Mixed Nuts

The Use of Atypical Classroom Techniques in Teaching
Entry Level Computer Science to Non CS-Majors

Sid Stamm
Indiana University
sstamm@indiana.edu

ABSTRACT

AAAAAAAAAAHHHHHHHHHHHHHHHHH!

Didn't expect that, did you? Neither did Steve Wolfman's class when he let loose with a blood-curdling scream on the first day. His goal was to demonstrate the strength of community and pooled resources. This point was clearly illustrated when he asked the entire lecture hall of two hundred students to scream at the top of their lungs.

Not everybody can learn from a CS lecture. In fact, I bet that even students who can get the most out of a lecture grow tired of the same thing every day; perhaps they would be more interested in class if it were taught in a different way or with a different twist.

I believe instructors should use what I call "atypical classroom techniques" in order to effectively capture as much of students' interest as possible, keep their attention, and target their preferred learning styles. Unlike lecturing and giving homework, these techniques are helpful, out-of-the- ordinary, unorthodox, unusual, or uncommon and sometimes even shocking teaching methods or activities. I present some experimental and anecdotal evidence to support my theory that the use of these techniques improves students' learning in an introductory Computer Science class.

1. THE PROBLEM

Students don't always pay attention, they lose interest, they may not learn most effectively the way they are taught, and sometimes they just don't care what's going on. Being an effective teacher is not a simple task; it can, in fact, be quite a burden.

1.1 Interest

One of the complaints I hear from first-year college students who take an entry level computer science course is that the material isn't interesting. Many people studying pedagogical techniques agree that "students who are new to computer science typically find the field full of theoretical, technical, or even tedious concepts" [6].

I believe much of this lack of interest is due to the large number of building blocks (such as procedures, syntax, and algorithms) that are taught to students so that they can apply what they've learned. Since the students are exposed to this dry or dull information at the beginning of their Computer Science (CS) studies, they get bored. It ends up being the instructor's responsibility to keep the students interested.

This difficulty in getting the students inspired to learn is only heightened when they are at the beginning of their CS studies; non-CS majors who have no initial motivation to learn the CS concepts cannot be easily inspired by telling them they will use it later; in many cases the intro-to-CS class is a part of their general education and not within their major.

1.2 Intuitive Concepts not Easily Taught

Lack of applicability is a big turn-off for students and can sometimes even make a course seem pointless, but it is not the only factor that limits their motivation to learn. Many CS concepts can seem exceedingly difficult because the students have not seen anything at all like it before. It often takes students lots of time and exposure to CS concepts in order to get them to the point where they understand and can apply the concept.

In many disciplines, this can be solved by requiring the students to spend lots of their time in the laboratory doing experiments. Experimentation is difficult for entry-level CS studies, however, because in order to apply theoretical concepts as taught in the classroom, the students must know a programming language to apply them. This adds to the frustration of learning a bunch of things they cannot immediately apply. The fact that many of these concepts are labeled with strange or ambiguous names (such as objects, threads, pointers) only adds to the confusion.

1.3 Short Attention Span

In order to adequately expose the students to CS concepts, they need to play with it or get their hands dirty in the proverbial Computer Science dirt; to get them into the "dirt," they have to be taught both the theory and the syntax. With the average attention span of a college freshman dwindling in subjects where they show little interest, the instructor often has to work very hard to keep them interested.

1.3.1 Shifting Gears

Steve Wolfman of the University of Washington says "I feel there is a need to jar people at the beginning of each class to shift them from what they have been doing to what we are going to be doing" [5]. This attention shift is especially important for instructors who teach first year college students; usually these students take classes in many different subjects at the same time and might have difficulty switching modes of thinking. For example, an instructor's effectiveness decreases if some of the students are worrying about a physics exam they took earlier that day. Gaining their attention becomes a significant part of a course when it is taught in short periods of time.

Recently, the computer science department at Rose-Hulman has moved the entry-level CS course from four 50 minute blocks to two 100-minute blocks, because it took the students too long to start up their computers and get shifted into "learn CS" mode. This has proven beneficial, and gives the instructor more time to teach, since the students are spending less of the class period setting up and adjusting to the new subject matter.

1.4 Various Learning Styles

Bonwell and Eison claim that "some cognitive research has shown that a significant number of individuals have learning styles best served by pedagogical techniques other than lecturing" [1]; the lecture/homework approach may not be the optimal teaching method to use in any discipline, and instructors can often lose students' attention during a lecture.

Getting students to say "Aha!" is often a good indication that the instructor will keep their attention–at least for a little while longer. Each student is different, though, and the "Aha!"s can come from completely different parts of a classroom education. A good way to keep the attention of most of the students in any class would therefore be to understand their learning styles, and teach to accommodate the diversity.

1.4.1 Learning Styles

Richard M. Felder of North Carolina State University published in 1996 a paper [4] that describes four different learning style models and classifications of students. He says "If professors teach exclusively in a manner that favors their students' less preferred learning style modes, the students' discomfort level may be great enough to interfere with their learning." In any given class, a professor may have students with a myriad of preferred learning styles. Felder suggests that instructors do what he calls "teaching around the cycle;" he describes this cycling through the different learning styles (described in his paper) and, one at a time, teaching in a way to benefit each style. I agree; concepts should be taught with as many different styles and methods as possible in order to reach a broader audience.

2. THE SOLUTION

So, clearly, the classical lecturing technique is not as effective as it should be. What can be done to make the learning experience better? First, the students need to be interested to learn. Second, concepts should be covered more than once and in different ways. Third, the instructor should try as hard as possible to gain and keep the students' attention. These three goals will help an instructor make an introductory CS class more effective.

2.1 Interest the Students

Get the students to do things they wouldn't do in other classes they take; show CS concepts at work in the real world using unusual stories or jokes. Have the students pretend to be part of a computer. Do something unpredictable to them and make introductory Computer Science fun to learn!

2.2 Reiterate Concepts

I think one of the best ways to learn introductory CS concepts is to be exposed to them over and over–almost to the point where they become repetitive. The obvious problem with this is that they do become repetitive after a while. If an instructor uses many different teaching methods for illustration and explanation the repetition will be much more bearable, since the instructor brings in a new point of view or way of looking at a concept for each method used.

2.3 Keep Their Attention

A good way for instructors to get their students' attention is to shock them. Whenever any of my professors says something off-color or out of the blue, no matter what I was thinking about, my attention instantly focuses to them. Dr. Gary Sherman infrequently says things (such as "If you don't like this, you don't like America") that I completely don't expect, and that draws my attention back from whatever I

was thinking of to what he is doing. Students are fascinated with unpredictability, and are more likely to watch an instructor if they can "expect the unexpected."

3. MY THEORY

By using unusual, uncommon, unorthodox, strange, or atypical classroom techniques, an instructor can help interest students, reiterate concepts without seeming repetitive, gain and even keep their attention in the subject of introductory computer science.

3.1 What is Atypical?

I define an atypical technique to be something unusual, or different from the common style of classroom instruction that consists of lecture, homework, group work, and tests. Examples of atypical techniques include role-playing exercises, giving bonus points for jokes students can tell that are related to the subject, in-class debates about the outcome of a situation, and extended analogy [6].

3.2 For Example

I have gathered some examples of atypical techniques (definitely not all of them) that are currently implemented or described. I have categorized them.

3.2.1 Active Learning

The techniques in this category heavily involve student involvement and often require the students to get up and move around. These put the students "into" the CS concept being taught. These techniques primarily target the students who learn best by doing something instead of those who learn best from watching or listening.

Set up a debate [1]: Break the class into groups or select a few students to take opposing sides of an issue or different solutions to a problem. Have them argue for the accuracy or use of their side.

Role-play [1]: Put students in an activity where they are parts of a computer program, or have similar behaviors to some aspects of CS. They can learn why things behave the way they do, or discover as well as possible benefits for a type of implementation or system.

Simulate algorithms using the class [7]: Use the class as data members in an algorithm or process. Distribute papers based on a set algorithm, or have the students alphabetize themselves.

3.2.2 Lecture Modifications

These techniques are just simple modifications on the delivery or protocol for a lecture. They have some form of shock or entertainment value and are primarily used to get or keep the students attention and interest.

Jar people at the beginning of each class : Steve Wolfman [5] makes a point to do something unpredictable and usually surprising at the beginning of each class, such as scream. Not only does this get the students' attention, but also captures students interest in his classes.

Extended analogy [6]: Make up a story that technically has nothing to do with any CS concept, but somehow allegorically or metaphorically relate it to CS. Such a story could start with "One time my father and I went fishing..." and can be completely fabricated and exaggerated to make a point. Camp, Hooper and Matocha, in their Extended Analogy paper [6], write "an item, exaggerated to the point of silliness, becomes easier to recall."

3.2.3 Supplemental

This topic helps the students tie a CS concept to something else in life (or in this case, pop culture). It is conducted as an addition to the class rather than an alteration of a teaching or lecture style.

Conduct unusual study sessions : Wolfman has been known to conduct a study session in which his Data Structures class watches Star Wars [5]; the students are given bonus points if they can come up with data structures related jokes based on the movie.

4. SOME TRIALS

What better way to test a theory than to actually expose a class of undergraduate freshmen non-CS majors to some of these atypical techniques in their introductory computer science class? I conducted two small experiments to test my theory that atypical classroom techniques would aid in the students' learning.

For my experiment, I concentrated on the technique of role-playing. I feel that this active learning technique should help students get more interested in something outside their major; for the students that have trouble learning from a lecture or from trying to write a program in a language as complex as Java, this should provide an alternate method to understand some CS concepts.

4.1 Setup

I had access to two independent sections of the entry-level CS course (CS120) at Rose-Hulman to use with my experiment. The two sections met at different times and were equal in size at initial enrollment with 30 students each. Almost all of the students had a declared undergraduate major of study outside of computer science: out of the 60 students, only one had declared CS.

The instructors for each section of CS120 (Dr. Cary Laxer and Dr. Andrew Kinley) have taught entry-level courses in the field and each have their own preferred teaching styles.

Since the section that Laxer teaches met earlier than Kinley's, I decided to use Laxer's class as a control variable: this way, students could talk between sections about how the concept was taught, and it would not affect the experiment. I will call his section "section C." I will refer to Kinley's class as "section E," or the experimental section.

4.2 Initial Bias

In a small college like Rose-Hulman, almost every class and professor has a reputation of some sort. I decided to gauge

the students' bias towards the class and professor before they started class. This way, if the two sections had different expectations for the class or professors, I could take that into consideration when reviewing the results of my experiments.

On the first day of class for each section, I passed out a simple anonymous questionnaire asking questions of the students about what they have heard and expect from the professor and the course itself.

According to the anonymous questionnaire, no student had taken a course from their instructor.

Section	Control	Experimental
Course Relevance	4.23	4.17
Subject Interest	3.83	3.66
Instructor Opinion	2	11

Table 1: Initial Student Bias

For the course relevance and subject interest, students were asked to rank how relevant they think the course is and how interested they are in computer programming, each on a scale of one to five, one being lowest and five being most interest or relevance. The numbers in Table 1 are the arithmetic mean of the result set. The results show that students in the control section are slightly more interested in and see the applicability of the course more than the experimental section; on the whole, however, the students are only slightly interested in computer programming.

The students were also asked to state what they have heard from other students about the professor who taught their section. I gave any negative response a negative point (-1), a neutral or no response a zero, and a positive response a positive point ($+1$). The numbers in Table 1 show the sum of the students' opinions; a positive number indicates an overall good opinion. On average, the students in the experimental section had been given a much higher opinion of their professor.

Section	Control	Experimental
C++	27%	24%
Java	2%	3%
C++	57%	66%

Table 2: Student Previous Experience

Finally, the students were asked what languages they knew or had used. Table 2 shows the experience of each section by percentage. A large percentage (roughly a quarter) of the students had been exposed to Java (which is used in the course) or C++, and well over half had been exposed to programming of some variety. The base skill level was pretty close to even in both cases, so I did not expect the students' programming experience to much influence the results of the experiments.

4.2.1 Expected Effect of Bias

The students were decently optimistic towards the course's relevance, and had about the same interest in programming; they were somewhat equally motivated to learn the subject matter, and I thought they should not have been easily discouraged or turned off by a professor doing something out of the ordinary in the classroom.

The students in the experimental section had been given a significantly higher opinion of their instructor which could make them more receptive to an atypical technique. On the other hand, they may have had higher expectations for the professor and could have been less motivated to do the learning on their own; this could lead to a class of spectators that would spend less time learning outside of the class than the other section.

Very few of the students knew Java (the language used to teach the course); and since the percentage that did know Java was so small, this should also not greatly affect the results. Overall, the students in both sections brought a similar amount of programming knowledge and motivation to the class as it started, so the experiment should not be significantly biased as far as motivation and programming skills are concerned.

4.3 Experiment 1: Interfaces

I conducted the first experiment on the concept of Java Interfaces. The control class was taught this concept using the standard lecture and group work method that was used for the rest of the classes. The experiment class was subjected to a simple role-playing exercise to supplement the standard lecture. During the execution of this experiment, Dr. Kinley taught both sections of the course.

4.3.1 Goal Statement

My goal for this experiment was to show, through having the students actively participate in a role-play that illustrated the functionalities of Java Interfaces, they would more easily and fully grasp the concept of interfaces and their uses than a class that was just taught with a lecture and some bookwork.

4.3.2 Execution

On the day where the students were to learn about Java Interfaces, both sections were given an identical quiz that polled questions pertaining to the concepts they were to learn. This pre-lecture quiz contained questions that the class should not have known before the class as well as some they could have been able to extrapolate based on previous homework assignments. The quizzes were marked for accuracy and I calculated the class's overall mean, median and standard deviation of accuracy.

The students were then taught the concept of Java Interfaces as defined in the course curriculum. Dr. Kinley conducted a role-play exercise with the experimental class. The next time the class met, they were given a nearly identical quiz with slightly reorganized questions to poll the same information as the pre-lecture quiz. The quizzes were once again marked for accuracy and I calculated the class's overall mean, median and standard deviation of accuracy.

4.3.3 The Role-Play

One student (we'll call him Jeff) was told to leave the room, and another (we'll call her Jill) was told to come to the front of the room.

Jill was given a list of activities they would "know" how to do if asked. Once they felt comfortable with the list of activities, Jeff was invited back. The instructor told Jeff to ask Jill to do some things: give her instructions.

After a few moments of Jeff asking Jill to do things that she could not do, the instructor explained that this is how life is without Java-style interfaces. Once again, Jeff was told to leave the room.

Jill was this time given a set of things she had to be able to do and what kind of parameters went with each instruction; they were written on the board. Jeff was invited back in and found himself greeted by a list of things that Jill could do as well as what she needed to know to do them.

The instructor asked if it would be easier to tell Jill what to do in the right way this time. As expected, Jeff confirmed it was easier, and found that Jill would react to his instructions in the way he expected. The instructor explained that the writing on the board was a kind of Java-style interface.

4.3.4 Initial Instructor Reaction

Immediately after the classes, I talked to the instructor; according to Kinley, the students were very receptive to the role-play exercise. He said their enthusiasm toward class participation increased when they got to stand up and do something other than sit and listen during class. The mood in the classroom was almost cheerful, and he felt that he had the attention of the entire class through the exercise.

Kinley said that he believed the students in the experimental section learned the concept of Java-style interfaces much more quickly than the control class. He also believed that they were more able to apply their knowledge of interfaces than the class that had not participated in the role-play. He expressed his confidence in the effectiveness of the role-play in getting the students to understand and apply interfaces.

4.3.5 Statistical Results

The statistical results are not as straightforward as a professor's anecdotal confirmation of my hypothesis.

Control Section	Pre-Quiz	Post-Quiz	Change
Mean Accuracy	50%	61%	11%
Median Score	60%	60%	5%
Standard Deviation	33%	15%	30%
Experiment Section			
Mean Accuracy	56%	59%	3%
Median Score	67%	60%	-10%
Standard Deviation	39%	23%	45%

Table 3: Results of the experiment

In short, the control section seemed to show more improvement than the experiment section. Table 3 shows that, on average, the students in the control section improved their scores 11%; the experimental section showed a much smaller mean improvement of 3%. [1]

The experiment section started with half of the scores above 67% but finished with a median of 60%. Seven percent of the students in that class moved below the average, indicating that either the tail end increased (fewer really low scores) or there were more really high scores than before the lesson. Combining this 7% decrease in median with the fact that half of the students' scores decreased significantly (-10% median in change) leads me to conclude that many students did worse than average but fewer students did really horribly. This spread in the spectrum is verified by the large deviation in the change of each student's score. In short, the poor scores aren't as poor, but there are more of them.

The median change for the control section is positive, which suggests that more than half of the students in the class had scores that increased. The mean increased quite a bit (11%) as well; thus, there are more high scores, but fewer perfect scores.

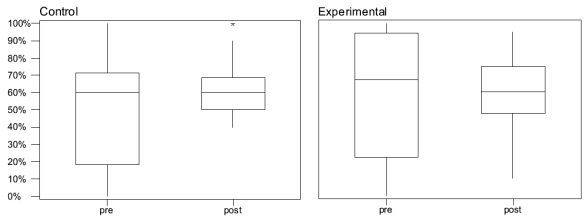

Figure 1: Interfaces Test Result

The standard deviations indicate that both sections scores clumped together much more closely around the mean after the lesson. This signifies that more of the students are ending up with a closer to average mastery of the class; each individual's skill in Java-style interfaces is becoming more "normal."

In summary (and as illustrated in Figure 1), both sections' scores coalesced into closer-knit groups around their averages, but many more students in the experimental section experienced a decrease in scores. In both cases, there is a variance of scores below the average that shrinks significantly after the lesson, but this area in the experimental section started off larger. It seems that the people who were having the most trouble in the experimental section were helped significantly by the role-play, but some people in that section suffered as well.

4.4 Experiment 2: Threads

The second experiment I conducted spanned the concept of threads. The control class was taught this concept, once again, using the standard lecture and group work method

[1] In Table 3, percent change was calculated for each student, the data in this chart reflects the mean, median and standard deviation of the changes, not the change in mean median and standard deviations.

that is standard in the CS120 curriculum. The experiment class was once again subjected to a simple role-playing exercise to supplement the standard classroom routine.

During this experiment, Dr. Laxer taught the control section and Dr. Kinley taught the experimental group.

4.4.1 Goal Statement

Once again, my goal for this experiment was to show how a role-play illustrating the functionalities and behaviors of threads would affect the students' understandings of the concept.

I hoped to show that adding the role-play benefited the students above a standard lecture and group-work exercise.

4.4.2 Execution

On the day where the students were to learn the most about threads, both sections were given an identical quiz about the concepts they were to learn. This time the pre-lecture quiz contained questions that the students may or may not know answers to, since they had already been exposed to threads a small amount in previous sessions. Some of the questions required more in-depth knowledge (or concept mastery) in order to be answered correctly, and I had hoped that such concept mastery would be gained during the session following the pre-lecture quiz. The quizzes were marked and I calculated the class's overall mean, median and standard deviation of accuracy.

During the session when the experimental section met, Dr. Kinley conducted a role-play exercise with the students.

Exactly like in the first experiment, the students were given a nearly identical quiz when the class next met with slightly reorganized questions about the same information as the pre-lecture quiz.

The quizzes were once again marked and I calculated the class's overall mean, median and standard deviation of accuracy.

4.4.3 The Role-Play

This second role-play was conducted in the manner of a live Guess-Who? [2] game: one person would hunt for a specific person by asking questions and eliminating students based on physical traits.

One student was chosen to be the leader of the exercise, I'll call him Jeff. He was told to leave the room while the game was set up.

The instructor chose one student at random, and gave them the "treasure" (in this case it was a Miss Piggy doll) to hide on their person. Once the student had hidden the treasure, the instructor asked the entire class to stand up by their seats.

Jeff was asked to come back in, and was told he needed to find the treasure by process of elimination. The process was to be iterated until only one student remained:

1. Ask a question about the person's appearance. For example, "Is the treasure held by a student with blond hair?"

2. Go through the class, person by person, and tell each person that fit the criteria to sit down and each person that didn't to remain standing.

Once the treasure was found, Jeff was asked to leave the room once again.

The instructor chose three people who would act as Jeff's threads, and they were told to join him outside the room. Again the treasure was hidden on a random student, and the entire class was asked to stand up.

When the treasure was hidden, Jeff and his threads were asked to re-enter the room. He was told to do the same thing, except this time he could dispatch an idle "thread" to sort through the class. This added the option to have multiple people working at once.

Jeff found out the benefits of multitasking and clearly illustrated under- standing of the benefits of having threads in a computer program.

4.4.4 Initial Instructor Reaction

After the role-play class, Kinley told me that the students really enjoyed it. He said that for the first time in a while, the entire class was energized, awake, and learning about threads. The change of pace was refreshing for the students as well as the instructor and everyone was more interested than usual.

Once again, the instructor said he believed the experiment was effective because the students seemed to learn a large amount from the exercise. Not only did it energize them and get them involved, the analogy of dispatching "minions" seemed to help the students understand the purposes, benefits, and functionality of threads.

4.4.5 Stasticial Results

At first glance, Table 4 shows that on the average the experimental section improved more (28% mean increase versus 5%), but at the end of the exercise the control section still had a slightly higher mean score and thus a deeper understanding of threads. In both cases, the standard deviation shrunk to 13%, which indicates that the spread of scores lessened after the class on threads. [2]

The experimental class, however, started with a bigger spread and essentially coalesced more. This would indicate that more of the students moved closer to the mean.

The median score in the control section did not change at all, which implies that the lower scores just moved up a little bit, and the half of the scores that started above the median didn't go anywhere. For that half of the students in the class, this session didn't affect their knowledge of threads

[2] Percent change was calculated for each student, the data in Table 4 reflects the mean, median and standard deviation of the changes, not the change in mean median and standard deviations.

Control Section	Pre-Quiz	Post-Quiz	Change
Mean Accuracy	53%	82%	28%
Median Score	50%	83%	33%
Standard Deviation	23%	13%	28%
Experiment Section			
Mean Accuracy	78%	83%	5%
Median Score	83%	83%	3%
Standard Deviation	16%	13%	17%

Table 4: Results of the experiment

much at all. The median in changes for the control section is very low which indicates that most of the students had little if any change in scores whatsoever.

The median score in the experimental section went up much in the same way that the mean did, which indicates that the increase happened class-wide and not just with a select few of the students. The fact that the standard deviation dropped by about 10% indicates that the benefit, though class-wide, affected mostly the people below the average. Their scores grew a lot closer to the mean.

Overall, the experimental section seems to have clearly made the most progress over one day with the lecture on threads.

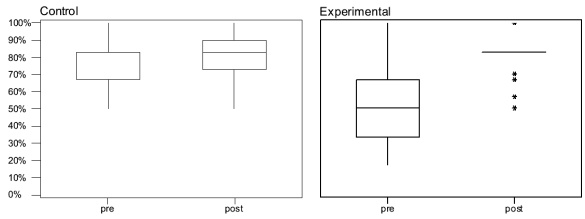

Figure 2: Threads Test Result

This time, two different instructors were teaching this class, which could be why the students in the control section started at a higher level of understanding than the ones in the experimental section: they were just at slightly different places in the course. The experimental section, however, showed significant gains in one session that may have taken the other session two.

Figure 2 shows the significant change that occurred in the experimental section. The wide distribution shrunk to what appears to be a line instead of a box for the post test. Also visible is the climb in low scores: most of the scores grew significantly whereas in the control section, the lower scores stayed low.

5. FURTHER ANALYSIS

There are many different types of atypical techniques; role-playing as exhibited here is only one example. Each technique will have different target learning styles, and will also have a different capacity to entertain students. I think the role-play targeted the active learners in the course who were not learning as effectively due to the high frequency of lecturing. The experimental section in my experiments, although motivated and entertained by the activity, may have not been the most receptive to the style of active learning.

In both experiments, the students who first scored below their section's average (I will call them lower-half students) were able to make significant gains in scores after a role-play. As Figure 3 shows, the lower-half students in the experimental section showed more increase than those in the control section; this shows how the role-play exercises benefit the students who began with less than average mastery of the CS concepts.

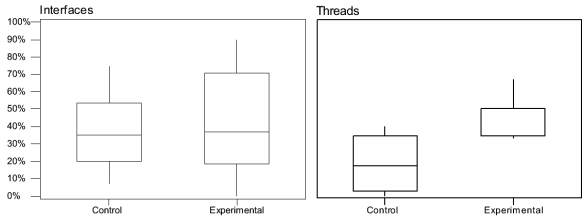

Figure 3: Increases for students who started below average

Not all the students benefit from this however; the average gain in lower-half students is much larger than the average gain for the whole class. This would imply that the students who began above the average (upper-half students) did not benefit from the role-play and perhaps actually suffered from such a concentration on active learning.

This could be one reason active-learning techniques are not implemented more frequently. In their paper on active learning, Bonwell and Eison write: "Perhaps the single greatest barrier of all, however is the fact that faculty members' efforts to employ active learning involve risk–the risks that students will not participate, use higher-order thinking, or learn sufficient content, that faculty members will feel a loss of control, lack necessary skills or be criticized for teaching in unorthodox ways" [1]. My experiment shows that the role-play did not have any obvious negative side-affects, so I say to all the professors out there: give it a shot, it can't hurt!

6. SIGNIFICANCE

So why should professors do unconventional, unorthodox, strange or atypical things in the classroom? There are many reasons, but it all boils down to motivation. If a professor can get students interested and motivated within a discipline, they will learn much more of what an instructor wishes to teach them than if they don't care.

While I only tested one type of atypical technique, there are many different types out there; some of them are active-learning related, some are not.

6.0.6 Gain & Keep Attention

In order to motivate students, the instructor must gain and keep their attention. Without this attention, the students will not learn anything. Atypical classroom techniques can help an instructor gain and keep the attention of students by

shocking them, varying the teaching styles or even by just presenting something unexpected in the classroom setting.

Since each student is interested in something different, it can be difficult to get the attention of the entire class; an instructor must vary methods of teaching so that all the students are interested at least part of the time.

6.0.7 Teach to Various Learning Styles

Atypical techniques can help instructors vary their target teaching styles. By just bringing a role-play into class for my experiments, I added an element of active learning to a lecture that otherwise might have remained passive or reflective. A larger variety of activities conducted in the classroom will enact what Felder calls, "teaching around the cycle" [4].

6.0.8 Reiteration

A good way to keep explaining a concept over and over is to present it in as many different ways as one can. A good example of typical reiteration comes in lecture with supporting homework.

Most mathematics classes include lectures over a concept using simple examples; later, the students are expected to work through some similar problems on their own in order to grind the concepts deeper into their head. Atypical techniques, when used often and as a supplement to the traditional lecturing, can help reiterate concepts to the students.

6.0.9 Immersion

Sometimes the best way to teach students an abstract concept is to immerse them in an example or metaphor. Getting them involved in something like a story, then telling them afterwards how it relates to the subject can spark the "Aha!"s that professors frequently look for.

Lots of concepts in computer science require a large amount of time in order to implement them. Instead of just providing a description of the (intuitive) theory, a concrete example is required to help the students understand; in a situation where there is limited time, a concrete example is not possible. So an instructor should construct a metaphor or other example that can be compared to an implementation. Once the students are involved, they can be asked how they behave or how they feel, and it can be related to attributes of the desired concrete example. Sometimes, this process is referred to as "extended analogy" [6].

6.0.10 For the Instructor

I can imagine that the same exact routine, day after day, semester after semester can become just as boring for an instructor as it can for the students. Atypical techniques can provide relief from the monotonous routine.

They can also provide a way for the students to help direct the class so that the instructor can find out how to teach them. Each class of students will be different and have diverse learning styles, so an approach that works for one section of students may not work quite as well for others.

Using atypical techniques with a capacity for student feedback (such as, in the role-playing exercises, when the students were trying to relate what they were doing to the concepts of threads and interfaces) can help the instructor pinpoint areas of difficulty so that he or she may concentrate on those areas.

7. RELATED WORK

There is a significant amount of work being done and published in the area of atypical classroom techniques used in entry-level college courses, but most of it is not related to computer science education. Being an education-focused institution, Rose-Hulman contains many professors of diverse disciplines who implement these techniques. In my search for some information about their effect, I talked to many professors who did something unusual in the classroom.

7.1 RHIT

Many faculty members at Rose-Hulman Institute of Technology regularly use atypical classroom techniques. Not only are they employed in the Computer Science and Software Engineering department, but also in many other departments throughout the institute.

7.1.1 In the CSSE Department

Dr. Andy Kinley

Assistant Professor of Computer Science

In the spring of 2001, I took an operating systems course from Andy Kinley. I remember the course well because he did strange things in the class every so often: he would sometimes bring in some Miss-Piggy dolls and illustrate a concept, or have the entire class participate in a powers-of-two hot-potato game. He claims to invent things like this off-the-cuff and uses new techniques as often as he can so the students learn to "expect the unexpected" and have some fun in the class. Kinley keeps the students on their toes by giving class exercises a twist: for example, he will often assign groups competing tasks and give them country names ("you be the US, you be Iraq. If the US is a multitasking kernel...").

The most interesting thing he did, however, was to discourage giving anyone (or receiving from anyone) unearned respect. He started the class off informing us that he expected to be treated as a peer, and that we should only give him our respect as a professional once he's earned it.

Many people in the class were shocked by this, but it broke down the wall between student and professor, providing a more colloquial and discussion- oriented classroom.

I asked him why he did atypical things, and he had lots of reasons: they're fun, it builds an instructor-student relationship that helps to alleviate the fear of asking for help. He also uses as many analogies and metaphors ("almost exclusively" he says) to describe concepts in many different ways. When I asked him why, he said, "You can't learn everything brand new;" things must be related to stuff you understand.

Dr. Laurence Merkle

Assistant Professor of Computer Science

Talking briefly with Merkle, I learned he does something unusual in the classroom: he puts obviously and intentionally wrong information into his lecture. Generally fairly traditional, this stirs up the class and gets them to pay attention to see when he will next "screw up." He says he gets a kick out of the reactions students have to his intended mistakes; some laugh, some ask questions, some correct him–overall, he says it gets students to listen and participate.

Merkle involves a lot of physical activity if he can. One exercise he uses often in a data structures class illustrates binary trees by having all the students stand up and point at two other people. He does this for the same reason: to get the students involved.

Another activity he conducts is arguments. If people in class have different answers to a question he asks, he helps develop the conflict into a full- fledged argument until the class figures out the right answer. This way, the students can discover something on their own without being fed directly.

7.1.2 Outside the Department

Dr. Caroline Carvill

Professor of American Literature

One thing that stands out about Carvill's teaching practices is her tendency to give students responsibility to teach the class. Often she will assign a few students to lead a class discussion, present a topic, or teach the class for the day. She says this keeps them involved in the subject to the point of deep thinking.

Carvill concentrates on two things that are unusual for a literature class when teaching: startling the students, and digressing. She says she "loves a little digression" because it keeps the students' attention. Realizing that the students' attention span is short, she allows the class discussion or lecture to wander wherever the students want, so long as it comes back. She says that "varying the discussion is good" and if she can get the students to laugh, that means that not only are they paying attention, but they are also interested.

Recently, I had the opportunity to help Dr. Carvill out with a good atypical technique. She wanted to convey the concept of literary point-of-view to the students (for example: first person, third person, omniscient, etc.) but she also wanted to show how different people will have a different point of view towards an event.

To illustrate this, she arranged for me to burst into her classroom on the first Friday of the term and pretend to be taking the class. With the students completely uninformed of our plan, I simply walked into class late and sat down in the front. We had a quick discussion about whether or not I should be allowed to join in late for the already-full course, and I left to go "talk to the registrar."

She then explained to the class that they had witnessed an objective view of an event. She and I then proceeded to re-enact the scene in two more ways: one from my point of view, and one from hers. In my point of view, she behaved completely unreasonably; in hers, I was the student from hell.

Within this activity, we made a point to be as humorous as possible so that the students would remember the exercise on point-of-view. I know I will never forget calling her (in jest of course) a "Faulkner-loving, flaming-liberal, flower-child, femmi-nazi from the South."

She claims to do all this to keep a "relaxed but non-chaotic environment." By riding the fine line between relaxation and chaos in the classroom, she keeps the learning experience fun and quite fast-paced.

Her techniques have possible applications in the CS field, since they are not restricted to their subject matter; Kinley does similar things in the classroom, and he agrees they keep the students interested.

Hearsay

Dr. Arthur Western has been known to throw candy bars at people who ask questions. He's told me about other faculty who tell off-topic stories on Fridays, use hand puppets, bring in an autoharp, or draw cartoons on the whiteboard. He talks about gimmicks (like putting gold stars on high-scoring tests) and the importance of pausing a lot to let things soak in.

One thing that is common among all the professors I talked to is their desire to use metaphor. I believe that Kinley hit the bulls-eye when he said, "You can't learn everything brand new."

7.2 The Larger Professional Community

So there are lots of cases of professors using non-traditional classroom techniques at Rose-Hulman; but what are people doing in the rest of the professional community?

7.2.1 Learning Styles

Students have many different preferred ways of learning stuff. Richard M. Felder writes about the classifications and models for learning styles in his 1996 paper, "Matters of Style" [4]. He explains that "if professors teach exclusively in a manner that favors their students' less preferred learning style modes, the students' discomfort level may be great enough to interfere with their learning."

Felder then presents four different models of learning styles, along with descriptions of how to "teach around the cycle." These models support my theory that people each have their own preferred methods of learning and that simply lecturing may not be enough to teach all students. The learning styles models are heavily used (and encouraged for use) at Rose-Hulman, since results are quite apparent when an instructor implements various styles; I have participated in classes where the instructor teaches "around the cycle," and have reaped the benefits of a multi-style learning environment.

7.2.2 Steve Wolfman

Wolfman's ideas are being heard loud and clear (literally) at the University of Washington. An article published in University Week [5] illustrated the effectiveness of his atypical classroom techniques. There he is recognized for "[his] own

blood-curdling scream the first day of a lecture class of more than 200 students as he explained that large classes can do some things smaller groups can't. To illustrate the point, he then instructed his TA's to shriek in unison–a bit louder than his own shout. Then he had the class stand and do the same."

He writes about this event as well as his use of role-playing in his study of the advantages of large lecture halls entitled "Making Lemonade" [7]. With a large classroom, he is able to simulate algorithms, having the students pretend they are data; "the large class size made demonstrations of the efficiency of algorithms compelling."

According to the U Week article, during a study session where they watched Star Wars, "the students could earn bonus points if they thought up a data structures joke based on the movie" [5]. Wolfman goes on to explain that he thought it seemed kind of "silly in one sense" but that it helped the students to take their newly gained knowledge to a deeper level and apply it to something with which it would normally not be associated. This allows the students to understand the concept more deeply and not just "compartmentalize" their knowledge.

The University Week article then goes on to explain that "such strategies are a part of the unorthodox approach that makes Wolfman's classes so memorable — and effective — according to students and professors in the department." He is a clear-cut example of the success of atypical classroom techniques in computer science education.

7.2.3 Active Learning

In 1991 Charles C. Bonwell and James A. Eison conducted an extensive study of Active Learning. They reaffirm my belief that, although "traditional lecture methods, in which professors talk and students listen, dominate college and university classrooms," [1] it doesn't mean that students' learning styles are being catered to enough. According to their paper, "some cognitive research has shown that a significant number of individuals have learning styles best served by pedagogical techniques other than lecturing:" atypical techniques would help. This article discusses in depth what active learning is and how to implement active techniques.

The authors also explore why professors are reluctant to implement new styles (because there is too much risk involved) and the reasons for a lack of empirical research data to support contemporary uses of active learning.

Bonwell and Eison explore one of the large fields that I put inside the area of "atypical techniques," or things that are not as common in education as lecture and homework are. They show how useful active learning can be as well as how easy it can be to implement, encouraging professors to get students involved in the classroom and not simply turn learning into a spectator sport.

7.2.4 English Classes

Along the same lines as active learning, Richard Buckland's study "Can we improve teaching in Computer Science by looking at how English is taught?" [3] implies that an interactive or discussion-oriented classroom will benefit CS students, especially women. His study shows that students enjoyed it too:

> "I feel that not only was the more interactive and caring humanities style approach well received in general, it was particularly effective for our female students. Computer science traditionally has problems attracting and retaining female students. Perhaps this teaching approach can serve a role in changing this pattern" [3]

7.2.5 Extended Analogy

Jeff Matocha, Tracy Camp and Ralph Hooper take analogy and metaphor to another level in "Extended Analogy: An Alternative Lecture Method" [6]. They discuss using a story that is topically unrelated to the pertinent subject in order to get the students to understand at a higher level. Just as Kinley says that people can't learn everything new, this draws out a metaphor to a longer story; and it also draws students into the story, capturing their interest. They justify such an extreme use of an off-topic story since "an item, exaggerated to the point of silliness, becomes easier to recall" [6]. In direct support to my theory about atypical techniques, the authors say that "in order to aid students' understanding and retention of technical concepts, instructors must devise and utilize unorthodox teaching methods."

8. STRINGING IT ALL TOGETHER...

In summary, there are lots of other studies and papers that back up my theory that atypical, and specifically non-lecture techniques will help out students who are learning introductory computer science. Some computer science concepts are tough to learn without being submerged inside.

Sometimes learning a concept requires a long span of attention that some students are not willing to give. Some students simply learn in ways that others do not. Though the students may find CS dry, the instructors can spice up a class and make it more effective by doing something shocking, obtuse, strange, unorthodox, or simply atypical to motivate them and then get, keep, and use the students' attention.

9. ACKNOWLEDGMENTS

Many people were involved with helping me construct this paper. The long list of people includes, but is not limited to the following.

- Dr. Mark A. Ardis helped me construct a plan for my research and writing, and provided direction to me as a new student in the field of CSE.

- Dr. Andrew Kinley helped me construct and implement my experiments on the students in CS120.
- Dr. Cary Laxer allowed me to conduct quizzes in his classroom, and provided support and feedback for my experiments.
- Dr. Gwen Lee-Thomas helped me to design and think through the implementation and evaluation of my experiments.
- Steve Wolfman helped point me to good sources of information on Computer Science Education, and provided me with a prime example of implementing atypical techniques in CS classes.

The following people provided me with some time to interview them about their teaching techniques and general classroom practice:

- Dr. Caroline Carvill
- Dr. David Finn
- Dr. Andy Kinley
- Dr. Daniel Morris
- Dr. Laurence Merkle
- Dr. Mark Yoder
- Dr. Arthur Western

Dr. Gary Sherman, Dr. Carvill, Dr. Kinley and other professors provided me with the inspiration to see how atypical techniques would affect CS education.

10. REFERENCES

[1] C. C. Bonwell and J. A. Eison. *Active Learning: Creating Excitement in the Classroom*. ERIC Clearinghouse on Higher Education, 1991.

[2] M. Bradley. Guess who?

[3] R. Buckland. Can we improve teaching in computer science by looking at how english is taugh? In *Proceedings of the second Australasian conference on Computer science education*, pages 155–162. ACM Press, 1996.

[4] R. M. Felder. Matters of style. *ASEE Prism*, 6(4):18–23, December 1996.

[5] R. Harrill. University of washington recognition awards 2001-02, 2002.

[6] J. Matocha, T. Camp, and R. Hooper. Extended analogy: an alternative lecture method. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, pages 262–266. ACM Press, 1998.

[7] S. A. Wolfman. Making lemonade: Exploring the bright side of large lecture classes. *ACM SIGCSE Bulletin*, pages 257–261, 2002.