The Teacher's Dilemma: A game-based approach for motivating appropriate challenge among peers

A Dissertation

Presented to
The Faculty of the Graduate School of Arts and Sciences
Brandeis University
Computer Science
Prof. Jordan B. Pollack, Dept. of Computer Science, Advisor

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

by
Ari Bader-Natal
May, 2008

This dissertation, directed and approved by Ari Bader-Natal's committee, has been accepted and approved by the Graduate Faculty of Brandeis University in partial fulfillment of the requirements for the degree of:

DOCTOR OF PHILOSOPHY

Adam B. Jaffe, Dean of Arts and Sciences

Dissertation Committee:

Prof. Jordan B. Pollack, Dept. of Computer Science, Chair

Prof. Timothy J. Hickey, Dept. of Computer Science

Prof. James Pustejovsky, Dept. of Computer Science

Prof. Jack Mostow, School of Computer Science, Carnegie Mellon University

In memory of my mother Karen

and my grandparents Jackie and Bob

Acknowledgments

I wish to acknowledge the many people who have helped and supported me throughout the past several years.

First, to Jordan Pollack, who envisioned this project years ago and has provided me with continuous encouragement and support ever since. I thank him for this, and also thank him for assembling a wonderful group of researchers. My colleagues within the DEMO Lab created a dynamic environment in which to work: Anthony Bucci, Keki Burjorjee, Paul Chiusano, Sevan Ficici, Kristian Kime, John Rieffel, Shivakumar Viswanathan. Particular thanks to Shivakumar Viswanathan for many hours of engaging discussions and critical feedback about this work.

The BEEweb project benefitted from the hard work of many individuals. Kristian Kime, Max Ekstrom, and Ann Marion initially created the system and set it on its course, and Peter Macko has since added much additional functionality to it. Boris Kerzner, Tien Hoang, Daniel Ivanov, Aleksey Mafusalov, Tapiwa Mushove, Zachi Klopman, Marina Virnik, and Abe Winograd are among the those who have contributed to the development of the various BEEweb activities. And many thanks to all of the students and teachers who have used the SpellBEE and BEEweb activities in their free time and in their classrooms.

For the SpellBEE project, thanks to Don McCabe and the AVKO Educational Re-

search Foundation for providing us with a digitized copy of Greene's study data, and thanks to Carina Canaan for reading aloud and recording several hundred sentences from *The Little Prince*.

Portions of the research in this dissertation were made possible by funding from National Science Foundation (NSF REC #0113317), the Hewlett Foundation (Grant #2003-8966), and the Spencer Foundation (via research grants awarded through the Discipline-Based Scholarship in Education Initiative at Brandeis University).

Special thanks to my dissertation examining committee: Jordan Pollack, Tim Hickey, James Pustejovsky, and Jack Mostow, who have generously offered their time and their feedback over the past few months. This work has benefitted particularly from Jack Mostow's careful reading and thoughtful comments on earlier drafts.

Finally, many thanks to my friends and family for their love and support. To my father, Sam, who always offers words of encouragement. To my sister, Danya, from whom I learned a fair share. To Patty, to Matt and Stef, and to Susan and Stanley, all of whom have helped me in various ways to reach this point. Finally, to Liz, who has been incredibly supportive and understanding throughout this entire process.

Abstract

The Teacher's Dilemma: A game-based approach for motivating appropriate challenge among peers

A dissertation presented to the Faculty of
the Graduate School of Arts and Sciences of
Brandeis University, Waltham, Massachusetts

by Ari Bader-Natal

In classroom-based studies, peer tutoring has proved to be an effective learning strategy, both for the tutees and for their peer tutors. Today, the increasingly widespread availability of computers and internet access in the homes and after-school programs of students offers a new venue for peer learning. In seeking to translate the successes of peer-assisted learning from the classroom to the Internet, one major hurdle to overcome is that of motivation. When teachers are no longer supervising student activity and when participation itself becomes voluntary, peer tutoring protocols may stop being educationally productive. In order to successfully leverage these peer interactions, we must find a way to facilitate and motivate learning among a group of unsupervised peers. In this dissertation, we respond to this challenge by reconceptualizing the interactions among peers within the context of a different medium: that of games. In designing a peer tutoring experience as a two-player game, we gain a valuable set of tools and techniques for affecting student participation, engagement, goals, and strategies.

Our contributions: 1) We define a criteria for games – the Teacher's Dilemma criteria – that motivates players to challenge one another with problems of appropriate difficulty; 2) We show three games that satisfy the Teacher's Dilemma criteria when played by rational players under idealized conditions; 3) We demonstrate, using computer simulations of strategic dynamics, that game-play will converge towards meeting these criteria, through time, under more realistic conditions; 4) We design a suite of software that incorporates a Teacher's Dilemma game into several web-based activities for different learning domains; 5) We collect data from thousands of students using these activities, and examine how the games actually affected the game-play strategy and learning among these students.

The game-theoretic analysis establishes the possibility for a game-based mechanism for motivating appropriate challenges, the simulations support the plausibility of this approach given non-optimal players, the implemented software systems demonstrate the scalability of this model, and the data analysis supports the real-world applicability of this game-based approach to motivating appropriate challenges for learning among unsupervised peers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

When a well-intentioned parent divides a dessert into two pieces for two children, it is rare for both to find the split fair, no matter how even-handed the effort. For the children, a much more satisfactory solution – and an entry point into the mathematical literature of cake-cutting algorithms [69, 76] – is one in which one cuts the dessert and the other chooses from among the two pieces. Viewed as a game, the first "player" to act is motivated to adopt a cutting strategy of maximizing the size of the smaller piece, and the second "player" is motivated to adopt a choosing strategy of selecting the piece they deem preferable. The children arrive at a solution that they both find fair, and neither is envious of the other's piece.

Once the children finish eating, they run off to another room to play some games online. In one of these games, the ESP Game [79, 78], they race to describe the contents of an image using the same descriptive words as their unseen and unknown partner does, playing from elsewhere in the world. The rules and rewards in this game result in a different desirable outcome: given the independence of the players, any agreements between them suggests that the agreed-upon descriptions are likely to be accurate. The children enjoy playing, and the accurate descriptive image labels are of value to the game's creators.

Now it is time for the children to study. Can we, in the same spirit as the cake cutting game for fair division and the description matching game for accurate labeling, construct a game in which the children provide one another with challenges that stimulate learning? This thesis explores the idea of a game-based approach to peer-assisted learning, and argues that an "appropriate challenge" game for study is, indeed, possible.

Towards this end, we motivate and describe a set of game criteria, which we call the *Teacher's Dilemma* criteria. We claim that games that meet these criteria have the desirable property of motivating participating players to identify and provide one another with challenges of appropriate difficulty for learning. We will support this claim with game-theoretic analysis, computer simulation of repeated play dynamics, and the examination of implemented web-based activities building on Teacher's Dilemma games.

1.1 Context and Prior Work

Benjamin Bloom [11] observed that the summative achievement scores of the average student under one-to-one tutoring is, in certain domains, two standard deviations above that of the average student under conventional group instruction. Noting the prohibitively high cost of providing one-to-one tutoring for all students, Bloom challenged researchers to devise practical methods that generate this level of achievement without the associated high cost of personal tutoring. His "2 sigma" challenge provides a context for exploring two bodies of research upon which we build: *peer tutoring* and *intelligent tutoring*. We briefly review both of these areas, and then discuss some recent efforts to combine these two approaches.

In *peer tutoring*, students are grouped, often in dyads, and are tasked with providing some form of support to their fellow learners. By providing these peers with structured supports, in the form of a protocol or script to follow, the students may more effectively assist one another in learning in the classroom setting. Researchers and practitioners define and implement these interactions in a variety of ways, and apply them to a wide range of contexts. In some cases, peers are of the same age, while in other cases, tutoring occurs cross-age. The value of peer-driven learning has been explored and validated in many classroom-based studies. In their meta-analysis of 65 of these published studies, Cohen, Kulik, and Kulik [21] found tutoring programs to have positive effects both on the tutees and on the peer tutors. Some of the work on peer tutoring focuses on the cooperative nature of collaboration (as opposed to a competitive or individualistic dynamic), including that of Johnson and Johnson [42, 44, 43] and Slavin [74, 75]. Other work focuses on introducing and analyzing new protocols for reciprocal tutoring among learners in various classroom settings, including ClassWide Peer Tutoring (CWPT), Peer-Assisted Learning Strategies (PALS), Classwide Student Tutoring Teams (CSTT), and START protocols, as discussed and compared by Maheady, Mallette, and Harper [55]. The various protocols explored by Fantuzzo et al. [28], Greenwood [34], King [46, 47, 48] and the collection edited by O'Donnell and King [64] sketch out a wide range of approaches to the over-arching vision of engaging students as naturalistic tutors for one other.

While peer tutoring seeks to support the learner through the individualized attention of a peer, *intelligent tutoring* systems (ITS) seek to support the learner using an interactive computer program. The goal of much of the research in the field of intelligent tutoring system, beginning with SCHOLAR in 1970 [13], has been to build a software-based tutor with sufficient artificial intelligence (AI) to effectively help the student learn. VanLehn [77] describes tutoring systems as consisting of an "outer loop" and an "inner loop", the former concerned with planning the sequence of problems to pose, and the latter concerned with providing scaffolding support to the student in the form of hinting and feedback. Both sets of activities are ideally tailored to each individual student, based on models of the student and the learning task domain. The variety of artificial intelligence techniques used to construct, revise, and apply these models is as wide as the field of AI itself, including production rule-based systems [2, 3], Bayesian networks [56, 17], statistical models [41, 22], and clustering methods [7], among many others.

One significant draw for the intelligent tutoring approach is that, once a good tutor has been developed, the marginal cost for supporting additional students is very low. This makes it attractive for its affordability and scalability. Unfortunately, the initial cost to develop an effective tutor is quite high, both in terms of money, time, and expertise, and the resulting software may be highly domain-dependent. Given the relative ease in developing an effective (classroom-based) peer tutoring environment, some ITS researchers have explored hybrid approaches designed to combine useful aspects of peer tutoring into an intelligent tutoring system. Three other recent research efforts can be viewed as such, also. Walker and colleagues have explored techniques for, and the effects of, incorporating peer tutoring into a cognitive tutoring architecture [82, 81, 58, 57]. Kumar, McCalla, and Greer [51] adopted a "human-in-the-loop" design in their peer help network. Chan and colleagues have explored tutoring interactions beyond the computer-tutor human-tutee standard paradigm in ITS research, including peer tutoring approaches [16, 14, 86, 15] and synchronous game-based approaches [18, 87]. Our own approach also lies at this intersection of intelligent tutoring and peer tutoring.

1.2 Outline

We seek to draw on the scalability of the computer-based systems and on the relative simplicity of supporting learning among peers in the systems that we build. Beyond representing a useful combination of these existing approaches, the hybrid approach enables a new possibility for peer tutoring, since network-based software allows the peers to interact beyond classroom walls. The increasingly widespread availability of computers with internet access – at home and in after-school programs – offers a new venue for peer-assisted learning. But when teachers are no longer supervising and when participation becomes voluntary, the peer tutoring protocols may stop being educationally productive (or may stop entirely). In order to leverage this new venue successfully, we must find a way to facilitate and motivate a learning environment among an internet-connected group of peers. One effective tool for structuring interactions and influencing behavior is through the design of formal games [80]. We propose that a formal game can overlay a peer tutoring protocol, and that the combination tutoring-game can effectively provide a learning-conducive environment for a network of unsupervised peers. *We suggest that this tutoring-game can form the basis for a "lightweight" intelligent tutoring system, in which the intelligence originates not from domain experts, but rather from the knowledge, beliefs, and common-sense reasoning of the participating peers.*

In Chapter 2, we introduce a model of appropriate challenge as an indicator of the student's probability of learning. We then define the Teacher's Dilemma criteria for games that, if met, motivate appropriate challenges. Among these criteria are the requirements that the dominant strategy for a rational tutee must be to provide a best-effort response to any challenge posed, and that the dominant strategy for a rational tutor must be to seek to identify and pose appropriate challenges given their tutee's abilities. Rather than using zero-sum games (in which one player's success is necessarily another's failure), we decouple the motivational structure for peer tutors and their peer tutees. We motivate the tutor to identify and pose problems at the cusp of the student's abilities. This includes both difficult problems that they believe their tutee may be capable of solving and simple problems that they believe the student cannot solve (even though they ought to be able to). We motivate the tutee to provide her best effort in responding, and reward her based on performance. Many different games can fit this profile, and we detail three of them. Each is presented and analyzed as an formal game based on game-theoretic assumptions. We provide proofs that each meets the Teacher's Dilemma criteria.

In Chapter 3, we illustrate, via computer simulations, that symmetric repeated play of one of these games converges to player strategies in which the tutor poses challenges of appropriate difficulty for their tutee, and the tutee replies with a best-effort response, consistent with the Teacher's Dilemma criteria.

In Chapter 4, we describe two software systems that we have built in order to enable students to participate in Teacher's Dilemma games across the internet. Both systems build on a common interaction paradigm illustrated in Figure 1.1. The Spell-BEE system represents our first attempt to build a web-based activity based on a Teacher's Dilemma game, applied to the task domain of American-English spelling. The activity is currently accessible online at http://SpellBEE.org/. Since we publicly released it four years ago, we have accumulated data from over 25,000 completed fourteen-question matches. The BEEweb, a subsequent system, expanded on this model and provided a more general scalable platform for game-based learning activities. A growing suite of activities are based on this model. We discuss the design goals and implementation decisions involved in constructing both of these systems.

In Chapter 5, we summarize and analyze the data collected over the past four years from thousand of students who have actively used learning activities built on the SpellBEE and BEEweb systems. We present statistics about the usage of these systems, providing an overall picture of how much data has been collected, who participates, and for how long. Over 14,000 people have actively participated in the SpellBEE activity alone, posing and responding to over 400,000 challenge problems. Based on this participation, we pose four research questions. First, we explore a core assumption of our model, that a game can be used to affect how peer tutors select challenges, by asking: "Does the game's payoff structure significantly affect the challenge selection strategies of tutors?" Second, we examine the collective student-modeling ability of the tutors in predicting the probability of a correct response from their tutees by asking: "How does the predictive performance of tutors, on the aggregate, compare to known difficulty-based performance expectations?" Third, we ask whether tutor or tutee grade levels (as a rough indicator of ability) affect the level of difficulty of the challenges posed: "Do the main effects of tutor grade or tutee grade (or the interaction effect of both) significantly affect the difficulty level of challenges posed?" Fourth, we examine if and where tutees improve at the task domain with use of our systems: "Does the response accuracy of tutees collectively improve with use of the system?" These four questions provide an empirical basis for evaluating the effectiveness of our web-based systems built on a Teacher's Dilemma game.

In Appendix A, we discuss an alternative system design for supporting Teacher's

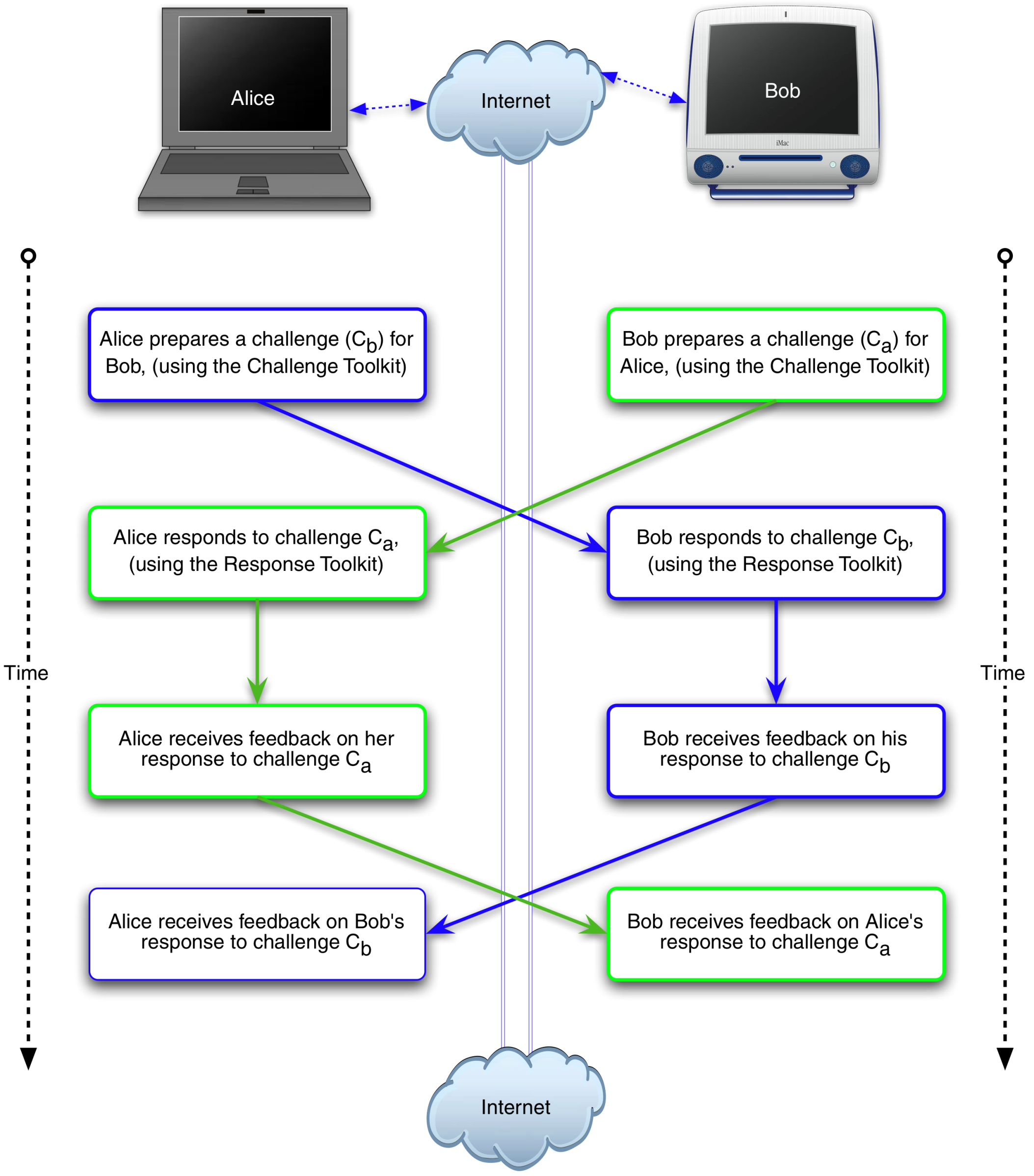

Alice
Internet
Bob
iMac
Alice prepares a challenge (C_b) for Bob, (using the Challenge Toolkit)
Bob prepares a challenge (C_a) for Alice, (using the Challenge Toolkit)
Alice responds to challenge C_a, (using the Response Toolkit)
Bob responds to challenge C_b, (using the Response Toolkit)
Time
Time
Alice receives feedback on her response to challenge C_a
Bob receives feedback on his response to challenge C_b
Alice receives feedback on Bob's response to challenge C_b
Bob receives feedback on Alice's response to challenge C_a
Internet

Figure 1.1: A web-based reciprocal tutoring activity, created by interleaving the steps of two instances of a Teacher's Dilemma game.

Dilemma games, dubbed “BEEmail.” This minimal proof-of-concept was designed to show that a Teacher’s Dilemma game can be built on a decentralized architecture, allowing it to transcend the scalability constraints of systems built around a centralized server.

In Appendix B, we detail the construction and contents of two data sets collected from the SpellBEE system that we are releasing publicly to the research community. In these data sets, we collect, categorize, and count spelling errors. The sets are based on two different techniques for formatting and labeling the data, which we presume to be appropriate for different types of applications. The first set details 99,498 instances of 44,450 misspellings of 2,984 English words, filtered by error type. The second set details 102,181 instances of 18,150 misspellings of 2,764 English words, filtered by frequency. Sample data (for one word) is included for each, and the data sets, in their entirety, are available online at http://www.cs.brandeis.edu/~ari/dissertation/.

Figure 1.2 provides an outline of the Teacher’s Dilemma games, system architectures, and learning activities that are introduced and discussed in this dissertation.

1.3 Contributions

We summarize the main contributions of this thesis as follows:

1. We define a criteria for games – the Teacher’s Dilemma criteria – that motivates players to challenge one another with problems of appropriate difficulty;

2. We show three games that satisfy the Teacher’s Dilemma criteria when played

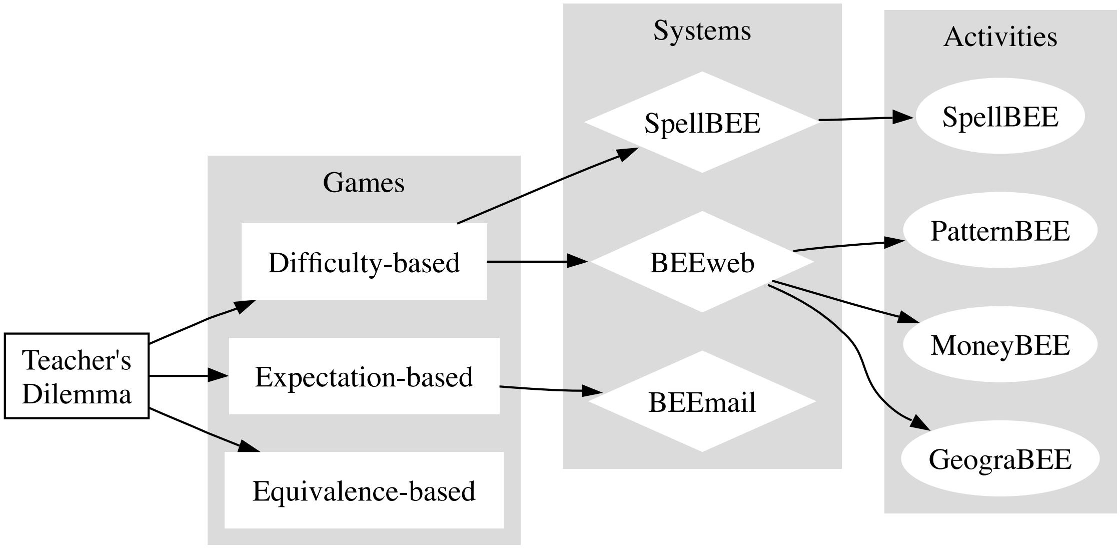


Figure 1.2: An outline of the Teacher's Dilemma games, system architectures, and learning activities introduced and discussed in subsequent Chapters.

once among rational players under idealized conditions;

3. We demonstrate, using computer simulations of strategic dynamics, that game-play will converge towards meeting these criteria, through time, under more realistic conditions;

4. We design a suite of software that incorporates a Teacher's Dilemma game into several web-based activities for different learning domains;

5. We collect data from thousands of students using these activities, and examine how the games actually affected the game-play strategy and learning among these students.

Our game-theoretic analysis establishes the possibility for a game-based mechanism for motivating appropriate challenges, the simulations support the plausibility of this approach given non-optimal players, the implemented software systems demonstrate the scalability of this model, and the data analysis supports the real-world applicability of this game-based approach to motivating appropriate challenges for learning among unsupervised peers.

Chapter 2

Games as a mechanism for learning

In seeking to translate the successes of classroom-based peer-assisted learning from the classroom to the Internet, one major hurdle to overcome is that of motivation. When teachers are no longer supervising student activity and when participation itself becomes voluntary, peer tutoring protocols may stop being educationally productive. In order to successfully leverage this out-of-classroom venue, we must find a way to facilitate and motivate a learning environment solely among a group of peers. We respond to this challenge by viewing the interaction among peers from the context of a different medium: that of games. In re-conceptualizing the peer tutoring experience as a two-player game, we gain a valuable set of tools and techniques for affecting student participation, engagement, goals, and strategies.

While there are many different ideas about what constitutes a “game,” we draw on two in particular: The first is the mathematical concept of games, developed within the field of game theory [80]. We draw on this for such concepts as dominant strategies, expected utility, and rationality, which together offer a means for reasoning about how the structure and payoffs of a game relate to the strategies that players will adopt. The second topic that we draw on, digital games, is less precisely defined, but is perhaps more familiar to students. Kirriemuir & McFarlane suggest that computer games have become the most frequently used interactive media among children [49]. The immense popularity of some of these games reflects that the medium can offer remarkably engaging experiences [50]. We suggest that by drawing on and combining both of these types of games, a web-based peer tutoring activity can provide sufficient motivational and strategic structure to offer a viable learning environment.

In this chapter, we develop a basis for constructing games for learning. We begin by introducing a probabilistic model of appropriateness for problem-solving challenges. We then describe a set of criteria that define the class of *Teacher's Dilemma* games, each of which provides the learner with challenges of appropriate difficulty for practice. We construct three formal games, and prove that, under standard game-theoretic assumptions, each satisfies the Teacher's Dilemma criteria. As a whole, this chapter offers a theoretical basis for constructing and analyzing games as a mechanism for motivating appropriate challenge among peers.

2.1 Appropriate Challenge

While there are many different ways that a software system can aid student learning, we examine one in particular: a system can challenge the student with problems to solve, provide the opportunity to respond, and offer useful feedback on performance. All such challenge-response-feedback experience is not equally valuable, however, as both the opportunity for, and likelihood of, learning can vary by student and by problem. We will introduce *appropriateness* as a term that quantifies the likelihood of the student learning from a given problem. Challenges for which the student is almost certain to respond incorrectly are minimally appropriate, because, while the skills involved are most likely not yet known (which offers room for growth), the likelihood that the student will be able to learn from the feedback provided is low. Similarly, a challenge that the student is almost certain to solve correctly is also minimally appropriate because, while the likelihood of the student being able to effectively understand and learn from the feedback provided may be high, the skills involved are most likely already mastered, leaving no opportunity for learning. More appropriate challenges, on the other hand, test skills that the student has not yet mastered, but are able to master with some feedback and practice. Our definition of challenge appropriateness is a response to the *Meta-Game of Learning* (MGL) concept suggested by Pollack & Blair [65], and explored further by Blair [10], Davies and Sklar [25], and Sklar and Parsons [73]. We define appropriateness probabilistically, as described in the following section.

2.1.1 Defining Appropriate Challenge

In quantifying learning opportunities, we seek to capture the probability that, from a particular challenge-response-feedback sequence, the student will learn the skills required to solve that challenge (and others like it.) We begin by making two observations about learning, and then construct an operational definition of appropriate challenge based on a probabilistic interpretation of these observations. First, we note that a student can only learn what they do not already know. Second, we note that a student is more likely to successfully internalize (i.e. learn from) feedback on easier problems than on harder problems. For learning to occur, the challenge must be one that the student does not already know but is able to successfully internalize based on the feedback provided. Specifically, we can use the product rule to frame this relationship. Letting $\mathrm{P}(A)$ represent the probability that the student does not yet know the skills involved, and letting $\mathrm{P}(B)$ represent the probability that the student is able to learn from the feedback provided, we can define challenge appropriateness as:

$$APPR_s(c) = \mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B|A) \tag{2.1.1}$$

A very simple model of appropriateness can be derived by representing $\mathrm{P}(A)$ and $\mathrm{P}(B|A)$ each in terms of a single variable: the likelihood that the student is able to provide an accurate response to the challenge posed. For student s, given some challenge c and response r, we let $\mathcal{A}_{r,c}$ denote the accuracy of the student's response to the challenge (where $\mathcal{A}_{r,c} = 1$ denotes a correct response and $\mathcal{A}_{r,c} = 0$ denotes an incorrect response.) We define $\mathrm{P}(A)$, the probability that the student does not yet know how to solve a challenge, as $(1 - \mathrm{P}[\mathcal{A}_{r,c}])$, the probability that the student provides an incorrect response to that challenge. We define $\mathrm{P}(B|A)$, the probability that the student is able to learn how to solve a challenge that they do not yet know, as $\mathrm{P}[\mathcal{A}_{r,c}]$, the probability that the student provides a correct response to that challenge:[1]

$$\begin{align} \mathrm{P}(A) &= 1 - \mathrm{P}[\mathcal{A}_{r,c}] \tag{2.1.2} \\ \mathrm{P}(B|A) &= \mathrm{P}[\mathcal{A}_{r,c}] \tag{2.1.3} \\ APPR_s(c) &= \mathrm{P}(A)\mathrm{P}(B|A) \notag \\ &= (1 - \mathrm{P}[\mathcal{A}_{r,c}])\,\mathrm{P}[\mathcal{A}_{r,c}] \tag{2.1.4} \end{align}$$

[1] In this model, we assume that the probability of "guessing" (i.e. student does not know the skill but their answer happens to be correct) and "slipping" (i.e. student does know the skill but their answer happens to be incorrect) are both negligible.

Given the shape of the appropriateness function (i.e. that the second derivative of the function is negative), the point at which the first derivative is zero indicates a maximum value for challenge appropriateness:

$$\begin{aligned}\frac{dAPPR_s(c)}{d\mathrm{P}\left[\mathcal{A}_{r,c}\right]} &= 0 \\ \mathrm{P}\left[\mathcal{A}_{r,c}\right] &= 0.5\end{aligned}$$

Thus, challenge appropriateness under this model is maximized when $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = 0.5$. Given that $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = 1 - \mathrm{P}\left[\neg\mathcal{A}_{r,c}\right]$, this also implies that $\mathrm{P}\left[\neg\mathcal{A}_{r,c}\right] = 0.5$ occurs at maximally appropriate challenges, and so we also have:

$$\mathrm{P}\left[\mathcal{A}_{r,c}\right] = \mathrm{P}\left[\neg\mathcal{A}_{r,c}\right] \tag{2.1.5}$$

The most appropriate challenges are therefore those for which the student's response is equally likely to be correct or incorrect.[2]

2.1.2 Contextualizing Appropriate Challenge

Our student-specific model of problem appropriateness shares certain similarities with several established models of instruction, motivation, and assessment.

From their studies of learner-determined study time allocation, Metcalfe and Kornell formed the Region of Proximal Learning framework, which offered a better fit for their experimental results than the dominant discrepancy reduction model [60]. Where the discrepancy reduction model suggests that self-directed learners will choose to study the hardest problems first (since they offer the most opportunity for growth),

[2] Assuming that the probability of an accurate response based on guessing is negligible.

the Region of Proximal Learning framework models a different approach. Learners should first briefly address “easy” problems, then spend the bulk of their study time on the “medium” difficulty problems, and use any remaining time on the “hard” problems. Metcalfe and Kornell observe that during medium-difficulty study, the length of time during which learning gains continued to occur was greater than the length of active learning while studying the easy problems. The origins of their model are traced back to a number of other theories of learning, including those of Vygotsky’s Zone of Proximal Development [68], which provides a model of development with instructional implications. Vygotsky suggests that for every student, there are problems that the student is currently unable to solve alone but is able to solve in collaboration with another person, and argues that instruction is most productive when it targets problems in this zone [68].

Among models of motivation, Csikszentmihalyi suggests that when the difficulty level of a challenge matches the level of a learner’s skill, a positive “flow” experience can result. Mismatches, on the other hand, result either in boredom (given easy challenges and high skills) or in anxiety (given hard challenges and low skills) [23]. In order to maintain this flow state once skills improve, challenge difficulty must increase to compensate. Koster situates this notion within his approach towards game design [50], and Hunicke and Chapman suggest a more direct application of flow theory to games, through the dynamic adjustment of game difficulty to constantly match the player’s observed skill level [40]. Lepper discusses how it is the activities that provide an intermediate level of difficulty that stimulate the most intrinsic motivation for learning within the student [52]. When extrinsic motivations are used, they should ideally be incorporated into the activity itself.

Within the literature on assessment, item response theory provides a model for estimating student ability based on observations of performance on problems of known difficulty [54]. Assuming the simplest one-parameter Rasch model [67, 29] and a dichotomous model of response accuracy, the amount of information about the student's ability gained by posing a particular challenge item aligns exactly with our functional definition of challenge appropriateness. Item information is greatest when the probability that the student will respond correctly is 0.5. Chen and colleagues have explored how an IRT-based tutoring system can this use item information as a basis for "curriculum sequencing" (i.e. recommending future content for study) [19, 20].

2.2 Game Theoretic Concepts and Definitions

The techniques, terminology, and assumptions that we will use in examining games designed to provide learners with challenges of appropriate difficulty are drawn from game theory. Many excellent resources on game theory are available (e.g. Gintis [31], Fudenberg & Tirole [30], von Neumann & Morgenstern [80]), so we present here only a brief overview of the specific terminology, techniques, and assumptions most relevant to the discussions that follow.

In the following paragraphs, we will elaborate on the italicized terminology: In the three games discussed throughout the rest of this chapter, we focus on *two-player*, *non-zero sum*, *extensive form games*. We assume that both players act *rationally*, attempting to maximize *utility* and *expected utility* (given their *privately-held beliefs* about *player types*.) For each game, we seek to show that certain *strategies* of interest are *dominant* for each player, resulting in *strategy profiles* that constitute an *equilibrium* with a desired system-wide property: that tutors will pose tutees with challenges of appropriate difficulty.

In each *two-player* game, the two participating peers adopt asymmetric roles: one is a “Teacher” and the other is a “Student.” We note that these roles refer to game-play only, and do not imply that the Teacher is an instructor in a classroom. As we shall see later, a particular learner may play the Teacher role in one game and the Student role in another, perhaps even simultaneously. Within the context of the game, the Teacher is the player tasked with selecting problems, and the Student is the player tasked with solving these problems.

The game itself is considered *non-zero sum* because, at the conclusion of the game when both players are awarded payoffs, the sum of these payoffs does not total to zero. In strictly competitive two-person games, on the other hand, one player’s loss is necessarily the other player’s gain.

The games that we introduce are *extensive form games*, in which player actions occur sequentially (rather than simultaneously), with full knowledge of previous actions taken. Sequential form games are traditionally represented by a game-tree, and illustrations of extensive-form games (such as Figure 2.1) can be interpreted as follows: Activity proceeds from the root node (at top) to the terminal nodes (at bottom.) Each node represents an action made by the player labeled above it (or, in the case of non-player labels, by the game itself.) The branches represent choices among the specified labeled options, and the gray triangles represent a large set of options, as specified. The comma-separated values displayed below the terminal nodes indicate the payoff values awarded to each player if that node is reached. The first value of the pair indicates the payoff to the Teacher, and the second indicates the payoff to the Student.

The player *rationality* assumption implies that, when faced with the choice between a smaller and a larger end-game *utility* value payoff, the rational player will always choose the larger of the options.

In games that involve uncertainty, utility-maximization is not sufficient to determine rational player actions. The notion of *expected utility* fills this gap, by weighting each uncertain outcome by the probability with which it occurs. When choosing among several actions leading to uncertain outcomes, the expected utility values of these actions can be compared, and we assume that the rational player will always select the action offering the largest expected utility.

While the probability of uncertain events may be agreed upon by players in some situations (e.g. two players may both agree that the probability of a coin tossed will land on its head is 0.5), there are other situations in which probabilities may differ, based on *privately-held beliefs*. When querying a player about a privately-held belief, that player may or may not honestly reveal it. In our games, the beliefs themselves pertain to *player types*, in which the types indicate whether or not the particular player will be able to correctly solve a particular problem.

For games in which a player must choose a single action, the player's *strategy* is defined by that action. For games involving sequences of actions, the player's strategy is defined by the sequence of actions that they choose. One player strategy is said to *strictly dominate* another if it yields a higher payoff (or is said to *weakly dominate* another if it yields at least as high of a payoff) as the other, regardless of the strategy chosen by the other game player(s). A strategy is said to be *dominant* if it dominates all other strategies (with strict and weak variations.) The selected strategies of each player in the game, taken as a set, constitute a *strategy profile*.

Each *equilibrium* concept specifies a set of qualifying conditions for strategy profiles. A Nash equilibrium, for example, is a strategy profile in which neither player can become better off (in terms of utility) by switching strategies, unless the other player also switches strategies. We are primarily concerned with an equilibrium that offers a desired educational property: that tutors will pose tutees with challenges of appropriate difficulty, and tutees will try their best to solve these challenges.

2.3 The Teacher's Dilemma criteria

In order to construct games in which students are provided with challenges of appropriate difficulty on which to practice, we begin by formalizing this goal as criteria for inclusion in a special class of *Teacher's Dilemma* games. We introduce the *appropriateness-dominance* criteria: the tutor's dominant strategy in a Teacher's Dilemma game must be to pose challenges of appropriate difficulty for the tutee. We supplement the *appropriateness-dominance* requirement with a second Teacher's Dilemma criteria, affecting the tutee. The tutee's dominant strategy in a game must be to provide a "best-effort" response: one that they believe most likely to be correct. This *effort-dominance* requirement prevents situations in which the tutee is provided with motivation to purposefully answer a question incorrectly for strategic reasons. Just as the effort-dominance criterion is needed to maintain the integrity of the appropriateness-dominance criterion given the tutee's freedom to strategize, other game-specific criteria may be necessary to maintain the integrity of these criteria in games that introduce additional assumptions and resources. We will discuss these additional criteria as necessary.

2.4 Three Teacher's Dilemma games

One of the advantages offered by defining the Teacher's Dilemma in terms of criteria rather than as a specific game, is that we leave room for significant variations in implementation. We have not specified the number of players that participate, nor have we specified what responsibilities or decisions are tasked to each player (aside from challenge-response by the tutee.) Additional domain-specific resources or game-play assumptions may be present or absent. We exhibit this flexibility by introducing three different games, perhaps best-suited for practicing problems in different task domains, each provably satisfying the Teacher's Dilemma game criteria.

2.4.1 A *difficulty*-based Teacher's Dilemma game

The first game that we introduce is a simple two-player game, which we call the *difficulty*-based game. Figure 2.1 illustrates this game. In this game, the Teacher first selects a challenge to pose to the Student. The Student then provides a response, and the accuracy of this response is objectively assessed (by the game, as discussed below.) Both players receive payoffs based on the accuracy of the response and the difficulty of the problem, as it is objectively assessed by the game, within the context of a well-defined task domain. The task domain is a set or grammar of challenges that test a student's abilities in a single topic. For a task-domain to be well-defined, the accuracy of any legal response r to any legal challenge c must be calculable.

The Teacher's payoff, π_t, and the Student's payoff, π_s are as follows:

$$\pi_t = \begin{cases} 1 - \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 0 \\ \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{2.4.1}$$

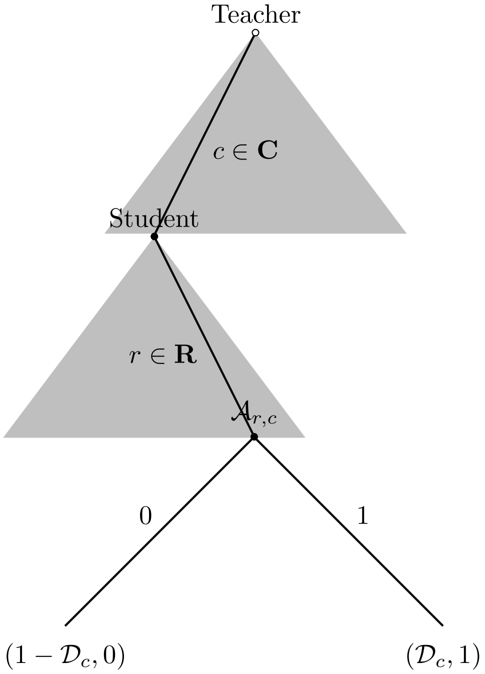


Figure 2.1: A *difficulty*-based Teacher's Dilemma game, in which the Teacher's payoff is determined by challenge difficulty and response accuracy, and the Student's payoff is determined solely by response accuracy. The Teacher chooses some challenge, c, from the space of legal challenges in the task domain, $\mathbf{C}$. The Student then chooses some response, r, from the space of legal responses in the task domain, $\mathbf{R}$. $\mathcal{D}_c$ indicates the difficulty of challenge c, ranging from 0 for the easiest challenges to 1 for the most difficult challenges. $\mathcal{A}_{r,c}$ represents the game-determined accuracy of response r to challenge c, where $\mathcal{A}_{r,c} = 1$ for correct responses or $\mathcal{A}_{r,c} = 0$ for incorrect responses. Players are rewarded points as shown in parentheses (with the Teacher's payoff listed before the Student's payoff).

$$\pi_s = \begin{cases} 0 & \text{if } \mathcal{A}_{r,c} = 0 \\ 1 & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{2.4.2}$$

The Student receives 1 point if the response is accurate, or 0 if it is not. The Teacher receives $\mathcal{D}_c$ points (where $\mathcal{D}_c$ measures the difficulty of the challenge, ranging from 0 for easiest to 1 for most difficult) if the response is accurate, or $1 - \mathcal{D}_c$ if it is not. As this game relies on the objective assessment of problem difficulty $\mathcal{D}_c$ and of response accuracy $\mathcal{A}_{r,c}$, the game is only applicable to task domains for which these assessments are attainable.

We point out that response accuracy $\mathcal{A}_{r,c}$ does not constitute a game strategy for the Student. The complicating factor here is that the Student's payoff is based on response accuracy, but the Student can only indirectly affect accuracy, via choice of response r. The "trembling hand" construct in game theory provides a way to model a situation in which a player does not have full control over their own actions [31], but this does not sufficiently describe the current situation. We note that while a Student does not always have the ability to provide a response that she is sure is accurate, the Student *does* always have the ability to provide a response that she is sure is *incorrect.* So the Student's available choice of strategy is limited by ability in one direction, but not in the other. Depending on the Student's payoff function π_s, the Student may have an incentive to exercise this asymmetry, by preferring a response guaranteed to be incorrect over a response likely (but not certain) to be correct. For some games, such strategic under-performance may be a form of "gaming the system" [6]. With this in mind, the *effort-dominance* criteria is included in the Teacher's Dilemma formulation. For a game to be effort-dominant, the rational Student must always prefer (based on expected utility values) to provide their "best-effort" response. If the Student were to identify the probabilities with which each response will be assessed as accurate, no response is deemed more likely to be accurate than a "best-effort" response.

We can show that this game meets the criteria of a Teacher's Dilemma game. Namely, that providing a best-effort response is the dominant strategy for the Student, and that providing a maximally-appropriate challenge is the dominant strategy for the Teacher.

Best-effort strategies dominate

Expected utility is a linear combination of the probability of occurrence and resulting payoffs of subsequent game tree branches. For the Student:

$$\mathrm{E}_{\pi_s} = \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c} = i\right] (\pi_s | \mathcal{A}_{r,c} = i) \tag{2.4.3}$$

While the payoff values here are known from the statement of the game, the corresponding probabilities are not. It is left to the Student to estimate the likelihood of reaching each terminal node. We refer to this estimation as the Student's "true expectation" of this probability, and denote it as $\dot{\mathcal{E}}_s(r, c)$. This estimate is based entirely on the player's privately-held beliefs regarding the likelihood of the outcome. By definition of true expectation, the Student believes that $\mathrm{P}[\mathcal{A}_{r,c}] = \dot{\mathcal{E}}_s$ and that $\mathrm{P}[\neg\mathcal{A}_{r,c}] = 1 - \dot{\mathcal{E}}_s$. The Student's expected utility can be restated as a linear combination of these true expectations and the payoffs from the game statement:

$$\begin{aligned}\mathrm{E}_{\pi_s} &= \left(1 - \dot{\mathcal{E}}_s\right) (\pi_s | \neg\mathcal{A}_{r,c}) + \left(\dot{\mathcal{E}}_s\right) (\pi_s | \mathcal{A}_{r,c}) \\ &= \left(1 - \dot{\mathcal{E}}_s\right) (0) + \left(\dot{\mathcal{E}}_s\right) (1) \\ &= \dot{\mathcal{E}}_s \end{aligned} \tag{2.4.4}$$

Given a posed challenge c, we call the response that the Student believes most likely to be correct a "best-effort" response, and denote it as r_{BE}. We call another response that they believe less likely to be correct a "less-effort" response, and denote it as r_{LE}. In the expectation notation used above, we denote this as $\dot{\mathcal{E}}_s(r_{BE}, c) > \dot{\mathcal{E}}_s(r_{LE}, c)$. We abbreviate this by using a tilde to denote the "less-effort" case, so we can restated this as $\dot{\mathcal{E}}_s > \tilde{\mathcal{E}}_s$. A payoff function is strictly *effort-dominant* if the expected utility associated with a "best-effort" response is greater than that associated with any non-best-effort response (i.e. $\forall c \forall r_{BE} \forall r_{LE} \left[\mathrm{E}_{\pi_s}(r_{BE}, c) > \mathrm{E}_{\pi_s}(r_{LE}, c) \right]$.) Since $\mathrm{E}_{\pi_s}(r_{BE}, c) = \dot{\mathcal{E}}_s$ and $\mathrm{E}_{\pi_s}(r_{LE}, c) = \tilde{\mathcal{E}}_s$, the best-effort strategy yields a higher expected utility than the less-effort strategy iff $\dot{\mathcal{E}}_s > \tilde{\mathcal{E}}_s$. As $\dot{\mathcal{E}}_s > \tilde{\mathcal{E}}_s$ holds true by definition of $\tilde{\mathcal{E}}_s$ a best-effort strategy strictly dominates all other "less-effort" strategies.

Appropriate-challenge strategies dominate

As the structure of the game (as shown in Figure 2.1) is known by all players, the previous analysis informs the Teacher's strategy. Since the best-effort strategy is strictly dominant for the Student, the Teacher can assume that a rational Student will adopt it.[3] The Teacher's task, then, is to select a challenge that maximizes the Teacher's expected payoff given a best-effort response from the Student. This expected utility is:

$$\mathrm{E}_{\pi_t} = \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c} = i\right] \left(\pi_s | \mathcal{A}_{r,c} = i\right) \tag{2.4.5}$$

As above, the probability of accuracy must be estimated by the player. Using similar notation as above, we let $\dot{\mathcal{E}}_t$ represent the Teacher's privately-held expectations regarding $\mathrm{P}\left[\mathcal{A}_{r,c}\right]$.

$$\begin{aligned} \mathrm{E}_{\pi_t} &= \mathrm{P}\left[\mathcal{A}_{r,c} = 0\right] \left(\pi_t | \neg \mathcal{A}_{r,c}\right) + \mathrm{P}\left[\mathcal{A}_{r,c} = 1\right] \left(\pi_t | \mathcal{A}_{r,c}\right) \\ &= \left(1 - \dot{\mathcal{E}}_t\right) \left(1 - \mathcal{D}_c\right) + \left(\dot{\mathcal{E}}_t\right) \left(\mathcal{D}_c\right) \end{aligned} \tag{2.4.6}$$

[3] Thus, during repeated game-play, the Teacher can safely attribute the Student's performance directly to their ability, without concern that it may instead reflect their strategy.

Assuming that $\dot{\mathcal{E}}_t$ accurately reflects $\mathrm{P}\left[\mathcal{A}_{r,c}\right]$ and that $\mathcal{D}_c$ accurately reflects $1-\mathrm{P}\left[\mathcal{A}_{r,c}\right]$, the Teacher's expected utility for posing challenge c to the Student is a function of the appropriateness of that challenge for that Student:

$$\begin{aligned}\mathrm{E}_{\pi_t} &= \left(1-\dot{\mathcal{E}}_t\right)\left(1-\mathcal{D}_c\right)+\left(\dot{\mathcal{E}}_t\right)\left(\mathcal{D}_c\right)\\ &= \left(1-\mathrm{P}\left[\mathcal{A}_{r,c}\right]\right)\left(\mathrm{P}\left[\mathcal{A}_{r,c}\right]\right)+\left(\mathrm{P}\left[\mathcal{A}_{r,c}=1\right]\right)\left(1-\mathrm{P}\left[\mathcal{A}_{r,c}\right]\right)\\ &= 2\mathrm{P}\left[\mathcal{A}_{r,c}\right]\left(1-\mathrm{P}\left[\mathcal{A}_{r,c}\right]\right)\\ &= 2APPR_s(c) \qquad (2.4.7)\end{aligned}$$

For a maximally-appropriate challenge, c_{APPR}, $\mathrm{P}\left[\mathcal{A}_{r,c_{APPR}}\right]=0.5$ holds by definition, and the Teacher's expected utility is $\mathrm{E}_{\pi_t}(r,c_{APPR})=0.5$. For a less-appropriate challenge, c_{OTHER}, $\mathrm{P}\left[\mathcal{A}_{r,c_{OTHER}}\right]\neq 0.5$, and the Teacher's expected utility $\mathrm{E}_{\pi_t}(r,c_{OTHER})<0.5$. Thus, the strategy of selecting a maximally-appropriate challenge strictly dominates any other challenge-selection strategy. Since the Teacher's expected utility varies linearly with the challenge appropriateness of the problem posed, we can go one step further. For all c_1, c_2 for which $APPR_s(c_1)>APPR_s(c_2)$, $\mathrm{E}_{\pi_t}(r,c_1)>\mathrm{E}_{\pi_t}(r,c_2)$ must hold. Thus, any strategy involving the selection of a specific challenge is strictly dominated by all strategies involving the selection of a more appropriate challenge. We have motivated appropriate challenge by aligning the Teacher's expected utility function with a Student-specific measure of challenge appropriateness.

While we have sufficiently addressed the case in which $\dot{\mathcal{E}}_t=\mathrm{P}\left[\mathcal{A}_{r,c}\right]$ and $\mathcal{D}_c=1-\mathrm{P}\left[\mathcal{A}_{r,c}\right]$, other dynamics may arise when one or both of these statements do not hold. On challenges for which $\mathrm{P}\left[\mathcal{A}_{r,c}\right]\neq\dot{\mathcal{E}}_t$, Figure 2.2 indicates that less-appropriate challenges may appear preferable. While this may undermine appropriateness in the one-shot game, repeated play between a particular pair of players provides the Teacher

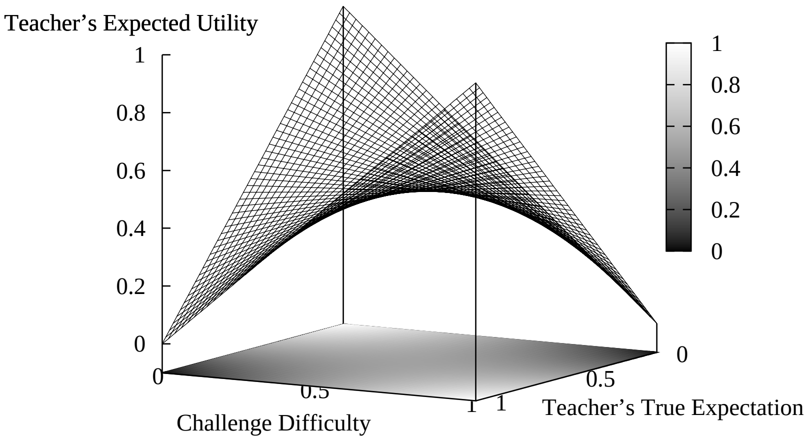


Figure 2.2: The Teacher's expected utility, E_{π_t}, is a function of the Teacher's true expectation, $\dot{\mathcal{E}}_t$, and the challenge's difficulty value, $\mathcal{D}_c$.

with the opportunity to observe the error in their estimation of the Student, and revise $\dot{\mathcal{E}}_t$ to incorporate, and better account for, these observations. Through repeated play, a learning Teacher will converge upon the true probability over time, ultimately valuing $\dot{\mathcal{E}}_t = \mathrm{P}\left[\mathcal{A}_{r,c}\right]$. This learning process is in the Teacher's best interests, as it provides an increasingly accurate mapping (from actions to payoffs) based on which the Teacher can take action. Similarly, if the challenge difficulty function does not initially reflect student response accuracy (i.e. $\mathcal{D}_c \neq 1 - \mathrm{P}\left[\mathcal{A}_{r,c}\right]$), it, too, may be realigned by introducing an observation-based updating process. When a dynamic process is used to update the $\mathcal{D}_c$ difficulty metric, inaccuracies may be corrected over time. Given repeated play, difference between $\mathcal{D}_c$ and $1 - \mathrm{P}\left[\mathcal{A}_{r,c}\right]$ will be reduced, ultimately resulting in the equivalent-functions case discussed in the previous paragraph. Thus, mis-aligned functions will align over time, and the dominant strategy for the Teacher will then be to select maximally-appropriate challenges for their Student.

By meeting the Teacher's Dilemma game criteria, this difficulty-based game offers one approach to motivating the selection of challenges of appropriate difficulty for learners.

2.4.2 An *expectation*-based Teacher's Dilemma game

While the previous section shows that the structure of the difficulty-based game will, under certain assumptions, converge to appropriate challenges being posed, the game assumes the existence of a suitable difficulty metric and the use of a learning algorithm to update that metric based on observed data. In this section, we introduce a second game. This game is designed to motivate appropriate challenges without the need for – or existence of – the dynamically-adjusted difficulty metric, as used in the previous game. Figure 2.3 shows an *expectation*-based Teacher's Dilemma game.

We will show that the tutee's payoff is structured in such a way as to motivate honesty in the statement of expectation while simultaneously motivating best-effort responses (i.e. the tutee never has an incentive to purposefully miss an answer), and the tutor's payoff is structured to motivate the selection of appropriate challenges.

The intuition behind the expectation-based game is that estimates of the probability of response accuracy – previously obtained through the difficulty metric – can alternatively be obtained by directly querying the players themselves. So, in this game, we assume that each player can form an opinion regarding the probability that the Student's response will be correct. Building on this notion of "true expectation" ($\dot{\mathcal{E}}$, as introduced above), we introduce a second notion, "stated expectation" $\mathcal{E}$. Where true expectation reflects a player's privately-held beliefs, stated expectation reflects their publicly-shared beliefs. For the Teacher, the statement of expectation is a response to the student modeling question: "With what probability do you expect

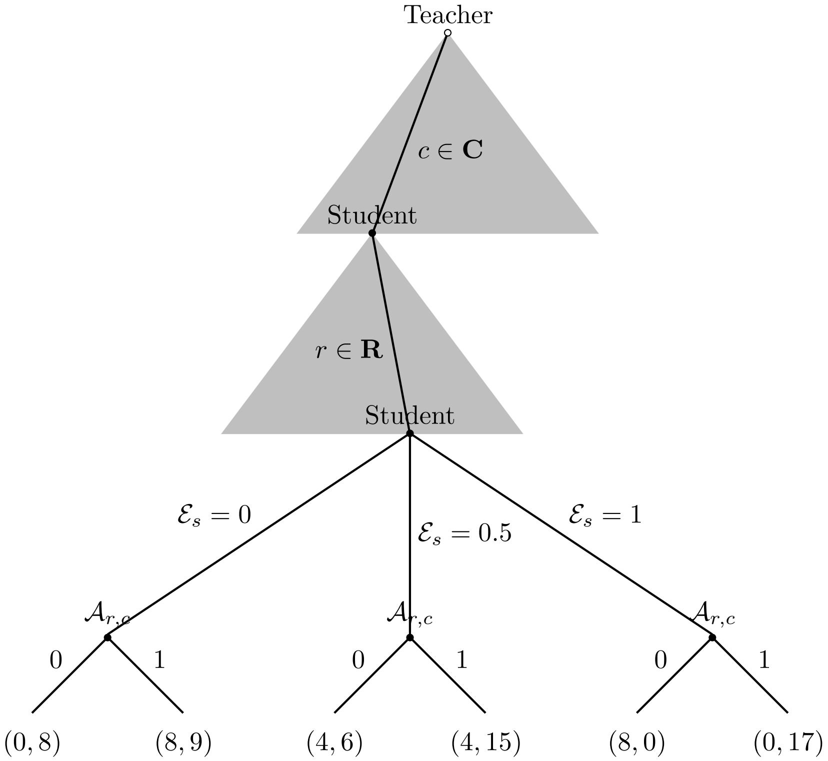


Figure 2.3: An *expectation*-based Teacher's Dilemma game, in which both players' payoffs are determined by the accuracy of the Student's response and the Student's level of confidence in that response. The Teacher first chooses some challenge, c, from the space of legal challenges in the task domain, $\mathbf{C}$. The Student then chooses some response, r, from the space of legal responses in the task domain, $\mathbf{R}$. The Student then states the probability with which they believe their response to be correct. The accuracy of the response to the challenge is objectively assessed by the game, $\mathcal{A}_{r,c}$ (where $\mathcal{A}_{r,c} = 1$ for correct responses or $\mathcal{A}_{r,c} = 0$ for incorrect responses.) Finally, both players are rewarded points as shown in parentheses (with the Teacher's payoff listed before the Student's payoff).

the tutee to accurately respond to the challenge question?" For the Student, the statement of expectation requires metacognitive reflection: "With what probability do you believe your response to the challenge question to be accurate?" As we will show, stated and true expectations may differ from one another, as the stated expectation may vary based on the context in which the student is asked to share. Players are free to misrepresent their true beliefs, and can be expected to do so whenever some strategic advantage (i.e. increase in expected utility) can be gained.

In order to show that our expectation-based game meets the Teacher's Dilemma criteria, we must show that, in this game, misrepresenting true beliefs can never offer any strategic advantage, and so players have no incentive to distort or otherwise misrepresent their true expectations. Thus, an expectation-based game must simultaneously meet three criteria to qualify as Teacher's Dilemma games. Response effort and statement truth must both dominate Student strategies, and challenge appropriateness must dominate Teacher strategies.

We note that Figure 2.3 is an instantiation of the game presented in Figure 2.4, in which the expectation statement is discretized into three levels ($\mathcal{E}_s = 0$, $\mathcal{E}_s = 0.5$, or $\mathcal{E}_s = 1$). The five w parameters from Figure 2.4 are set as follows: $w_1 = 8$, $w_2 = 0$, $w_3 = -8$, $w_4 = 9$, and $w_5 = 8$.

We can show that for the general form of the game (and thus, for all instances, including the one shown in Figure 2.3), the Teacher's Dilemma criteria are all met. We begin by stating the game's payoff functions, noting that we impose the following restrictions on the w-values: $w_3 < 0$, $w_3 + w_4 > 0$, and $w_1 > 0$:

$$\pi_t = \begin{cases} w_1\mathcal{E}_s + w_2 & \text{if } \mathcal{A}_{r,c} = 0 \\ w_1(1 - \mathcal{E}_s) + w_2 & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{2.4.8}$$

$$\pi_s = \begin{cases} w_3{\mathcal{E}_s}^2 + w_5 & \text{if } \mathcal{A}_{r,c} = 0 \\ w_3(1 - \mathcal{E}_s)^2 + w_4 + w_5 & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{2.4.9}$$

By definition of true expectation, the Teacher believes that $\mathrm{P}[\mathcal{A}_{r,c}] = \dot{\mathcal{E}}_t$ and $\mathrm{P}[\neg\mathcal{A}_{r,c}] = 1 - \dot{\mathcal{E}}_t$, and the Student believes that $\mathrm{P}[\mathcal{A}_{r,c}] = \dot{\mathcal{E}}_s$ and $\mathrm{P}[\neg\mathcal{A}_{r,c}] = 1 - \dot{\mathcal{E}}_s$. The expected

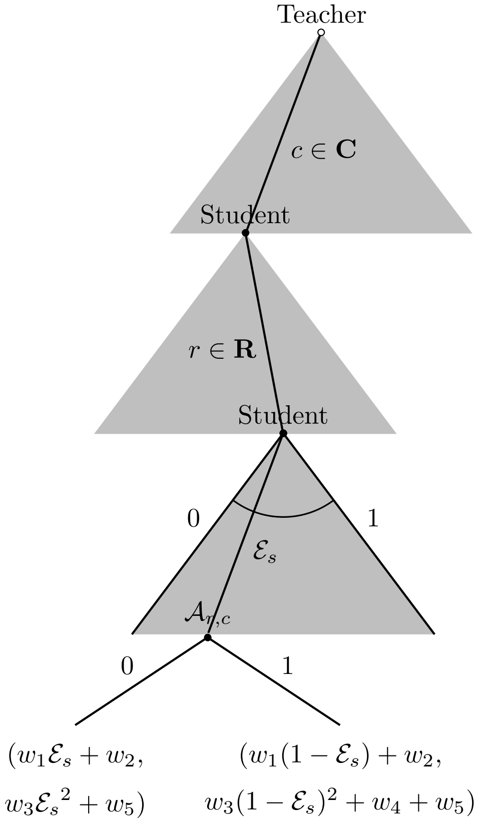

Teacher
$c \in \mathbf{C}$
Student
$r \in \mathbf{R}$
Student
0
1
$\mathcal{E}_s$
$\mathcal{A}_{r,c}$
0
1
$(w_1\mathcal{E}_s + w_2,$
$w_3{\mathcal{E}_s}^2 + w_5)$
$(w_1(1-\mathcal{E}_s) + w_2,$
$w_3(1-\mathcal{E}_s)^2 + w_4 + w_5)$

Figure 2.4: The generalized form of the *expectation*-based Teacher's Dilemma, in which stated expectation $\mathcal{E}_s$ is not discretized. Payoffs values are parameterized into five w-values, and we impose the restrictions that $w_3 < 0$, $w_3 + w_4 > 0$, and $w_1 > 0$.

utility of each player can thus be stated as a function of these $\dot{\mathcal{E}}$ probabilities and the payoff above, as plotted in Figures 2.5 and 2.6. For the Teacher:

$$\begin{aligned} \mathrm{E}_{\pi_t} &= \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c} = i\right] (\pi_t | \mathcal{A}_{r,c} = i) \\ &= \left(1 - \dot{\mathcal{E}}_t\right)(w_1\mathcal{E}_s + w_2) + \left(\dot{\mathcal{E}}_t\right)(w_1(1-\mathcal{E}_s) + w_2) \\ &= w_1\left(\dot{\mathcal{E}}_t + \mathcal{E}_s - 2\dot{\mathcal{E}}_t\mathcal{E}_s\right) + w_2 \qquad (2.4.10) \end{aligned}$$

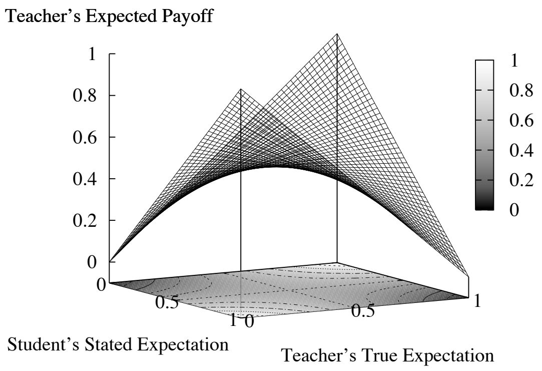


Figure 2.5: The Teacher's expected utility is a function of their true expectation $\dot{\mathcal{E}}_t$ and the Student's stated expectation $\mathcal{E}_s$. Shown here with $w_1 = 1$ and $w_2 = 0$.)

For the Student:

$$\begin{aligned} \mathrm{E}_{\pi_s} &= \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c} = i\right] \left(\pi_s | \mathcal{A}_{r,c} = i\right) \\ &= \left(1 - \dot{\mathcal{E}}_s\right) \left(w_3 \mathcal{E}_s^2 + w_5\right) + \left(\dot{\mathcal{E}}_s\right) \left(w_3 \left(1 - \mathcal{E}_s\right)^2 + w_4 + w_5\right) \\ &= w_3 \mathcal{E}_s^2 - 2 w_3 \dot{\mathcal{E}}_s \mathcal{E}_s + w_3 \dot{\mathcal{E}}_s + w_4 \dot{\mathcal{E}}_s + w_5 \qquad (2.4.11) \end{aligned}$$

We can show that for this game, the Student's payoff function π_s is simultaneously truth-dominant and effort-dominant, and the Teacher's payoff function π_t is appropriateness-dominant.

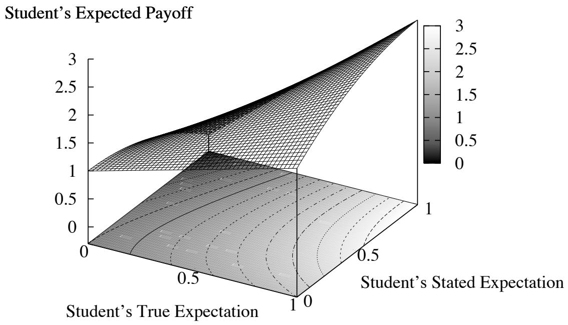


Figure 2.6: The Student's expected utility is a function of their true expectation $\dot{\mathcal{E}}_s$ and stated expectation $\mathcal{E}_s$ regarding response accuracy. Shown here with $w_3 = -1$, $w_4 = 2$, and $w_5 = 1$.)

Best-effort, truthful strategies dominate

As we showed in the previous section, the student's expected utility is:

$$E_{\pi_s} = w_3\mathcal{E}_s^2 - 2w_3\dot{\mathcal{E}}_s\mathcal{E}_s + w_3\dot{\mathcal{E}}_s + w_4\dot{\mathcal{E}}_s + w_5 \tag{2.4.12}$$

Using the same approach as in the effort-dominance proof for the first game, we recall that a payoff function is strictly *effort-dominant* if the expected utility associated with a "best-effort" response is greater than that associated with any non-best-effort response (i.e. $\forall c \forall r_{BE} \forall r_{LE} \left[\mathrm{E}_{\pi_s}(r_{BE}, c) > \mathrm{E}_{\pi_s}(r_{LE}, c)\right]$.) Based on the Student's expected utility for this game, this requires:

$$\begin{gathered}
\mathrm{E}_{\pi_s}(r_{BE}, c) > \mathrm{E}_{\pi_s}(r_{LE}, c) \\
\left[w_3\mathcal{E}_s^2 - 2w_3\dot{\mathcal{E}}_s\mathcal{E}_s + w_3\dot{\mathcal{E}}_s + w_4\dot{\mathcal{E}}_s + w_5\right] > \left[w_3\mathcal{E}_s^2 - 2w_3\tilde{\mathcal{E}}_s\mathcal{E}_s + w_3\tilde{\mathcal{E}}_s + w_4\tilde{\mathcal{E}}_s + w_5\right] \\
(-2w_3\mathcal{E}_s + w_3 + w_4)\left(\dot{\mathcal{E}}_s - \tilde{\mathcal{E}}_s\right) > 0
\end{gathered} \tag{2.4.13}$$

Given that $\dot{\mathcal{E}}_s > \tilde{\mathcal{E}}_s$, we can say that there exists some $\Delta > 0$ such that $\dot{\mathcal{E}}_s = \tilde{\mathcal{E}}_s + \Delta$. Substituting this into 2.4.13, we get:

$$\begin{aligned}(-2w_3\mathcal{E}_s + w_3 + w_4)\left((\tilde{\mathcal{E}}_s + \Delta) - \tilde{\mathcal{E}}_s\right) &> 0\\ (-2w_3\mathcal{E}_s + w_3 + w_4)\,\Delta &> 0\\ w_3\mathcal{E}_s &< \frac{w_3 + w_4}{2} \qquad (2.4.14)\end{aligned}$$

Recall that, as stated in Figure 2.4, $w_3 < 0$ and $w_3 + w_4 > 0$ limits the w-values. Based on the signs of these terms, we have $\frac{w_3+w_4}{2w_3} < 0$. And since $\mathcal{E}_s$ is a probability, we know that $0 \leq \mathcal{E}_s \leq 1$ must hold. Combining these:

$$\frac{w_3 + w_4}{2w_3} < 0 \leq \mathcal{E}_s \leq 1 \qquad (2.4.15)$$

This inequality holds for every value for $\mathcal{E}_s$, making a best-effort strategy strictly dominate all non-best-effort strategies. Thus, the Student's strategy is effort-dominant.

Next, we wish to establish that the truthful revelation of privately-held beliefs (i.e. $\mathcal{E}_s = \dot{\mathcal{E}}_s$) yields a payoff higher than any untruthful revelations (i.e. $\mathcal{E}_s \neq \dot{E}_s$). In this expectation-based game, we wish to use the Student's expectation of response accuracy $\dot{\mathcal{E}}_s(r, c)$ as an approximation for the actual (but unknown) probability of response accuracy $\mathrm{P}[\mathcal{A}_{r,c}]$. Since this expectation is privately-held by the Student, it cannot be directly accessed to determine the Teacher's payoff. Instead, we must rely on what the student reveals this expectation to be, which may or may not accurately reflect their private beliefs. To avoid any discrepancies, we wish to design the context in which the Student states his expectation in such a way as to encourage honest reporting and discourage any distortion. Assuming player rationality, if the truth-revealing strategy dominates untruthful strategies, we have succeeded. Within the mechanism design literature, such strategies are discussed in terms of the concepts of incentive compatibility, truthful mechanisms, and proper scoring rules [62, 63, 85, 61]

We will identify the highest-paying stated expectation corresponding to each true expectation, and show that the two are always equal. Given the downward-facing orientation of the expectation surface (due to the restriction that $w_3 < 0$, as illustrated in Figure 2.6), we can solve for these maxima by identifying when the partial derivative of E_{π_s} with respect to $\mathcal{E}_s$ is zero:

$$\begin{aligned} E_{\pi_s} &= w_3\mathcal{E}_s^2 - 2w_3\dot{\mathcal{E}}_s\mathcal{E}_s + w_3\dot{\mathcal{E}}_s + w_4\dot{\mathcal{E}}_s + w_5 \\ 0 &= \frac{\partial \mathrm{E}_{\pi_s}}{\partial \mathcal{E}_s} = 2w_3\mathcal{E}_s - 2w_3\dot{\mathcal{E}}_s \\ \mathcal{E}_s &= \dot{\mathcal{E}}_s \end{aligned} \tag{2.4.16}$$

The Student attempting to maximize expected utility will therefore always state their expectation ($\mathcal{E}_s$) exactly as they truly believe it ($\dot{\mathcal{E}}_s$.) Any misleading statement of true expectation leads to a lower expected utility, and so π_s is truth-dominant.

Appropriate-challenge strategies dominate

We note that based on the results of the previous section, a Teacher can assume that their Student will adopt a best-effort response strategy and a truthful expectation-statement strategy, and should form their own strategy accordingly. Our primary criteria for the Teacher's strategy is that it be appropriateness-dominant. We show that, when the players' true expectations agree, the Teacher's strategy is immediately appropriateness-dominant, and when the players' true expectations do not initially agree, the Teacher's strategy converges on appropriateness-dominance over the course of repeated play between the pair.

If the Teacher believes that he will agree with the Student's stated expectation of response accuracy, we have $\dot{\mathcal{E}_t} = \mathcal{E}_s$. We can rewrite the Teacher's expected utility purely in terms of $\dot{\mathcal{E}_t}$, and then solve for the highest expected utility. Noting that since $w_1 > 0$ by definition of the game, the resulting derivative indicates a maximum value of the function:

$$\begin{aligned} \mathrm{E}_{\pi_t} &= w_1 \left(\dot{\mathcal{E}_t} + \mathcal{E}_s - 2\dot{\mathcal{E}_t}\mathcal{E}_s \right) + w_2 \\ &= w_1 \left(2\dot{\mathcal{E}_t} - 2\dot{\mathcal{E}_t}^2 \right) + w_2 \\ \frac{d\mathrm{E}_{\pi_t}}{d\dot{\mathcal{E}_t}} &= 0 = w_1 \left(2 - 4\dot{\mathcal{E}_t} \right) \\ \dot{\mathcal{E}_t} &= 0.5 \end{aligned} \tag{2.4.17}$$

In this case, the optimal strategy is to select challenges for which $\dot{\mathcal{E}_t} = 0.5$, and so π_t is appropriateness-dominant.

On the other hand, if the Teacher believes that their own expectations will differ with those of the Student, we have $\dot{\mathcal{E}_t} \neq \dot{\mathcal{E}_s}$. The Teacher has the opportunity to out-perform the appropriateness strategy, by selecting the challenge that maximizes the anticipated difference between $\dot{\mathcal{E}_t}$ and $\dot{\mathcal{E}_s}$. In this case, either the Teacher or the Student has provided a poor estimation of expectation, and the results from the following accuracy assessment will lead that player's expectations back in line. In this case, the one-shot game may not be appropriateness-dominant, but the repeated game converges to an appropriateness-dominant payoff as the two players independently converge on increasingly accurate expectation models. Feedback from off-diagonal ($\dot{\mathcal{E}_t} \neq \mathcal{E}_s$) challenges serve to sharpen either the Teacher's student model of the Student, the Student's model of himself, or both.

In showing that the payoff functions for the expectation-based game could be restricted in such a way as to arrive at a Teacher's Dilemma, we surpassed one of the significant limitations of the difficulty-based game: the need for an existing difficulty metric. We were able to organize the game in such a way as to obtain approximations of this information from the players themselves. Errors in their estimations serve as opportunities for learning, and dissipate as that learning occurs.

2.4.3 An *equivalence*-based Teacher's Dilemma game

The third game that we introduce follows a different approach to motivating appropriate challenges. Where the previous two games leverage available *proxies for appropriateness* (such as problem difficulty or player-stated expectations of accuracy), the third game instead leverages available *evidence of appropriateness.* Recalling our definition of challenge appropriateness as the probability that the student will be able to learn from a challenge-response-feedback sequence, we can alternatively motivate appropriateness by rewarding Teachers most when their Students show evidence of learning. We do this by combining a simple technique for the assessment of learning – identifying change in response accuracy between a pre-test and a post-test – with a game-based reward structure. The key assumption for this model is one of challenge-equivalence: For any challenge c, we must be able to generate another challenge c' that requires all of the same skills to solve. For example, in a domain of mathematical word-problems, equivalent challenges may be generated by changing the values in the original challenge. In a spelling domain (in which challenges are words to spell), equivalent challenges may be other words that share the same root. The assumption of challenge equivalence is suitable for at least some learning domains. Guzmán and Conejo [35] discuss how the SIETTE web-based system for knowledge assessment can generate “isomorphic items,” and contextualizes this within a larger body of work on automatically-generated problems for tutoring systems [9].

Given the ability to generate challenge equivalents, we allow the Teacher to select a challenge, present it to the Student as a pre-test, provide the Student with feedback on their response accuracy and information about the correct response, then present the Student with the equivalent challenge as a post-test, and then reward both players. Figure 2.7 shows this third Teacher's Dilemma game. In this game, while the tutor's task still focuses on selecting appropriate challenges for their tutee, they are rewarded directly for the observed learnability of skills in challenges posed. We will show that the tutor's reward is maximized for selecting challenges involving skills that the tutee does not yet have but, given some feedback and an opportunity to attempt a similar challenge, the tutee becomes able to solve. The tutee's reward, again, motivates best-effort responses for both attempts. Additionally, we note that if the players are given the opportunity to communicate between the first and second challenges (as is the case in the collaborative testing environment described by Barros, Conejo and Guzman [8]), both players are motivated to try to help the tutee learn the skills involved.

As with the expectation-based game above, this game (as shown in Figure 2.7) is an instance of a more general game, defined in terms of w-parameters, which is shown in Figure 2.8. The instance was derived from the generalized form by setting the w parameters as follows: $w_1 = 2$, $w_2 = 1$, $w_3 = 3$, $w_4 = 0$, $w_5 = 1$, and $w_6 = 1$. In the general case, the following constraints must hold:

$$w_3 > w_1 > (w_2 = w_5 = w_6) > w_4 = 0 \tag{2.4.18}$$

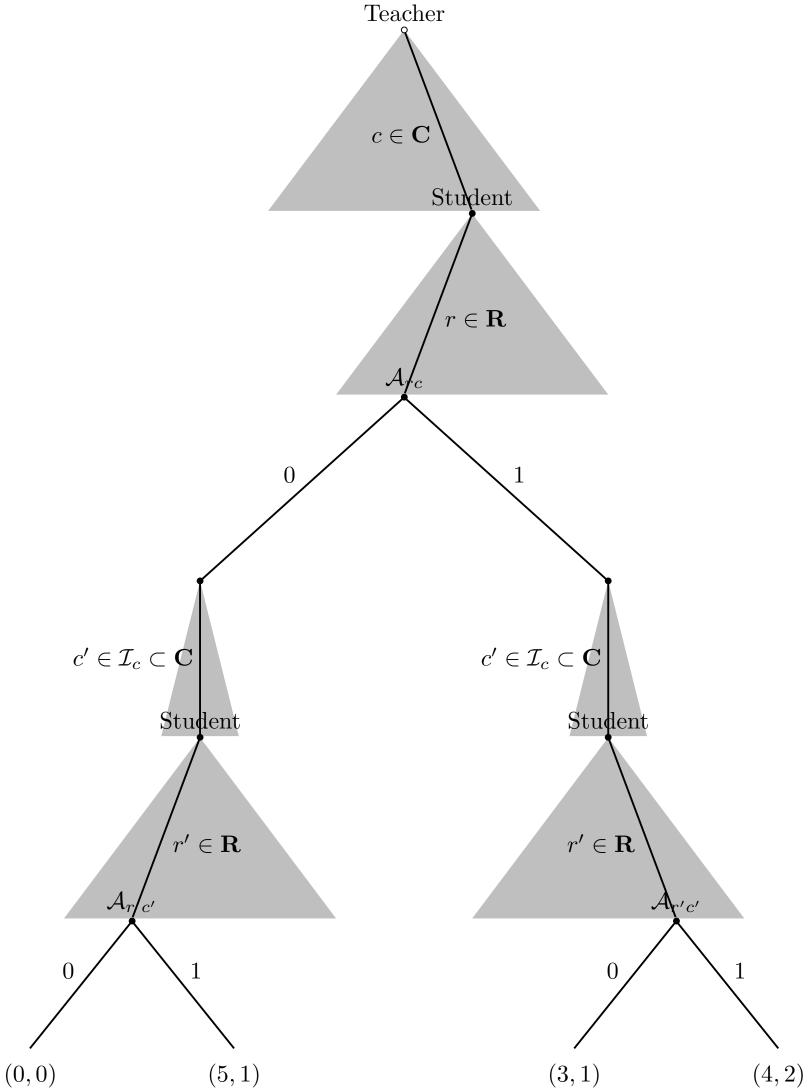


Figure 2.7: An *equivalence*-based Teacher's Dilemma game, in which the Teacher's payoff is a function of the response accuracies and changes in accuracy, and the Student's payoff is a function of the response accuracies. The Teacher first chooses some challenge, c, from the space of legal challenges in the task domain, $\mathbf{C}$. The Student then chooses some response, r, from the space of legal responses in the task domain, $\mathbf{R}$. The accuracy of the response to the challenge is objectively assessed by the game, $\mathcal{A}_{r,c}$ (where $\mathcal{A}_{r,c} = 1$ for correct responses or $\mathcal{A}_{r,c} = 0$ for incorrect responses.) The student is then provided feedback on their performance: the accuracy of their response is revealed, and information about the correct answer is presented. The Student is provided with a second challenge, $c' \in \mathbf{C}$, which is an element in the set of challenges equivalent to c, $\mathcal{I}_c$. The student provides a response r' to this challenge, and the accuracy information is again shared. Finally, both players are rewarded payoffs as specified above.

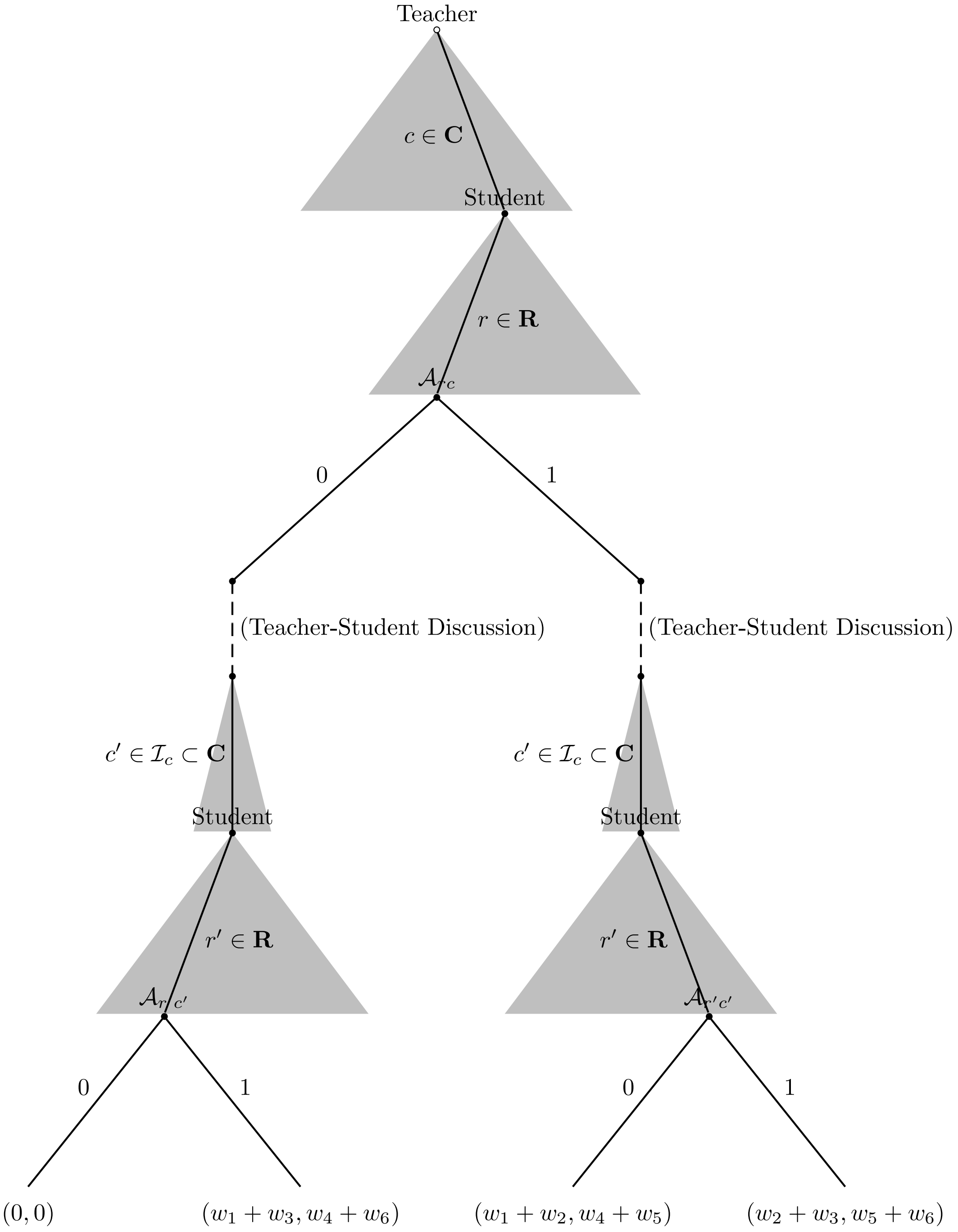


Figure 2.8: The generalized form of the *equivalence*-based Teacher's Dilemma. Teacher-Student discussion is optionally included between the first and second challenges. Payoffs are parameterized based on six w-values, restricted as follows: $w_3 > w_1 > (w_2 = w_5 = w_6) > w_4 = 0$.

The payoff functions for the players are specified as follows:

$$\pi_t = w_1 \left|\mathcal{A}_{r',c'} - \mathcal{A}_{r,c}\right| + w_2(\mathcal{A}_{r,c}) + w_3(\mathcal{A}_{r',c'}) \tag{2.4.19}$$

$$\pi_s = w_4 \left|\mathcal{A}_{r',c'} - \mathcal{A}_{r,c}\right| + w_5(\mathcal{A}_{r,c}) + w_6(\mathcal{A}_{r',c'}) \tag{2.4.20}$$

We will show that the best-effort strategy is dominant for the Student (both when selecting r and r'), the appropriate-challenge selection strategy is dominant for the Teacher, and, given the opportunity to communicate between the first and second challenges, both players are provided with strategic incentives to facilitate learning.

Best-effort strategies always dominate

In order establish whether a best-effort strategy is dominant, we follow the approach taken for the previous games. Namely, we derive the Student's expected utility function, and compare the expected utility of a "best-effort" and "less-effort" strategy. We do this for all three Student responses in the game tree, starting with the two second-response cases. For these analyses, we introduce the notation $\dot{\mathcal{E}}'_s$ to indicate the Student's true expectation of response accuracy, $\mathcal{A}_{r',c'}$. In choosing a second-response strategy following an *incorrect* first response, expected utility is:

$$\begin{aligned}
\mathrm{E}_{\pi_s}|\neg\mathcal{A}_{r,c} &= \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r',c'} = i\right](\pi_s|\mathcal{A}_{r',c'} = i) \\
&= \left(1 - \dot{\mathcal{E}}'_s\right)(0) + \left(\dot{\mathcal{E}}'_s\right)(w_4 + w_6) \\
&= \left(\dot{\mathcal{E}}'_s\right)(w_4 + w_6) \qquad (2.4.21)
\end{aligned}$$

Using the same approach as in the effort-dominance proof for the first two games, we recall that a payoff function is strictly *effort-dominant* if the expected utility associated with a "best-effort" response is greater than that associated with any non-best-effort response (i.e. $\forall c' \forall r'_{BE} \forall r'_{LE} \left[\mathrm{E}_{\pi_s}(r'_{BE}, c') > \mathrm{E}_{\pi_s}(r'_{LE}, c') \right].)$ For this to be the case, it must hold that:

$$\begin{aligned} \mathrm{E}_{\pi_s}(r'_{BE}, c')|\neg\mathcal{A}_{r,c} &> \mathrm{E}_{\pi_s}(r'_{LE}, c')|\neg\mathcal{A}_{r,c} \\ \left(\dot{\mathcal{E}}'_s\right)(w_4 + w_6) &> \left(\tilde{\mathcal{E}}'_s\right)(w_4 + w_6) \end{aligned} \tag{2.4.22}$$

By definition of these constraints (as stated in 2.4.18), $w_4 = 0$ and $w_6 > 0$, so $w_4 + w_6 > 0$. This reduces the inequality above to $\dot{\mathcal{E}}'_s > \tilde{\mathcal{E}}'_s$, which is true, by definition of $\tilde{\mathcal{E}}'_s$. Thus, a best-effort strategy is dominant for the second response following an incorrect first response.

Similarly, in choosing a second-response strategy to follow a *correct* first response, the Student's expected utility is:

$$\begin{aligned} \mathrm{E}_{\pi_s}|\mathcal{A}_{r,c} &= \sum_{i=0}^{1} \mathrm{P}\left[\mathcal{A}_{r',c'} = i\right](\pi_s|\mathcal{A}_{r',c'} = i) \\ &= \left(1 - \dot{\mathcal{E}}'_s\right)(w_4 + w_5) + \left(\dot{\mathcal{E}}'_s\right)(w_5 + w_6) \\ &= \left(1 - \dot{\mathcal{E}}'_s\right)(w_5) + \left(\dot{\mathcal{E}}'_s\right)(w_5 + w_6) \end{aligned} \tag{2.4.23}$$

For effort-dominance, it must hold that:

$$\begin{aligned} \mathrm{E}_{\pi_s}(r'_{BE}, c')|\mathcal{A}_{r,c} &> \mathrm{E}_{\pi_s}(r'_{LE}, c')|\mathcal{A}_{r,c} \\ \left(1 - \dot{\mathcal{E}}'_s\right)(w_5) + \left(\dot{\mathcal{E}}'_s\right)(w_5 + w_6) &> \left(1 - \tilde{\mathcal{E}}'_s\right)(w_5) + \left(\tilde{\mathcal{E}}'_s\right)(w_5 + w_6) \end{aligned} \tag{2.4.24}$$

Since $w_5 > 0$ and $w_6 > 0$ by definition of the constraints (as stated in 2.4.18), and since $\dot{\mathcal{E}}'_s > \tilde{\mathcal{E}}'_s$ by definition of $\tilde{\mathcal{E}}'_s$, we can simplify this inequality:

$$\begin{aligned}\left(\dot{\mathcal{E}}'_s - \tilde{\mathcal{E}}'_s\right)(w_5 + w_6) &> \left(\dot{\mathcal{E}}'_s - \tilde{\mathcal{E}}'_s\right)(w_5)\\ w_5 + w_6 &> w_5\\ w_6 &> 0 \end{aligned} \tag{2.4.25}$$

Again, as $w_6 > 0$ is implied by definition of the constraints in 2.4.18, the conditions necessary for best-effort to be a dominant strategy for the Student are always met.

Finally, when evaluating Student response strategies for the first response r, we take advantage of the fact that we now know that the rational Student will provide a best-effort response to the second question, regardless of her performance on the first. Thus, the expected utility at the first step can be stated as follows (and simplified since the parameter constraints dictate that $w_4 = 0$):

$$\begin{aligned}\mathrm{E}_{\pi_s} &= \sum_{i=0}^{1}\sum_{j=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c} = i\right]\mathrm{P}\left[\mathcal{A}_{r',c'} = j\right](\pi_s | \mathcal{A}_{r,c} = i, \mathcal{A}_{r',c'} = j)\\ &= \left(1 - \dot{\mathcal{E}}_s\right)\left(\dot{\mathcal{E}}'_s\right)(w_4 + w_6) + \left(\dot{\mathcal{E}}_s\right)\left(1 - \dot{\mathcal{E}}'_s\right)(w_4 + w_5) + \left(\dot{\mathcal{E}}_s\right)\left(\dot{\mathcal{E}}'_s\right)(w_5 + w_6)\\ &= w_6\left(1 - \dot{\mathcal{E}}_s\right)\left(\dot{\mathcal{E}}'_s\right) + w_5\left(\dot{\mathcal{E}}_s\right)\left(1 - \dot{\mathcal{E}}'_s\right) + \left(\dot{\mathcal{E}}_s\right)\left(\dot{\mathcal{E}}'_s\right)(w_5 + w_6)\\ &= w_5\dot{\mathcal{E}}_s + w_6\dot{\mathcal{E}}'_s \end{aligned} \tag{2.4.26}$$

As before, for "best-effort" to dominate "less-effort", the following must hold:

$$\begin{aligned}\mathrm{E}_{\pi_s}(r_{BE}, c) &> \mathrm{E}_{\pi_s}(r_{LE}, c)\\ w_5\dot{\mathcal{E}}_s + w_6\dot{\mathcal{E}}'_s &> w_5\tilde{\mathcal{E}}_s + w_6\dot{\mathcal{E}}'_s\\ w_5\dot{\mathcal{E}}_s &> w_5\tilde{\mathcal{E}}_s \end{aligned} \tag{2.4.27}$$

This inequality holds true as long as $w_5 > 0$, which is the case by game constraints.

So we have now shown that a best-effort response strategy dominates all less-effort response strategies game-wide. The rational Student will adopt such a strategy, and the rational Teacher can expect the Student to do so.

Learnable-challenge strategies dominate

Given the pre- and post-testing built into this game, our discussion of appropriate challenge can move from the realm of quantifying the probability of learning to quantifying changes in accuracy, an indicator of learning itself. Thus, for this game, what we show will not be the dominance of appropriate challenge strategies (based on the probability of change in accuracy), but rather the dominance of learned challenge strategies (based on the change in probability of accuracy). We wish for a payoff function which rewards the Teacher based on how much more likely the Student is to produce a correct response on the equivalent problem as compared to on the original problem. For these problems, as before, we begin by identifying the Teacher's expected utility:

$$\begin{aligned}
\mathrm{E}_{\pi_t} &= \sum_{i=0}^{1}\sum_{j=0}^{1} \mathrm{P}\left[\mathcal{A}_{r,c}=i\right]\mathrm{P}\left[\mathcal{A}_{r',c'}=j\right]\left(\pi_t|\mathcal{A}_{r,c}=i,\mathcal{A}_{r',c'}=j\right)\\
&= \left(1-\dot{\mathcal{E}}_t\right)\left(\dot{\mathcal{E}}'_t\right)(w_1+w_3)+\left(\dot{\mathcal{E}}_t\right)\left(1-\dot{\mathcal{E}}'_t\right)(w_1+w_2)+\left(\dot{\mathcal{E}}_t\right)\left(\dot{\mathcal{E}}'_t\right)(w_2+w_3)\\
&= (w_1+w_3)\,\dot{\mathcal{E}}'_t+(w_1+w_2)\,\dot{\mathcal{E}}_t-2w_1\dot{\mathcal{E}}_t\dot{\mathcal{E}}'_t \qquad (2.4.28)
\end{aligned}$$

Figure 2.9 plots this expected utility, as a function of the Teacher's true expectation for the first and second response accuracies.

The Teacher strategy that maximizes expected utility is to pose a question for which the Teacher expects the Student to answer incorrectly the first time (i.e. $\dot{\mathcal{E}}_t =$

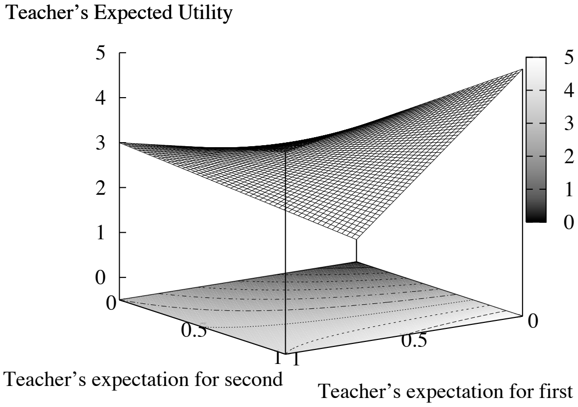


Figure 2.9: The Teacher's expected utility, E_{π_t}, is a function of the Teacher's true expectation for the first challenge, $\dot{\mathcal{E}}_t$, and for the second challenge, $\dot{\mathcal{E}}'_t$.

0), but given feedback on their response, expects the Student to answer correctly on the second (i.e. $\dot{\mathcal{E}}'_t = 1$). The expected utility corresponding to this strategy is:

$$\begin{aligned}\mathrm{E}_{\pi_t} &= (w_1 + w_3)\,\dot{\mathcal{E}}'_t + (w_1 + w_2)\,\dot{\mathcal{E}}_t - 2w_1\dot{\mathcal{E}}_t\dot{\mathcal{E}}'_t \\ &= (w_1 + w_3)\,(1) + (w_1 + w_2)\,(0) - 2w_1\,(0)\,(1) \\ &= w_1 + w_3 \qquad (2.4.29)\end{aligned}$$

In general, we can use an approach similar to our effort-dominance proofs, by introducing notation to represent the amount of change in the Teacher's true expectation for the first and second attempts. Letting $\delta = \dot{\mathcal{E}}'_t - \dot{\mathcal{E}}_t$, we call a challenge that the Teacher expects to yield a higher δ a "more-learning" challenge, and denote it as c_{ML}. We call another challenge that the Teacher expects to yield a lower $\tilde{\delta}$ a "less-learning" challenge, and denote it as c_{LL}. Based on these definitions, $\delta|(c_{ML}) > \tilde{\delta}|(c_{LL})$. We wish to show that $\forall c_{ML} \forall c_{LL} \left[\mathrm{E}_{\pi_t}(r, c_{ML}) > \mathrm{E}_{\pi_t}(r, c_{LL}) \right]$. For this to hold,

$$\begin{aligned}
\mathrm{E}_{\pi_t}(r, c_{ML}) &> \mathrm{E}_{\pi_t}(r, c_{LL}) \\
(w_1 + w_3)\left(\dot{\mathcal{E}}_t + \delta\right) - 2w_1\dot{\mathcal{E}}_t\left(\dot{\mathcal{E}}_t + \delta\right) &> (w_1 + w_3)\left(\dot{\mathcal{E}}_t + \tilde{\delta}\right) - 2w_1\dot{\mathcal{E}}_t\left(\dot{\mathcal{E}}_t + \tilde{\delta}\right) \\
(w_1 + w_3)\,\delta - 2w_1\dot{\mathcal{E}}_t\delta &> (w_1 + w_3)\,\tilde{\delta} - 2w_1\dot{\mathcal{E}}_t\tilde{\delta} \\
(w_1 + w_3)\left(\delta - \tilde{\delta}\right) &> 2w_1\dot{\mathcal{E}}_t\left(\delta - \tilde{\delta}\right) \\
\dot{\mathcal{E}}_t &< \frac{w_1 + w_3}{2w_1} \qquad (2.4.30)
\end{aligned}$$

Since the game constrains these w-parameters such that $w_3 > w_1 > 0$, $\frac{w_1+w_3}{2w_1} > 1$ must hold. Since probability $\dot{\mathcal{E}}_t \leq 1$, this $\dot{\mathcal{E}}_t < \frac{w_1+w_3}{2w_1}$ holds for all values of $\dot{\mathcal{E}}_t$. Thus, the rational Teacher will always prefer to pose a challenge for which they expect the greatest possible learning gain to occur during the game.

Productive discussion strategies dominate

One important additional dynamic to recognize is that if the two players are provided with the opportunity to communicate between the first and the second challenges – such as is the case in the collaborative testing environment described by Barros, Conejo and Guzman [8] – both players are motivated to use the opportunity to assist the Student in learning the skills involved in solving the challenge posed. For the Student, this follows from effort-dominance. If the Student is able to better learn the skills involved, the Student can effectively increase their expectation for the accuracy of their second response, $\dot{\mathcal{E}}'_s$. Within the framework of the effort-dominant strategy that a rational Student adopts, communication effectively opens a venue for generating a better “best-effort” response. Similarly, when presented with the opportunity to communicate between responses, the Teacher is motivated to assist the Student in learning the skills involved. In doing so, the Teacher can increase $\dot{\mathcal{E}}'_t$, and thereby generate a preferable "more-learning" challenge. In no case is either player provided with any incentive to discourage or undermine the Student in learning.

2.5 Summary

We introduce the concept of challenge appropriateness as a simple model of the likelihood of a student learning a particular challenge problem, and show how the class of Teacher Dilemma games can be used as a mechanism for focusing peer study on maximally appropriate challenges. We present three novel games, and show how each meets the criteria of a Teacher's Dilemma under a different set of assumptions. The difficulty-based game serves as the foundation for the implemented web-based systems that we discuss in future chapters. The expectation-based game provides a model for adapting the difficulty-based game for task domains for which an adaptive difficulty metric is not readily available. This effectively supports domains in which the space of problems grows over time, as is the case with user-generated content. Finally, the equivalence-based game offers a different approach that measures and rewards learning directly. Mid-game communication is introduced, the game motivates productive student dialogue without undermining the Teacher's Dilemma criteria. Using Teacher Dilemma games, we offer a way to effectively motivate peer learners to provide one another with appropriate challenges for practice.

Chapter 3

Simulating game dynamics

In the previous chapter, we presented a set of criteria for games that motivate appropriate challenge among peers, and we introduced three games that meet these criteria. For each of these games, our proofs that the game is appropriateness-dominant and effort-dominant rely on the standard game-theoretic assumption of player rationality. While this assumption may be reasonable for game interactions among autonomous agents [83], it is not necessarily appropriate for human players. Research in Evolutionary Game Theory has shown that such results are often still meaningful if we replace the assumption of strict rationality with a different assumption: that of repeated play over time [31]. Given repeated play, players may not select the optimal strategy *immediately*, but may instead reach it *eventually* as the result of a series of smaller strategic changes. Such repeated play dynamics offers a more realistic picture of what we would expect to observe from students engaging in a Teacher's Dilemma game.

In this Chapter, we explore the question: What can we expect to happen when two students repeatedly play a Teacher's Dilemma game? We simulate two different

Teacher's Dilemma games, and use these simulations to offer insight into two issues raised in Chapter 2. First, our previous analysis focused on the "asymmetric" form of each game, in which each player was either a Teacher or a Student. The "symmetric" form – in which two instances of the game are played simultaneously, with player roles reversed in the two games – introduces additional complexity. We use the simulation in Section 3.1 as an opportunity to observe if the symmetry introduces unexpected collusive strategies. Second, our analysis in Section 2.4.2 of the expectation-based game claimed that differences in the expectations of Teacher and Student could be resolved with the passage of time. As we are now introducing a temporal component to the game-play, we use the simulation in Section 3.1 as an opportunity to examine if it is reasonable to expect player expectations to converge through repeated play.

3.1 Simulating repeated play in the symmetric difficulty-based game

In constructing a simulation of repeated gameplay, we strive to offer more plausible support for the claim that the game will motivate the selection of appropriate challenges and best-effort responses from players. As the results and interpretations of any simulation depend on the implementation decisions made in constructing the simulation, we will attempt to motivate and describe implementation decisions whenever possible. We begin with a discussion of these assumptions, and then move to a discussion of the results of the simulation.

3.1.1 Simulation assumptions

We make a number of assumptions about the game itself:

- We assume that the game is played exclusively between a pair of players. The strategies that each player evolves is determined only by these games (i.e. neither player adjusts their strategies based on interactions with other players.)
- We assume that the game is iterated an infinite (or at least unknown) number of times, so impending end-games do not affect player strategy.
- We assume that gameplay is symmetric. The implication for the players is that a strategy must encompass a Teacher strategy and a Student strategy.

We also make several assumptions about each player:

- We assume that a one-parameter Rasch (Item Response Theory) Model [29] is applicable. This model assumes that we can characterize every challenge problem j according to a difficulty value β_j, can characterize each student i by some ability value θ_i, and can model the probability of an accurate response X_{ij} as follows:

  $$X_{ij}(\beta_j, \theta_i) = \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)} \tag{3.1.1}$$

  We modify this to allow the level of a student's response effort γ_i to affect the probability that a response is accurate. We therefore model effort-modulated response accuracy as follows:

  $$X_{ij}(\beta_j, \theta_i) = \gamma_i \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)} \tag{3.1.2}$$

- We assume that a Teacher strategy consists of a level of problem difficulty, and that a Student strategy consists of a level of effort exerted. Thus, a symmetric game strategy must specify both a problem difficulty to pose and a level of effort to exert in response.

- We assume that each student maintains a set of different strategies, and evolves this set over time using a hill-climbing process. Once per iteration, a slight variation on each strategy is generated (in terms of challenge difficulty, response effort, or both), and the player chooses between the original strategy and the variation based on which performs better in games against the other player. This is in contrast to the strategy-selection process assumed in Chapter 2, in which we assumed that a player selects a strategy through an evaluation of all possible strategies. Here, we simply assume that the player maintains *some* set of strategies, and improves this set over time, as (randomly-generated) opportunities arise.

- We assume that player strategies are initially randomly distributed throughout the two-dimensional space of effort and difficulty. We also randomly determine the ability level of each player before the first round.

- We assume that the hill-climbing procedure for generating variations on strategies consists of some probability that the generated strategy will vary in each dimension, and some upper limit on how large these variations may be. (The size and direction of each variation is randomly sampled within this limit.)

- We assume that student ability remains fixed over time.[1]

[1] We have performed simulations in which we assume that student ability increases at a constant

3.1.2 Simulation Observations

We now turn to simulating the symmetric-role difficulty-based game. We discuss one example to help clarify this discussion, which is illustrated in Figure 3.1.

In this example, we assume that strategy space is bounded. Specifically, we limit the parameters ranges such that challenge difficulty is bound by $-3 \geq \beta_j \geq 3$, student ability is bound by $-3 \geq \theta_i \geq 3$, and response effort is bound by $0 \geq \gamma_i \geq 1$.

Each player maintains 40 strategies at each point in time. Initially, these strategies are randomly selected (uniformly over the bounded space of challenge difficulty and response effort.) The ability level of each player is also randomly determined, and in the example illustrated, we note that Player 1 has an ability level of 0.835 and Player 2 has an ability level of -1.788. At each iteration, a variation on each strategy is generated. The probability of this variant changing the response effort level or the challenge difficulty level of a strategy is 0.05, and each change adds somewhere between -0.15 and 0.15 to the changed dimension.[2] Every strategy-variant pair is compared each iteration, and the "better" of the two is retained for the next iteration. This decision is made based on which one out-scores the other against more of the other player's strategies: Against each of the other player's 40 strategies, the current strategy and the variant strategy each receive a certain number of points (the sum of Teacher points and Student points from the difficulty-based game, as defined in Section 2.4.1.) So in each of these 40 situations, one of the two is preferred.[3] The

rate, and have performed others in which student ability increases at a rate proportional to challenge appropriateness, but examine the simplest case here for clarity of interpretation. We note that in the simulation detailed in Section 3.2, we look at a dynamic model in which student learning affects the difficulty of a particular challenge (rather than the overall ability of the student.)

[2] The size of this change is randomly selected from within this range. A ceiling and floor are imposed, to constrain the parameters within their ranges ($-3 \geq \beta_j \geq 3$ and $0 \geq \gamma_i \geq 1$.)

[3] Ties are awarded to the original strategy, rather than the variant.

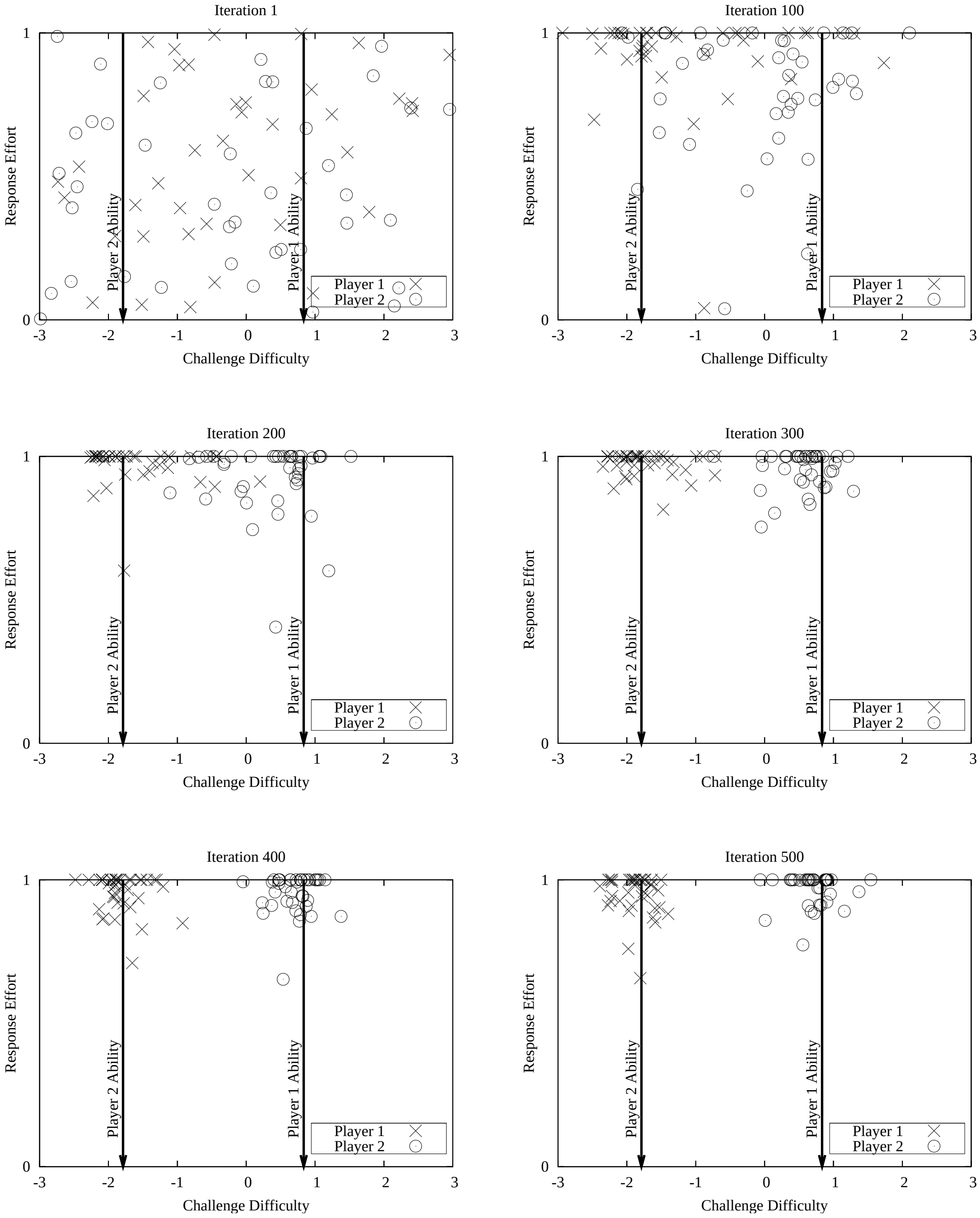


Figure 3.1: Snapshot sequence from simulation of the symmetric-role difficulty-based Teacher's Dilemma game. The ability levels of each player are represented by an arrow pointing to the level of challenge difficulty at which the player has a 50% chance of producing an accurate response. Strategies are initially generated randomly, and each player evolves these strategies over time. We note that the evolved strategies of each player consist of maximum effort and challenges at the other player's level (i.e. at which there is a 50% chance of response accuracy.)

strategy with the majority of these preferences is retained for the next iteration.[4]

In Figure 3.1, we see that the strategies evolve roughly as anticipated. While the process is slow and noisy, we note that the challenge difficulty levels of Player 1's strategies have all evolved to roughly the ability level of Player 2, and vice versa. For both players, the level of response effort has increased over time, with the majority of strategies specifying the maximum effort level. While the parameters specified in the previous paragraph affect the speed and the amount of noise in this process, the simulations consistently result in the same sort of dynamic. So while the strategies adopted by the players in the first iterations do not match the optimal strategy of a player who can simultaneously evaluate the entire space of strategies, as the analysis in Chapter 2 assumes, the strategies adopted in later iterations do approximate this optimal strategy.

3.2 Simulating repeated play in the *expectation*-based game

Previously, when we analyzed the expectation-based game in Section 2.4.2, we claimed that differences between the expectations of the Teacher and Student would be resolved with the passage of time. We now use a simulation of repeated play of that game to support this claim. As in the previous section, we begin by stating the assumptions underlying our simulated model, and then discuss a sample run of the simulation.

[4] Again, if there are equal numbers of votes for both, the current strategy is retained.

3.2.1 Simulation assumptions

We make a number of assumptions about the game itself:

- As before, we assume that the game is played exclusively between a pair of players. The strategies that each evolves is determined only by these games.
- We assume that the game is iterated an infinite (or at least unknown) number of times, so impending end-games do not affect player strategy.
- We assume that gameplay is asymmetric, and that player roles do not change between iterations.
- The game consists of a fixed set of challenge questions, which are randomly generated. At each iteration, the Teacher selects one of these questions to pose to the Student.

We also make several assumptions about the players:

- Both players maintain expectations regarding response accuracy for each question. These expectations are initially randomly generated, but are later based on observed performance. As such, the Teacher's expectation, Student's expectation, and true probability are entirely independent values at the beginning of the game. In later iterations, players' expectations are based (with some noise) on their recent memory of the Student's performance on the problem.
- The Student's response accuracy on a challenge question is determined by a probability associated with that question. This probability may increase over time, given practice. The amount of learning that occurs is a function of the notion of challenge appropriateness from the previous Chapter.

- While we now allow for learning (as described in the following section), the effect of learning is restricted to the learned problem, and does not transfer to other problems.

- At each iteration, the Teacher poses the challenge with the greatest expected utility (as illustrated in Figure 2.5.)

3.2.2 Simulation observations

As before, we provide a specific implementation of this model. Here we look at a game repeated between two players that includes 100 possible challenge questions. We assume here that each player constructs their expectations on the Student's performance on up to five of the most recent attempts, when available, with some noise added (up to 10% variation.) The amount of learning is proportional to challenge appropriateness. Recalling Equation 2.1.4, we define appropriateness as: $APPR_s(c) = \mathrm{P}\left[\mathcal{A}_{r,c}\right]\left(1 - \mathrm{P}\left[\mathcal{A}_{r,c}\right]\right)$ The amount of learning (i.e. change in true probability of an accurate response) is defined as $0.5APPR_s(c)$.

We will again discuss one example simulation. Figures 3.2 illustrates the state of the game at six different points in time, sampled at iterations 1, 50, 100, 150, 200, and 250. At each iteration, we plot the 100 challenge questions according the Teacher's expectation of response accuracy (along the x-axis), the Student's expectation of response accuracy (along the y-axis), and the actual, determining probability of response accuracy (as the color of each data point.)

From the iterations shown in this Figure, we see that the first challenges selected were those that the two players disagreed on the most, with one player expecting that the response was more likely to be correct than incorrect, and the other player

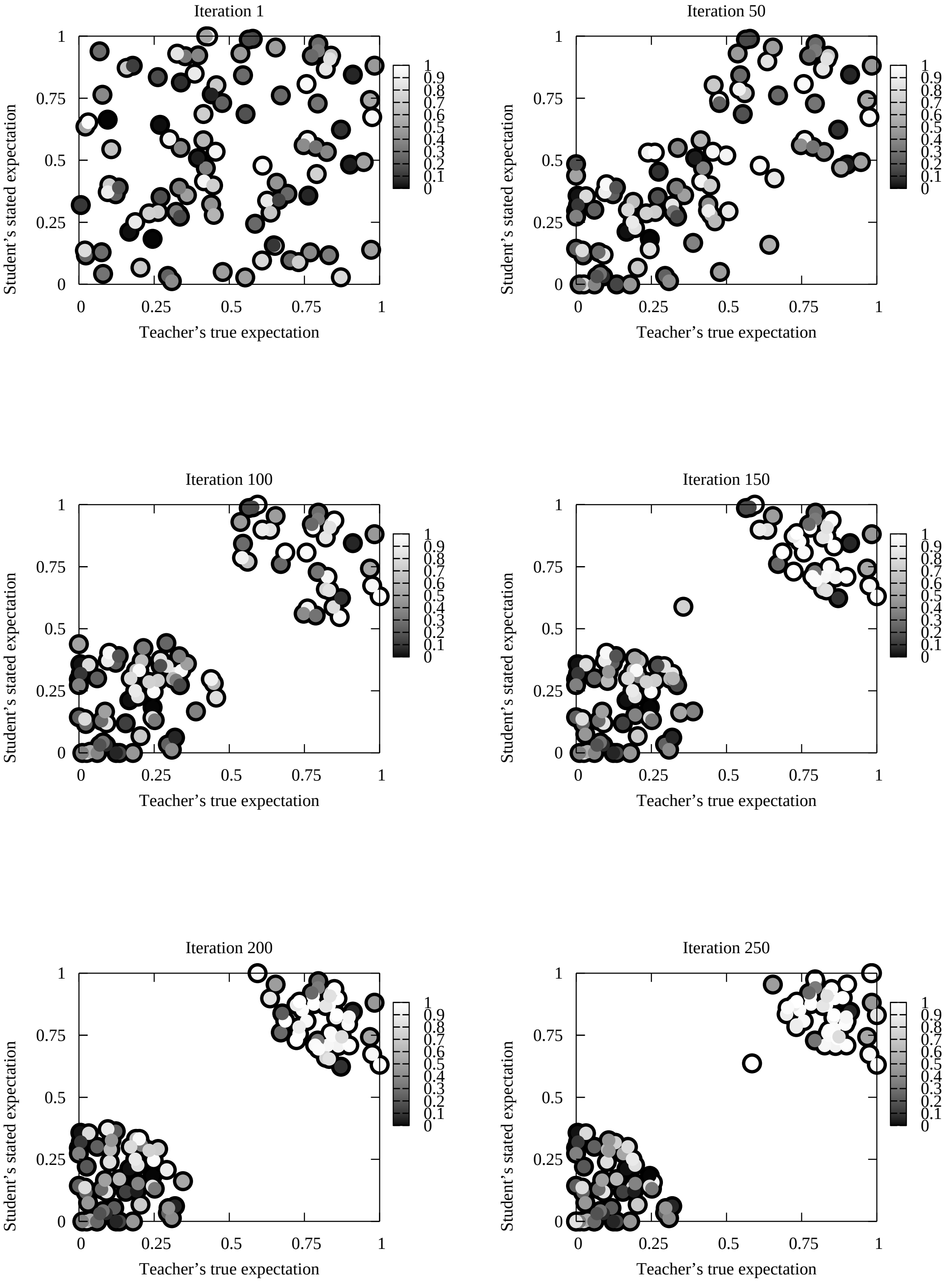


Figure 3.2: Snapshot sequence from simulation of the asymmetric-role *expectation*-based Teacher's Dilemma game.

expecting the opposite. As the two players' initial (i.e. randomly-generated) expectations for these challenges became increasing based on their common observations, these challenges moved closer to the $x = y$ diagonal line representing agreement. Between the illustrations of iteration 50 and iteration 100, we observe that the Teacher primarily selects challenges in the center of the plot, on which both players expect to be near the probability of 0.5. As the Student learns these problems, they migrate to the upper right-hand corner, leaving only the challenges that both players agree to be very difficult or very easy. Note the distribution of data point colors, indicating the true probability of an accurate response. Over the course of the six iterations shown, and continuing in unseen future iterations, the challenges in the upper right-hand corner are increasingly light (indicating high true probability) and those in the lower left-hand corner are increasingly dark. Thus, the Student and Teacher expectations increasing align with the true probability of the challenges.

In Figure 3.3, we offer a second view of this type of simulation, in which we show only the path that challenges travel over the course of a game. In this Figure, an arrow is used to indicate changes in player expectations following a response to the challenge being posed. By compiling all such arrows into a single plot, we see that expectations of the two players have largely converged for most challenges, as indicated by the concentration of arrows along the $y = x$ line. This suggests that the expectations of players repeatedly playing an expectation-based Teacher's Dilemma game do converge over time. Once these beliefs converge, the best strategy for the Teacher is to select problems that both now agree to be of appropriate difficulty.

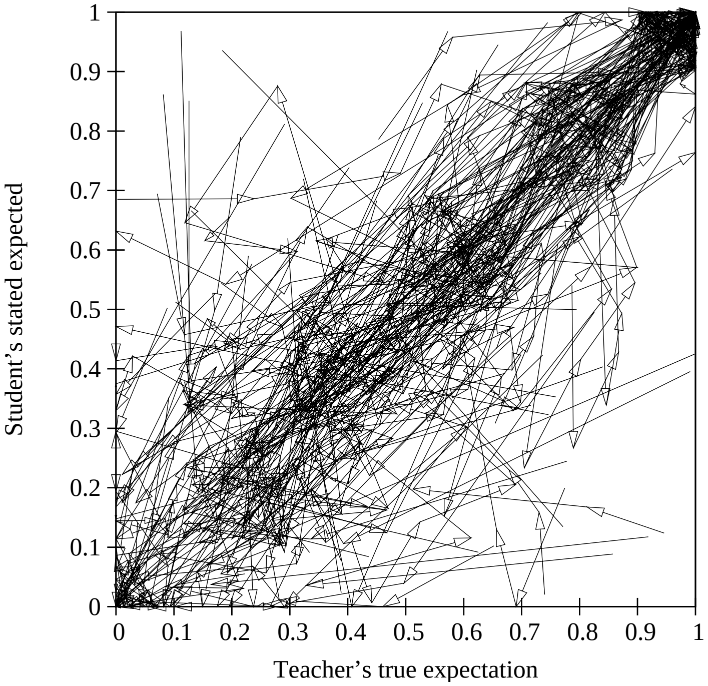


Figure 3.3: Expectation convergence in the asymmetric-role *expectation*-based Teacher's Dilemma game. Arrows indicate how the location of a challenge, with respect to the expectations of both players, changes over the course of a game.

3.3 Summary

In this Chapter, we designed a series of simple computer simulations, of Teacher's Dilemma games played repeatedly by a fixed pair of players. We used these simulations as a means to conceptualize what we might expect to see from student players. In these simulations, players varied their strategies over time in response to how they were observed to have performed, rather than attempting to initially select a globally optimal strategy (as was the case in the previous Chapter). In Section 3.1, we observed that the strategies of players in a reciprocal difficulty-based Teacher's Dilemma game (determined by challenge difficulty levels and response effort levels) converged on maximizing response effort and posing appropriate challenges (i.e. for which $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = 0.5$). In Section 3.1, we observed that the strategies of players in an expectation-based Teacher's Dilemma game had the effect of first preferring challenges for which the two players disagreed most, then preferring challenges for which the two players agreed were most appropriate, and only then preferring challenges for which the two players agreed were less appropriate. Over the course of this process, the Teacher's true expectation and the Student's stated expectation converged with the true probability of an accurate response, and also converged with one another. As such, discrepancies were primarily resolved early in the game (offering a means for improving expectations), and attention quickly shifted to posing challenges based on appropriateness, with the most appropriate challenges posed first. These simulations suggest that Teacher's Dilemma games can retain their motivational value even if the players do not meet strict game-theoretic assumptions, if the players interact repeatedly over time.

Chapter 4

Web-based systems for Teacher's Dilemma games

While the Teacher's Dilemma games offer a theoretic basis for motivating learning among pairs of players, the games themselves remain abstract constructs, offering no specification of venue, implementation, or even domain content. In this chapter, we describe several systems that we have built in order to incorporate these games into web-based activities, and make them widely available to all learners with internet access. To date, we have developed three such systems. Two are detailed in this Chapter, and a third is discussed in Appendix A. The SpellBEE activity was the first system built, designed specifically as a learning game for those interested in improving their (American-English) spelling. The second system, the BEEweb, generalized the architecture underlying the SpellBEE software into a platform on which we developed and deployed Teacher's Dilemma game-based activities for a variety of different task domains. The third system, BEEmail, provides a minimal proof-of-concept that a Teacher's Dilemma game can be built using a decentralized peer-to-peer architecture (see Appendix A.) In this chapter, we discuss the design and implementation of each of the SpellBEE and BEEweb systems.

SpellBEE and BEEweb both support a synchronous activity built on the difficulty-based Teacher's Dilemma game introduced in Section 2.4.1, played by a pairs of users located in different places. In incorporating these abstract Teacher's Dilemma games into web-based activities, we have chosen to make player roles reciprocal and game-play symmetric. We accomplish this by having the two players simultaneously engaged in two instances of the game. Each player fills the "Teacher" role in one game and the "Student" role in the other. By interleaving the active steps of these games, players can participate in both games simultaneously, and experience these as a single reciprocal tutoring experience. Figure 4.1 illustrates how the steps in the difficulty-based games are arranged, and offers a picture of our TD game-based model of reciprocal peer tutoring across the internet. This outlines the steps of a single "turn," and our activities are each based on several such turns during each peer interaction.

Several other computer-based tutoring systems have also attempted to leverage peer interactions as a context for, or basis of, learning. Where most intelligent tutoring system designs focus on a computer tutor assisting a human learner, this arrangement is but one of many ways to approach the overarching goal of learning. Chan and Chou [14] map out a much larger space of options, based on the particular constellation of number of participants involved (e.g. 2 or 3), roles each plays (i.e. teacher and learner), type of player fulfilling these roles (i.e. real/human and virtual/machine), types of support provided to these players (i.e. scaffolding), and symmetry of roles (i.e. fixed or alternating). Of these options, we have focused on reciprocal tutoring systems for two human learners. Several other recent research efforts have studied

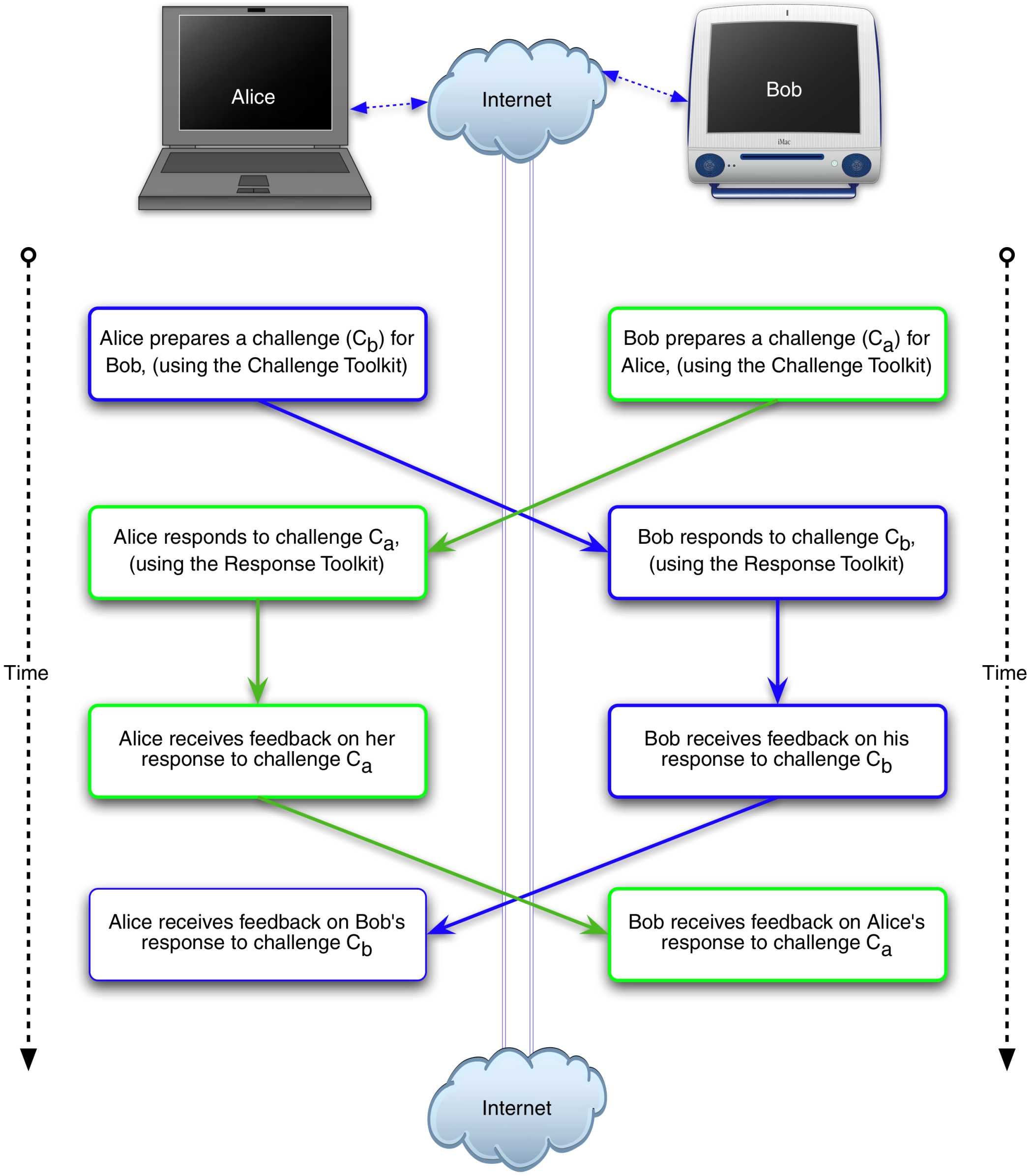

Alice
Internet
Bob
Alice prepares a challenge (C_b) for Bob, (using the Challenge Toolkit)
Bob prepares a challenge (C_a) for Alice, (using the Challenge Toolkit)
Alice responds to challenge C_a, (using the Response Toolkit)
Bob responds to challenge C_b, (using the Response Toolkit)
Alice receives feedback on her response to challenge C_a
Bob receives feedback on his response to challenge C_b
Alice receives feedback on Bob's response to challenge C_b
Bob receives feedback on Alice's response to challenge C_a
Time
Time
Internet

Figure 4.1: A web-based reciprocal tutoring activity, created by interleaving the steps of two instances of the difficulty-based Teacher's Dilemma game. The players engage in each step simultaneously, and progress through the four steps sequentially.

other tutoring system arrangements based on interactions among human learners. Walker et al. [81, 82] have explored the effects of extending a Cognitive Tutor [2] to incorporate peer tutoring. Wong et al. [86] have experimented with the effects of incorporating various types of cognitive tools in a reciprocal tutoring system. Chang et al. [18] have designed a system, Joyce, that seeks to motivate engagement through games with peers. This work is similar to ours in the shared goals of using a game to increase student motivation and of partially de-coupling a player's outcome (i.e. winning vs. losing) from their relative ability (i.e. more skilled vs. less skilled.) The work differs in that the challenges seen in our systems are selected by one student for the other student, while the challenges seen by students using Joyce system are always selected by the system itself. Joyce relies on randomness[1] in order to de-coupled the relative outcome from the relative ability of competing players, whereas we base the game scoring mechanism (for the Teacher) on the appropriateness of the challenges posed.

In this chapter, we discuss two of our own systems, presenting the goals of each, and describing the design and implementation decisions involved.

4.1 SpellBEE: A two-person spelling game

The SpellBEE system represents our first attempt to build a web-based activity based on a Teacher's Dilemma game, applied to the task domain of American-English spelling. The activity is currently accessible online at http://SpellBEE.org/, and since we publicly released it four years ago, we have collected data from over 25,000

[1] This is done in a variety of ways common in board games: random "dice rolls," path "shortcuts," and position "bumping" are three components incorporated into the dynamics of this game.

completed (i.e. fourteen-question) sessions. With this activity, we want the student to improve her ability to spell various words, after hearing the word spoken in the context of a full sentence and reading that sentence. Given the opportunity to attempt a words again at a later point in time, we would like the student to then be more likely to succeed on the later attempt. The extent to which spelling knowledge is transferable remains debatable – as we will discuss in Section 4.1.2 – but we do wish students to attempt to transfer knowledge of how certain sounds (e.g. phonemes or syllables) are written in the context of different words. Peer tutoring has been applied successfully to the spelling domain in classroom environments [26], where a game-based approach to collaboration was found to be easy to implement, inexpensive, and quite effective. Research on the difficulties involved in teaching and learning spelling has been ongoing for the past century [12], offering a rich background to draw on for the present study.

By organizing the spelling activity around the model illustrated in Figure 4.1, we provide the student with several opportunities for learning. In the challenge-selection step, the student is motivated to reflect on a set of seven words, and attempt to reason about the likelihood that their partner will be able to correctly answer each of these words. In doing so, the student may compare the structure of these words to that of other words that their partner has attempted in the past. If, for instance, one of the seven options is the word “perceive” and the student knows that their partner has recently attempted to spell “receive”, the student may leverage this recognized similarity in their decision-making process. Next, in the second step, the student hears a sentence spoken and sees the sentence on the screen. In reading along with the spoken sentence, the student is exposed to the (correct) spelling of several other words. When the student then attempts to spell the challenge word, they type in their response, and presumably examine and approve that response before submitting it. This generate-and-test process may occur several times for a particular response, until the student believes the response reflects their best effort (or until the student runs out of time.) Finally, the student is presented with feedback on the accuracy of their response. If correct, the feedback can reinforce their knowledge. If incorrect, the feedback presents the correct answer, which the student may then study briefly before the entire process repeats.

4.1.1 Applying the Teacher's Dilemma model

The SpellBEE software architecture enforces the structure of the difficulty-based Teacher's Dilemma game (introduced in Section 2.4.1) as follows: The space of legal challenges – the **C** in Figure 2.1 – is based on a list of about 3,000 words drawn from the word-list published in Greene's "New Iowa Spelling Scale" (NISS) study [33]. Each challenge problem is based on one of these words, and consists of an audio presentation of a sentence that contains the word[2], and the sentence itself, with the challenge word blanked-out, displayed visually on the student's computer screen. (The sentences were culled from a selection of children's books now in the public domain[3].) The space of legal responses – the **R** in Figure 2.1 – includes all strings of length up to twenty characters. Response accuracy – $\mathcal{A}_{r,c}$ in Figure 2.1 – is a case-insensitive string matching test, with 1 indicating a match between the

[2] Initially, approximately 300 sentence audio files were recorded while being read aloud by a person. We later added several thousand additional sentence audio files, generated automatically using commercial text-to-speech software.

[3] SpellBEE sentences were parsed from de Saint-Exupery's *The Little Prince*, and from several books accessed through Project Gutenberg [38]: *Aesop's Fables* (EText-No. 28), *Peter Pan in Kensington Gardens* (EText-No. 1332), *Dorothy and the Wizard of Oz* (EText-No. 420), *Alice in Wonderland* (EText-No. 11), *The Jungle Book* (EText-No. 236), and *Treasure Island* (EText-No. 120).

challenge word string and the typed response string, and 0 indicating some mismatch between the two. Finally, the difficulty metric for challenges – $\mathcal{D}_c$ – is also based on the data collected in Greene's study, as is described in detail in Section 4.1.2.

4.1.2 On problem difficulty in the spelling domain

English spelling, the learning domain addressed by SpellBEE, is known to be a challenging one. Cahen, Craun, and Johnson [12] offer an overview of earlier efforts to understand and predict spelling difficulty. One focus has been on the regularity in the mapping between phonemes (units of sound) and graphemes (the written form of each phoneme.) Hanna et al. [37] tested the performance of a model for predicting spelling based on phonetic information, and reported 49% whole-word spelling accuracy based on a 200-rule model. Simon and Simon [72] noted that at this accuracy level, memorizing these 200 rules would negligibly improve the spelling performance of fourth grade students. While more recent attempts have attained much better results, such as the Damper et al. [24] approach based on the expectation-maximization algorithm achieving 82.3% whole-word accuracy, American-English spelling remains an irregular and challenging task, both for researchers to model and for young students to learn.

Greene's 1954 New Iowa Spelling Scale study began with an effort to generate a core list of words that were widely used in written communication, primarily drawn from prior studies, resulting in a list of about 5500 words. Using this list of words, Greene launched a nationwide study in order to estimate the spelling difficulty of each of these words. This involved approximately 230,000 students in 8,800 classrooms across 645 school systems, with each student spelling 100 words, for a total of over 23 million word spellings [33]. The resulting data reports the percentage of students,

Table 4.1: Data from Greene's 1954 study is shown for the first 20 words in the SpellBEE dictionary. The table lists the percentage of students in the grade level specified in the columns who correctly spelled each of the words in the rows. Blank spaces indicate word-grade combinations for which no data was collected.

Grade	2	3	4	5	6	7	8
abandon	2	3	10	18	34	43	49
ability	1	2	6	19	30	57	71
able	2	21	56	76	86	95	97
about	9	51	78	91	97	97	99
above	4	29	59	76	84	94	97
abroad		10	13	37	54	69	86
absence		4	6	14	28	49	50
absent	1	2	6	13	25	41	56
absolute	1	2	6	13	25	41	56
absolutely		1	2	3	11	16	41
abstract				3	8	16	34
abundant		3	4	14	20	34	47
abuse	1	7	27	40	56	68	80
accept		1	4	20	42	60	61
acceptable			3	5	21	39	48
acceptance	1	2	3	6	11	25	44
accepted		3	5	13	33	52	65
accepting	2	3	3	6	22	38	49
accident		2	5	21	44	60	76
accidents		1	8	14	41	54	66

for each grade level from 2 to 8, who spelled each word correctly. Table 4.1 presents a subset of the resulting data set.

While Greene's study was published over 50 years ago, our analysis suggests that it remains a relatively good predictor of American-English spelling difficulty [5]. In SpellBEE, challenge difficulty is calculated based on the NISS data. For a student in grade level g, the difficulty of challenge word c is defined as the probability that a student in the NISS study at the specified grade level did not correctly spell the challenge word:

$$\mathcal{D}_c = 1 - \frac{NISS[g][c]}{100} \tag{4.1.1}$$

For example, based on the data included in Table 4.1, we say that the difficulty of the challenge word "absolutely" is 0.99 for a third grade student, but only 0.69 for an eighth grade student.

4.1.3 Design goals

Now that each of the variables and functionality necessary to implement the Teacher's Dilemma game has been established, we must provide a system in order to enable players to initiate and interact with these games. Two overarching goals informed the design of this system:

- *The system should enable synchronous games between users' computers with as few computer configuration requirements as possible.* We adopted a web-based client-server architecture to avoid requiring that the user has permission to download and run applications on their machine. The client portion of the code is contained in an unsigned Java applet, so a Java Virtual Machine must be installed, configured, and enabled within the web browser. As most browsers are already configured as such upon installation, this generally requires no additional effort on the part of the user, and as the Java applet is platform independent, we make no requirements about which browser, operating system, or machine type. As unsigned Java applets can only communicate with the server from which they were loaded, messages are routed between players via a centralized server. This connection does occur on a non-standard port, and so we do introduce the requirement that any firewall software must enable communication over this port with the SpellBEE server.[4] Finally, browser cookies are often used to keep track of a user's session, but if cookies are disabled we instead maintain session information within the URL query string. By implementing our system in these ways, we are able to provide users with easy access to the game from any computer, included those offering limited control over configuration settings, as is often the case with computers in schools and after-school programs.

- *The system should protect the personal safety and data privacy of the student participants.* Our student users are primarily minors, and so we are very conservative on safety issues. As a general rule, we don't allow any form of direct communication between users: There is no part of the game in which one player can type a message that is displayed on the screen of another player.[5] There is no chat-like client incorporated into the game environment and there are no message boards or other venues for asynchronous communications on the website. Interactions between players are strictly limited to the game-play itself, such as creating and solving challenges. These limited actions are, themselves, limited: The selection of a challenge word is made from among a sampling of pre-approved words in Greene's list (rather than being typed in by the players themselves), and instead of sharing responses directly with the tutor in the feed-

[4] Also implied is the requirement that the user has network access to connect to the SpellBEE server.

[5] Arguably, the one exception to this rule is that players see the login name of other players. Each names consists of a unique alphanumeric string of up to 10 digits in length. After being validated as both unique and not containing any obscenities, the username is immutable, and so cannot effectively serve as a venue for communicating messages.

back step, only the response accuracy (i.e. true or false) is shared.[6] In limiting the activity in these ways, we are able to allow anyone to participate (rather than restricting participation to classroom teachers and their pre-approved list of students), without compromising student safety. We have also taken a conservative approach to data privacy. We collect no personally-identifying information from students.[7] All players select a unique pseudonym[8] and a password when registering a user account, and all subsequent self-reported user data collected – grade level, location, and gender – are associated only with this pseudonym.

In designing the SpellBEE system around these two overarching goals – to enable synchronous games between users' computers with as few computer configuration requirements as possible, and to protect the personal safety and data privacy of the student participants – we have attempted to design a system for use primarily by elementary school students. In the following section, we look at the set of specific implementation decisions that define what the student sees when they participate in the SpellBEE activity.

[6] In the future, we could offer a finer-grained view of response accuracy by noting the accuracy of each sub-problem within the challenge. We will offer a detailed discussion of sub-problem accuracy in Section 5.5, and the concepts developed there could be applied here. For example, a word response could be labeled in such a way as to indicate the spelling accuracy of each syllable or grapheme within that word. We should note that while the response accuracy report is limited for the tutor, it is not limited for the tuteee. The tutee sees their own response, the accuracy of that response, and the correct response.

[7] The student may identify, during the registration process, the username and/or email address of a teacher or parent. The teacher or parent username specified is then granted access to view the spelling history of the student.

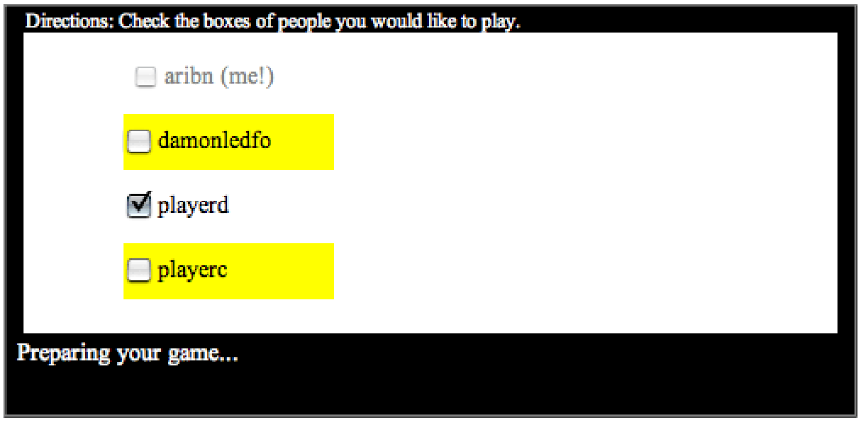
[8] Each names consists of a unique alphanumeric string of 3–10 digits in length. The server validates whether selected names are of the right length, are unique, and do not contain any obscenities.

4.1.4 User interface implementation decisions

After creating an account and using it to log into the SpellBEE website, the user is able to choose a partner with whom to begin a match. In order to guarantee that players are satisfied with the matches made, both parties must approve a match. We do this by presenting each user with an interface through which they may extend and rescind offers, and accept or ignore offers extended by other players. Figure 4.2 shows SpellBEE's player-matching interface for performing these actions. The interface is non-blocking, as the player need not wait for a response to one action before performing another. Once an offer is extended by one player and accepted by the other, both players are removed from all players' lists, and a game between the two is initiated. In order to bound the number of messages sent to all players being matched, the server imposes a maximum limit on the number of other players displayed on any one player's screen. This is achieved server-side: All available players (i.e. logged in and not currently in a match) are randomly assigned to a node in an ordered list (e.g. i), and for a limit of $2k$, each player is only made aware of – and sent notification messages regarding – the $2k$ other players whose node indices range from $i - k$ to $i + k$.

Once the game begins, the player is presented with a user interface for selecting a challenge for their partner, shown at the top of Figure 4.3. Seven words are randomly selected from the dictionary of about 3,000 words, and the user is prompted to click to choose one of them. Next to each word choice is a pair of point values, indicating the number of points that the player will receive if their partner provides an incorrect response (shown at left, in red) or a correct response (shown at right, in green.)[9]

[9] This color and bar interface is not ideal, but did remain consistent throughout all testing.



Directions: Check the boxes of people you would like to play.

□ aribn (me!)

□ damonledfo

☑ playerd

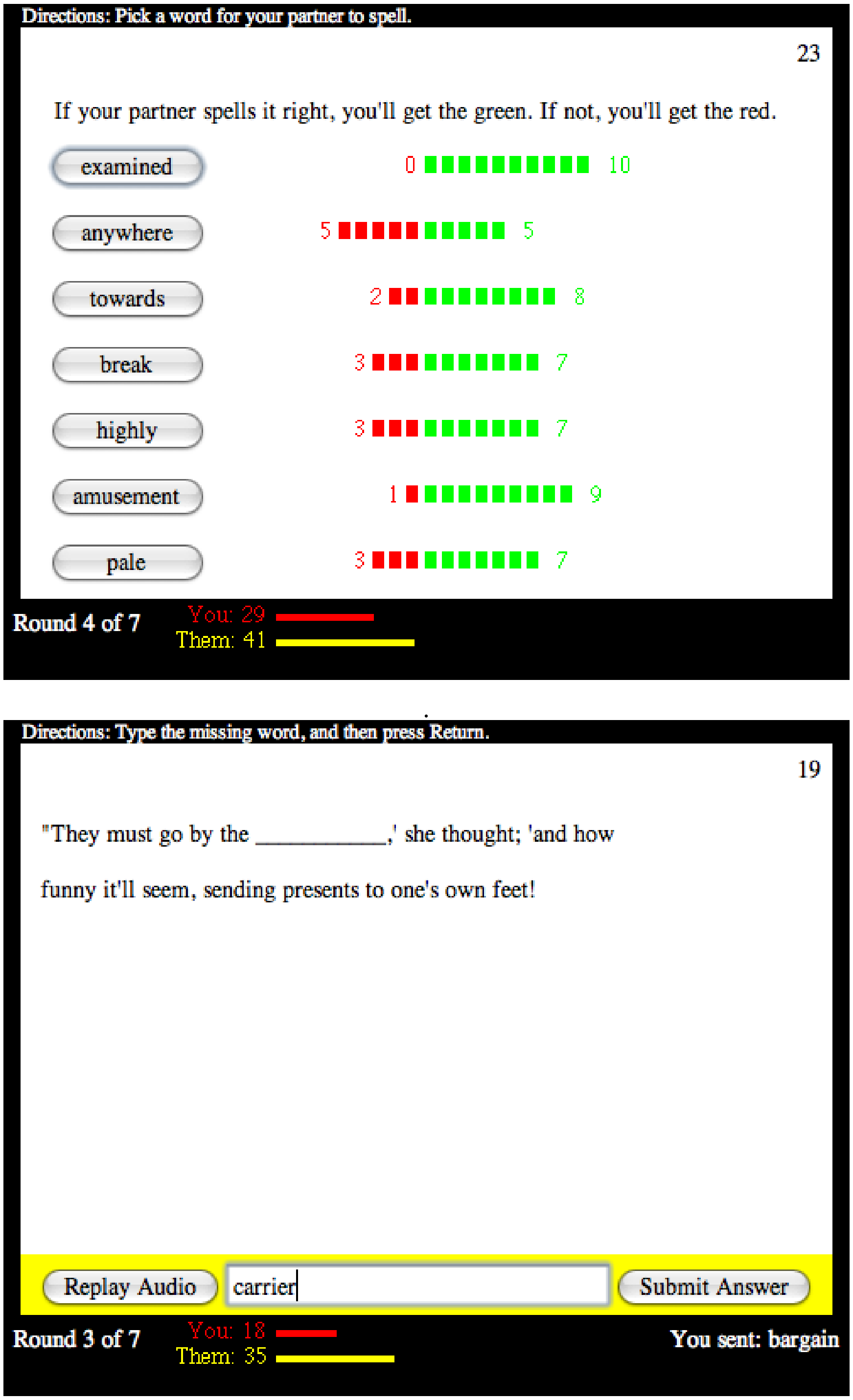
□ playerc

Preparing your game...

Figure 4.2: The player-matching interface in SpellBEE. Each available player (i.e. logged in, but not currently matched in a game) is listed. The checkbox next to each name can be used to extend an offer for a match (by checking the box) or rescind a previously-extended offer (by un-checking the box). When an offer is received from another player, their name is highlighted in yellow. By checking a box associated with one such player, a game is initiated.

The response interface is shown at the bottom of Figure 4.3. A sentence containing the challenge word is displayed, with the challenge blanked-out. An audio recording of the sentence being spoken is played (the user can re-play the audio recording, if desired.) The user types their response into a text field at the bottom. In both the challenge and response screens, a 30-second timer (in the upper-right corner) imposes a time limit on taking action.

Finally, two screens are used to provide feedback on performance. One tells the tutee if their response was correct (and, if not, what the correct response was). The second tells the tutor if their tutee correctly answered the challenge problem that they posed. This provides the tutor with a basis for revising their challenge-selection process for future questions. We note that learning about a tutee from this type of interactive feedback can be a slow process, and the end of the game may approach



Directions: Pick a word for your partner to spell.

23

If your partner spells it right, you'll get the green. If not, you'll get the red.

examined	0	10
anywhere	5	5
towards	2	8
break	3	7
highly	3	7
amusement	1	9
pale	3	7

Round 4 of 7

You: 29

Them: 41

Directions: Type the missing word, and then press Return.

19

"They must go by the ___________,' she thought; 'and how

funny it'll seem, sending presents to one's own feet!

carrier

Round 3 of 7

You: 18

Them: 35

You sent: bargain

Figure 4.3: The challenge selection interface (top) and response interface (bottom) screens in the SpellBEE activity.

before the tutor has constructed a picture of their tutee's abilities. In order to speed up this process, we provide tutors with their tutee's recently-spelled word list, in which response accuracy is noted for up to 20 recent problems. Figure 4.4 shows the information that we show players about their partner.

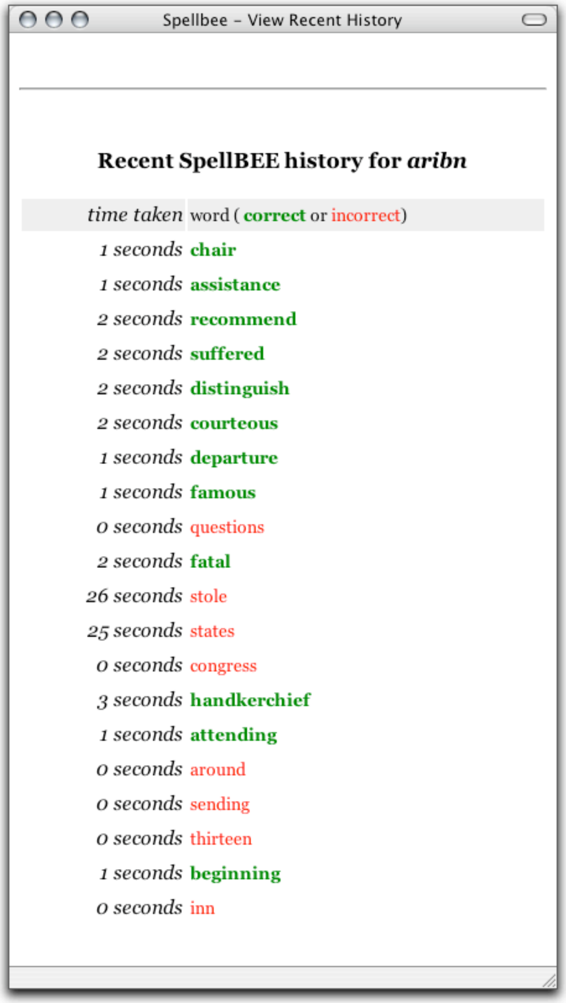
A number of additional implementation decisions affect game play.

- *Matches are repeated games.* We note that while the difficulty-based Teacher's Dilemma game was presented as a one-shot game, we implement it in SpellBEE as a repeated game. During each two-player match, seven rounds of challenge-response-feedback occur before the game concludes and the players are returned to the matching screen.

- *Challenge selections are sampled.* As the space of challenges in the SpellBEE activity encompasses about 3,000 different words, we limit the tutor's task to that of selecting from a much shorter list of choices. The challenge selection screen includes seven different words, randomly selected from the 3,000. The difficulty values associated with each of these words is used to calculate and show the tutor how many points they will receive if the response is correct or incorrect.

- *Payoff values are scaled.* The scores for both the Teacher and the Student range from 0–10 (rather than from 0–1, as defined in the abstract difficulty-based game.) Additionally, the points that a player earns in each of the seven rounds and both of the player roles (i.e. Teacher, Student) are accumulated into a single score.

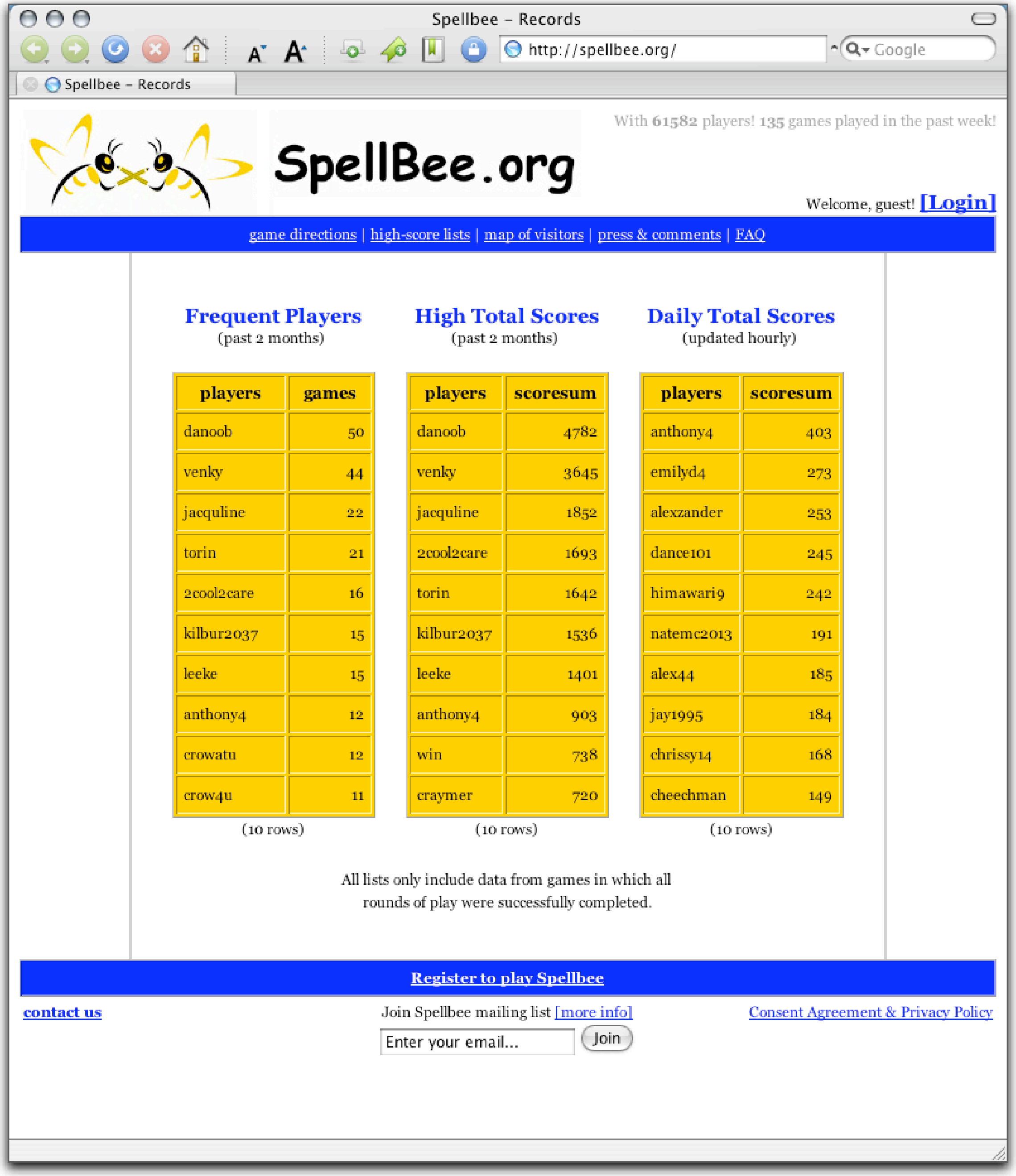
- *Actions are time-limited.* The challenge and response interfaces are each lim-



Recent SpellBEE history for *aribn*

time taken	word (**correct** or incorrect)
1 seconds	**chair**
1 seconds	**assistance**
2 seconds	**recommend**
2 seconds	**suffered**
2 seconds	**distinguish**
2 seconds	**courteous**
1 seconds	**departure**
1 seconds	**famous**
0 seconds	questions
2 seconds	**fatal**
26 seconds	stole
25 seconds	states
0 seconds	congress
3 seconds	**handkerchief**
1 seconds	**attending**
0 seconds	around
0 seconds	sending
0 seconds	thirteen
1 seconds	**beginning**
0 seconds	inn

Figure 4.4: A tutee's recent history is shared with their tutor, in order to provide the tutor with a starting point for assessing the tutee's abilities and choosing appropriate challenges. The words displayed in bold type in green signify correct responses, and the words in red signify incorrect responses. The "time taken" is counted from when the student begins typing[11] until when they submit their response.



With 61582 players! 135 games played in the past week!

Welcome, guest! [Login]

game directions | high-score lists | map of visitors | press & comments | FAQ

Frequent Players
(past 2 months)

players	games
danoob	50
venky	44
jacquline	22
torin	21
2cool2care	16
kilbur2037	15
leeke	15
anthony4	12
crowatu	12
crow4u	11

(10 rows)

High Total Scores
(past 2 months)

players	scoresum
danoob	4782
venky	3645
jacquline	1852
2cool2care	1693
torin	1642
kilbur2037	1536
leeke	1401
anthony4	903
win	738
craymer	720

(10 rows)

Daily Total Scores
(updated hourly)

players	scoresum
anthony4	403
emilyd4	273
alexzander	253
dance101	245
himawari9	242
natemc2013	191
alex44	185
jay1995	184
chrissy14	168
cheechman	149

(10 rows)

All lists only include data from games in which all rounds of play were successfully completed.

Register to play Spellbee

contact us

Join Spellbee mailing list [more info]

Enter your email... Join

Consent Agreement & Privacy Policy

Figure 4.5: Screenshot of the high score lists in SpellBEE. These lists help to encourage active participation.

ited to 30 seconds of activity (at which point challenges are randomly selected, responses are submitted as-is.) This bounds the amount of time a player ever has to wait for their partner before the game will continue.

The SpellBEE system design goals and implementation decisions described served to inform the development of a second system, the BEEweb.

4.2 BEEweb: A platform for developing Teacher's Dilemma games

In order to demonstrate that the Teacher's Dilemma game-based approach can be successfully applied to other task domains, the BEEweb was developed as a general platform on which to build web-based tutoring activities for various task domains. The BEEweb platform itself provides much of the functionality common to all of the hosted activities: an application programming interface provides game developers with a way to take advantage of the platform's relational databased-based persistent storage mechanism, intra-client messaging system, and player matching facilities. Each BEEweb activity is accessible through a unique URL (e.g. `http://PatternBEE.org`, `http://MoneyBEE.org`, `http://GeograBEE.org`), but all share the same rich web infrastructures (outside of the game client applet): account-based access supports both users and classroom groups, providing users with the ability to review their own historical activity, and providing teachers with the ability to manage groups and track progress for individual students.

4.2.1 BEEweb learning activities

To date, three learning activities have been built on top of the BEEweb platform. Each of these activities shares the interaction flow illustrated in Figure 4.1, but differs on how each of the Teacher's Dilemma game variables are defined, including $\mathbf{C}$ (the space of challenges), $\mathbf{R}$ (the space of responses), $\mathcal{A}_{r,c}$ (the response accuracy function), $\mathcal{D}_c$ (the challenge difficulty function), and the user interface toolkits for interacting with the challenges and responses. Thus, we will describe each activity in terms of how these variables are defined.

PatternBEE

PatternBEE – publicly accessible at `http://PatternBEE.org` – focuses on spatial reasoning, with an activity that is loosely based on Tangram puzzles. The task is, given a set of geometric tiles and an outline of a larger geometric shape, to fit the tiles into the outline without any tiles overlapping other tiles or extending outside of the outline. In making this a two-person activity, we ask the first participant, the Teacher, to create an outline (i.e. the challenge for the Student). The challenge toolkit that we provide for the Teacher to use in doing this allows them to select between one and six tiles, and to drag, rotate, and flip these as desired into a non-overlapping arrangement.[12] We then ask the second participant, the Student, to attempt solve this challenge by arranging tiles into a pattern that shares the outline of the challenge. The response toolkit shows this outline, and provides the Student with tiles to drag, rotate, and flip into position.[13] Response accuracy is computed

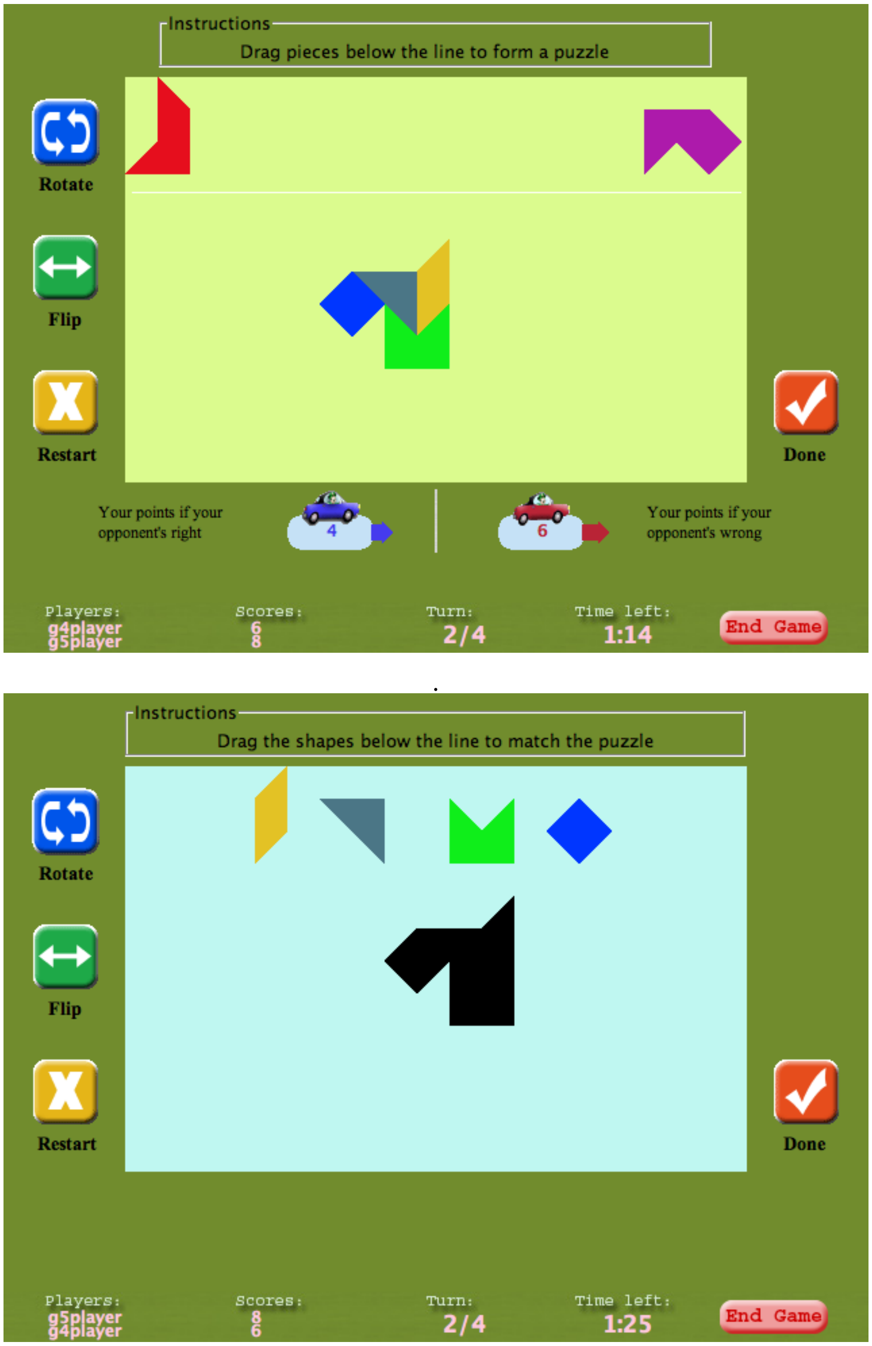
[12] We note that while SpellBEE challenge selection involved choosing from a list of seven items, challenge selection in PatternBEE is a much less restrictive, more open-ended, activity.

[13] Only those tiles used by the Teacher in forming the challenge are included, so every available tile will appear in the solution.

by comparing the outlines created by the tile patterns of both players. If they match exactly, the response is considered correct. If not, it is considered incorrect. We note that there are always several tile arrangements that cast the target outline, and any one of these is considered correct. Finally, the challenge difficulty in PatternBEE is estimated roughly, as no equivalent of Greene's study is available for this spatial-reasoning domain. Based on the observation that angular outlines reveal more about their makeup than smooth-edged outlines, we base our challenge difficulty estimates on the "edginess" of outlines. We estimate challenge difficulty based on the number of tiles used and the length of the perimeter of the overall outline. Figure 4.6 shows the toolkits for interacting with challenges and responses in PatternBEE, and Figure 4.7 shows a screenshot of part of the interface for reviewing past activity.

MoneyBEE

MoneyBEE (online at `http://MoneyBEE.org`) is a game that uses coins to create simple math problems of a particular form (e.g. "I'm thinking of 5 coins that sum to 55 cents, what are they?") A challenge is a tuple, specifying the combined value and count of some assortment of coins. A response is a 4-tuple, specifying the quantity of each type of coin (i.e. quarter, dime, nickel, penny). The challenge toolkit consists of an interface to add and remove up to 5 quarters, 5 dimes, 5 nickels, and 10 pennies from the assortment of coins. The response toolkit is similar, but also includes a statement of the challenge (in terms of number of coins and combined value.) Both toolkits are displayed in Figure 4.8. In this activity, challenge difficulty estimates are based on the number of steps required for a heuristic search algorithm to identify the solution. We note that, based on the constraints on the number of each type of coin, this domain includes 2376 unique challenges.



Instructions

Drag pieces below the line to form a puzzle

Your points if your opponent's right 4

6 Your points if your opponent's wrong

Players: g4player g5player

Scores: 6 8

Turn: 2/4

Time left: 1:14

Instructions

Drag the shapes below the line to match the puzzle

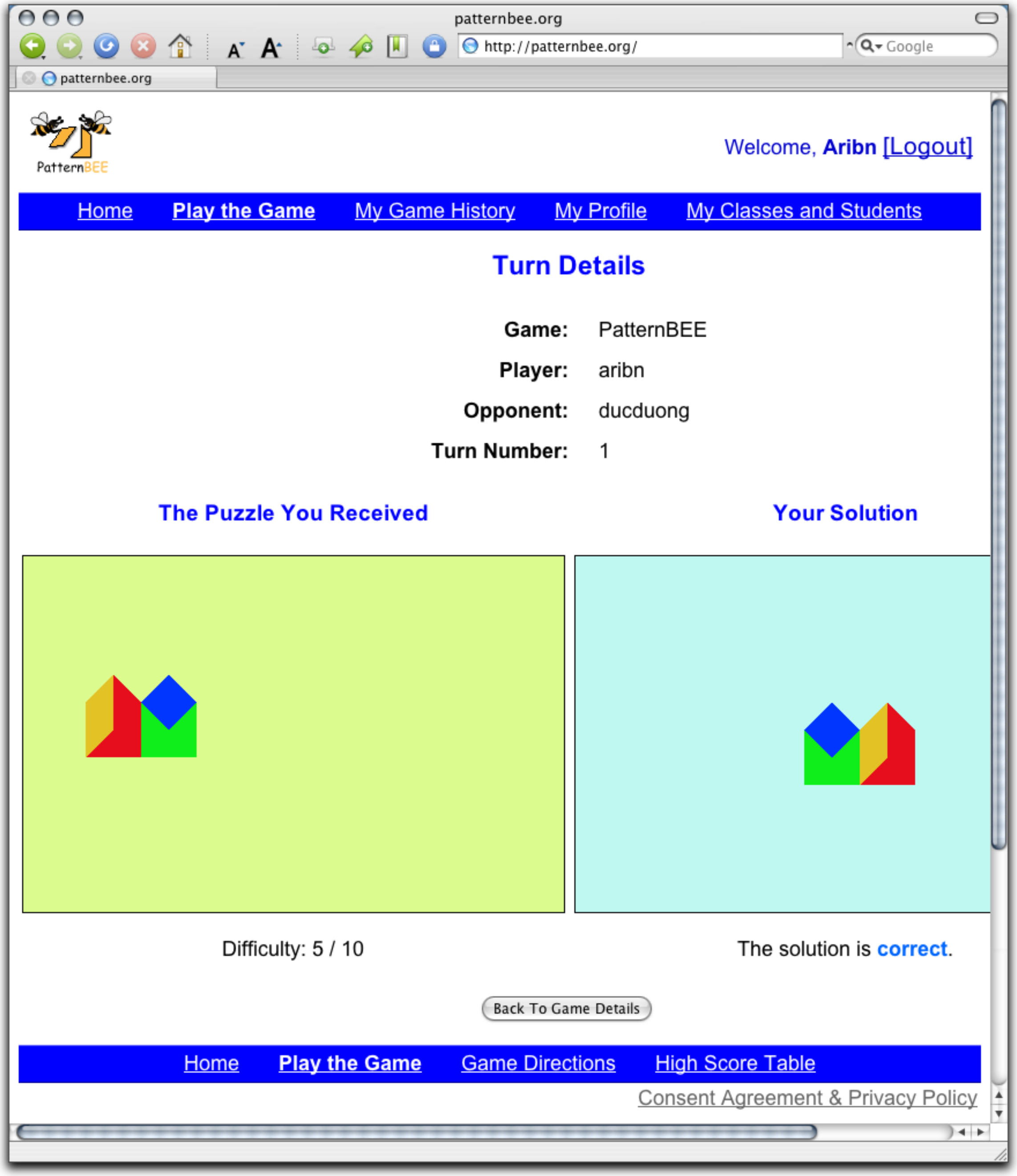
Players: g5player g4player

Scores: 8 6

Turn: 2/4

Time left: 1:25

Figure 4.6: The challenge selection interface (top) and response interface (bottom) for the PatternBEE activity.



Welcome, **Aribn** [Logout]

Turn Details

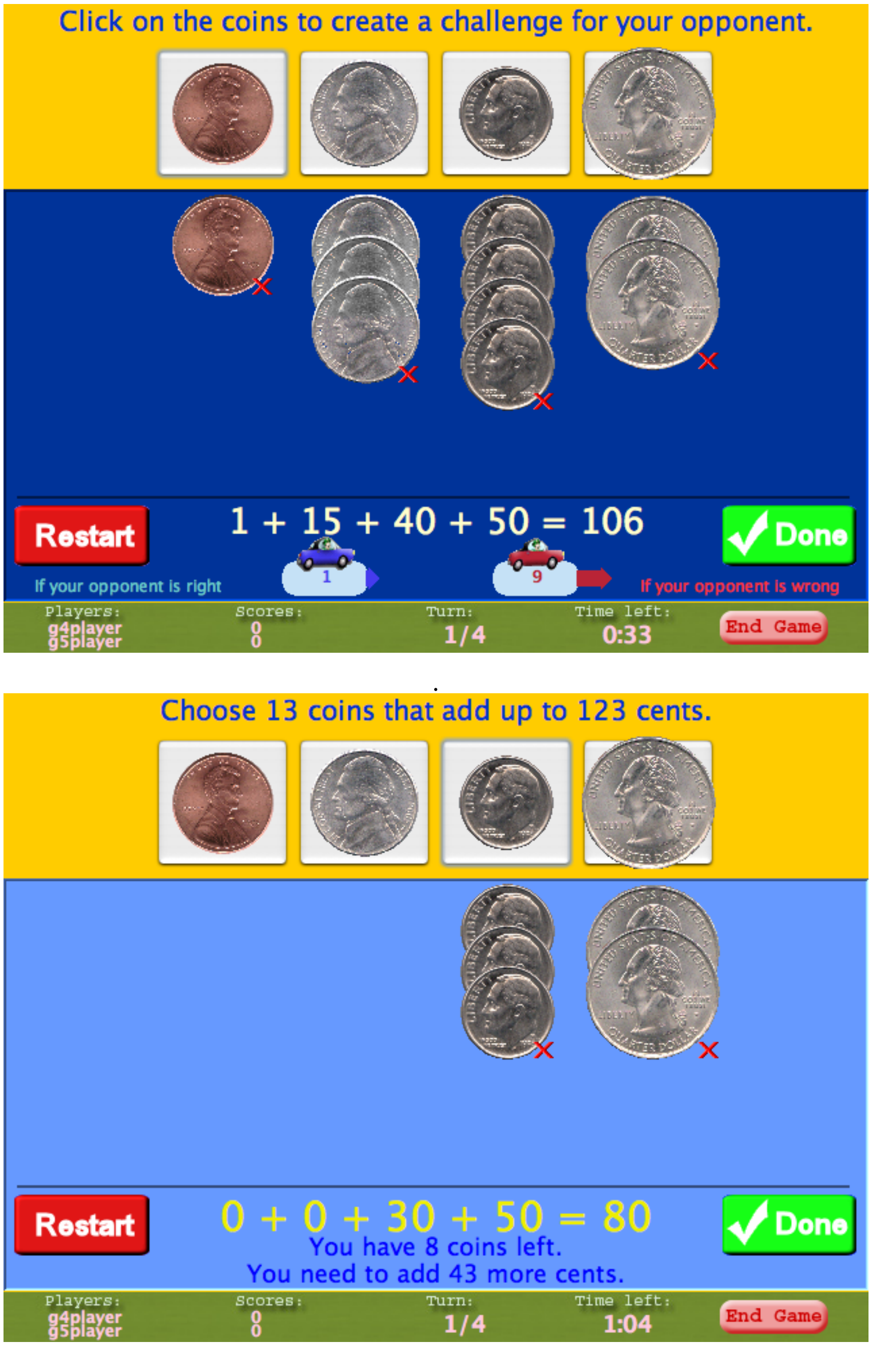
Game:	PatternBEE
Player:	aribn
Opponent:	ducduong
Turn Number:	1

The Puzzle You Received — **Your Solution**

Difficulty: 5 / 10

The solution is **correct**.

Figure 4.7: Screenshot of historical view of a PatternBEE challenge. Note that although the challenge and response were not identical, the solution is considered accurate.



Click on the coins to create a challenge for your opponent.

1 + 15 + 40 + 50 = 106

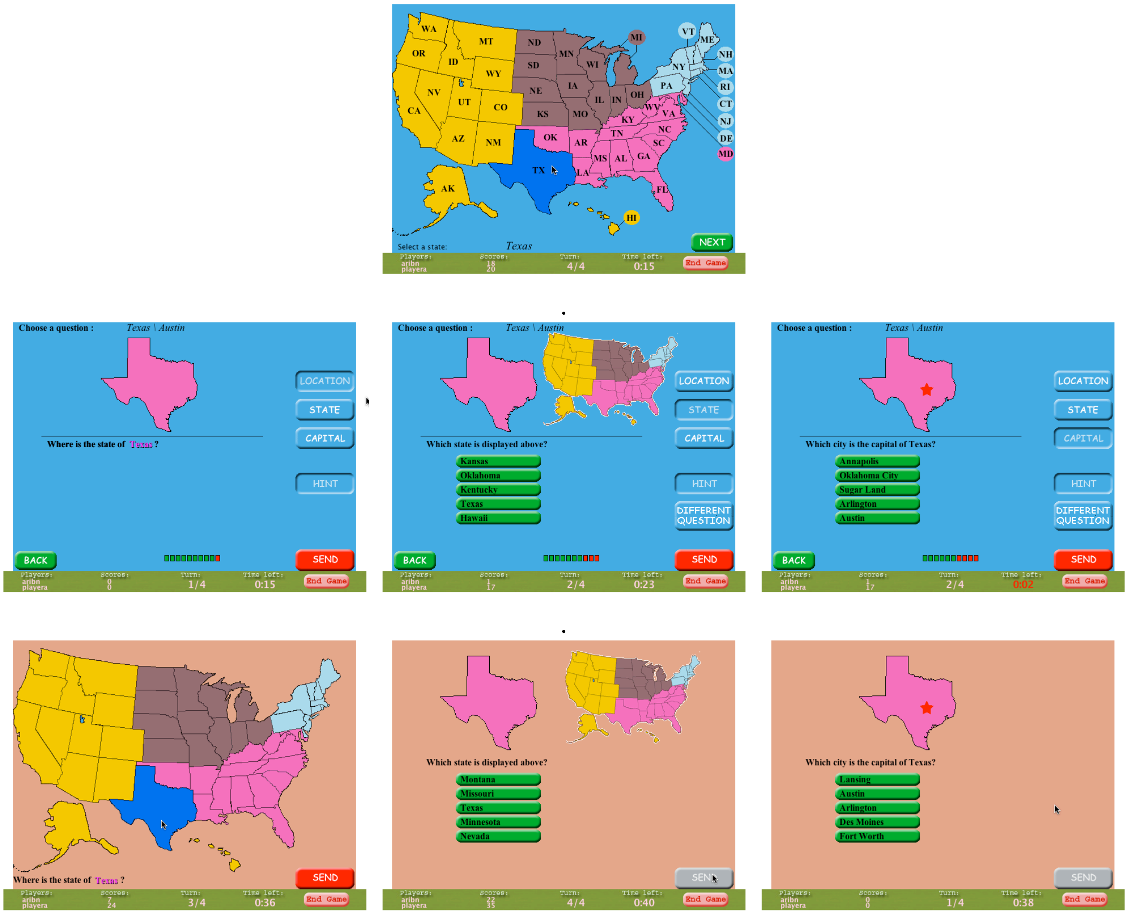
Choose 13 coins that add up to 123 cents.

0 + 0 + 30 + 50 = 80

You have 8 coins left.

You need to add 43 more cents.

Figure 4.8: The challenge selection interface (top) and response interface (bottom) for the MoneyBEE activity.



Where is the state of Texas ?

Which state is displayed above?

Kansas
Oklahoma
Kentucky
Texas
Hawaii

Which city is the capital of Texas?

Annapolis
Oklahoma City
Sugar Land
Arlington
Austin

Where is the state of Texas ?

Which state is displayed above?

Montana
Missouri
Texas
Minnesota
Nevada

Which city is the capital of Texas?

Lansing
Austin
Arlington
Des Moines
Fort Worth

Figure 4.9: After selecting a state (top), the tutor selects one of three types of questions for the GeograBEE activity. This determines which of the three challenge selection interfaces (middle) and corresponding response interfaces (bottom) are used.

GeograBEE

GeograBEE (online at `http://GeograBEE.org`) focuses on a geographical knowledge domain, in which challenges are each questions about one U.S. state. Three different challenges types are available for the Teacher to choose among. The GeograBEE challenge toolkit, displayed in Figure 4.9, is broken into two steps: the Teacher first selects a state from a map and then selecting one of three types of questions to ask about that state. These involve:

- Identifying the capital city of the specified state, from among a list of five U.S. cities (multiple choice question).
- Locating the specified state on a U.S. map.
- Identifying a state (by name) based on an illustration of its geographical shape, from among a list of five possible states (multiple choice question).

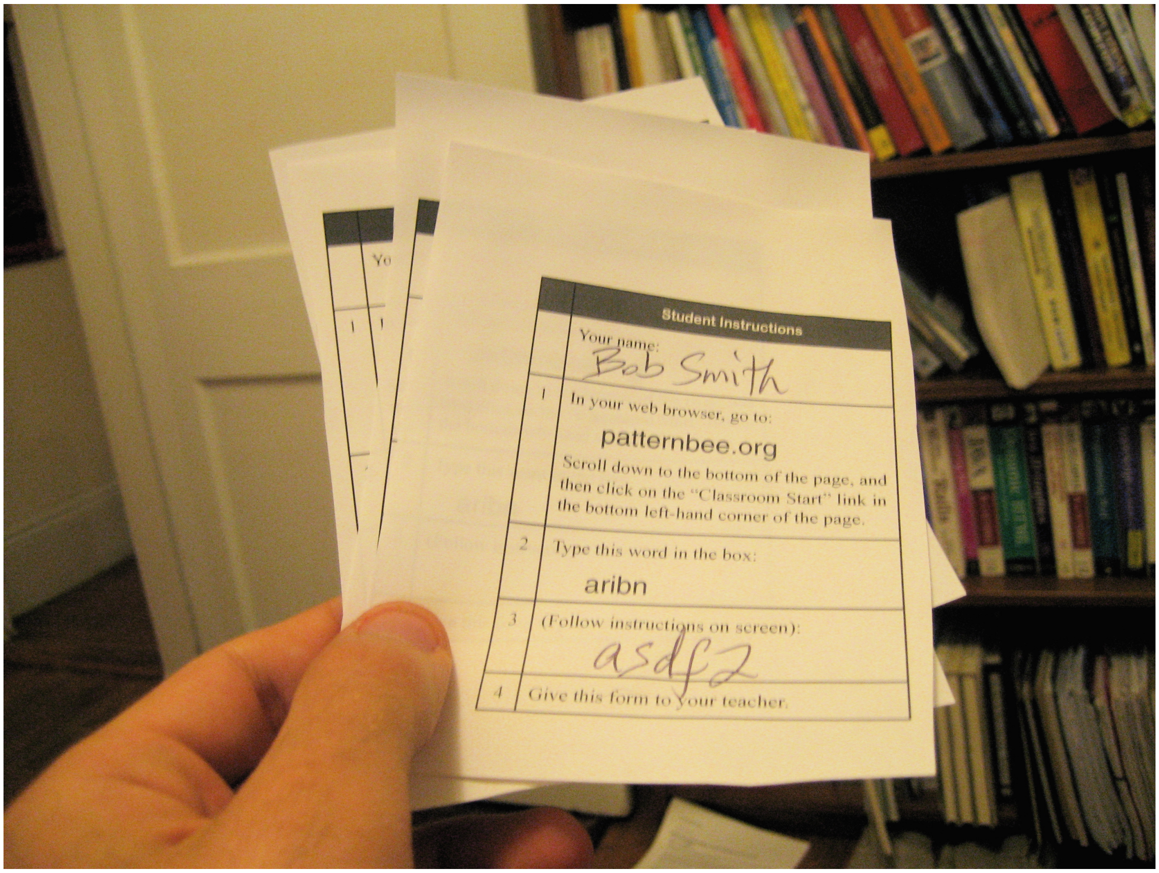
The response toolkit for the identification-based questions displays the question in multiple-choice form, from which the tutee must make a selection. The response toolkit for the location-based questions displays the question and presents the user with a map, upon which they click on a state to respond. Challenge difficulty estimation in GeograBEE takes into account the tutee's location, and the category, and the relative locations of the answer options: The distance from the user's home state to the challenge state affects difficulty, with challenges "closer to home" marked as less difficult than those further away. Additionally, for the capital identification questions, the geographical distance between the incorrect answers and the correct answers affect difficulty, so a multiple choice question with five cities in one state has a higher difficulty rating that one with five cities in five different states.

4.2.2 Design decisions

In developing a system for students to use either out-of-school, during free time, or in school as part of a teacher-directed classroom activity, we faced several unique design challenges. We present two of them here, and discuss the solutions that we designed and implemented as part of the system.

In designing the BEEweb platform, we sought to provide teachers with tools to view their student's participation and progress. While SpellBEE offers only rudimentary tools, the BEEweb provides teachers with a hierarchical set of views of their classroom groups, summary statistics and progress visualizations, and question-by-question challenge and response reports for each student match. In making student data available to their classroom teacher, we faced a usability problem. As user accounts are purely pseudonymous, teachers cannot necessarily recognize which username corresponds to which student. We solve this identity-mapping problem by implementing a modified sign-up process for classroom usage: After selecting their pseudonym, each student is prompted to write both this username and their own real name on their instruction sheet, and then to hand this sheet back to their teacher. This process generates an offline, paper-based mapping between student names and game pseudonyms for the teacher to refer to later, as needed. Figures 4.10-4.11 shows this process in action, after the completed forms have been re-collected.

A second challenge that we faced, indicated by user feedback from the SpellBEE activity, is that users were frustrated when they logged into the game and found no one else with whom to play. In response, the BEEweb provides access to "practice rounds" in this situation. This option becomes available only if no other players are present, and offers the user with a challenge to solve, drawn randomly from the database of challenges previously constructed by players in past games. Practice rounds are only available as long as other players are not present, and disappear once other players arrive. The goal here is both to lower user frustration and to increase the amount of time that players wait for a partner before giving up and logging out. A second technique that we have adopted to assist users in successfully starting a game with another player is to make game activity information (i.e. active player counts)



Student Instructions

Your name: Bob Smith

1 In your web browser, go to:
patternbee.org
Scroll down to the bottom of the page, and then click on the "Classroom Start" link in the bottom left-hand corner of the page.

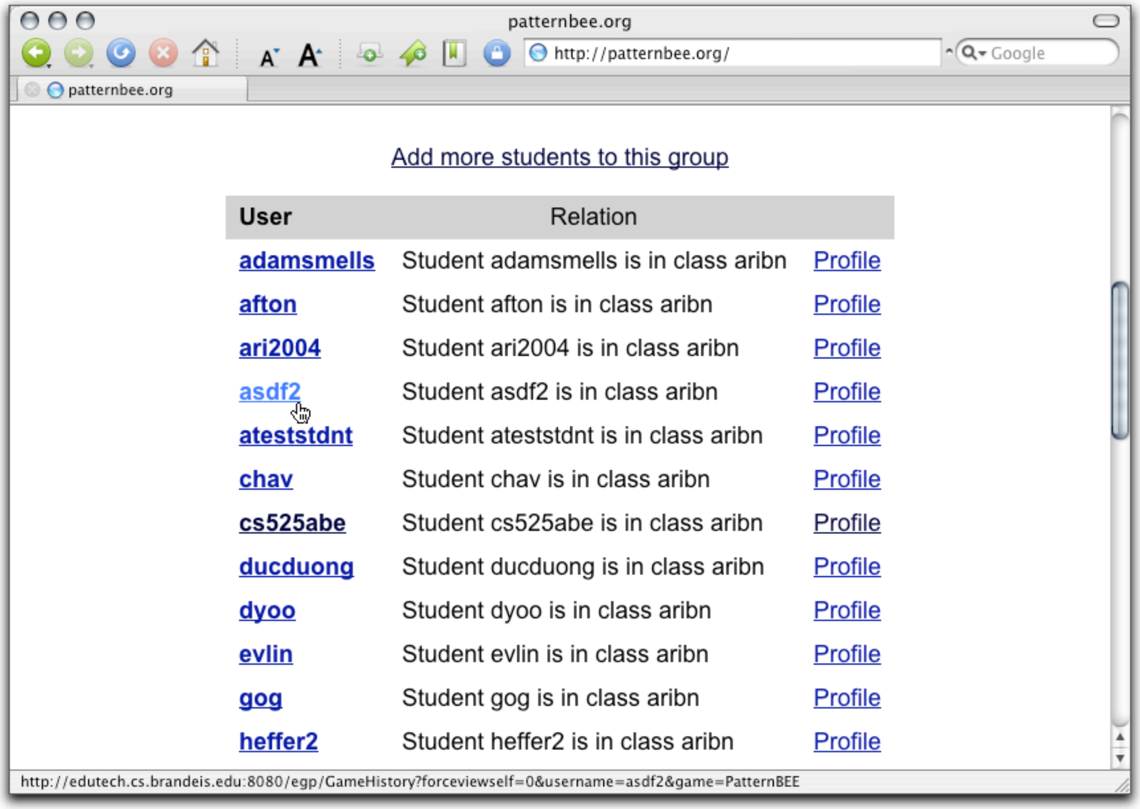
2 Type this word in the box:
aribn

3 (Follow instructions on screen):
asdf2

4 Give this form to your teacher.

Figure 4.10: When using the activity in a classroom, the teacher distributes customized instruction sheets to the students (top.) After registering a new account, each student is prompted to fill out the form with their name and new pseudonym (after it is approved by the system.) The teacher collects the completed forms, and can refer to these paper records later. This allows the teacher to associate the pseudonyms with students, without these names ever being entered into our system.

available outside of the game itself. By publishing a feed of this information that updates several times a minute, we enable other software (e.g. desktop applications, browser add-ons, widgets, other websites) to keep users informed about game activity. These tools can then serve to inform the user about when to log in to play. Figure 4.12 shows a screenshot of one such tool, that always remains visible in the user's desktop,



Add more students to this group

User	Relation	
adamsmells	Student adamsmells is in class aribn	Profile
afton	Student afton is in class aribn	Profile
ari2004	Student ari2004 is in class aribn	Profile
asdf2	Student asdf2 is in class aribn	Profile
ateststdnt	Student ateststdnt is in class aribn	Profile
chav	Student chav is in class aribn	Profile
cs525abe	Student cs525abe is in class aribn	Profile
ducduong	Student ducduong is in class aribn	Profile
dyoo	Student dyoo is in class aribn	Profile
evlin	Student evlin is in class aribn	Profile
gog	Student gog is in class aribn	Profile
heffer2	Student heffer2 is in class aribn	Profile

http://edutech.cs.brandeis.edu:8080/egp/GameHistory?forceviewself=0&username=asdf2&game=PatternBEE

Figure 4.11: Once the students have completed the process shown in Figure 4.10, their accounts are linked to their teacher's account (bottom.) This shows the teacher's screen for one of their classroom groups. The teacher can view the history of their students at various granularities (via the "Profile" links at right.) The most granular view is shown in Figure 4.5.

and provides one-click access to the player-matching screen for each game.

4.3 Summary

In this chapter we have detailed the goals, design, and implementation of two different systems for incorporating a Teacher's Dilemma game into a web-based system designed for peer tutoring across a variety of task domains. Since we first released

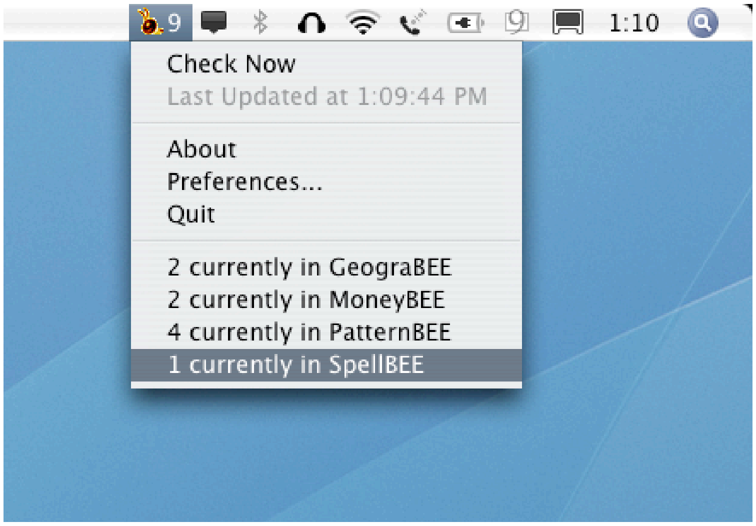


Figure 4.12: An optional download of an application displays an up-to-date feed of system activity (i.e. active user counts) in the operating system menu bar. This allows a user to visit the game website only when they know that others are currently online. When the menu is not open, the color of the bee icon and the accompanying number indicate overall system activity as follows: When some players are online, the icon turns yellow and the number of players are displayed; when no players are online, the icon turns black; and when the server is unreachable (i.e. if the user's computer is disconnected from the internet) the icon turns gray. When opened, per-game active user counts are visible. Clicking on a game menu item launches a web browser and loads the player-matching interface in a browser window.

SpellBEE four years ago, we have collected data on over 25,000 completed peer tutoring sessions. In Chapter 5, we will use this data to probe our model and explore its effect on participating tutors and tutees.

Chapter 5

Empirical Analysis of SpellBEE and BEEweb data

The game theoretic analysis presented in Chapter 2 and the simulations presented in Chapter 3 both suggest that a Teacher's Dilemma game will motivate peer tutors to provide their tutees with appropriate challenges. Using data collected from the SpellBEE and BEEweb systems discussed in Chapter 4, we now examine the effects of the Teacher's Dilemma on the activity of real students.

In this Chapter, we pose four research questions. First, we explore a core assumption of our model, that a game can be used to affect how peer tutors select challenges, by asking: "Does the game's payoff structure significantly affect the challenge selection strategies of tutors?" Second, we examine the collective student-modeling ability of the tutors in predicting the probability of a correct response from their tutees (i.e. the $\dot{\mathcal{E}}_t$ from Chapter 2) by asking: "How does the predictive performance of tutors, on the aggregate, compare to NISS-based performance expectations?" Third, we ask whether tutor or tutee grade levels (as a rough indicator of ability) affect the level of difficulty of the challenges posed: "Do the main effects of tutor grade or tutee grade (or the interaction effect of both) significantly affect the difficulty level of challenges posed?" Fourth, we examine if and where tutees improve at the task domain with use of our systems: "Does the response accuracy of tutees collectively improve with use of the system?" These four questions provide an empirical basis for evaluating the effectiveness of our web-based systems built on a difficulty-based Teacher's Dilemma game.

5.1 Summary of SpellBEE and BEEweb usage

We begin by presenting summary statistics about the usage of these systems, in order to provide an overall picture of how much data has been collected, who participates, and for how long.

When a student creates an account on the SpellBEE and BEEweb system, certain information is collected during the registration process, including the student's grade level, gender, and location. We use this self-reported grade level as the basis for grouping students when calculating certain summary statistics.[1]

Summary statistics on the SpellBEE and BEEweb activities are based on the number of active users[2] is displayed in Table 5.1, and the number of questions answered by these students is included in Table 5.2.[3] The numbers for the SpellBEE activity

[1] As we will discuss shortly, some students participate at the direction of a classroom teacher. For these students, grade level and location is not self-reported, but is instead specified by the classroom teacher.

[2] We note our definition of "active" players here: We consider users who register for an account and complete one or more (seven-question) games to be active, as distinguished from those users who register but never complete a game.

[3] We note that some students may be active players in more than one BEEweb activity, so while totals are summed within activity across grades, we do not tally totals within grade across activities.

Table 5.1: Active BEEweb and SpellBEE game users, counted by grade level.

Grade	GeograBEE	MoneyBEE	PatternBEE	SpellBEE
2	9	9	14	628
3	54	38	137	1,166
4	325	41	280	2,110
5	143	88	307	3,371
6	72	58	187	2,960
7	132	112	172	1,612
8	79	53	99	2,508
Total	814	400	1,196	14,355

Table 5.2: BEEweb and SpellBEE questions answered, counted by player grade level. (Note that this includes questions from incomplete games.)

Grade	GeograBEE	MoneyBEE	PatternBEE	SpellBEE
2	96	80	245	15,258
3	1,877	360	2,480	36,262
4	12,265	436	10,310	62,910
5	2,447	1,039	9,072	110,989
6	1,349	733	3,643	102,535
7	2,587	1,176	4,960	46,528
8	1,016	529	1,853	69,676
Total	21,637	4,365	32,563	444,158

are the highest, with over 14,000 active users completing over 400,000 challenges. The numbers for the SpellBEE activity are higher than those for the other three activities primarily because it was online first, and has been active for the longest period of time. SpellBEE was first released in November 2003, PatternBEE was first released in February 2005, MoneyBEE in May 2005, and GeograBEE in January 2006.[4]

By grouping and plotting users by the number of players that completed various

[4] The data presented in this section includes SpellBEE system usage from 2/1/2004–2/1/2008 and BEEweb system usage from the initial game release dates until 2/12/2008.

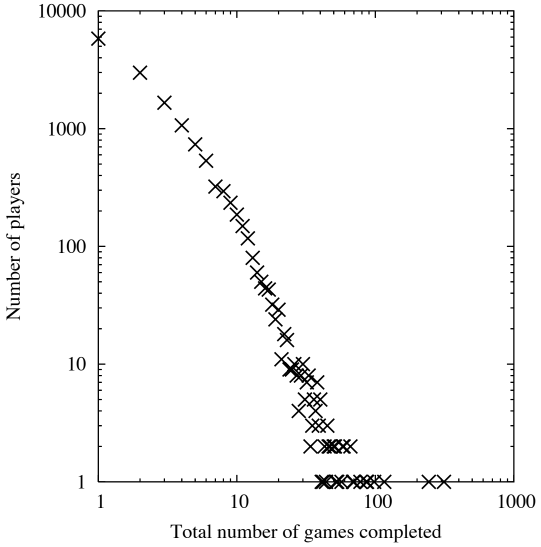


Figure 5.1: Log-log plot of SpellBEE system usage, based on the number of users that completed various numbers of games. (The first data point in the upper left-hand corner indicates that 5,818 players completed only one game, while the data point in the bottom right-hand corner indicates that one player completed 313 games.)

numbers of games, we gain a sense of how overall system activity is distributed among users. Figure 5.1 graphs SpellBEE usage in this way, on a log-log plot.

One factor hindering game play in SpellBEE was that a player could only participate if a second player was available and willing to start a match. When the BEEweb system was developed, it was designed to allow for “practice rounds”, in which a lone participant could solve challenges stored in the database of previously posed problems. By adding such functionality, we hoped to increase the length of time that a lone player would remain on the site, and thereby increase the likelihood that a second player would arrive before the first logged out. BEEweb practice rounds are included in Table 5.3 and Figure 5.2. Table 5.3 tallies the number of practice rounds completed in each activity, and the number of subsequent games started (and

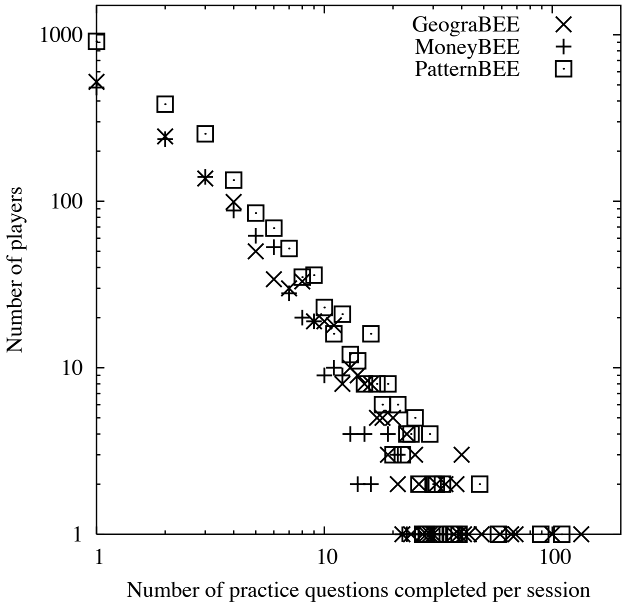


Figure 5.2: BEEweb practice round completion, by game. The data point in the upper-left-most corner indicates that there were 912 cases in which a player completed only 1 PatternBEE practice round during the session, while the data point in the lower-right-most corner indicates 1 case in which a player completed 134 GeograBEE practice rounds during the session.

Table 5.3: BEEweb practice rounds: the number of rounds completed and the number of subsequent games initiated by players following practice rounds.

Activity	Practice Questions	Subsequent Games
GeograBEE	4,905	437
MoneyBEE	3,079	99
PatternBEE	6,772	641
Total	14,756	1,177

Table 5.4: Active classroom-based SpellBEE users, by player grade level.

Grade	Classroom groups	Students
Multiple	83	(counted below)
2	16	223
3	15	412
4	24	907
5	29	1,246
6	17	642
7	10	347
8	10	216
Total	204	3,993

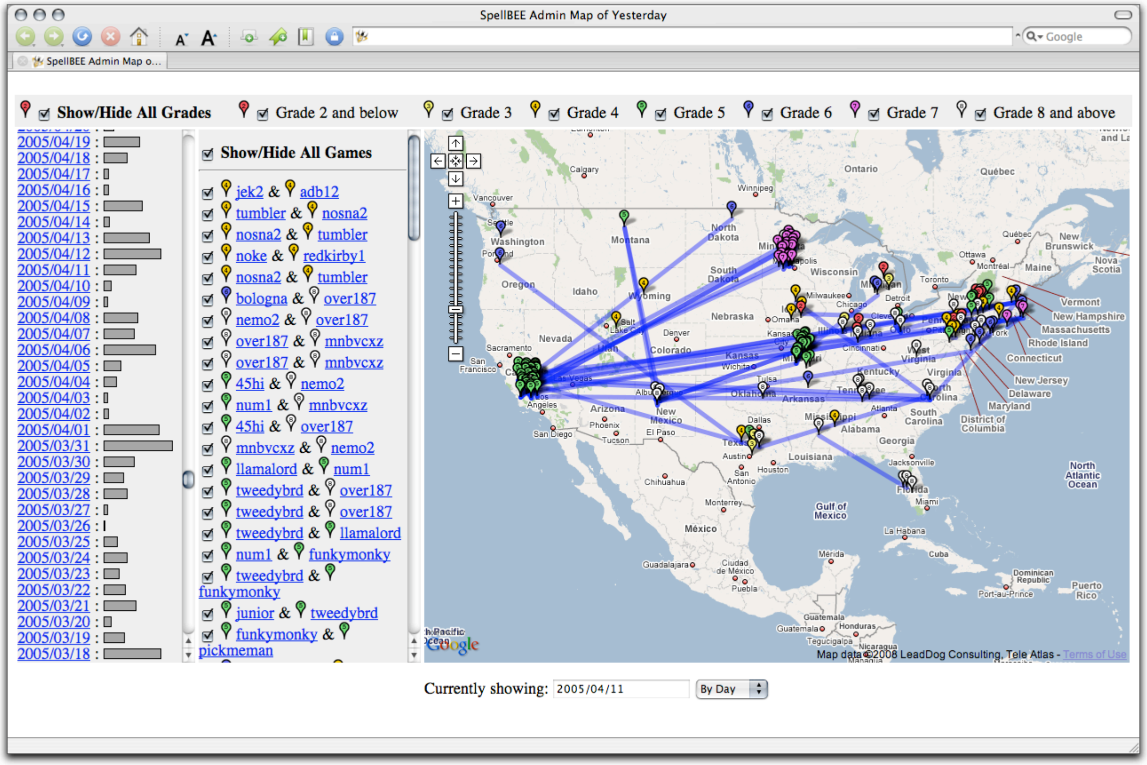
presumably enabled by) these practice rounds. Figure 5.2 shows how many players completed how many practice rounds per session[5], to give a sense of how practice round participation was distributed among players.

As described in Section 4.1.3, SpellBEE and BEEweb were designed to accommodate classroom-based usage. Table 5.4 summarizes the extent of teacher adoption of the SpellBEE activity in an organized group setting, listing the number of classroom groups participating and the number of active student users in those groups.[6] From this, we see that almost 4,000 students participated at the direction of one of about 200 classroom teachers.

Since the SpellBEE and BEEweb systems support both individual usage and classroom-based usage, we implicitly enable games between members of different classrooms, or between a classroom student and an out-of-classroom user. Figure 5.3 displays a visualization of student participation and partner matching for one day

[5] We consider a session to be a single day.

[6] For teachers that participated with groups of students spanning multiple grade levels, the students are tallied by grade level, and the teachers are tallied separately, under “Multiple.”



SpellBEE Admin Map of Yesterday

SpellBEE Admin Map o...

☑ **Show/Hide All Grades** ☑ Grade 2 and below ☑ Grade 3 ☑ Grade 4 ☑ Grade 5 ☑ Grade 6 ☑ Grade 7 ☑ Grade 8 and above

2005/04/19 :
2005/04/18 :
2005/04/17 :
2005/04/16 :
2005/04/15 :
2005/04/14 :
2005/04/13 :
2005/04/12 :
2005/04/11 :
2005/04/10 :
2005/04/09 :
2005/04/08 :
2005/04/07 :
2005/04/06 :
2005/04/05 :
2005/04/04 :
2005/04/03 :
2005/04/02 :
2005/04/01 :
2005/03/31 :
2005/03/30 :
2005/03/29 :
2005/03/28 :
2005/03/27 :
2005/03/26 :
2005/03/25 :
2005/03/24 :
2005/03/23 :
2005/03/22 :
2005/03/21 :
2005/03/20 :
2005/03/19 :
2005/03/18 :

☑ **Show/Hide All Games**

☑ jek2 & adb12
☑ tumbler & nosna2
☑ nosna2 & tumbler
☑ noke & redkirby1
☑ nosna2 & tumbler
☑ bologna & over187
☑ nemo2 & over187
☑ over187 & mnbvcxz
☑ over187 & mnbvcxz
☑ 45hi & nemo2
☑ num1 & mnbvcxz
☑ 45hi & over187
☑ mnbvcxz & nemo2
☑ llamalord & num1
☑ tweedybrd & over187
☑ tweedybrd & over187
☑ tweedybrd & llamalord
☑ num1 & funkymonky
☑ tweedybrd & funkymonky
☑ junior & tweedybrd
☑ funkymonky & pickmeman

Currently showing: 2005/04/11 By Day

Figure 5.3: Visualization of one day of activity (4/11/2005) in the SpellBEE community. Pushpins represent participants (colored by grade level and located within their reported home state), and lines connect game partners from that day.

(within the continental United States).[7] Clusters within a state generally indicate classroom group usage, and the bundles of lines connecting clusters suggest cross-classroom usage. (Closer inspection would make visible a large number of within-classroom activity.)

One question on our minds early on in the study was whether there would be any relationship between a student's cumulative success at the activity and the duration of that student's active participation in the activity. Does a high response accuracy rate correspond with a long participation duration? In Figure 5.4, we plot each

[7] Student locations on the map are not precise, and indicate only their state.

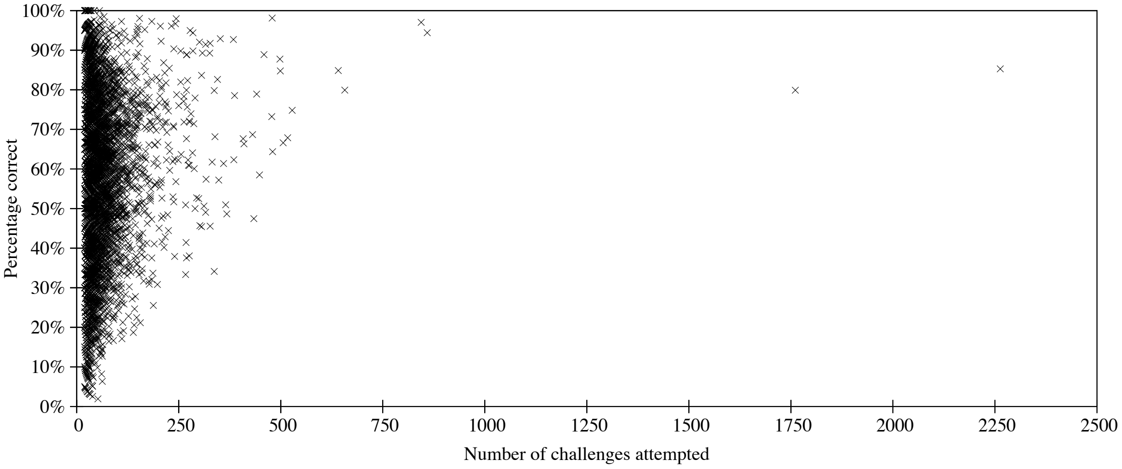


Figure 5.4: Students are plotted, based on the number of challenges attempted since registering (through March 2008), and on the percentage of these responses that were accurate.

student according to their cumulative percentage of correct responses and the number of challenges that they have attempted to date. This scatterplot provides a simple picture of this relationship.

Together these statistics suggest that the systems that we have built, and the activities supported by these systems, have been used by thousands of student, primarily accessed from out-of-classroom environments. As the number of participants and quantity of data collected from the SpellBEE activity is larger than from the three BEEweb activities combined, the research questions explored in the subsequent sections will look primarily at this game's data.

5.2 On tutor sensitivity to the game payoff function

In constructing a learning activity based on an abstract game-theoretic formulation of a participant behavior, we assume that the basic tools of this approach are applicable and relevant. The payoff functions, for example, were carefully designed to motivate students to adopt certain types of game-play strategies. While we recognize that the game-theoretic assumptions of perfect knowledge and rationality are not appropriate to expect from elementary school-aged students playing a game during their free time, we wish to understand if the reward-based motivational structure has any effect, whatsoever, on the students. If the reward structure were to change, would the challenge selection strategies of the tutors also change? In order to establish whether or not students exhibited any sensitivity to the game's payoff function, we organized a simple experiment in which students (in a classroom setting) were randomly assigned to one of four experimental conditions, differing only in the function used to determine the Teacher's payoff. As this experiment was performed using an early SpellBEE prototype, a number of variations existed from the game as described in the previous chapter: First, instead of including a fixed number of rounds, the game progressed for as many rounds as necessary for one player to accumulate 100 points. Second, the difficulty metric used in this prototype was based not on Greene's NISS data, but rather on the normalized "Scrabble score" of the various word options.[8] Participating

[8] A Scrabble score is the sum of the point value of the word's letters in the English-language editions of the board game "Scrabble." Namely, 1 point for the letters E, A, I, O, N, R, T, L, S, U; 2 points for letters D, G; 3 points for the letters B, C, M, P; 4 points for the letters F, H, V, W, Y; 5 points for the letter K; 8 points for the ltters J, X; and 10 points for the letters Q, Z. The Scrabble score of each option presented to the Teacher is calculated, and these scores are normalized to generate difficulty values (i.e. $\mathcal{D}_c = 0$ for the option with the lowest Scrabble score, $\mathcal{D}_c = 1$ for the option with the highest Scrabble score.)

classrooms were assigned to one of four groups: *MotivateEasy*, *MotivateAppropriate*, *MotivateDifficult*, and *MotivateSkewedAppropriate*. As players change their challenge selection strategies over time, we look only at challenge selection data from a point in time: each player's second game completed. All player matches included pairs of players within the same group. In all four groups, the reward function for the Student was the same, with the Student receiving 10 points for a correct response and 0 for an incorrect response (note that all rewards are scaled by a factor of 10):

$$\pi_s = \begin{cases} 0 & \text{if } \mathcal{A}_{r,c} = 0 \\ 1 & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{5.2.1}$$

The reward function for the Teacher varied by group. The 20 students in the *MotivateEasy* group were rewarded for asking easy questions, regardless of the accuracy of the tutee's response:

$$\pi_{t_{MotivateEasy}} = \begin{cases} 1 - \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 0 \\ 1 - \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{5.2.2}$$

The 24 students in the *MotivateAppropriate* group were rewarded for asking difficult questions that the student gets right and easy questions that the student gets wrong, as in the SpellBEE and BEEweb activities:

$$\pi_{t_{MotivateAppropriate}} = \begin{cases} 1 - \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 0 \\ \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{5.2.3}$$

The 44 students in the *MotivateDifficult* group were rewarded for asking difficult questions, regardless of the accuracy of the tutee's response:

$$\pi_{t_{MotivateDifficult}} = \begin{cases} \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 0 \\ \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{5.2.4}$$

The 11 students in the *MotivateSkewedAppropriate* group were rewarded for asking difficult questions that the student gets right, but rewarded more for asking easy questions that the student gets wrong:

$$\pi_{t_{MotivateSkewedAppropriate}} = \begin{cases} 2 - \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 0 \\ \mathcal{D}_c & \text{if } \mathcal{A}_{r,c} = 1 \end{cases} \tag{5.2.5}$$

While there are many potential ways to identify group-wide differences in challenge selection strategy, we explored two: First, we use a Kruskal-Wallis (non-parametric) one-way analysis of variance by ranks. This lets us determine if the four groups differed in terms of the relative difficulty of the challenges selected. The results show that the data from the groups does, in fact, differ significantly ($H(3) = 31.549$, $p < 0.001$), suggesting that the payoff function does affect the challenge-selection strategies adopted by the students when playing the games. Second, we classify each person according to whether the majority of the challenges that they posed were among the two most difficult options (*Asks Hard*), were among the three middle-difficulty options (*Asks Medium*), were among the two least difficulty options (*Asks Easy*), or none of the above (*Other*). Table 5.5 tabulates the distribution over these observed strategies for each group. We note that, among the four groups, a higher proportion of students in group *MotivateAppropriate* (for which the Teacher payoff function was the same as that of the difficulty-based Teacher's Dilemma game) were

Table 5.5: Distribution of students' observed challenge-selection strategies, by group.

Group	*Asks Hard*	*Asks Medium*	*Asks Easy*	*Other*
MotivateEasy	25%	10%	45%	20%
MotivateAppropriate	33%	29%	0%	38%
MotivateDifficult	70%	9%	7%	14%
MotivateSkewedAppropriate	46%	27%	0%	27%

observed to fit the *Asks Medium* challenge selection strategy.

So while the tutors do appear to alter their challenge selection strategies based on the payoff function of the game, these adopted strategies are not as clear as we would have expected, based on what strategies maximize these payoff functions. We note that across all groups, tutors seemed to have a bias towards posing the relatively difficult challenges.

5.3 On tutor student-modeling abilities

A second assumption of our model is that the peer tutor is capable of forming predictions about the likelihood that their tutee will be able to correctly answer the problem posed. While this assumption plays a much more significant role in the expectation-based Teacher's Dilemma game, it does form a part of the difficulty-based game (as used in the SpellBEE and BEEweb activities), as tutors weigh the value of different challenges on both the stated payoff and their own prediction regarding response accuracy. Given that these predictions ($\dot{\mathcal{E}}_t$) are held privately and never explicitly stated during the game, we can examine these predictions only indirectly. Another indirect exploration of the aggregate perception of those posing questions to students was described by Hadjidemetriou [36], in which the teachers' awareness of their students' knowledge is measured. Here, we take a slightly different approach, and examine the aggregate performance of students (at various levels of problem difficulty) in order to assess strategic biases in problem selection.

We begin by noting that in the difficulty-based game, as detailed in Section 2.4.1, the tutor calculates the expected utility associated with selecting a particular problem as follows:

$$\mathrm{E}_{\pi_t} = \left(1 - \dot{\mathcal{E}}_t\right)(1 - \mathcal{D}_c) + \left(\dot{\mathcal{E}}_t\right)(\mathcal{D}_c) \tag{5.3.1}$$

If the tutor is able to identify a hard problem that the difficulty function overestimates in difficulty for the tutee (i.e. for which $\mathcal{D}_c > 0.5$ and $\mathcal{D}_c > \left(1 - \dot{\mathcal{E}}_t\right)$) or an easy problem that the difficulty function underestimates in difficulty for the tutee (i.e. for which $\mathcal{D}_c < 0.5$ and $\mathcal{D}_c < \left(1 - \dot{\mathcal{E}}_t\right)$), the tutor's expected utility will be greater than if the difficulty function were accurate. Furthermore, there are two sets of discrepancies for which the tutor's expected utility would exceed that corresponding to any accurately-measured (i.e. $\mathcal{D}_c = \left(1 - \dot{\mathcal{E}}_t\right)$) challenge: when $\mathcal{D}_c > 0.5 > \left(1 - \dot{\mathcal{E}}_t\right)$ and when $\mathcal{D}_c < 0.5 < \left(1 - \dot{\mathcal{E}}_t\right)$. Thus, tutors can leverage their own insights about their tutee's abilities to perform better than expected, if these insights are correct. If these insights turn out to be flawed or otherwise incorrect, the tutee will perform worse than expected. Thus, by comparing the performance of tutors to the expected performance of tutors, we are able to see if their insights regarding tutee abilities were, on the whole, more accurate than the student model as predicted by the NISS data.

Figure 5.5 illustrates, for each student grade level and NISS-based difficulty grouping, the percentage of accurate responses observed within the SpellBEE data. The NISS-predicted level of response accuracy is also included, so that the observation summaries may be compared to the expected outcome. In this figure, we see that the seven grade levels produce similar patterns. In each case, the percentage of accurate responses observed decreases as NISS-based difficulty increases. We note that in each grade, the students correctly answered the easiest challenges *less* often than expected, and correctly answered the most difficult challenges *more* often than expected. We believe that this observation supports our claim that Teachers strategically bias the selection of challenges posed to their Students. In the next paragraph, we shall see that this particular bias does, in fact, reward the Teachers, in that it leads to higher-than-expected payoffs within the game.

Figure 5.6 presents five graphs plotting tutor data covering four years of challenges posed in the SpellBEE system.[9] The top-left graph (*TL*) plots the percentage of *correct* student responses as a function of the (NISS-based) problem difficulty. The top-right graph (*TR*) plots the percentage of *incorrect* student responses as a function of the problem difficulty. In both of these graphs, observed student response levels are compared to the (NISS-based) expected response levels. The middle-left graph (*ML*) plots the payoff awarded to a teacher, if the response is *correct*, as a function of the problem difficulty. The middle-right graph (*MR*) plots the payoff awarded to a teacher, if the response is *incorrect*, as a function of the problem difficulty. Finally, the bottom graph (*B*) combines the four graphs above it into a single view, quantifying payoffs awarded (both observed and expected) to tutors: $B = (TL \times ML) + (TR \times MR)$.

[9] To simplify the interpretation of this Figure, we will not distinguish among the grade levels of students, and will instead provide a unified plots covering all students together.

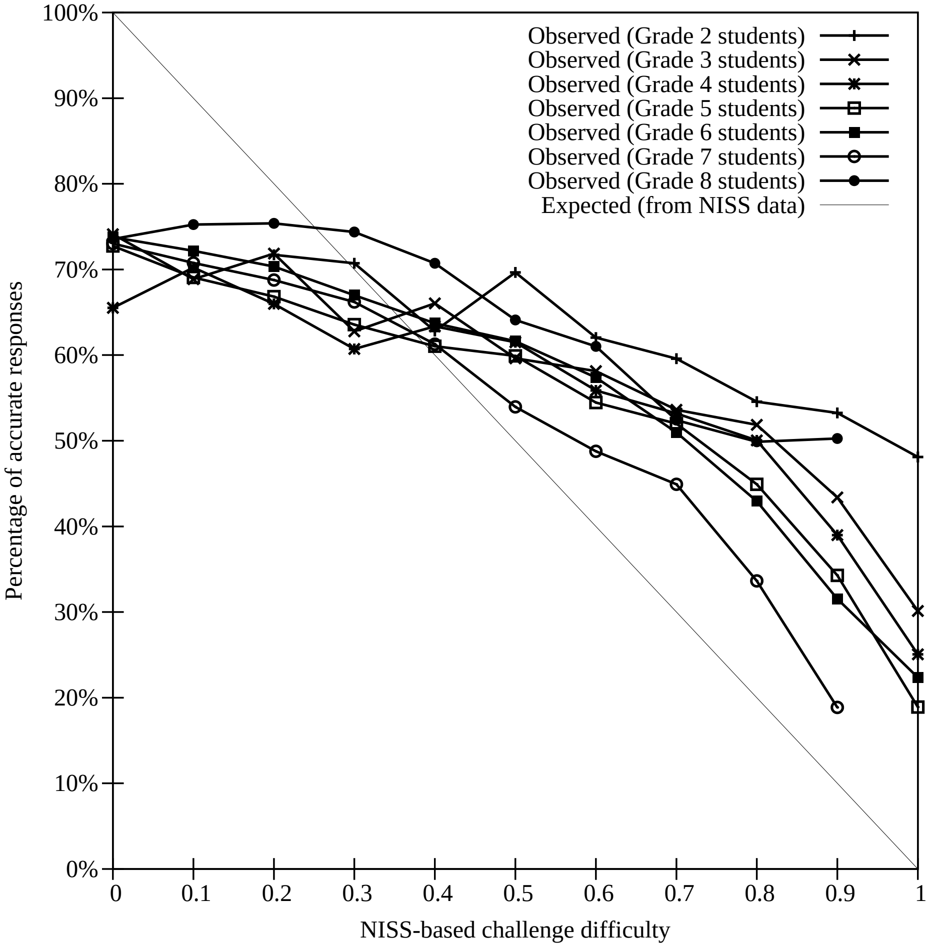


Figure 5.5: Expected vs. observed response accuracy in SpellBEE. For each grade level, the percentages of accurate observed responses are displayed by difficulty (rounded to the nearest 0.1). Data points are omitted for cases in which less than 50 samples were available.

In graph *B*, we observe that for challenges with a difficulty below 0.26 and for challenges with a difficulty above 0.5, the participants, on aggregate, earned more Teacher points than was expected based on NISS-predicted response accuracy. As such, the student participants predicted the likelihood of response accuracy better

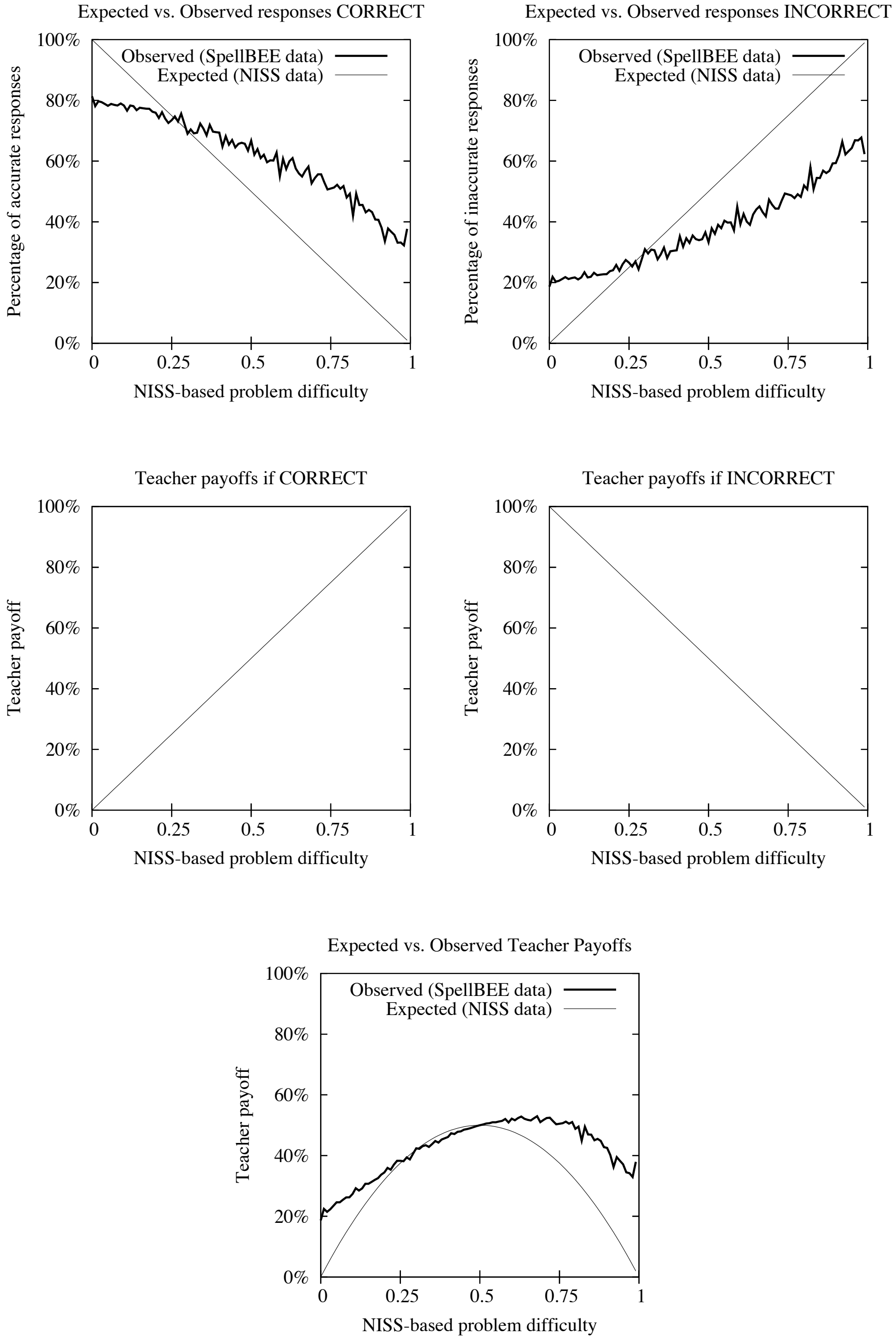


Figure 5.6: Expected vs. observed tutor payoff data in SpellBEE. The top graphs plot response accuracy, the middle graphs plot accuracy-dependent payoffs, and the bottom graph plots actual payoffs.

than expected likelihoods based on the grade-level NISS data. For challenges with difficulty values between 0.26 and 0.5, the peer tutors predicted slightly worse than the NISS-based model.

Alternatively, we can return to the observation at the beginning of this section that there are two sets of conditions for which a discrepancy between the stated problem difficulty and the Teacher's expected problem difficulty is particularly valuable: if the Teacher can identify a problem for which $\mathcal{D}_c > 0.5 > \left(1 - \dot{\mathcal{E}}_t\right)$ or if $\mathcal{D}_c < 0.5 < \left(1 - \dot{\mathcal{E}}_t\right)$, the Teacher's expected utility will be higher than that corresponding to any problem without a discrepancy. While we cannot know a Teacher's true expectation, $\dot{\mathcal{E}}_t$, we can use a contingency table to explore the relative frequency of challenges that likely represent a true expectation satisfying one of these two sets of conditions. In Table 5.6, we group and count the number of challenges posed according to the a dichotomous version of the NISS-based difficulty metric, and according to the observed accuracy of the response. In doing so, we find a surprising large number of "easy" challenges (i.e. $\mathcal{D}_c < 0.5$) for which the response was incorrect, and an even larger number of "difficult" challenges (i.e. $\mathcal{D}_c > 0.5$) for which the response was correct. Among all challenges posed for which the NISS data suggests that an incorrect response is more likely than a correct response (i.e. $\mathcal{D}_c > 0.5$), almost 48% of the responses observed were actually correct. That such a sizable fraction of these outcomes defied expectations suggests the degree to which participating students were able to productively model the abilities of their partners.

There are at least two alternative explanations that may potentially account for the differences between the data observed and expected. First, the differences may be attributed to errors in the self-reported grade levels of students. Second, the differences may be attributed to changes in the relevance or accuracy of the NISS data

Table 5.6: Contingency table tabulating the NISS-based challenge difficulty and the observed response accuracy for each response submitted within the SpellBEE activity.

		Difficulty (NISS)		
		$\mathcal{D}_c < 0.5$	$\mathcal{D}_c > 0.5$	
Accuracy (Observed)	$\mathcal{A}_{r,c} = 0$	52,598	132,061	184,659
	$\mathcal{A}_{r,c} = 1$	138,188	121,227	259,415
		190,786	253,288	444,074

since it was first collected over 50 years ago. We can assess the relevance of the first alternative explanation by exploring a subset of data for which grade level information was not self-reported: classroom usage. For these students, the grade level was reported by their classroom teacher, and so can be considered to be accurate. In Figure 5.7, we plot the response accuracy data for only this subset of users, comprising a total of 57,857 responses. We note that the trend in this subset of observed data reflects that of the larger set, suggesting that the self-reporting of grade level did not noticeably affect the results. In response to the second alternative explanation – that the trends observed might only reflect a now-outdated difficulty metric (i.e. the expected outcomes plotted above are no longer appropriate) – we note that the differences between the observations and expectations are not uniformly skewed in one direction. It is not the case that SpellBEE students performed better than expected across all difficulty levels. Instead, we find that the students tended to do better than expected on the difficult problems while simultaneously doing worse than expected on the easy problems. These two contradictory trends cannot be explained simply by changes over time in overall spelling abilities. If, for example, we suspect that student spelling abilities have slipped over time due to the prevalence of spell-checking functionality in computer software, this change would support one of the trends, but

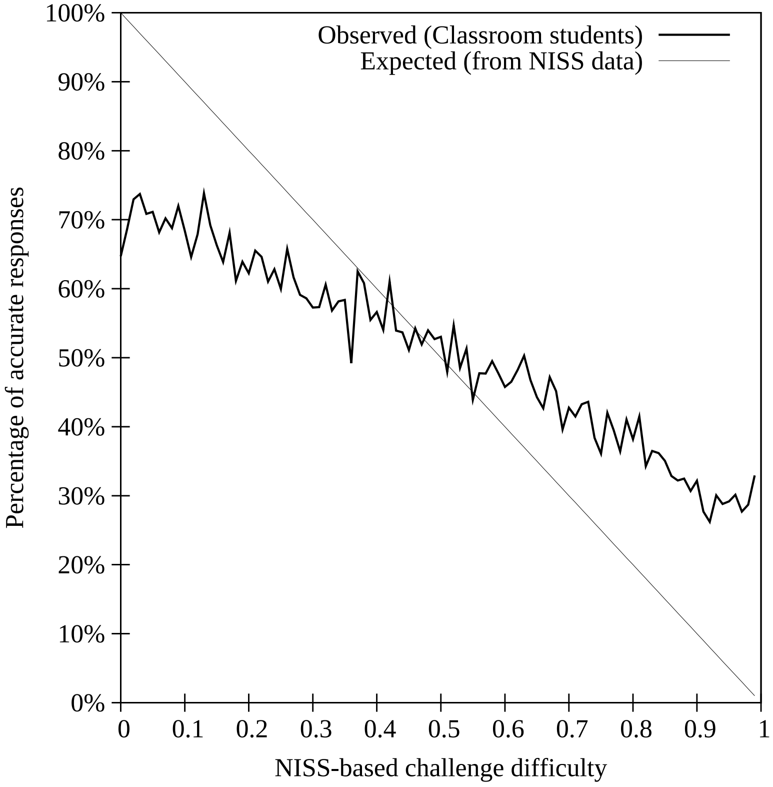


Figure 5.7: Expected vs. observed response accuracy among classroom users of SpellBEE, for whom grade level was specified by a classroom teacher.

would contradict the other. As such, we believe that the trends observed support the idea that peer tutors are capable of forming predictions regarding the response accuracy of their tutees, and that these predictions are, by and large, more accurate than predictions based on the grade-specific NISS study data.

5.4 On the effect of tutor and tutee age on challenge difficulty

Given that cross-age partnerships occur in the SpellBEE and BEEweb activities, we would like to see if the Teacher's Dilemma is effective in motivating cross-age tutors to adapt the difficulty of the problems that they pose to the skill level of their tutee. More specifically: Does the grade level of the tutee predict differences in difficulty of problems posed by tutors of varying grade levels? Additionally, we will examine the converse (whether tutor grade level can explain the difference in problem difficulties posed to tutees of varying grade levels) and the interaction (whether the effects of different tutor grades is the same for different tutee grades).

We use the SpellBEE data for this investigation, as the difficulty metric is more meaningful than those of the BEEweb activities. In order to provide a consistent measure of problem difficulty across all tutee grades, we use a metric derived from the NISS data: the word's "location" (used by Wilson and Bock [84], among others). This metric indicates the (fractional) grade level at which a student would have exactly 50% likelihood of a correct response, based on a piecewise linear interpolation of the graded NISS data.[10] Figure 5.8 illustrates how the grade-specific difficulty data is combined to derive this grade-independent statistic for two different words.

We note that in the SpellBEE activity, players are not made aware of the grade level (or age) of their partner, and learn about the ability level of their partner through the interactive process of the game itself. A facility exists in SpellBEE to show a player the last twenty questions attempted by their partner and whether each

[10] Wilson and Bock calculate the 50% threshold based on a logistic model fit to the discrete grade-level data, while we calculate the threshold slightly differently, based on a linear interpolation of the grade-level data.

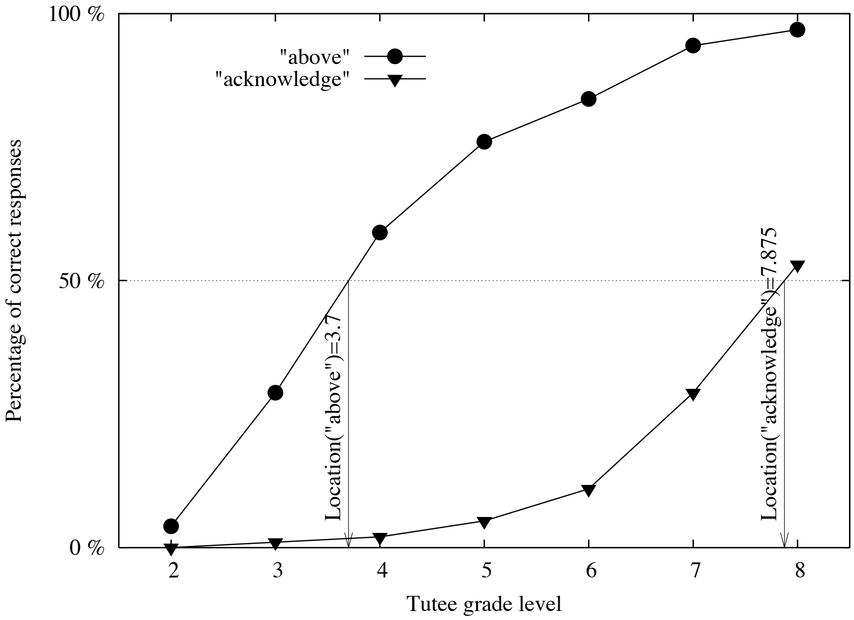


Figure 5.8: Examples of the grade-independent word location statistic.

response was correct or incorrect, which can provide an indication of ability before the game begins.

For each combination of tutor grade level and tutee grade level, Table 5.7 tallies the total number of challenges posed, and Figure 5.9 plots the distribution of word location (rounded to the nearest integer) among those challenges. The shape of these distributions suggests that challenge difficulty varies primarily with tutee grade. In order to examine this directly, we use a two-factor between-subjects 7x7 factorial ANOVA, in which the first factor is the grade of the tutor (varying from 2 – 8) and the second factor is the grade level of the tutee (also varying from 2 – 8). We found all three effects to be statistically significant at the $\alpha = 0.05$ level: The main effect of tutee grade level yielded an F ratio of $F(6, 38.84) = 141.221$, $p < 0.001$, the main effect of the tutor grade level yielded $F(6, 38.84) = 6.990$, $p < 0.001$, and the

Table 5.7: Cross-grade challenges posed in SpellBEE. For each combination of tutor grade and tutee grade, the total number of challenges posed is tallied.

		Tutee Grade Level						
		2	3	4	5	6	7	8
	2	7,289	1,280	1,145	1,888	1,410	598	1,957
	3	1,264	22,766	2,660	3,369	2,524	1,176	2,779
	4	1,065	2,578	42,081	6,464	4,356	1,945	4,122
Tutor Grade Level	5	1,818	3,275	6,506	78,560	9,016	2,958	8,551
	6	1,334	2,415	4,341	8,959	71,892	4,334	8,630
	7	571	1,181	1,943	2,948	4,314	30,841	4,451
	8	1,902	2,744	4,133	8,656	8,710	4,497	42,335

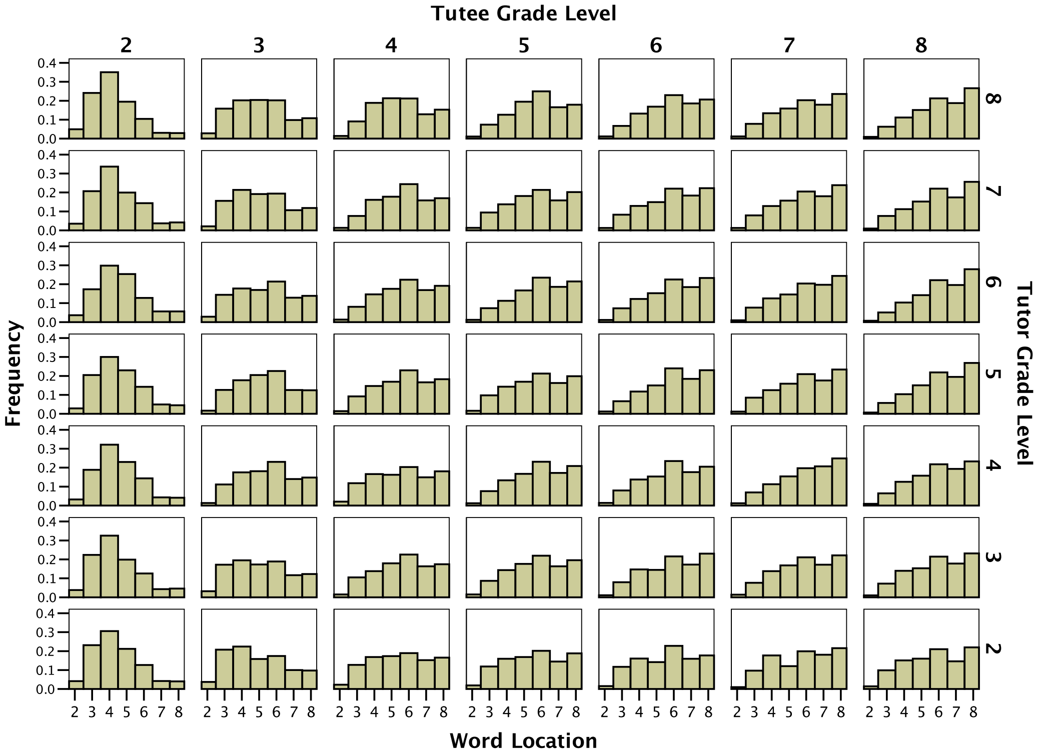


Figure 5.9: For each combination of tutor grade and tutee grade, the distribution of challenge difficulty is plotted. In these plots, difficulty is denoted by word location (as a grade-independent consistent measure of difficulty), rounded to the nearest integer.

interaction effect of the tutor and tutee grade levels yielded $F(36, 483844) = 17.329$, $p < 0.001$. The magnitude of the effect of the tutee's grade level (partial $\eta^2 = 0.956$) was much higher than that of the tutor's grade level (partial $\eta^2 = 0.519$) and that of the interaction (partial $\eta^2 = 0.001$). So while the problems posed by cross-age tutors may be affected by all three factors, the factor with the largest effect on the challenge difficulty of problems posed was the grade level of the tutee.

5.5 On tutee learning at multiple grain sizes

Given the peer-driven nature of this design, we wish to validate that students using the SpellBEE and BEEweb systems are indeed learning, and improving at the skills involved in the task domain. An evaluation of student learning is complicated by two factors, however. First, as our activities are designed to be played voluntarily by students during their free time, we chose not to incorporate static pre- and post-tests in the games, opting, instead, to base evaluations on data collected during student interactions. Second, we note that the sampling of data collected during these interactions is biased by strategic use of challenge-selection. So, in order to measure learning given these constraints, we focus our attention on cases in which a single student faces the same problem at two different points in time. In a sense, these two time-separated responses act as a very narrowly-defined pre-test and post-test, and by looking only at relative performance between the two (rather than the actual performance on either), we are able to look past biases in challenge selection. Here, we use McNemar's test [1, 59], a non-parametric statistical method that tests for change in a dichotomous trait for a group of subjects before and after an intervention, to examine student data from the SpellBEE activity. We perform several tests of student learning by identifying and aggregating this repeated-attempt data in various ways. Based on these tests, we see that students are, indeed, learning, and are able to map how this learning is distributed over challenges and sub-problems in the task domain.

In Chapter 2, we defined response accuracy in terms of the challenge c and the response r:

$$\mathcal{A}_{r,c} = \begin{cases} 0 & \text{if response } r \text{ is not the correct solution to challenge } c \\ 1 & \text{if response } r \text{ is the correct solution to challenge } c \end{cases} \tag{5.5.1}$$

When a student sees some challenge c at time t_i and again later at time t_j, we can compare the accuracy of their earlier response, $\mathcal{A}_{r_i,c}$, to the accuracy of their later response, $\mathcal{A}_{r_j,c}$. As response accuracy is binary, all repeated attempts fall into one of four categories:

(A) $\mathcal{A}_{r_i,c} = 0$ and $\mathcal{A}_{r_j,c} = 0$

(B) $\mathcal{A}_{r_i,c} = 0$ and $\mathcal{A}_{r_j,c} = 1$

(C) $\mathcal{A}_{r_i,c} = 1$ and $\mathcal{A}_{r_j,c} = 0$

(D) $\mathcal{A}_{r_i,c} = 1$ and $\mathcal{A}_{r_j,c} = 1$

By ignoring the students in categories A and D, we are able to set aside challenge-selection strategy bias. Letting Δ count the number of students in B, and letting ∇ count the number of students in C, McNemar's test uses Δ and ∇ to test the association between SpellBEE usage and response accuracy, using the statistic:

$$\chi^2_{McNemar} = \frac{(|\Delta - \nabla| - 1)^2}{\Delta + \nabla} \tag{5.5.2}$$

We include in this tally only pairs of attempts with a minimum time-span $(t_j - t_i)$ of one day and a maximum time span of one year. For students represented more than once in this tally, we include the pair of attempts with the longest time-span $(t_j - t_i)$ separating them. Using this approach, we found that, based on data collected prior to February 2008, Yates' continuity-corrected $\chi^2 = 58.4551$, $df = 1$, $p < 0.001$, odds ratio = 2.030. [11] We reject the null hypothesis that the marginal frequencies are homogenous (i.e. that Δ and ∇ are equally likely), and conclude that there is a statistically significant association between student spelling accuracy and SpellBEE usage. Since $\Delta > \nabla$, this association represents an *increase* in spelling accuracy with SpellBEE usage.

In order to gain a richer understanding of the nature of student improvements within the spelling domain, we can analyze spelling accuracy at a finer grain. Hanna et al. [37] thoroughly detailed the role and regularity of phoneme-grapheme[12] correspondences in American-English spelling, and we draw upon this work by examining spelling accuracy at the levels of graphemes and syllables.

We begin by extending our notion of response accuracy so that it may be applied to sub-problems. Intuitively, sub-problem accuracy is a measure of the correctness of a particular part or aspect of a response, irrespective of the accuracy of other parts (or the accuracy of the response as a whole.) For example, consider the case in which the tutor poses the word "accommodation" as a spelling challenge, and the tutee types in the string "acomodation" as their response. While the whole-word accuracy of this response is 0, if we consider only the sub-problem accuracy of the spelling

[11] Alternatively, if students are represented by the pair of attempts with the shortest time elapsed, Yates' continuity-corrected $\chi^2 = 48.3532$, $df = 1$, $p < 0.001$, odds ratio = 1.980.

[12] A *phoneme* is the smallest unit of sound in a language (e.g. /f/), and a *grapheme* is a written form of a phoneme (e.g. "ph").

of the syllable "tion", the sub-problem response is 1. In general, we define sub-problem accuracy with respect to some sub-problem structure s (such as a grapheme or syllable):

$$\mathcal{A}^*{}_{r,c,s} = \begin{cases} 0 & \text{if sub-problem } s \text{ of challenge } c \text{ was } \textit{not} \text{ correctly solved in } r \\ 1 & \text{if sub-problem } s \text{ of challenge } c \text{ was correctly solved in } r \end{cases}$$

The value of a notion of sub-problem accuracy is that a particular sub-problem appears in the context of many challenge problems, and by allowing repeated-attempt instances to compare the sub-problem accuracy across different challenge problems, a much larger set of instances is available:

(A) $\mathcal{A}^*_{r_i,c_i,s} = 0$ and $\mathcal{A}^*_{r_j,c_j,s} = 0$

(B) $\mathcal{A}^*_{r_i,c_i,s} = 0$ and $\mathcal{A}^*_{r_j,c_j,s} = 1$

(C) $\mathcal{A}^*_{r_i,c_i,s} = 1$ and $\mathcal{A}^*_{r_j,c_j,s} = 0$

(D) $\mathcal{A}^*_{r_i,c_i,s} = 1$ and $\mathcal{A}^*_{r_j,c_j,s} = 1$

Counting students in these $\mathcal{A}^*$-based groups, we can again use McNemar's test. As a much larger number of instances are available when examining sub-problems, we are able to look for evidence of learning on each individual syllable or grapheme (rather than the single analysis used across all repeated word-grained repeated spellings.)[13]

We compare the number of discordant pairs (i.e. groups B and C) for each grapheme in the English language. We used a list of 170 graphemes in American-English spellings, drawn from Dewey [27], based on the list used by Hanna et al. [37]. For each grapheme in this list, we identified all words containing the grapheme substring. We then identify all cases in which a student attempted two of these words.

[13] McNemar's test is only appropriate if sufficient data is available, specifically, that $\Delta + \nabla > 10$.

Table 5.8: Each grapheme is classified, using McNemar's test, according to how spelling accuracy changed over time (and SpellBEE usage.) Dashes occur in non-contiguous graphemes, for which some (unspecified) letter occurs in place of the dash.

Improved	—	Worsened
($p < 0.05$)	($p \geq 0.05$)	($p < 0.05$)
A, C, CC, CE, CQ, CQU, CT, D, DG, E, ED, EI-E, EIGH, EN, ES, F, G, GH, GI, H, I, I-E, IA, IA-E, IGH, IN, J, K, KN, L, M, N, NG, O, OL, ON, OO, OW, OW-E, P, PP, PT, Q, QU, R, S, SI, SSI, ST, T, TH, TI, U, U-E, W, WH, X, Y	A-E, AI, AI-E, AL, AU, AW, AY, B, CH, CI, CK, DD, DI, E-E, EA, EA-E, EE, EE-E, EI, EL, EO, ET, EW, EY, –EY, FF, FT, GG, GN, GU, GUE, IE, IE-E, IL, LD, LE, LL, LV, MB, MM, MN, NN, O-E, OA, OI, OU, OUGH, OWE, RR, SC, SCI, SH, SL, SS, SW, TCH, TT, UE, UI, UI-E, V, WR, Z	*none*

Letting s represent the grapheme "sub-problem", and letting c_i represent the word challenge that the student tried first and c_j the word challenge attempted second, we calculate grapheme sub-problem accuracies $\mathcal{A}^*_{r_i,c_i,s}$ and $\mathcal{A}^*_{r_j,c_j,s}$ based on whether the grapheme occurs in the student's response string.[14] We then categorize the response pair based on these values.

Using this approach to examine data collected as of April 2006, we found 58 graphemes upon which student spelling accuracy, based on McNemar's test, significantly changed (for the better) after practice ($p < 0.05$); 63 graphemes upon which no significant change was observed (i.e. $p \geq 0.05$); and 0 graphemes for which student spelling accuracy significantly changed (for the worse) after practice ($p < 0.05$)[15] Table 5.8 groups graphemes according to the statistical significance and directionality of the results of these McNemar tests. Similarly, we treat syllables as sub-problems and look at learning on each individual syllable. Using the same techniques as described above (e.g. syllable sub-problem accuracy is based on the existence of the syllable in

[14] A more nuanced sub-problem accuracy function would take into account the location of the grapheme within the response typed. We simply test for its presence.

[15] For the 50 remaining graphemes, $\Delta + \nabla \leq 10$, so not enough data was available to apply the test.

the response, regardless of its exact location), we find that, at the $\alpha = 0.05$ level, 79 syllables for which spelling accuracy significantly changed for the better, 304 syllables for which no significant change was observed, and 0 syllables for which spelling accuracy significantly changed for the worse.

Thus, for repeated words, we observed students improving at the spelling task, and for individual syllable- and grapheme-level sub-problems, all significant changes were in the direction of spelling improving. While student assessment is complicated by the inappropriateness of formal testing in a game and by the sampling bias introduced by the strategic selection of problems, we see the approach taken in this experiment as one suitable technique to test for student learning over time.

5.6 Summary

In this Chapter, we described SpellBEE and BEEweb activity participation in terms of the number of active users for each activity, and the grade level breakdowns among these users. We detailed how many questions these users answered, and how this usage was distributed among the population of users. We offered statistics on BEEweb practice rounds, and discussed the value of such rounds in subsequent player matching. We discussed and quantified teacher-directed classroom participation. We offered a geographical visualization of matches among players during a single day.

Based on this usage data, we examined four research questions. The first tested a core assumption of our model: that Teacher strategies for selecting challenges were sensitive to changes in the Teacher payoff function. We found that while there was a bias towards selecting difficult questions regardless of the payoff function, Teacher strategies were, in fact, sensitive to changes in the function, establishing the effectiveness of the game-based approach for influencing peer-tutoring strategies. The second question sought to test whether this strategic responsiveness was being leveraged productively by participants. We found that, on the aggregate, peer tutors did strategically bias their selection of challenges, and were effectively able to leverage knowledge about their tutees to perform better than expected in the game. While the majority of games were played among peers in the same grade level, the third question tested whether the game-based approach retained its value for cross-age student pairs. We found that the age of the tutee is a much stronger indicator of the difficulty of posed challenges than the age of the tutor, with the focus on challenges that are tutee-appropriate (rather than tutor-appropriate.) Finally, the fourth question examined if and when tutees improved at the spelling task with use of our system. We explored this at three levels of problem granularity, and all three analyses support the conclusion that students did improve with use of the system. Together, the results of these four inquiries support our argument that a Teacher's Dilemma game retains its motivational value when implemented in software and played by students.

Chapter 6

Conclusion

In exploring the idea of a game as a mechanism for motivating the selection of appropriate challenges for learning, and in then constructing and evaluating several systems to support games based on this idea, we have raised many issues that have yet to be resolved. Several of these issues offer opportunities for future research, and we discuss these here.

6.1 Discussion

While the Teacher's Dilemma offers a simple approach to structuring game-based learning, we observed discrepancies between the expected behavior of a "rational" player and the observed behavior of actual students using the SpellBEE and BEEweb activities. Understanding the nature of these gaps and identifying techniques that may close them provide one opportunity for improving learning outcomes. Real players do not act strictly rationally, for example. Perhaps a game could be constructed such that it converges on appropriate challenge given a different set of assumptions (i.e. other than Expected Utility theory) about player behavior.[1] Since, for example, Teachers seem to consistently overestimate the likelihood of accurate responses to very difficult challenges (from their Students), the game payoffs may be pre-adjusted to compensate. Subsequently, even though tutors still overestimate this likelihood, the frequency with which they choose these challenges may remain at their intended (i.e. low) level. One can conceive of a system in which this process – of observing aggregate student biases and adjusting the game payoffs to compensate – takes place automatically, forming the basis for a motivational meta-structure that adapts to the collective decision-making processes of its users. While this system may no longer meet the Teacher's Dilemma criteria under the assumption of rationality, it could still effectively meet the Teacher's Dilemma criteria assuming the observed non-rational decision-making strategies of the actual players in the game.

In Chapter 2, we introduced the notion of appropriate challenge, and constructed a probabilistic definition for which an optimal challenge yields an accurate response from the tutee with probability $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = 0.5$. We stress here that the core value of the model is derived from the goal being defined in terms of the probability of response accuracy, and the actual value of this probability may be varied if needed. In domains for which an incorrect response carries additional persistent cost (e.g. in a domain with physical challenges, an incorrect response may lead to physical injury), a higher probability may be preferable. In order to satisfy this new notion of appropriateness, we simply adjust the game to match. If we define appropriateness at the $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = \frac{2}{3}$ level, the only adjustment necessary to the difficulty-based game compatible is to alter one payoff value: we change the Teacher's reward for a correct response from $\mathcal{D}_c$ to $\frac{\mathcal{D}_c+1}{2}$. All proofs in Section 2.4.1 still hold, and the game meets the

[1] Kahneman and Tversky's Prospect Theory [45] offers one such alternative.

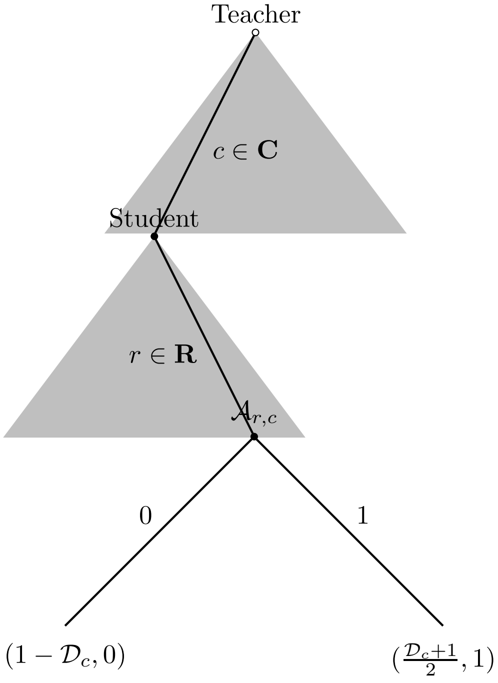


Figure 6.1: A modified *difficulty*-based Teacher's Dilemma game, in which the change to the Teacher's payoff for a correct responses to make it motivate a new notion of appropriateness, which is maximized when $\mathrm{P}\left[\mathcal{A}_{r,c}\right] = \frac{2}{3}$.

Teacher's Dilemma criteria under this new definition of appropriateness. Figure 6.1 illustrates this modified game. This flexibility suggests that the Teacher's Dilemma game-based approach can be applied to any task domains for which appropriateness can be expressed in terms of the probability of response accuracy. Experimenting with the model in such domains offers the opportunity to generalize this approach to learning domains for which $\mathrm{P}\left[\mathcal{A}_{r,c}\right] \neq \frac{1}{2}$.

Both the SpellBEE and BEEweb systems are built on a synchronous activity model. While the synchronicity adds immediacy and excitement to the game-play, it also brings with it a number of limitations that must be acknowledged. First, for a game to be initiated, two players must simultaneously be logged in and ready to be matched. Given the short span of time that a lone student is willing to wait for another player to arrive (generally less than a minute), that student will likely exit before the next player enters. We explored a variety of techniques to get players to arrive at the same time, ranging from implementing a daily tournament that begins at a specific time every afternoon, to suggesting to lone users to call a friend and ask them to login, to implementing a "practice round" mode in the BEEweb activities, to providing users with desktop status-bar applications to provide them with the information to log in only when they know they can be matched. Beyond this complication added to player matching, synchronicity places limits on the type of domain that can be used. By making the activity symmetric, both players are simultaneously occupied with the same step, either selecting challenges, constructing responses, or viewing feedback. One player may be faster than the other at completing the step, and must then wait for their partner to finish. By adding a time limit to each step (e.g. 30 seconds for challenge selection in SpellBEE), we effectively put a limit on the maximum length that a fast player may have to wait. But while this may limit the wait, it also limits the complexity of the problems that we may reasonably ask a student to attempt to solve. One viable alternative, which we have only recently begun to explore, is to embed a Teacher's Dilemma game into an asynchronous activity. The BEEmail system, as described in Appendix A, represents our first attempt at doing this. Players need not use the game simultaneously, games can be initiated by anyone with anyone (regardless of whether or not they have played before) and the task domain is not limited by the amount of time that users need to solve (or select) a challenge problem. We anticipate that, while the game may lose some of the excitement of a race, the rates of game activity, new user adoption, and participant retention would all be higher for an asynchronous activity.

While the expectation-based TD game and the equivalence-based TD game were each designed to provide certain advantages over the original difficulty-based TD game formulation, we have yet to observe how any of them perform in practice. The expectation-based game, for example, offers a mechanism for assessing challenge appropriateness that is designed to be dynamic and self-correcting (i.e. by basing it on the tutors' expectations), but we have yet to assess how accurate tutors' expectation statements are or how flexibly these expectations change in response to observed tutee performance. By deploying systems based on these newer games, we can begin to assess and compare their various affects on tutor challenge selection and tutee learning.

In keeping with our goal of protecting student safety as described in Section 4.1.3, we have allowed no open lines of communication between game players. If we were able to guarantee that communications would not compromise student safety, adding a facility for student dialogue could be very productive, as it has the potential to enhance a tutor's ability to help their tutee learn (subject to the limits of the tutors abilities). The equivalence-based Teacher's Dilemma game offers one example of how a game can motivate productive dialogue. While we can never guarantee that in-game communication cannot affect student safety, we can provide a weaker guarantee that may be sufficient in many contexts. Namely, we guarantee that the game poses *no additional risk* to student safety, if we require players to prove that they are already in contact before enabling any in-game communication between them. Such a proof might consist of an offline message-passing task (e.g. each player is provided with a different password that they must convey to the other, in person or by phone, and both must type in the other password in order to enable the chat functionality.) If suitably defined, such a proof is sufficient to enable dialogue without introducing any new risks to personal safety. In doing so, we can enable the type of collaborative dialogues discussed by Graesser et al. [32], as has been incorporated in several tutoring systems built for use in controlled classroom settings among trusted users [47, 82, 86].

6.2 Summary

We have organized the chapters of this dissertation to reflect our five main contributions:

In Chapter 2, we introduced a model of appropriate challenge to reflect the probability of learning, define the Teacher's Dilemma as set of a criteria for games that motivate appropriate challenges, and provided three examples of games that meet the Teacher's Dilemma criteria. We provided proofs for each of these games. In Chapter 3, we illustrated, via computer simulations, that symmetric repeated play of one of these games, under certain assumptions, converges to player strategies in which the tutor poses the tutee with challenges of appropriate difficulty and the tutee replies with a best-effort response, consistent with the Teacher's Dilemma criteria. In Chapter 4, we described two separate systems we have built in order to allow pairs of students to engage in a Teacher's Dilemma game across the internet. We described the design goals and implementation decisions involved in constructing each of these systems. In Chapter 5, we summarized and analyzed the data collected from several thousand students using two of these activities over the course of several years, and found that students were generally responsive to the Teacher's Dilemma mechanism (even in cross-age matches), that peer tutors were able to strategically leverage their own knowledge of the domain and of the tutee when selecting challenges to pose, and that tutees improved at the learning task over time with use of the game (at three levels of measurement granularity).

The game-theoretic analysis establishes the possibility for a game-based mechanism for motivating appropriate challenges, the simulations support the plausibility of this approach given non-optimal players, the implemented software systems demonstrate the scalability of this model, and the data analysis supports the real-world applicability of this approach. Together, these contributions suggest that Teacher's Dilemma games – and the two-person learning activities built upon them – offer a mechanism for motivating peer learners to provide one another with challenges of appropriate difficulty for learning.

Appendix A

BEEmail: Proof-of-concept of a decentralized architecture for an asynchronous game

Where the SpellBEE activity and the BEEweb platform have both been released publicly, a third system architecture for Teacher's Dilemma games has been implemented but not yet tested or released. This "BEEmail" system serves as a proof-of-concept of a decentralized architecture for an asynchronous[1] Teacher's Dilemma game. The primary advantage of a decentralized system architecture is in increasing scalability: If we can increase the number of concurrent active users without increasing the computational resources that we must provide to support those users, our resource limitations will not limit how many users can simultaneously participate. In demonstrating that this architecture can support activities, we support our claim that a Teacher's Dilemma game-based model is highly scalable, theoretically capable of sup-

[1] The game we describe is asynchronous in that the two players do not act at the same time.

porting large numbers of simultaneous users.

The centralized server in the SpellBEE and BEEweb architectures fills a variety of roles:

- *match-making services* allow players to find and initiate games with one another.
- *message-passing services* allows game messages to flow between matched players, where it would otherwise be restricted (based on the Java applet security policy.)
- *game-state persistence* maintains the coherence of a turn-taking game.

In abandoning the centralized server model, alternatives must be provided to fulfill each of these roles. With BEEmail, we do this in a somewhat novel way: we piggyback on an existing asynchronous decentralized communication network that, we assume, our players are already using: email. As most web browsers are configured to process *mailto* URLs in the default email client[2] and most email clients are configured to open clicked *http* URLs in the default web browser[3], we can treat email as a semi-automated mechanism for message-passing among web-based game clients. We note that, for the average user, this requires no additional software installation or network configuration. In the following section, we discuss how we designed a system for Teacher's Dilemma games based on this architecture.

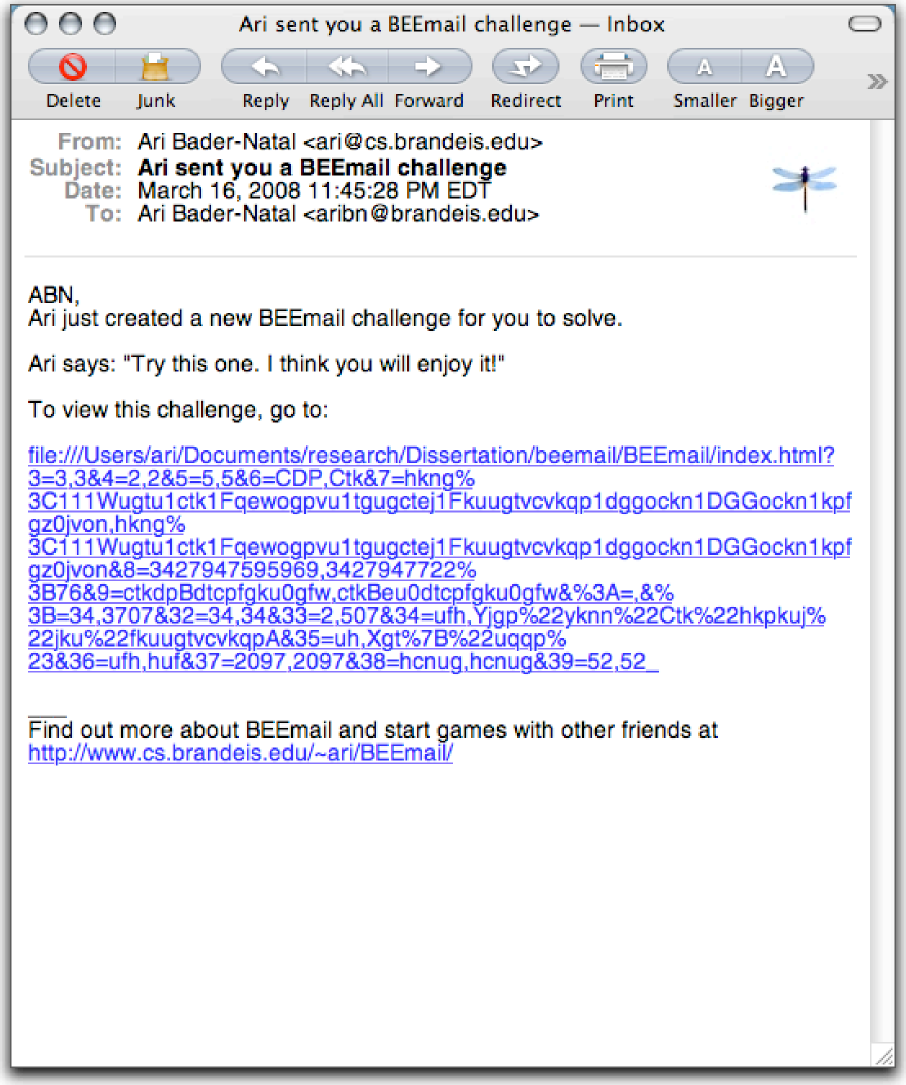
[2] Through the *mailto* URL scheme [39], web browsers offer a mechanism for composing a new email message and auto-filling certain fields of that message (e.g. the recipient's email address, the message subject, the message body.)

[3] Email clients (with some exceptions) render all URLs within plain-text emails as click-able links that open in the default web browser.

A.1 Implementing a decentralized architecture for an asynchronous game

The game may still be played within a browser, but actions that would have initiated a message being sent to the server will now instead auto-compose an email message, with the relevant data encapsulated in a specially-formed URL. A message can thus be passed from one client browser to another as follows: The first browser auto-generates an email message addressed to the user of the second browser, with the message encoded and appended as the query string portion of a URL. When this message is received, clicking on that URL opens a browser window, and visits the specified URL, which can, upon loading, decode the original message from the query string of that URL. Figure A.1 shows a sample BEEmail message, as it appears in the recipient's email inbox.

Figure A.2 provides a sketch of the client-side Javascript functions for encoding game state in a *mailto* URL, and decoding game state from the query string of a *http* URL. In this code, the `sendState()` function is invoked when a player completes their turn. This function first serializes the current state of the game into a URL-encoded string, then composes a new email message using the browser-specified email client. The email is automatically addressed to the other player, the message subject is set, and a basic message body is generated. This body includes the URL necessary for the other player to continue the game in their own preferred game client. The `renderState()` function is invoked after the game client HTML page finishes loading. This function first extracts the serialized game state string from the query string of the current URL. That string is then deserialized, and the resulting game state is rendered in the browser window.



From: Ari Bader-Natal <ari@cs.brandeis.edu>
Subject: **Ari sent you a BEEmail challenge**
Date: March 16, 2008 11:45:28 PM EDT
To: Ari Bader-Natal <aribn@brandeis.edu>

ABN,
Ari just created a new BEEmail challenge for you to solve.

Ari says: "Try this one. I think you will enjoy it!"

To view this challenge, go to:

file:///Users/ari/Documents/research/Dissertation/beemail/BEEmail/index.html?
3=3,3&4=2,2&5=5,5&6=CDP,Ctk&7=hkng%
3C111Wugtu1ctk1Fqewogpvu1tgugctej1Fkuugtvcvkqp1dggockn1DGGockn1kpf
gz0jvon,hkng%
3C111Wugtu1ctk1Fqewogpvu1tgugctej1Fkuugtvcvkqp1dggockn1DGGockn1kpf
gz0jvon&8=3427947595969,3427947722%
3B76&9=ctkdpBdtcpfgku0gfw,ctkBeu0dtcpfgku0gfw&%3A=,&%
3B=34,3707&32=34,34&33=2,507&34=ufh,Yjgp%22yknn%22Ctk%22hkpkuj%
22jku%22fkuugtvcvkqpA&35=uh,Xgt%7B%22uqqp%
23&36=ufh,huf&37=2097,2097&38=hcnug,hcnug&39=52,52_

__
Find out more about BEEmail and start games with other friends at
http://www.cs.brandeis.edu/~ari/BEEmail/

Figure A.1: A sample BEEmail message, as it appears in the recipient's email inbox. Clicking on the included link opens a web browser and renders the current state of this game.

```
function sendState() {
  queryString = serializeState(gameState);
  stateLink = pathToPartnersGameClient + "?" + queryString
  message = myName + " created a new BEEmail problem for you"
     + " to solve. To view this challenge, go to: " + stateLink;
  location.href = "mailto:" + partnersEmail + "?subject=" + myName +
     " sent you BEEmail&body=" + message
}

function renderState() {
  queryString = location.search;
  gameState = deserializeState(queryString);
  renderUI(gameState);
}
window.onload = renderState;
```

Figure A.2: A Javascript code sketch for the message-passing and message-receiving functionality in BEEmail.

While this email-based mechanism may be somewhat cumbersome, it is an entirely sufficient basis upon which to construct a decentralized approach to asynchronous message-passing. The game client simply consists of a web page (rendered by any modern web browser) that uses an available scripting language (such as JavaScript) to parse a URL in order to decode and render game state and to encode game state in a mailto URL. If the game client HTML page is written in a domain-independent manner (e.g. in which all URLs are relative), it will behave consistently regardless of which site hosts the page or even the filename of the game client itself. Assuming the game client HTML has been written in this domain-independent manner, it can even function properly when hosted locally on the user's own PC, and accessed in the browser via the `file` protocol, specifying the local path of the HTML document. If both players cache a copy of the game client locally and access it in this way, no web server is necessary at all. Alternatively, since no server-side processing is required, a third party could host a game client simply by placing the HTML file in any web-accessible directory.

The technique of encapsulating an application entirely within an HTML file that can be relocated (and even hosted locally) has been used in other contexts. The TiddlyWiki "reusable non-linear personal web notebook" [70] offers one powerful example of this technique. This wiki works when hosted online, run from the user's desktop, or plugged in and run from a USB "thumb drive." Installation involves simply moving a file.

The idea of piggy-backing communications on an existing distributed message protocol has been used previously in a tutoring system, as the EduBingo program [53] relies on the Extensible Messaging and Presence Protocol (XMPP), "an open XML technology for presence and real-time communication" [71]. XMPP-based game messaging enables transparent communications (i.e. the user does not ever have to send or receive messages manually), but in this case we chose email for the simplicity of the integration and the opportunity to experiment with asynchronous game play.

Where the SpellBEE and BEEweb systems are both designed around synchronous play, asynchronous games have a unique set of characteristics that are worth exploring for Teacher's Dilemma games. First, asynchronicity simplifies player matching, as anyone can initiate a game with anyone else at any time simply by using the game client to send a specially-formatted email. The recipient need not have participated before, whereas the recipient of a game request in the synchronous SpellBEE and BEEweb activities must have registered for an account and be logged in at the same time as the requesting player. Second, the time limits on each step of the synchronous game – imposed to bound the amount of time that a player might have to wait before being able to move to the next step – may be eliminated if desired. In doing so, we can experiment with task domains that require 5–10 minutes (or more) to respond to a challenge, where such a game would not be feasible within a fast-pace synchronous game framework. The slower pace may, however, have a negative affect on engagement, and we would need to examine and understand an such relationship.

A.2 Issues with email-embedded messaging and distributed client architecture

We note that in implementing BEEmail, we faced several issues specific to email-embedded messaging and a distributed client architecture. We mention briefly how we addressed five of these issues:

- *Game state can be manipulated in transition.* If the serialization process for encoding game state into a URL does not encrypt or obfuscate the game state, the game state may be easily altered. If the URL includes the correct answer in plain-text, for example, by simply viewing the URL, the recipient may undermine the activity. If the URL includes the players' scores, a player can alter these scores before visiting the URL. In BEEmail, we avoid these issues by obfuscating the game state before appending it to the game client URL, as illustrated in Figure A.1. We believe that removing the human-readability of the game state should be sufficient to prevent most such abuses.
- *Game-state messages persist after play.* Given that the game state is entirely encapsulated in the email messages sent and received by players, if an outdated email-embedded link is clicked, game-play reverts to that outdated game state. In order to detect when this happens, we cache time-stamped copies of game's

state (one per partner) locally in a browser cookie, and refer to these cookies upon rendering a game state, to identify stale links and notify the user.

- *Email clients may alter long URL links with line-wrapping.* Some email clients limit the length of links, and others insert spaces when line-wrapping. We identify partial links (and notify the user), by appending a special token to the end of every game URL, and checking for this token before rendering a game page. Another approach is to generate HTML emails rather than plain-text emails, since links are never broken and are handled more consistently. As there is no mechanism for composing HTML emails via a *mailto* URL, HTML emails must be sent indirectly, through a server-hosted script.[4]

- *Two players may have different versions of the game client.* We embed the game client version number as part of the game state, and if the version numbers of the two players in a game do not agree, the user running the older version is prompted to update to the latest version. A game can successfully be initiated only once both players are running the same version of the game client.

- *Game state messages must embody both of the Teacher-Student games.* In order for each game to have a single score per player in our model in which the complete game state is embodied in the messages sent between users, both of the simultaneously-played Teacher's Dilemma games must be encapsulated in the game state. We do this by aggregating the challenge, response, and feedback activities so that each player interaction incorporates activities from both of the games. Specifically, once the game is underway, each turn involves four such

[4] In this case, the game client would submit a GET or POST request to the server that encapsulated the game state, and the server would compose and send an HTML email on behalf of the player.

activities: the player is presented with feedback on how they performed on the previous challenge, the player solves a new challenge, the player is presented with feedback on how their partner performed on the last challenge posed, and the player poses a new challenge for their partner.[5] Once all four parts of the turn are completed, a game state message is sent to the other player. In aggregating turn activity in this way, a single message can encapsulate game state from the two simultaneous Teacher's Dilemma games.

By integrating a locally-hosted web application with an email-based distributed technique for maintaining game state, we demonstrate a proof-of-concept of a decentralized, scalable architecture for an asynchronous Teacher's Dilemma game.

[5] In the first few rounds, the steps for which no data is yet available are skipped.

Appendix B

SpellBEE Misspelling Data

We wish to make public to the research community two sets of data that we have collected, both from the SpellBEE system, and both related to American English spelling errors. Both data sets are described, and sample data (for one word) is included for each. The data sets in their entirety, covering all SpellBEE words, are available online at http://www.cs.brandeis.edu/~ari/dissertation/.

When examining the SpellBEE data, we can distinguish among different categories of spelling errors. Here, we use two tools – the Porter2 word stemmer [66] and the GNU Aspell spell-checking program (version 0.60.4) [4] – to assist in classifying each spelling error into one of four groups. The stemmer is used to determine if the challenge word and the response string share a common word stem, and the spell-checker is to see if the response string is "close" to the challenge word.[1] The categories are determined as follows:

- *Category 1*: includes all correct responses, and is not included in this data.

[1] For each incorrect spelling, Aspell generates a list of possible words and we check if the challenge word appears in that list. Specifically, we use the "bad-spellers" mode of suggestion generation for American English.

- *Category 2*: includes incorrect responses for which the response is a valid spelling of an English word *other than* the challenge word, which bears no resemblance to the challenge word, according to the two tools used.[2] This may indicate that the student did not understand the instructions or could not hear the audio, and entered their response based on appropriate part of speech only.)

- *Category 3*: includes incorrect responses for which the response is not a valid spelling of any English word, but is similar to the challenge word, based on the spell-checker. This likely indicates the student understood the challenge, but was not able to respond accurately.

- *Category 4*: includes incorrect responses for which the response is not a valid spelling of any English word and is not similar to the challenge word. This likely indicates that the student did not try, for some reason, to answer the challenge posed.

- *Category 5*: includes incorrect responses for which the response is a valid spelling of an English word *other than* the challenge word, and this word bears a resemblance to the challenge word, according to one of the two tools used.[3] This may indicate that the student could not clearly hear the word spoken.

While this classification scheme may suggest the source of the error, we have not verified the categorization of individual errors with the students who made them. Thus, we offer two different views of our collected data, one filtered by classification,

[2] Since the spell-checker does not generate suggestions for valid English words, we first break the spelling of the typed word by duplicating the last letter, and then use the spelling suggestion facility.

[3] Like in Category 2, the response word is first broken by duplicating the final letter. But, in this case, the spell-checker includes in its list of suggestions the challenge word.

and the other filtered by frequency. In the first data set, we include only Category-3 errors, providing a data on the relative frequencies of true misspellings. In the second data set, we include (and label) misspellings from all categories, but omit those misspellings that were observed only once.

We omit correct spellings (Category-1 responses) from both data sets, as the relative frequency of these spellings is directly affected by the payoff structure of the game. As suggested by the analysis in Section 5.3, the number of correct responses to the most difficult challenges is disproportionately high, and the number of incorrect responses to the least difficult challenges is disproportionately low. In setting aside accurate responses, we believe that the relative frequencies among the remaining (i.e. incorrect response) data are not biased by the challenge selection process resulting from the Teacher's Dilemma.

B.1 Dataset 1: Category-3 errors

Table B.2 includes an excerpt from the `spellbee_errors1.txt` file online. This dataset lists, for each word in the SpellBEE dictionary, every student response to it that is classified in Category-3. Frequency data is included for each challenge-response combination, in descending order. In total, this data set details 99,498 instances of 44,450 Category-3 misspellings of 2,984 words.

B.2 Dataset 2: Non-unique errors

Table B.2 includes an excerpt from the `spellbee_errors2.txt` file online. This dataset lists, for each word in the SpellBEE dictionary, every incorrect response with a frequency greater than 1. The category and frequency of each response is listed, ordered descending by frequency. In total, this data set details 102,181 instances of 18,150 non-unique misspellings of 2,764 words.

Table B.1: Category-3 misspellings of the word “accommodation”, listed by response frequency.

Misspelling	Freq.
accomidation	66
accomadation	44
accomodation	38
accomodations	37
acomidation	36
comidation	32
acomadation	26
accomidations	16
accomadations	14
acommadation	11
accomedation	9
comadation	8
commadation	5
acomadations	5
acomination	5
acomedation	5
commidation	5
acomodation	4
acombination	4
acomanation	3
ocomidation	3
comodation	3
accommidation	3
acommedation	2
acamadation	2
acommidation	2
acommodation	2
accodimation	2
commodations	2
comedation	2
accomination	2
acomidashion	2
accomidashin	2
ecomidations	2
acomidations	2
comidashin	1
comidashon	1
comidatoin	1
comindation	1
commadations	1
commadionshon	1
commedatio	1
commedition	1
commidatassion	1
commidattion	1
commodation	1
coumiedashaon	1
echomonation	1
ecomidation	1
ecomondonation	1
icomidation	1
icomination	1
ocamadation	1
occomidation	1
ocomadashion	1
ocomidations	1
ocommedition	1
ocommination	1
accomodation	1
aaccomination	1
acamadshons	1
acamidation	1
accmoidations	1
accodamation	1
accodamedation	1
accoidation	1
accomaddition	1
accomadiction	1
accomadition	1
accomaditions	1
accomadtion	1
accomanation	1
accombidation	1
accomdadation	1
accomdation	1
accomdations	1
accomedations	1
accomedatoin	1
accomeedation	1
accomendation	1
accomendations	1
accomidak	1
accomidashions	1
accomidassion	1
accomidat	1
accomidatio	1
accomidatiom	1
accomidatons	1
accomidfation	1
accomindation	1
accomitdation	1
accommadation	1
accommendation	1
accommodati	1
accommodatoin	1
accommondation	1
accommondition	1
accomondation	1
acodimation	1
acoidashon	1
acomadacoin	1
acomadaitoin	1
acomadaon	1
acomadashons	1
acomadasion	1
acomadatin	1
acomadayshen	1
acomadtion	1
acomantion	1
acombatoin	1
acombdation	1
acombitation	1
acombnation	1
acomdiation	1
acomdition	1
acomedashon	1
acomedations	1
acomedaytion	1
acomedtion	1
acomendation	1
acometion	1
acomidaion	1
acomidaition	1
acomidaiton	1
acomidashone	1
acomidatoin	1
acomidayshen	1
acommadtion	1
acommdions	1
acommendations	1
acommitation	1
acommonation	1
acondimation	1
akomidesion	1
camadation	1
comadashon	1
comadtion	1
combadation	1
combidation	1
comedashin	1
comedashon	1
comedatoin	1
comeddation	1

Table B.2: Non-unique misspellings of the word “accommodation”, listed by response frequency and error category.

Misspelling	Category	Freq.
accomidation	3	66
accomadation	3	44
accomodation	3	38
accomodations	3	37
acomidation	3	36
comidation	3	32
acomadation	3	26
accomidations	3	16
accomadations	3	14
acommadation	3	11
accommodations	5	11
combination	2	10
accomedation	3	9
comadation	3	8
commadation	3	5
acomadations	3	5
acomination	3	5
acomedation	3	5
commidation	3	5
comination	4	4
acomodation	3	4
comadations	4	4
acombination	3	4
acomanation	3	3
place	2	3
ocomidation	3	3
comodation	3	3
accommidation	3	3
comadasons	4	2
com	2	2
acommedation	3	2
acamadation	3	2
acommidation	3	2
co	2	2
accom	4	2
acommodation	3	2
water	2	2
accodimation	3	2
acc	4	2
finest	2	2
food	2	2
commodations	3	2
acomad	4	2
acom	4	2
acomida	4	2
comenation	4	2
comedation	3	2
accomination	3	2
acomidashion	3	2
accomidashin	3	2
ecomidations	3	2
acomidations	3	2
combanation	4	2
a	4	2

Bibliography

[1] Alan Agresti. *Categorical Data Analysis*. Wiley, New York, 1990.

[2] John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray Pelletier. Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences*, 4(2):167–207, 1995.

[3] J.R. Anderson, C.F. Boyle, and B.J. Reiser. Intelligent Tutoring Systems. *Science*, 228(4698):456–462, 1985.

[4] Kevin Atkinson. GNU Aspell, 2006. http://www.gnu.org/software/aspell/.

[5] Ari Bader-Natal and Jordan B. Pollack. Evaluating problem difficulty rankings using sparse student data. In *Educational Data Mining: Supplementary Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-2007)*, pages 1–10, Marina del Rey, CA, July 2007. IOS Press.

[6] R.S. Baker, A.T. Corbett, and K.R. Koedinger. Detecting student misuse of intelligent tutoring systems. In *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, LNCS 3220, pages 531–540, Berlin Heidelberg, 2004. Springer-Verlag.

[7] Tiffany Barnes. The q-matrix method: Mining student response data for knowledge. Technical Report WS-05-02, AAAI-05 Workshop on Educational Data Mining, Pittsburgh, 2005.

[8] B. Barros, R. Conejo, and E. Guzman. Measuring the Effect of Collaboration in an Assessment Environment. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-2007)*, volume 158, page 375. IOS Press, 2007.

[9] M.V. Belmonte, E. Guzmán, L. Mandow, E. Milan, and J.L.P. de Ia Cruz. Automatic generation of problems in web-based tutors. In L.C. Jain, R.J. Howlett, N. S. Ichalkaranje, and G. Tonfoni, editors, *Virtual Environments for Teaching & Learning*, chapter 7, pages 237–281. World Scientific, London, 2002.

[10] Alan D. Blair. Co-evolutionary learning: Lessons for human education? In *Proceedings of the Fourth Conference of the Australasian Cognitive Science Society*, Newcastle, Australia, 1999.

[11] Benjamin S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, Jun. - Jul. 1984.

[12] Leonard S. Cahen, Marlys J. Craun, and Susan K. Johnson. Spelling difficulty – a survey of the research. *Review of Educational Research*, 41(4):281–301, October 1971.

[13] J.R. Carbonell. AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):190–202, 1970.

[14] Tak-Wai Chan and Chih-Yueh Chou. Exploring the design of computer supports for reciprocal tutoring. *International Journal of Artificial Intelligence in Education*, 8:1–29, 1997.

[15] T.W. Chan, YL Chung, R.G. Ho, W.J. Hou, and G.L. Lin. Distributed Learning Companion Systems—West Revisited. *The 2nd International Conference of Intelligent Tutoring Systems, C. Frasson, G. Gauthier & G. McCalla (Eds.). Lecture Notes in Computer Science*, 608:643–650, 1992.

[16] T.W. Chan, C.W. Hue, C.Y. Chou, and O.J.L. Tzeng. Four spaces of network learning models. *Computers & Education*, 37(2):141–161, 2001.

[17] K. Chang, J.E. Beck, J. Mostow, and A. Corbett. Does Help Help? A Bayes Net Approach to Modeling Tutor Interventions. *AAAI2006 Workshop on Educational Data Mining*, 2006.

[18] Li-Jie Chang, Jie-Chi Yang, Tak-Wai Chan, and Fu-Yun Yu. Development and evaluation of multiple competitive activities in a synchronous quiz game system. *Innovations in Education and Teaching International*, 40(1):16–26, 2003.

[19] Chih-Ming Chen, Hahn-Ming Lee, and Ya-Hui Chen. Personalized e-learning system using item response theory. *Computers & Education*, 44:237–255, 2005.

[20] Chih-Ming Chen, Chao-Yu Liu, and Mei-Hui Chang. Personalized curriculum sequencing utilizing modified item response theory for web-based instruction. *Expert Systems with Applications*, 30, 2006.

[21] Peter A. Cohen, James A. Kulik, and Chen-Lin C. Kulik. Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, 19(2):237–248, Summer 1982.

[22] Ricardo Conejo, Eduardo Guzmán, Eva Millán, Mónica Trella, José Luis Pérez-De-La-Cruz, and Antonia Ríos. Siette: A web-based tool for adaptive testing. *International Journal of Artificial Intelligence in Education*, 14:29–61, 2004.

[23] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience.* Harper & Row, New York, 1990.

[24] R.I. Damper, Y. Marchand, J.D.S. Marsters, and A.I. Bazin. Aligning text and phonemes for speech technology applications using an EM-like algorithm. *International Journal of Speech Technology*, 8(2):147–160, 2005.

[25] Mathew Davies and Elizabeth Sklar. Modeling human learning as a cooperative multi agent interaction. In *AAMAS Workshop on Humans and Multi-Agent Systems, at the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003.

[26] Joseph C. Delquadri, Charles R. Greenwood, Kathleen Stretton, and R. Vance Hall. The Peer Tutoring Spelling Game: A Classroom Procedure for Increasing Opportunity to Respond and Spelling Performance. *Education and Treatment of Children*, 6(3):225–239, Summer 1983.

[27] Godfrey Dewey. *Relative Frequency of English Spellings.* Teachers College Press, New York, 1970.

[28] John W. Fantuzzo, Judith Alperin King, and Lauren Rio Heller. Effects of reciprocal peer tutoring on mathematics and school adjustment: A component analysis. *Journal of Educational Psychology*, 84(3):331–339, 1992.

[29] Gerhard H. Fischer and Ivo W. Molenaar, editors. *Rasch Models: Foundations, Recent Developments, and Applications.* Springer-Verlag, New York, 1995.

[30] Drew Fudenberg and Jean Tirole. *Game Theory.* MIT Press, Cambridge, MA, 1998.

[31] Herbert Gintis. *Game Theory Evolving.* Princeton University Press, Princeton, New Jersey, 2000.

[32] Arthur C. Graesser, Natalie K. Person, and Joseph P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9:495–522, 1995.

[33] Harry A. Greene. *New Iowa Spelling Scale*. State University of Iowa, Iowa City, 1954.

[34] Charles R. Greenwood. Monitoring, improving, and maintaining quality implementation of the classwide peer tutoring using behavioral and computer technology. *Education & Treatment of Children*, 16(1):19–48, February 1993.

[35] Eduardo Guzmán and Ricardo Conejo. A library of templates for exercise construction in an adaptive assessment system. *Technology, Instruction, Cognition and Learning (TICL)*, 2(1-2):21–32, 2004.

[36] Constantia Hadjidemetriou. Using rasch models to reveal countours of teachers' knowledge. *Journal of Applied Measurement*, 5(3):243–257, 2004.

[37] Paul R. Hanna, Jean S. Hanna, Richard E. Hodges, and Jr. Edwin H. Rudorf. Phoneme-grapheme correspondences as cues to spelling improvement. Research Report OE-32008, Office of Education / U.S. Department of Health, Education, and Welfare, 1966.

[38] Michael Hart. Project Gutenberg. http://www.gutenberg.org/.

[39] P. Hoffman, L. Masinter, and J. Zawinski. The mailto URL scheme. http://www.ietf.org/rfc/rfc2368.txt.

[40] Robin Hunicke and Vernell Chapman. AI for dynamic difficulty adjustment in games for dynamic difficulty adjustment in games. In *In Proceedings of the Challenges in Game AI Workshop, Nineteenth National Conference on Artificial Intelligence (AAAI '04)*, San Jose, 2004. AAAI Press.

[41] Jeff Johns, Sridhar Mahadevan, and Beverly Woolf. Estimating student proficiency using an item response theory model. In M. Ikeda, K. Ashley, and T.-W. Chan, editors, *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS-2006)*, pages 473–480, 2006.

[42] David W. Johnson and Roger T. Johnson. *Learning together and alone: cooperative, competitive, and individualistic learning*. Allyn & Bacon, Boston, 1994.

[43] D.W. Johnson and R.T. Johnson. *Cooperation and Competition: Theory and Research*. Interaction Book Company, Edina, Minnesota, 1989.

[44] Roger T. Johnson, David W. Johnson, and Mary Beth Stanne. Effects of cooperative, competitive, and individualistic goal structures on computer-assisted instruction. *Journal of Educational Psychology*, 77(6):668–677, December 1985.

[45] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, March 1979.

[46] Alison King. ASK to THINK-TEL WHY: A Model of Transactive Peer Tutoring for Scaffolding Higher Level Complex Learning. *Educational Psychologist*, 32(4):221–235, 1997.

[47] Alison King. Transactive peer tutoring: Distributing cognition and metacognition. *Educational Psychology Review*, 10(1):57–75, March 1998.

[48] Alison King. Structuring peer interaction to promote high-level cognitive processing. *Theory into Practice*, 41(1):34–41, Winter 2002.

[49] John Kirriemuir and Angela McFarlane. Literature review in games and learning.

[50] Raph Koster. *A Theory of Fun for Game Design.* Paraglyph Press, 2004.

[51] V.S. Kumar, G.I. McCalla, and J.E. Greer. Helping the peer helper. In *Proceedings of the International Conference on AI in Education*, pages 325–332, 1999.

[52] Mark R. Lepper. Motivational considerations in the study of instruction. *Cognition and Instruction*, 5(4):289–309, 1988.

[53] Hui-Chun Liao. EduBingo: A bingo-like game for skill building. Master's thesis, National Central University, Jhongli, Taiwan, 2005.

[54] Frederic M. Lord. *Applications of Item Response Theory to practical testing problems.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

[55] Larry Maheady, Barbara Mallette, and Gregory F. Harper. Four classwide peer tutoring models: similarities, differences, and implications for research and practice. *Reading and Writing Quarterly*, 22:65–89, 2006.

[56] M. Mayo and A. Mitrovic. Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, 12(3), 2001.

[57] BM McLaren, E. Walker, K. Koedinger, N. Rummel, H. Spada, and M. Kalchman. Improving Algebra Learning and Collaboration through Collaborative Extensions to the Algebra Cognitive Tutor. *Poster Presented at CSCL-05, Taipei, Taiwan*, 2005.

[58] Bruce M. McLaren, Lars Bollen, Erin Walker, Andreas Harrer, and Jonathan Sewall. Cognitive tutoring of collaboration. In *Proceedings of the 2005 conference on Computer support for collaborative learning*, 2005.

[59] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June 1947.

[60] Janet Metcalfe and Nate Kornell. The dynamics of learning and allocation of study time to a region of proximal learning. *Journal of Experimental Psychology: General*, 132(4):530–542, 2003.

[61] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting information feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, September 2005.

[62] Noam Nisan. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[63] Noam Nisan. Introduction to mechanism design (for computer scientists). In E. Tardos N. Nisan, T. Roughgarden and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press, Cambridge, 2007.

[64] Angela M. O'Donnell and Alison King, editors. *Cognitive Perspectives on Peer Learning*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 1999.

[65] Jordan B. Pollack and Alan D. Blair. Co-evolution in the successful learning of backgammon strategy. *Machine Learning*, 32(3):225–240, 1998.

[66] Martin Porter. The English (Porter2) Stemming Algorithm, 2002. http://snowball.tartarus.org/algorithms/english/stemmer.html.

[67] Georg Rasch. *Probabilistic Models for Some Intelligence and Attainment Tests*. The University of Chicago Press, Chicago, 1980.

[68] Robert Rieber and Aaron Carton, editors. *The collected works of L. S. Vygotsky*, volume 1: Problems of General Psychology. Plenum Press, 1978.

[69] Jack Robertson and William Webb. *Cake-Cutting Algorithms*. A K Peters, Ltd., Natick, Massachusetts, 1998.

[70] Jeremy Ruston. TiddlyWiki. http://www.tiddlywiki.com/.

[71] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Core. http://www.ietf.org/rfc/rfc3920.txt.

[72] Dorothea P. Simon and Herbert A. Simon. Alternative uses of phonemic information in spelling. *Review of Educational Research*, 43(1):115–137, Winter 1973.

[73] E. Sklar and S. Parsons. Towards the Application of Argumentation-Based Dialogues for Education. *International Conference on Autonomous Agents: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-*, 3:1420–1421, 2004.

[74] R.E. Slavin. Co-operative learning. *The Social Psychology of the Primary School*, 1990.

[75] Robert E. Slavin. Research on cooperative learning and achievement: What we know, what we need to know. *Contemporary Educational Psychology*, 21(1):43–69, January 1996.

[76] H. Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.

[77] K. VanLehn. The Behavior of Tutoring Systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.

[78] Luis von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon University, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, December 2005.

[79] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, April 2004. ACM Press.

[80] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1947.

[81] E. Walker, K.R. Koedinger, B.M. McLaren, and N. Rummel. Cognitive Tutors as Research Platforms: Extending an Established Tutoring System for Collaborative and Metacognitive Experimentation. *8th international conference on intelligent tutoring systems, Jhongli, Taiwan*, pages 26–30, 2006.

[82] E. Walker, N. Rummel, B.M. McLaren, and K.R. Koedinger. The student becomes the master: Integrating peer tutoring with cognitive tutoring. In *Proceedings of the Conference on Computer-Supported Collaborative Learning*, 2007.

[83] Gerhard Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1997.

[84] Mark Wilson and R. Darrell Bock. Spellability: A linearly ordered content domain. *American Educational Research Journal*, 22(2):297–307, Summer 1985.

[85] R. L. Winkler. Scoring rules and the evaluation of probabilities. *Test*, 5(1):1–60, 1996.

[86] W.K. Wong, T.W. Chan, C.Y. Chou, J.S. Heh, and S.H. Tung. Reciprocal tutoring using cognitive tools. *Journal of Computer Assisted Learning*, 19:416–428, 2003.

[87] F.Y. Yu, L.J. Chang, Y.H. Liu, and T.W. Chan. Learning preferences towards computerised competitive modes. *Journal of Computer Assisted Learning*, 18:341–350, 2002.