# COMPS Computer Mediated Problem Solving: A First Look

**Melissa A. Desjarlais**
Valparaiso University
melissa.desjarlais@valpo.edu

**Jung Hee Kim**
North Carolina A&T
jungkim@ncat.edu

**Michael Glass**
Valparaiso University
michael.glass@valpo.edu

## Abstract

COMPS is a web-delivered computer-mediated problem solving environment designed for supporting instructional activities in mathematics. It is being developed as a platform for student collaborative exploratory learning using problem-specific affordances. COMPS will support computer-aided monitoring and assessment of these dialogues. In this paper we report on the first use of COMPS in the classroom, supporting an exercise in quantitative problem-solving. We have identified a number of categories of dialogue contribution that will be useful for monitoring and assessing the dialogue and built classifiers for recognizing these contributions. Regarding the usability of the interface for problem-solving exercises, the primary unexpected behavior is an increase (compared to in-person exercises) in off-task activity and concomitant decrease in shared construction of the answer. Its first large deployment will be for Math 110, a quantitative literacy class at Valparaiso University.

## Introduction

COMPS is a web-delivered computer-mediated problem solving environment designed for supporting instructional activities in mathematics.

In its initial classroom use COMPS supports groups of students engaging a particular exercise in quantitative literacy: figuring out a winning strategy, should one exist, for a Nim-like game. It has problem-related affordances for the students to manipulate, shows the instructor the conversations in real time, permits the instructor to intervene, and records all events for analysis. The intelligent part of COMPS, which has not been deployed in classroom use, has the computer itself participate in the supervisory task: monitoring the conversation status for bits of knowledge and other markers of progress or lack of progress and displaying its findings to the supervising instructor.

COMPS gives us a platform for deploying AI techniques in mathematics dialogues. Immediate applications include:

- *Exploratory learning*. COMPS is an environment with affordances for computer-supported collaborative exploratory-learning dialogues. Plug-in modules provide problem specific aids and affordances. The Poison game we report on here comes with a visualization of the game state and buttons for playing.
- *Computer-monitored dialogues*. COMPS has provisions for an instructor to oversee and intervene in the student conversations. In the style of, e.g. Argunaut [De Groot et al. 2007], COMPS will provide a status screen for the instructor, showing what knowledge the students have discovered in their inquiry learning as well as measures of affective state (e.g. are they on-task or frustrated) and other measures of progress. Experiments toward computer-generated status are described in this paper.
- *Assessment reports*. Using similar techniques as for monitoring, COMPS will provide the instructor with assessment reports of the conversations. This will permit the instructor to have the students engage in the exercises out of class, on their own time.
- *Observation and data collection*. COMPS collects transcripts and data that will be useful both in understanding the student problem-solving behaviors and in producing better computer understanding of COMPS conversations.

In this paper we report on the interaction model of COMPS, the educational context for its initial deployment, results from first use in a classroom setting, and first results toward having it monitor the progress the student conversation.

## The COMPS Model

The common threads to COMPS applications are a) dialogue, b) solving problems, and c) third parties. It is intended to facilitate and capture the kinds of interactions that would occur in mathematics problem-solving conversations. We have a simplified keyboard-chat communication channel instead of in-person face-to-face and voice communication. This permits us to readily log all interaction, more importantly it facilitates having the computer understand, monitor, assess, and potentially intervene in the dialogue. Because the problem domain is mathematics COMPS includes facilities for interpreting and rendering “ASCII Math,” expressions typed in-line using ordinary keyboard characters [MathForum 2012a].

COMPS conversations can be tutorial or they can be peer-to-peer explorations. Our view of how to support interactions is informed by the tutorial problem-solving dialogue studies of [Fox 1993] and the Virtual Math Team problem-solving dialogue studies of [Stahl 2009]. Wooz, the immediate predecessor to COMPS, has been used for recording and facilitating tutorial dialogues in algebra and differential equations, experiments in structured tutorial interactions, and exploratory learning with differential equations visualization applets [Kim and Glass 2004][Patel et al. 2003] [Glass et al. 2007].

The other element of COMPS conversations is possible third parties: teachers figuratively looking over the shoulders of the students as they work, computers also looking over the shoulders, teachers and computers intervening in the conversation, reports generated afterward with assessments of the student learning sessions, and analyses of the transcripts of interactions.

The common elements of COMPS applications are thus:

- *Interactivity*. Just as in in-person interactions, participants can behave asynchronously: interrupting and chatting over each other. Participants can see the other participants' keystrokes in real time, they do not need to take turns or wait for the other person to press *enter*. One use for this is documented by Fox who found tutors using *transition relevance points* [Sacks et al. 1974]. These are places within a dialogue turn where the other party is licensed to take over. For example, the tutor can provide scaffolding by starting to say an answer. Using prosodic cues (rising voice, stretched vowels), the tutor provides the student opportunities to take over the dialogue and complete the thought.
- *A problem window*. The problem is configurable, but generally there are areas of the screen window that keep the problem statement and elements of the solution in view without scrolling them off the screen. These items are assumed to be within the dialogue focus of all participants at all times, the objects of team cognition (Stahl) and shared construction (Fox).
- *A central server*. The server routes interaction traffic between the participants and optional third parties to the conversation (both human and machine), and records all interactions in log files.

Figure 1 at the end of this paper illustrates COMPS at work.

COMPS runs as a Flash application within a web browser, the server is a Java program. The application is configurable: plug-in modules written in Flash provide custom environments tailored for particular mathematics problems and learning modalities.

COMPS is similar in spirit to the Virtual Math Teams (VMT) chat interface [MathForum 2012b]. The VMT interface supports a generalized graphical whiteboard instead of having specialized interfaces for particular exercises. However many of the exercises that COMPS is intended to support are currently executed in class with manipulatives. For example the Poison game described in this report uses piles of tiles. It was incumbent on us to have COMPS provide software affordances that mimic the manipulatives.

## Math 110 Background

A goal of this project is to introduce COMPS computer-mediation to the group collaborative exercises in the Valparaiso University (VU) Math 110 class. Math 110 delivers the quantitative literacy skills expected of an educated adult [Gillman 2006] along with the mathematics skills expected in quantitative general education classes in a liberal arts curriculum. It achieves this by using modern pedagogical techniques and a selection of topics and problems that are quite different from, more motivating than, and we hope more successful than the typical bridge or college algebra class.

It is the style of instruction that matches Math 110 to COMPS, viz:

- Problems are explored by experimentation, using manipulatives and written instructions.
- Four person groups collaborate on the in-class explorations, with students adopting special assigned roles in the collaborative process.
- During the class period the instructor observes the group interactions and offers suggestions or guiding questions, as needed.

These are aligned with the three threads of COMPS: solving problems, dialogue, and third parties. During a semester, students solve twenty in-class problems. An emphasis is placed on problem-solving strategies.

Math 110 in its current form has been the established bridge class in the VU curriculum for 15 years. Students enrolled in Math 110 performed poorly on the mathematics placement exam and must successfully complete the course before they can enroll in quantitatively-based general education courses. Data show that completing Math 110 has a positive effect on retention and success at the university [Gillman 2006].

Math 110 differs from simply repeating high school algebra not only in teaching style but also in content. There are five topical themes: Pattern Recognition, Proportional Reasoning, Fairness, Graphs and Decision Science, and Organizing Information. Together these themes provide a background in logical reasoning, quantitative skills, and critical thinking.

Writing skills are exercised by requiring students to write up each problem in a narrative format. Each written solution includes the statement of the problem in the student's own words, the solution of the problem, and an explanation of the solution. Often this entails a description of the experimental activities and results. The students are assessed on the written aspect of the solution in addition to the mathematical aspect.

## Poison Exercise

An example of a Math 110 collaborative exercise—the first we have implemented in COMPS—is the Poison problem. The prompt is shown in Figure 2 at the end of this paper. Poison is a Nim-like two-person game. Starting from a pile of tiles, each person removes one or two tiles per turn. The last tile is "poisoned," the person who removes the last tile loses. The question before the students is to figure out how to play perfectly, to find an algorithm for either person A or person B to force a win. In a classroom setting the manipulative for this exploratory learning exercise in pattern

| | |
|---|---|
| A | well everytime ive had 4, or 7 i lose. |
| C | huh? |
| A | Oh wait, that's every round >:( |
| C | i dont think it matters |
| B | hahaha |
| | (playing game) |
| B | lets do 23 again and ill pick a 1 to start instead of a 2? |
| A | FINE |
| | ⋮ |
| D | i just tried to avoid 7 and still got stuck with 4 |

**Figure 3:** Dialogue from Poison Exercise Using COMPS

recognition is a box of tiles. Students also have pencil and paper.

For purposes of moving this exercise to the computer-mediated environment, we wrote a COMPS module that simulates the manipulatives: the pile of tiles. There are buttons for each of the two teams to remove one or two tiles. There is an option to arrange the tiles into small groups, a useful way to visualize the game and its solution. Students sometimes discover this method while playing with the tiles on the table-top. There is an option to restart the game with an arbitrary number of tiles. Students often find that they can better analyze the game if they consider a simpler problem, with only a few tiles. Finally, there is a record of the moves played, since in the face-to-face regime students typically write down the sequences of moves for study.

The current status of this COMPS plug-in is that students can play Poison, the teacher can monitor all the ongoing conversations in the computer lab, and the teacher can intervene. The computer is not yet monitoring the conversation.

## First Usage

### Setup

In November 2011 students in an elementary education mathematics course used the COMPS version of the Poison exercise. These were not Math 110 students, but education students who would normally engage in quantitative literacy classroom exercises as part of both learning the mathematics and experiencing how it is taught.

Twenty-five students were arranged in six groups in a computer lab so that group members were not near each other and verbal conversations were discouraged. The students were accustomed to working in groups sitting around a table. Keyboard chat was a new element. Each student was given a copy of the problem. The instructor logged in as a member of each group so that she could monitor and contribute to the conversations. A sample from a conversation is shown Figure 3. The session ran for approximately 40 minutes, at which time the students stopped where they were and gathered together offline to share notes for their written reports.

| | |
|---|---|
| A | How? |
| //D | If you take 2, then whatever you do on the next turn, you can do the opposite to leave 1. |
| B | If you take 1 or 2, then you can take 1 or 2 to counter balance that// |
| A | OK |
| C | OK |
| //C | So if I take 2, whatever they do ... |
| B | So basically if the other team ends up 4 left, then you can win. // |
| D | Yes |
| B | And that's if the other team ends up with 4 left |
| B | OK |
| A | We could maybe abbreviate opponent as OPP or something. Whatever, you might be writing a lot. |
| B | So yeah. um |
| | (*sounds of mumbling*) |
| C | Ok. Um |
| B | Oh boy |
| A | We don't need grammar. |
| B | Um so, if they 4 left you can win have how can you get it so that .. |
| D | If you have 5 or 6 on your turn, you can either take 1 or two to get it to that situation. |
| B | Ok you you want to get to 4, that's kind of a stable point where you can force them |

**Figure 4:** In-Person Poison Dialogue

### Observations

Both from experience observing Poison exercises, and from prior audiotaped sessions, differences between COMPS-mediated and in-person versions of Poison were evident.

- The COMPS students spent considerable time off-task, chatting about things not related to the problem. From the start, when students were logging in and greeting each other, it took some time for them to focus on the problem. Off-task conversation was almost negligible in our audio tapes, and not extensively observed in the classroom before the problem is solved.
- The COMPS groups spent much time playing the game for entertainment value, without advancing toward the goal of deducing whether a winning strategy existed.
- In the COMPS environment there was more team rivalry between the two teams within a group. There was even an instance where a student was reluctant to share the winning strategy with the rest of the group.

A consequence of all three of these behaviors is that incidences of shared construction of the winning strategy are less often observed in the COMPS transcripts, compared to their transcribed verbal ones. Figure 4 (in-person) and Figure 3 (computer-mediated) illustrate the typical difference. The in-person group engages in long exchanges where group cognition is evident. In the computer-mediated group the students rarely engage with each other for more than several turns at a stretch.

### The student experience

Students were surveyed the next day in class. There were 8 Likert questions (strongly-disagree to strongly-agree) and 6 short-answer questions. The students told us the following.

- Using the computer seemed easy: 19 of the 25 students either agreed or strongly agreed.
- Students were divided over whether it was easier to play Poison on a computer than with tiles on a table.
- Eleven students were neutral with regard to whether it was easier to find a winning strategy for Poison on a computer than with tiles on a table, while 10 students either agreed or strongly agreed that the computer was easier.

  This finding stands in contrast with our observation that the computer-mediated groups were less successful in finding a winning strategy.
- Responding to open-ended questions, students enjoyed the chat option in COMPS and the fact that the activity was different from other class activities.
- On the other hand, when comparing using COMPS to solving problems face-to-face around a table, the students commented that it took time to type their ideas (which were sometimes difficult to put into words) and they could not show things to the others.

  One student did comment that the chat environment made the student try to solve the problem individually rather than sharing the solution right away among the group members.
- Aspects of the Poison module were troublesome. Students were confused about the L/R buttons (they were for the two teams), they would have preferred images of tiles to the `@` symbol, and they found keeping up with the conversation difficult at times.

  This was congruent with our own observation of students using the interface. Images of tiles, and perhaps even a way to drag them with a mouse cursor, would be a better model for the manipulatives than the simple row of `@` symbols and buttons. It took students a while to learn to use the interface in this respect.
- The students would have liked to have a way to have a private chat between members of a team so that the other team could not see their conversation.

Other observations of student use of the interface:

- The physical tiles are limited to 20, but the computer placed no limit on virtual tiles. Combined with the Poison game's evident play value, this resulted in some COMPS groups playing longer games with more tiles than the physical-tiles groups do. Such games did not contribute to understanding.
- In person student groups picked and maintained teams a bit more readily. We think COMPS should allow students to pick a team, and have the software display the current team rosters.
- We observed students using the full-duplex chat communication constantly. They often do not take turns, and they react to the other students' developing turns as they are typed.

## Studies in Computer Monitoring

The first COMPS application of intelligence is to figuratively look over the shoulder of the students as they work, then display a real-time summary for the instructor. We have initially approached this task by writing shallow text classifiers. The work in this section is described in an unpublished report [Dion et al. 2011].

### Background

We created a preliminary set of categories and classifiers based on two sources of language data

- Tape-recorded dialogues of upper-class students working the poison exercise. Figure 4 shows an extract of recorded verbal interaction.
- Written reports of the Poison problem that Math 110 students provided in earlier semesters. These reports exhibit many of the mathematical realizations that student exhibit while solving the Poison problem, but none of the dialogue or problem-construction phenomena.

This work was completed before the initial collection of COMPS-supported Poison dialogues, so does not include the COMPS data.

For the COMPS Math 110 project we are concentrating first on identifying epistemic knowledge and social co-construction phenomena. This is congruent with the results of a study of the criteria that teachers use for assessing student collaborative efforts [Gweon et al. 2011]. We categorized the dialogue data according to the following types of phenomena we deemed useful for real-time assessment along these axes:

- Bits of knowledge: domain-specific realizations that are either needed or characteristically occur during the path toward solving the problem.
- Varieties of student activities that were on-task but not part of the cognitive work of constructing the solution: e.g. picking sides, clarifying rules, playing the game.
- Student utterances related to constructing a solution: e.g. making observations, hypothesizing, wrong statements.
- Off-task statements, filler.

Altogether we annotated the student utterances with 19 categories, shown in Table 1. In this study, individual dialogue turns or sentences were assigned to one of these categories.

### Experiment in machine classification

For our classifiers we chose two numerical methods: non-negative matrix factorization (NMF) and singular value decomposition (SVD). SVD is the most common numerical technique used in latent semantic analysis (LSA). Both of these methods rely on factoring a word-document co-occurrence matrix to build a semantic space: a set of dimensionality-reduced vectors. The training set for these experiments—the text used for building semantic spaces—was 435 sentences from the written corpus. The test sets

Table 1: Dialogue Categories from Poison Conversations

| | **Dialogue Category** |
|---|---|
| 1 | 4 tiles is important |
| 2 | 2 and 3 are good tiles |
| 3 | You want to leave your opponent with 19 tiles |
| 4 | Going first gives you control of the game |
| 5 | You want to take 1 tile on your first move |
| 6 | 1, 4, 7, 10, 13, 16, 19 are the poison numbers |
| 7 | "Opposite" strategy |
| 8 | "3 pattern" |
| 9 | Wrong statements |
| 10 | Exploring |
| 11 | Playing the game |
| 13 | Making an observation |
| 14 | Clarifying observations |
| 15 | Clarifying rules |
| 16 | Exploring further versions of the game |
| 17 | Hypothesizing |
| 18 | There is a winning strategy |
| 19 | Filler |

were taken from approximately 100 sentences from the written corpus and 500 spoken dialogue turns. All our semantic spaces had 20 dimensions. Our feature sets included unigrams (individual words) and bigrams.

We report here on three computer-tagging methods: SVD, NMF-s, and NMF-u.

The **SVD** and **NMF-s** methods are supervised. They match test sentences to manually accumulated bundles of exemplar sentences. This technique is much the same as the latent semantic analysis algorithm used successfully by Auto-Tutor [Graesser et al. 2007].

In the NMF-s method the vector for a test sentence was built by solving a set of linear equations in 20 unknowns, which effectively computed what the vector for the test sentence would have been had that sentence been a part of the training set. We believe that this technique for using non-negative matrix factorization to build text classifiers is novel.

The **NMF-u** method is unsupervised. The reduced dimensions of the factored matrices are assumed to correspond directly to semantic dimensions within the data. This approach was described by [Segaran 2007] for classifiying blog posts. Our training documents (sentences) were sorted according to a) their manually-assigned category and b) which of the 20 dimensions in the NMF vector representation of the document had the largest value. The dimensions were then manually associated with individual tags, if possible.

## Results

Table 2 summarizes the classification success rates of the two supervised methods, using unigram, bigram, and combined uni- and bi-gram feature spaces. We report the percentage of sentences that were correctly tagged from $n = 113$ test sentences. Test sentences represented all categories. Overall classification accuracy varied from 45% to 55%. Some categories occurred very infrequently in both the training and test corpora, resulting in very low success rates. Thus we also report the percent correct among the most common three categories in the test corpus: numbers 6, 11, and 15 in Table 1. Together these represented $n = 59$, more than half the test sentences.

A $\chi^2$ test on tagging sentences in the top three categories shows that the computer tagging success rates are indeed not due to random chance. All values are significant at the $p < 0.05$ level and some at the $p < 0.01$ level. We found no consistent advantage to using unigrams, bigrams, or both together. In this our result is similar to [Rosé et al. 2008], where differences among these conditions are slight. That study of classifiers for collaborative learning dialogues evaluated its results using $\kappa$ interrater reliability between human and computer annotaters. We have not computed $\kappa$, as the number of categories is large and the number of test sentences is small, rendering the statistic not very meaningful [Di Eugenio and Glass 2004].

In the NMF-u method many dimensions did not correlate with any tag. It was thus not capable of categorizing a test sentence into all the possible categories, leaving most of the categories unrecognized. Table 3 summarizes the most prominent categories that the NMF-u method found. For some of the most attested categories NMF-u was successful at correctly tagging the sentences in those categories, at the cost of a high rate of false positives. It had high recall but the precision was startlingly low.

## Data Collection for Analysis

One of the benefits of COMPS is the ability to gather data on students, their interactions, and the exercise that they engage in.

An advantage of recording group problem-solving is that ordinary obligations and discourse pragmatics dictate that the participants signal when they achieve some understanding or some common ground. This means that not only are all the learnable knowledge components visible, but participants in the discussion should be making recognizable signs of whether the components are understood [Koschmann 2011]. In short, student thinking is forced out into the open in ways that an assessment test, a cognitive experiment, or a think-aloud protocol might never get at.

Our study of Poison collaborative dialogues [Dion et al. 2011] has already uncovered knowledge components that students realize and express *before* they arrive at a closed-form solution but are *not themselves* part of the solution. Examples are: 2 and 3 tiles force a win, 4 tiles is a simple completely-analyzable case. There is no good way besides observation to find out the ancillary realizations that students characteristically pass through as they explore the problem. And it is necessary to understand these ancillary realizations in order to assess the state of the knowledge-construction task.

## Conclusions and Future Work

COMPS is being developed with several uses in mind, viz: a platform for student collaborative exploratory learning us-

Table 2: Accuracy of Supervised Classifiers

| | % Correct | | | Top 3 Tags | | |
|---|---|---|---|---|---|---|
| | All Tags<br>$n = 113$ | Top 3 Tags<br>$n = 59$ | $\chi^2$<br>$p$ value | Tab 6<br>$n = 19$ | Tag 11<br>$n = 13$ | Tag 15<br>$n = 27$ |
| NMF-s Unigrams | 47% | 61% | .003 | 58% | 31% | 78% |
| NMF-s Bigrams | 45% | 58% | .027 | 37% | 38% | 81% |
| NMF-s Both | 48% | 64% | .024 | 52% | 54% | 78% |
| SVD Unigrams | 51% | 66% | .0002 | 52% | 86% | 85% |
| SVD Bigrams | 55% | 68% | .028 | 63% | 15% | 96% |
| SVD Both | 53% | 59% | .003 | 42% | 0% | 100% |

Table 3: Unsupervised NMF Classifier Results

| | Class | N | Correctly classified | False positives |
|---|---|---|---|---|
| Unigrams | #7 Opposite Strategy | 13 | 13 (100%) | 63 |
| | #6 Poison Numbers | 13 | 12 (92%) | 2 |
| Unigrams no-stopwords | #15 Clarifying Rules | 27 | 16 (59%) | 8 |
| | #1 Four Tiles Important | 9 | 5 (56%) | 15 |
| Bigrams | #7 Opposite Strategy | 13 | 11 (85%) | 19 |
| | #15 Clarifying Rules | 27 | 23 (85%) | 23 |
| | #6 Poison Numbers | 13 | 12 (92%) | 10 |
| | #1 Four Tiles Important | 9 | 5 (56%) | 15 |

ing problem-specific affordances, computer-aided monitoring and assessment of these dialogues, and recording dialogues for study. Its first large deployment will be for Math 110, a quantitative literacy class at VU.

First use with 25 students students exercising the Poison exercise in six teams shows that COMPS is quite usable. What seemed like a fairly straightforward translation of the Poison exercise manipulatives to software affordances will, however, benefit from updating and experimentation.

Analyzing dialogues collected before COMPS, we have identified a number of categories of dialogue contribution that will be useful in monitoring and assessing the dialogue. With regard to epistemic knowledge in the Poison problem domain, we have identified realizations that students pass through on the way toward building the final solution. These realizations may not appear in the final solution, but having students engage in dialogue and team cognition seems to successfully force the cognitive processes into the open.

We have classifiers based on latent semantic analysis and non-negative matrix factorization that can recognize a few of the most important of these epistemic categories in solving the Poison exercise. One of our classifiers relies on a somewhat novel method of using NMF. It entails discovering where a test sentence would be in the factor matrices by solving a system of linear equations. It performed about as well as LSA on our data set, but more testing would be needed. Our classifiers are trained on student written reports, we expect that accuracy will improve once we train them on student dialogue data.

Regarding the usability of the interface for problem-solving exercises, the primary unexpected behavior that we will address in future tests is the increase (compared to in-person exercises) in off-task activity and concomitant decrease in shared construction of the answer. Certain updates, such as making the interface more explanatory and reducing the maximum number of tiles, may reduce the evidently enhanced play value provided by the computer mediated environment. Also specifically addressing this goal we have two improvements on offer:

- Unlike earlier Wooz exercises, the Poison problem prompt was not on permanent display in a COMPS window. The students have it on paper. Possibly putting the problem on display will serve to keep the students more on-task. In short, we may be suffering the consequence of not following our own COMPS interaction model strictly enough.
- In Math 110 team members are assigned roles. For example one student is a moderator, one is a reflector, and so on. These are not represented in the COMPS interface. Possibly displaying which students are assigned to which role will foster more focused interactions.

We note that in addition to the epistemic tags, teachers have been found to evaluate student collaborative activities on a number of axes such as goal-setting, division of labor, and participation [Gweon et al. 2011] [Gweon et al. 2009]. Accordingly, we have been annotating our dialogues using the VMT threaded markup scheme [Strijbos 2009] which shows when a turn addresses previous turns and annotates the discourse relationship between them. Future work on the text classifiers needs to address these discourse relations. The VMT Chat interface [MathForum 2012b] permits users to explicitly link their dialogue utterances: a user can indicate that a particular dialogue turn responds to a different, earlier, turn, possibly uttered by somebody else. COMPS does not have this functionality, but it might be useful.

## Acknowledgments

This work could not have been done without our hardworking, patient, and resourceful students. Special thanks to Nicole Rutt, Lisa Dion, and Jeremy Jank from the 2011 VU mathematics REU who worked on analyzing and classifying Poison dialogues, Scott Ramsey from NC A&T who designed and implemented much of COMPS in Actionscript, and Bryan Lee who helped update the server.

This work is supported by the National Science Foundation REESE program under awards 0634049 to Valparaiso University and 0633953 to North Carolina A&T State University and the NSF REU program under award 0851721 to Valparaiso University. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

## References

R. De Groot, R. Drachman, R. Hever, B. Schwarz, U. Hoppe, A. Harrer, M. De Laat, R. Wegerif, B. M. McLaren, and B. Baurens. Computer supported moderation of e-discussions: the ARGUNAUT approach. In Clark Chinn, Gijsbert Erkens, and Sadhana Puntambekar, editors, *Mice, Minds, and Society: The Computer Supported Collaborative Learning (CSCL) Conference 2007*, pages 165–167. International Society of the Learning Sciences, 2007.

Barbara Di Eugenio and Michael Glass. The kappa statistic: A second look. *Computational Linguistics*, 32:95–101, 2004.

Lisa Dion, Jeremy Jank, and Nicole Rutt. Computer monitored problem solving dialogues. Technical report, Mathematics and CS Dept., Valparaiso University, July 29 2011. REU project.

Barbara Fox. *The Human Tutoring Dialogue Project*. Erlbaum, Hillsdale, NJ, 1993.

Rick Gillman. A case study of assessment practices in quantitative literacy. In *Current Practices in Quantitative Literacy*, MAA Notes 70, pages 165–169. Mathematical Association of America, 2006.

Michael Glass, Jung Hee Kim, Karen Allen Keene, and Kathy Cousins-Cooper. Towards Wooz-2: Supporting tutorial dialogue for conceptual understanding of differential equations. In *Eighteenth Midwest AI and Cognitive Science Conference (MAICS-2007), Chicago*, pages 105–110, 2007.

Art Graesser, Phanni Penumatsa, Matthew Ventura, Zhiqiang Cai, and Xiangen Hu. Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language. In Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch, editors, *Handbook of Latent Semantic Analysis*, pages 243–262. Lawrence Erlbaum, 2007.

Gahgene Gweon, Rohit Kumar, Soojin Jun, and Carolyn P. Rosé. Towards automatic assessment for project based learning groups. In *Proceedings of the 2009 conference on Artificial Intelligence in Education*, pages 349–356, Amsterdam, 2009. IOS Press.

Gahgene Gweon, Soojin Jun, Joonhwan Lee, Susan Finger, and Carolyn Penstein Rosé. A framework for assessment of student project groups on-line and off-line. In Sadhana Puntambekar, Gijsbert Erkens, and Cindy E. Hmelo-Silver, editors, *Analyzing Interactions in CSCL*, volume 12, part 3 of *Computer-Supported Collaborative Learning Series*, pages 293–317. Springer, 2011.

Jung Hee Kim and Michael Glass. Evaluating dialogue schemata with the wizard of oz computer-assisted algebra tutor. In James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu, editors, *Intelligent Tutoring Systems, 7th International Conference, Maceió, Brazil*, volume 3220 of *Lecture Notes in Computer Science*, pages 358–367. Springer, 2004.

Tim Koschmann. Understanding understanding in action. *Journal of Pragmatics*, 43, 2011.

MathForum. Math notation in email messages or web forms. Web help page from Math Forum: Virtual Math Teams project, 2012a. URL `http://mathforum.org/typesetting/email.html`.

MathForum. VMT software orientation. Web help page from Math Forum: Virtual Math Teams project, 2012b. URL `http://vmt.mathforum.org/vmt/help.html`.

Niraj Patel, Michael Glass, and Jung Hee Kim. Data collection applications for the NC A&T State University algebra tutoring dialogue (Wooz tutor) project. In Anca Ralescu, editor, *Fourteenth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-2003)*, pages 120–125, 2003.

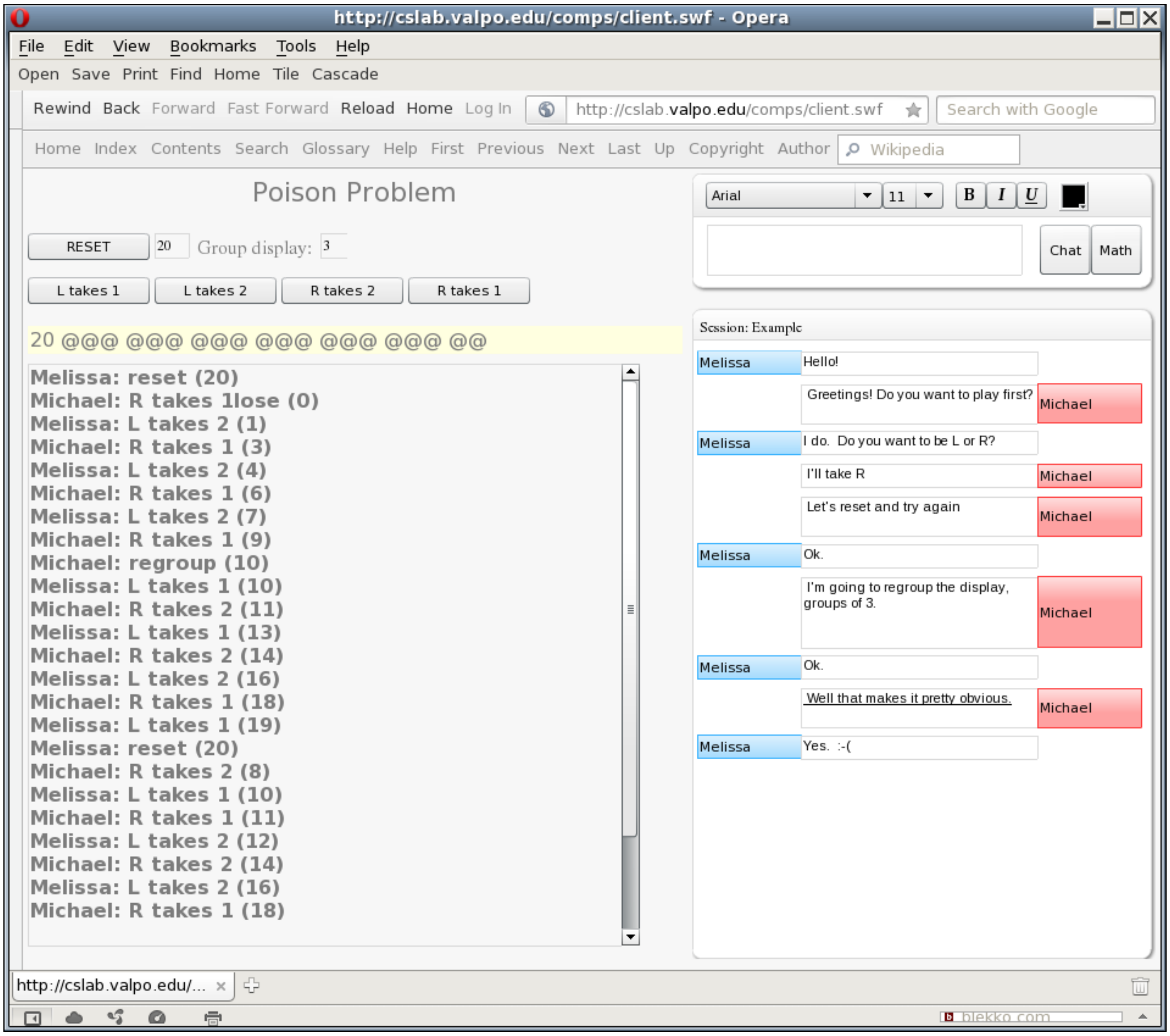
Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in CSCL. *Int. J. of Computer-Supported Collaborative Learning*, 3(3), 2008.

H. Sacks, E.A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, pages 696–735, 1974.

Toby Segaran. *Programming Collective Intelligence*. O'Reilly, 2007.

Gerry Stahl. *Studying Virtual Math Teams*. Springer, 2009.

Jan-Willem Strijbos. A multidimensional coding scheme for VMT. In Gerry Stahl, editor, *Studying Virtual Math Teams*, chapter 22. Springer, 2009.



20 3

20 @@@ @@@ @@@ @@@ @@@ @@@ @@

Melissa: reset (20)
Michael: R takes 1lose (0)
Melissa: L takes 2 (1)
Michael: R takes 1 (3)
Melissa: L takes 2 (4)
Michael: R takes 1 (6)
Melissa: L takes 2 (7)
Michael: R takes 1 (9)
Michael: regroup (10)
Melissa: L takes 1 (10)
Michael: R takes 2 (11)
Melissa: L takes 1 (13)
Michael: R takes 2 (14)
Melissa: L takes 2 (16)
Michael: R takes 1 (18)
Melissa: L takes 1 (19)
Melissa: reset (20)
Michael: R takes 2 (8)
Melissa: L takes 1 (10)
Michael: R takes 1 (11)
Melissa: L takes 2 (12)
Michael: R takes 2 (14)
Melissa: L takes 2 (16)
Michael: R takes 1 (18)

Melissa Hello!
Greetings! Do you want to play first? Michael
Melissa I do. Do you want to be L or R?
I'll take R Michael
Let's reset and try again Michael
Melissa Ok.
I'm going to regroup the display, groups of 3. Michael
Melissa Ok.
Well that makes it pretty obvious. Michael
Melissa Yes. :-(

**Figure 1:** COMPS with Poison problem.

The people in each group are to form two teams. One team will play against the other team in the group. To begin, place 20 tiles between the two teams. Here are the rules:

1. Decide which team will play first.
2. When it is your team's turn, your team is to remove 1 or 2 tiles from the pile.
3. The teams alternate taking turns.
4. The team that is forced to take the last tile – the poison tile – loses the game.

Play this game a number of times, alternating which team plays first. As you play these games, keep track of your moves/choices. Eventually, you want to be able to determine how your team should play to force the other team to lose. In order to make this determination, you will need to look for a pattern. In order to find a pattern, you will need data, and so you will need to decide how to collect and organize these data to see if a pattern will appear.

**Figure 2:** Poison Assignment