noting physical limitations and programming around them as needed, but without the option (as in the traditional courses) of changing the mechanical design to simplify the software control. Performance evaluation could now include consideration of differences between the simulated robot and the physical robot in a real environment. Because each student had a local copy of the simulator, initial development and testing could take place off-line; when ready, the student could then test the same controller on the physical robot. In principle, the simulator would be an accurate reflection of the real-world environment. Discrepancies could lead students to develop more robust control algorithms, and/or feed back to the instructors in order to improve the simulator itself. Students could work completely according to their own schedules using the simulator, and with minimal coordination, using the physical robot, although some common meeting times (through the virtual environment) were scheduled to maintain a semblance of a “classroom”

Third, electronic document formats were required for classroom observations, for convenience of network exchange between remote students and the instructor. These could be native word processing documents (the most straightforward to produce and yielding the smallest files), but any combination of typed text, scanned writing or drawing, screen dumps of simulator windows, or even digital movies of simulator sessions or the physical robot in action, was acceptable. The

object was to maintain the richness of handwritten lab notebooks from the traditional courses while making them into cross-platform electronic documents, without excessive effort.

### 3.1. WWW Autonomous Robotics (CEG 499)

WWW Autonomous Robotics<sup>6,7,8,9,10,15</sup> at Wright State University (WSU) is a guided project-based course, modelled after LEGO 375 (Table 3). There are no tests and no structured classroom interaction. Rather, students are judged based on their performances in completing a series of increasingly-difficult projects. The instructors provide the tools, technical knowledge (including an online textbook<sup>3</sup>), and equipment needed to complete each project. Students are required to manage their own efforts and make appropriate use of available resources to complete their projects in a timely manner. They are also required to engage fellow students in discussions about the techniques which they applied on a class discussion list. Students document their work in electronic project journals (in PDF format) that are submitted for review on a regular basis. In addition, students are required to answer a series of discussion questions every week, submit those answers, and participate in group discussion over the discussion questions at regular intervals. There is no special open capstone project, but students are given the opportunity to demonstrate (through the virtual classroom environment) an example of a problem solution which they feel is particularly successful or elegant.

Table 3. Overview of WWW Autonomous Robotics (CEG 499) at WSU.

Class	Year	Participants	Course Structure	Contest Structure	Classroom Type	Course Grading	Journal Format	Extant Journals
1	2005	5	A	B	virtual	C	electronic	5
2	2006	8	A	B	virtual	C	electronic	8
3	2007	5	A	B	virtual	C	electronic	5

Key:

Participants:

Primarily computer engineering and computer science undergraduates. Two first-year electrical engineering graduate students participated in 2006. All students were male.

Course Structure:

A Scheduled, guided practicum projects. Asynchronous access to robot and chat rooms 24/7. Interaction with other students and staff via ad-hoc scheduling, or via E-mail. Additional access to staff via virtual office hours and by appointment. Whole-class online meetings once per week with required online attendance once every 2 weeks.

Contest Structure:

B Informal contests held online at end of term. Students demonstrated their “best of show” controllers for a class problem of their choice. Students were encouraged to show controllers for the more difficult problems at the end of the course.

Classroom Type:

virtual No physical classroom or laboratory. Online access to a chat client with whiteboard (Coccinella), streaming video feed (QuickTime) from the lab containing the physical Khepera and its arena, Khepera robot/arena simulator with shared-session capabilities. Students provided their own computers and high-bandwidth network access (DSL or better).

Course Grading:

C 100 points total, lab notebook (60), class participation (40). Lab notebook: regularity of design notebook entries (10), neatness (10), mechanical design (20), design testing and critical review (20). Class participation: participation in discussion of questions supplied before class (24), instructor's subjective assessment contribution to the group discussions (16). Modelled after criteria from LEGO 375 (Table 1, M and N).

Journal Format:

electronic Regardless of how originally generated (e.g., typed, scanned, screen dumps, digital images), journal entries were required to be submitted in PDF format. Freeware multi-platform authoring tools were provided.

Extant Journals:

All student journal files (PDFs) are extant.

### **3.2 Inquiry-Based Approaches to Autonomous Robotics (BIOL 803b)**

In Summer 2004, BIOL 803 began a 3-year transition from its traditional laboratory format to a distance-learning virtual classroom environment (Table 2, Classes 6–8), termed here *BIOL 803b*. Each offering had unique features that merit discussion.

Class 6 replaced a free-exploration LEGO mechanical design exercise with 3 half-day sessions on teleoperated robotics, using the Khepera simulator and remote Khepera at WSU. This initial curriculum did not require Java programming skills, and merely allowed pairs of participants to observe the behavior of pre-written controllers (wall-following, obstacle avoidance, keyboard-input remote control) in both the simulator and the physical robot. Streaming video allowed everyone to follow the progress of the remote robot. A Java-based chat client accessed via web browser (Spaniel Chat) was used to allow participants to communicate with each other and with the instructors (both at CWRU and WSU). Participants were encouraged to use the chat client exclusively for communication during the exercises, even though all were present in the same physical laboratory as the local instructor. According to the extant chat logs, this request was mostly ignored, except for exclamations of glee or approval, or specific requests directed at the remote instructor (who could be reached only via the chat client). Most of the conversation between participants was ordinary speech, and the local instructor circulated among the workstations to give personal assistance or intervention. Since the bulk of the Class 6 curriculum was still the traditional in-person group-based format (including the capstone Light Tag contest), this result is not surprising. Design notebooks were also in the traditional handwritten format.

For Class 7, the curriculum of BIOL 803b was changed significantly from the traditional format. Except for the 2-day LEGO-/Handy Board-based obstacle-avoidance robot exercise (which was held in the traditional laboratory), all other course activities used the Khepera robot (both remote and simulated) and took place in a large computer laboratory. Participants were introduced to programming in Java, and all the former exercises in programming, motor behavior, sensor properties, and feedback control of behavior, were adapted to the Java/Khepera environment. The end-of-course robot contest was changed to timed maze navigation instead of Light Tag, since neither the simulated nor the physical Khepera environments could support multiple robots. All course materials were made available in electronic form. Design notebooks were required to be in electronic format (document files in Microsoft Notepad or Microsoft Word). Streaming video and chat clients were provided for communication. The instructor was gently insistent upon exclusive use of the chat for communications and queries rather than speech or physically

visiting someone's workstation (colloquially referred to as "beaming over" when it seemed unavoidable; obviously this would be impossible if everyone were at a unique remote site). As a result, after the first few days, the lab environment became very much a virtual classroom, each participant absorbed in his/her own task (teachers and students were encouraged to work separately as much as possible, rather than in pairs, though this was sometimes relaxed). Compared to previous traditional versions of BIOL 803, the physical environment was eerily quiet, the only sounds being the clicking of keyboards and mice, and an occasional self-exclamation. During the old LEGO robot exercise, the lab resembled the traditional form, and all participants did seem to be at their most active and enthusiastic during these 2 course days.

The last offering of BIOL 803b (Class 8) represented the final break with the LEGO robotics curriculum of BIOL 803a. All 8 days of the course utilized the virtual classroom and Khepera simulator/remote robot environment. (Due to various participant scheduling issues, it was not possible to run the course for the usual 10 days. However, the "missing" 2 days correspond to the omitted traditional-format obstacle avoidance exercise, so Class 8 had the same online curriculum as Class 7.) The 3 student participants were the youngest to date (all rising 8th graders), and it was much more difficult to enforce the work-alone and chat-client-only communications rules, compared to Class 7, although it was still much quieter than in any of the traditional-format courses. The 2 teachers who were from the same school system as the 3 students mostly paired up with the students, to help them with their tasks, and to encourage them to type entries in their design notebook documents. The Java programming exercises were very difficult for these students to complete, even with teacher and instructor assistance. Only one teacher attempted an autonomous solution to the timed maze navigation contest; all other participants used the keyboard-entry direct-drive controller provided by the instructors. Significant instructor "beaming over" for direct intervention and troubleshooting was still required. All participants (and the local instructor) were mentally very weary by the end of the course.

#### **4. Comparison of Traditional vs. Online Courses**

The traditional and online autonomous robotics courses reflect the consequences of their fundamental differences in design choices for the laboratory environment: in-person, hands-on, group-based work involving construction and software control of robots vs. remote-access, independent work utilizing a detailed robot simulator as the primary development system. For example, physical lab notebooks cannot be exchanged in a virtual classroom, but electronic documents can, so student journals must be authored in, or at least converted from traditional forms to, electronic formats. Moreover, given the variety of computer platforms in the marketplace, the electronic formats chosen must be usable under a wide range of hardware (*e.g.*, from 200 MHz Pentium IIs to 3.6 GHz multi-core Pentium Ds) and operating systems (*e.g.*, Windows, MacOS, Linux), or even a range of versions of the "same" operating system (*e.g.*, Windows 2000, XP, and Vista).

Since the traditional course is the reference of judging "success", one strategy to translate the traditional course into an online format is to identify the critical features of the traditional course that make it successful, and re-implement them in the online format, using appropriate tools. If there is not a straightforward translation available, then multiple workarounds must be used.

Consider classroom communication. The traditional laboratory setting allows high-bandwidth communications in multiple simultaneous modes (voice, sight, and hearing; verbal and non-verbal), some of which may be imperceptible but are always operating in the background. Current technology for virtual classroom delivery has greatly reduced communications bandwidth: a computer screen is the “window” into the classroom environment, and dedicated, prioritized channels for communicating information must compete for space and user attention in that window. Written text lacks subtlety: auditory and non-verbal cues that clarify meaning during in-person communication are absent. Careful writing can compensate, but not all users have the training or temperament to communicate as effectively in writing as in ordinary speech. As another example, the utility of a streaming video feed of a remotely-operated robot depends upon the bandwidth available (*e.g.*, network speed, computer speed, video size and resolution). Thus, unless course participants are made aware of the known limitations, or given the opportunity to learn how to function within them, serious misunderstandings can arise that degrade the quality of communication.

#### **4.1 University Courses (LEGO 375 vs. CEG 499)**

Despite a greater curriculum scope (both mechanical and software design *vs.* software only), LEGO 375 was perhaps an easier course for typical students than CEG 499. First, traditional laboratories are still more familiar learning environments for contemporary students; even though they may have extensive experience with virtual environments in online social or gaming situations, these are very different from classrooms. Second, LEGO 375 students worked in groups of 3, so no individual student was required to be competent at both mechanical design and software design — other members of the group were available for assistance. In CEG 499, all students work alone and must be competent in everything required for the course; structurally, there is no opportunity to share task responsibilities, or to trade expertise in one area for a groupmate’s expertise in another, which is a benefit of group-based work. While the instructor and other students are technically available (through chat or E-mail), the effect is still potentially isolating. Weaker students may be unable to effectively resist the isolating effects and react by withdrawal. Only focussed, well-organized, and highly-motivated students can succeed in the current CEG 499 virtual laboratory environment; these students revel in the medium and exploit it to increase access and opportunity. Finally, it is likely that some students who received high grades in LEGO 375 would fare poorly in CEG 499, even if their expertise or preference was for software design, without the traditional social support available in the in-person laboratory. Indeed, in the virtual environment, we note that it is more difficult to detect students who are drifting away and not participating, because the online communications channels are so narrow (as narrow for the instructors as for the students); this can delay instructor intervention until it is too late for a floundering student.

Students in LEGO 375 shared a common physical laboratory that was set up and maintained in working order by the instructors; this was a primary instructor responsibility, that the students could take as given, a guaranteed infrastructure. In CEG 499, establishing a functional local virtual environment has been partially offloaded to each student. Students must install the common software on their own computers, and they are responsible for all system administration duties. Even though all the tools required for the course are easily downloadable from both the open Internet and local servers, the installation and configuration of these tools has proved

surprisingly problematic, even for technically-minded students. The amount of time “wasted” on configuration and testing on a per-student basis still detracts significantly from the course. It is apparent that not all students are competent to optimally configure their own computers; indeed, the modern marketplace trend is toward the computer-as-appliance with no user-serviceable internals (hardware or software), so this situation will likely worsen. Future offerings of CEG 499 will provide pre-configured thumb drive and/or CD images that contain all necessary tools, perhaps even being entire bootable systems in their own right (which would avoid affecting a student’s existing configuration).

The computer workspace environment was much simpler for LEGO 375 (Fig. 1) than for CEG 499 (Fig. 2). This is mainly due to the fact that, in LEGO 375, computers were used primarily for code development and incidental web browsing (e.g., to look at assignment text on the course webpage), whereas in CEG 499, the computer screen was a student’s only (though multi-channel) window into the entire virtual classroom (Khepera simulator, robot streaming video feed, chat client, code editor, compiler window, lab journal word processing document, and course textbook viewer), and thus could get very cluttered.

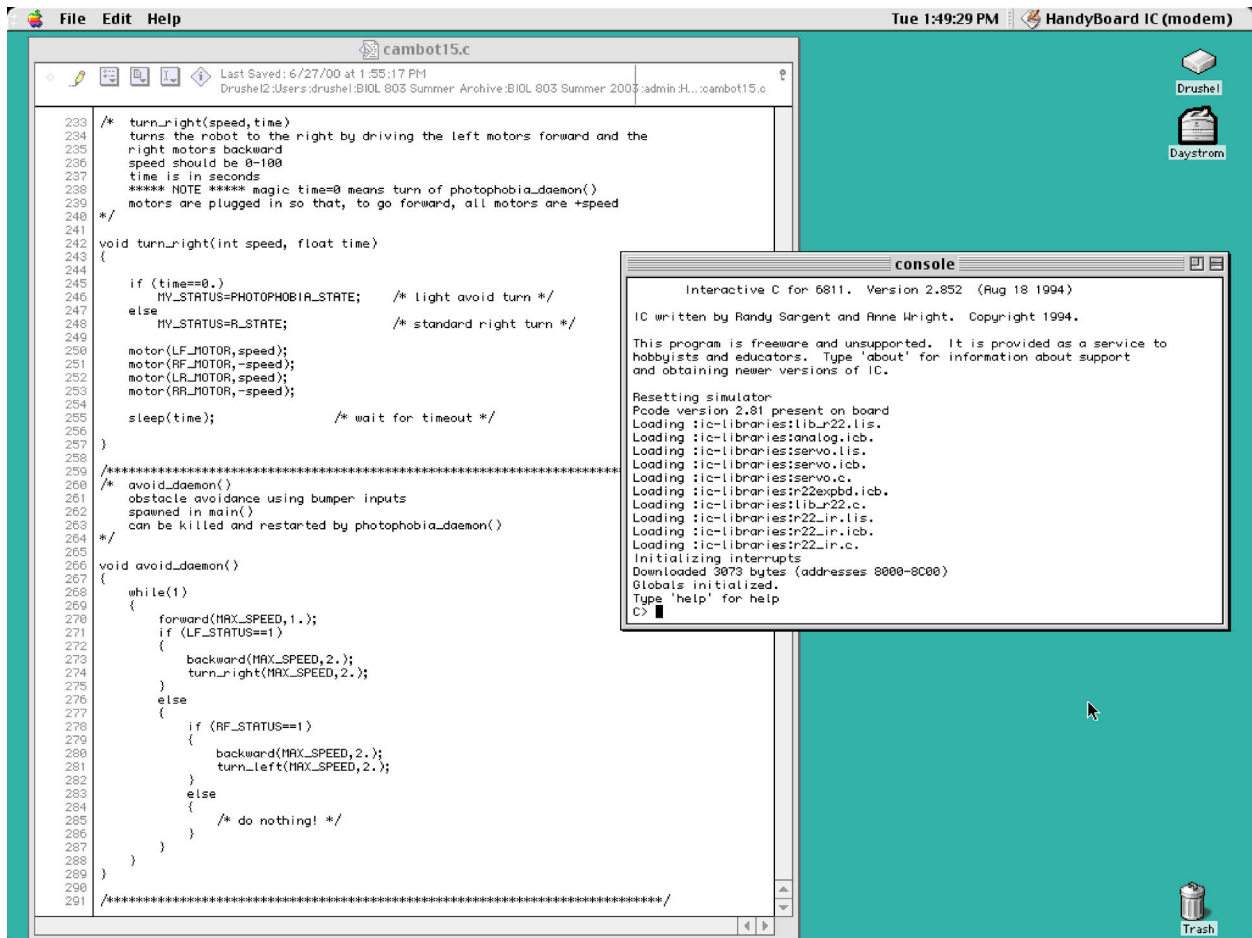


Figure 1. Screenshot of typical work environment from the traditional autonomous robotics courses (LEGO 375 and BIOL 803a). Lab computers were various models of Apple Macintosh (PowerPC) desktops running MacOS 7.x, 8.x, or 9.x (or 9.2 “Classic mode” under MacOS X versions less than 10.5.x). This example is from a Power

Macintosh 7600, MacOS 9.2, showing the MIT Handy Board Interactive C (IC) interpreter (a Motorola 68K binary running under emulation) and a code file opened in the BBEdit 4.5 text editor. Screen size is 1024 × 768 pixels.

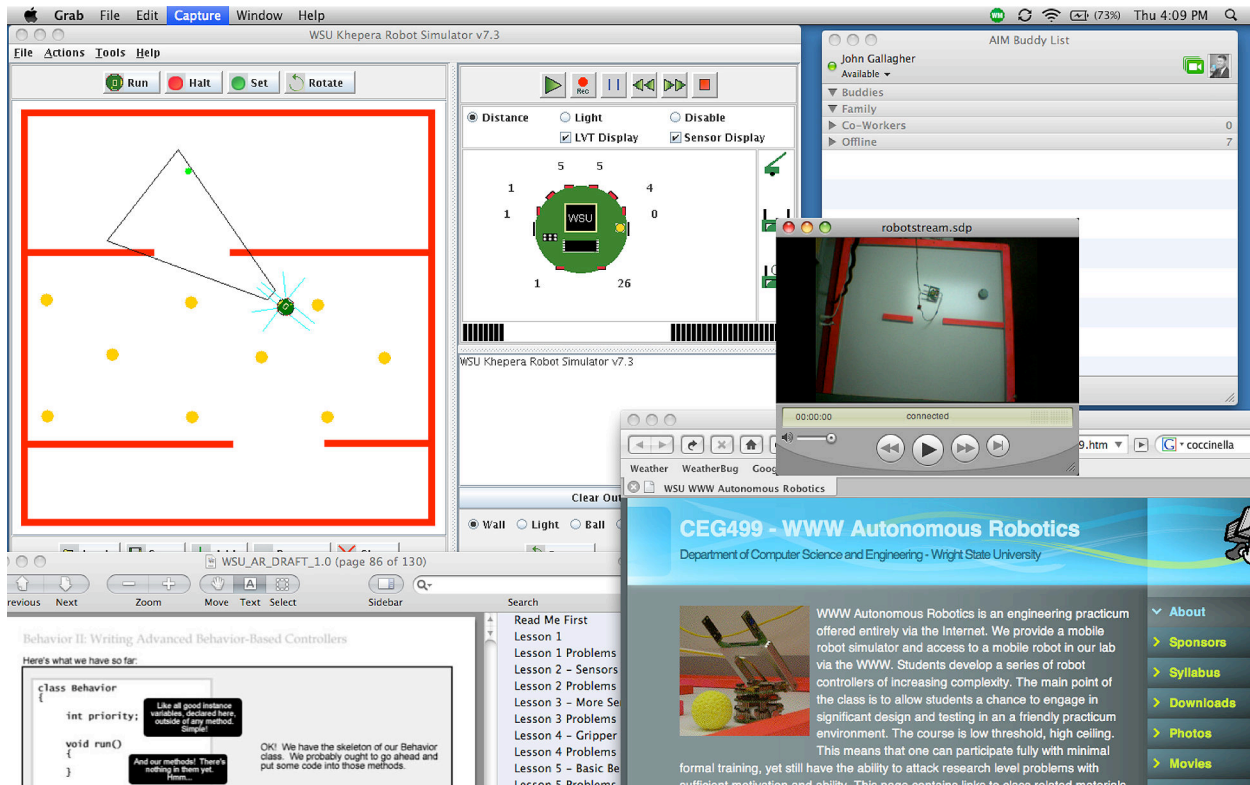


Figure 2. Screenshot of typical work environment from the online autonomous robotics courses (CEG 499 and BIOL 803b). CEG 499 student computers and BIOL 803b lab computers were usually commodity (Intel x86) desktops or laptops running Microsoft Windows 2000 or XP. This example, however, is from an instructor's Apple 17-inch MacBook Pro, demonstrating the cross-platform capabilities of the course software. Illustrated are the Khepera simulator, a streaming video feed (QuickTime) from the WSU robotics lab, a chat client (iChat, though typically Coccinella), a web browser open to the course webpage, and a PDF viewer displaying part of the electronic textbook<sup>3</sup>. Other windows that could also be open on this extremely cluttered desktop include a text editor for Java code development and a word processor for keeping the class journal. Screen size is 1400 × 900 pixels.

## 4.2 STEM Educator and Student Courses (BIOL 803a vs. BIOL 803b).

The traditional and online versions of the summer teacher/student courses show many of the same differences as the university-level courses, and these differences are magnified. In the traditional course (BIOL 803a), both teachers and students had much more success with LEGO mechanical design problems than programming or behavioral control problems. Indeed, participants were more tolerant of mechanical difficulties than programming bugs, even if the time spent troubleshooting was the same for each. In teacher-student pairs, the student frequently did the bulk of the LEGO construction, while the teacher assumed responsibility for the control software, especially for middle-school students. In the online course (BIOL 803b), especially in Courses 7 and 8, everyone became restless and frustrated at the picky syntax of Java, and cryptic compiler warning and error messages. Participants seemed to “get” programming principles much more easily in IC than in Java, and certainly could get IC code to compile without errors much more easily than Java code. Only a few teachers and one student

had any prior exposure to Java, and even these participants had some programming difficulties. For the truly novice programmer, IC seems markedly simpler to learn than Java. Both versions of BIOL 803 also showed the same differences in computer usage as between the traditional and online versions of the university courses, namely, the traditional course used the computer as code editor and debugger, while the online course used it for all the communications channels (chat, video, simulator, *etc.*). To preserve working screen space, as well as reduce network bandwidth consumed, for Classes 7 and 8, the instructor used a computer video projector to display some course materials on a large screen at the front of the computer laboratory, visible to all participants at their workstations. Information displayed included (as appropriate) live robot video feed from the remote lab at WSU, simulator demonstrations, and a text editor window used as a whiteboard for typing lecture notes and Java programming examples. This allowed participants to reduce the number of open or tiled application windows needed on their own computer screens, usually allowing the chat window and one or two others to be open without too much clutter.

Chat communications were terse by all participants, and frequently of low information content, requiring repeated direct questions from the instructor in order to provide assistance. Students frequently used the chat client for incidental conversation, but only rarely for technical assistance, except at the direct inquiry of the instructor or another teacher. Content improved somewhat with practice, but no formal training or guidelines were provided. This issue should be carefully considered for subsequent offerings of any of the online versions, university or teacher/student.

## **5. Discussion**

We have identified a number of issues that detract from the efficacy of online courses, compared to the traditional in-person laboratory courses:

(1) Communications bandwidth is low in online courses. Extra effort must be expended to compensate. We had, it seems incorrectly, thought that students who had become more socialized to text messaging (via cell phones, Blackberries, *etc.*) and Internet tools would be sufficiently expressive in online media to at least partially offset the problems of limited communication bottlenecks. We have found this to be untrue. Student descriptions of phenomena (especially problems encountered) often lack specificity and immediacy, and the instructors often find themselves pulling meaning out of students as if they were being deposed in a courtroom. In live classroom environments, this does not seem to be as much of a problem.

(2) The instructor cannot “see the whole class” from the terminal the way he/she can in a real classroom. It becomes all too easy for the instructor to lose immediacy and to inadvertently ignore those who are quiet.

(3) Serial chatlog makes multi-thread conversations confusing. It is difficult to filter out the background chatter as is possible in a real class when 10 groups are talking about 10 different topics at once. Various communities (online gamers, ham radio operators, NASA flight controllers, *etc.*) have developed communication protocols that ameliorate this problem. However, there is no standard that students already know. Furthermore, many students seem

disinclined to use such protocols and/or feel unduly restricted in their communications when they do.

(4) The screen window on the world is too small for all the open windows needed. This is in part a direct result of targeting our class to be accessible even to those with modest home computers. However, there may be no other alternative than to move to larger screens and/or higher bandwidth connections.

(5) In the traditional versions of the course, one need only be physically present for others to detect and help to fulfill needs. In the online versions, there is a bar of entry (getting tools up and configured) that must be cleared before reaching minimal visibility to students and instructors. This already disadvantages a large class of students. Even those who do clear the bar do so at different rates, meaning that latecomers risk falling behind before the race even starts. We will attempt to lessen this problem by providing self-contained class tool environments that require zero configuration. Furthermore, since the design of the online classes does not explicitly reward group efforts, there is little reason to collaborate save altruism. In the future, we will also redesign the course so that students are explicitly rewarded for group efforts and are explicitly paired off with other members of the class for some work tasks, with the goal of making most students work with most other students at least once during the course.

(6) The loss of mechanical design tasks increases disinterest for some people, especially the K-12 students in the summer course. Code and simulations seem too boring for them, and they have difficulty focussing. Having LEGOs to play with makes at least some of the tasks less abstract and more interesting.

(7) Java is much harder than IC, especially for novice programmers. This objection is difficult to overcome with the Khepera robotic platform, given our current investment in the simulator environment. Java is platform-independent, while IC is not.

We remain committed to our explorations of the possibilities of online delivery of autonomous robotics practica. Despite some difficulties, we are confident that the limitations we have described in this paper can be overcome, at least somewhat, through both curriculum redesign and improved student orientation to the virtual laboratory environment.

### **Acknowledgements**

This work was supported by the National Science Foundation under grants DUE 0341150 (RFD) and DUE 0341263 (JCG). Additional support has been provided by Wright State University, the Ohio Board of Regents, and the Howard Hughes Medical Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the page author(s) and do not necessarily reflect the views of the National Science Foundation or other sponsors.

### **Bibliography**

- 1 Alexander, S. [no byline] (2003). Tag, you're *IT*. Howard Hughes Medical Institute News: Science Education News. <<http://www.hhmi.org/news/legorobotsgia.html>>
- 2 Beer, R.D., Chiel, H.J., and R.F. Drushel (1999). Using autonomous robotics to teach science and engineering. *CACM* **42**: 85-92.
- 3 Bolick, D.S., Gallagher, J.C., and R.F. Drushel (2005). *Learning autonomous robotics*. Unpublished textbook manuscript.
- 4 Drushel, R.F. (2007). Autonomous LEGO robotics at Case Western Reserve University: BIOL/EECS 375/475. Historical course website archive (Class 19, Fall 2004). <<http://drushel.cwru.edu/375/>>
- 5 Drushel, R.F., and J.C. Gallagher (2006). Comparing a distance learning and a traditional offering of an autonomous robotics practicum: Lessons and observations. *Proc. ASEE National Conference and Exposition*, Chicago, IL, paper 2063.
- 6 Gallagher, J.C., Bolick, D.S., and R.F. Drushel (2005). Infrastructure supporting an internet-connected mobile robot supporting a WWW-based robotics practicum. *Proc. ASEE North Central Conference*, Ohio Northern University, Ada, OH.
- 7 Gallagher, J.C., and R.F. Drushel (2005). Supporting engineering practica for undergraduates and STEM educators: An online robotics laboratory. Abstract, *NSF Engineering and Computing Education Grantees Meeting*, Washington, DC.
- 8 Gallagher, J.C., Drushel, R.F., and D. Bolick (2005). Increasing accessibility to a first-year engineering course in mobile autonomous robotics. *Proc. ASEE National Conference and Exposition*, Portland, OR, paper 2192.
- 9 Gallagher, J.C., and S. Perretta (2002). WWW autonomous robotics: Enabling wide area access to a computer engineering practicum. *Proc. 33rd ACM technical symposium on computer science education*, ACM Press, pp. 13–17.
- 10 Gallagher, J.C., Perretta, S., and R.F. Drushel (2002). Teaching autonomous robotics over the World Wide Web: An online computer engineering practicum. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* **4** (2). <<http://imej.wfu.edu/articles/2002/2/03/index.asp>>
- 11 Mars Pathfinder home. <<http://mars.jpl.nasa.gov/MPF/index1.html>>
- 12 Mars Pathfinder mission — home page. Historical 1997 website archive from NASA/Jet Propulsion Laboratory. <<http://mars.jpl.nasa.gov/MPF/index0.html>>
- 13 Martin, F.G. (1992). *The 6.270 robot builder's guide*. Media Lab, Massachusetts Institute of Technology, Cambridge, MA.
- 14 Martin, F.G. (2007). *The Handy Board technical reference*. Computer Science Department, University of Massachusetts, Lowell, MA.
- 15 Perretta, S., and J.C. Gallagher (2003). A portable mobile robot simulator for a world wide web robotics practicum. *Proc. ASEE National Conference and Exposition*, Nashville, TN, paper 2327.
- 16 The history of 6.270. <<http://web.mit.edu/6.270/www/about/history.shtml>>
- 17 K-TEAM Corporation (1999) *Khepera user manual*. Version 5.02. Prévelange, Switzerland. <<http://www.k-team.com>>