From Response Systems to Distributed Systems for Enhanced Collaborative Learning

Jeremy ROSCHELLE, Patricia SCHANK, John BRECHT
SRI International
333 Ravenswood Avenue, Menlo Park CA, USA

Deborah TATAR
Virginia Tech
Center for HCI
660 McBryde Hall, MC 106
Blacksburg, VA 24061

S. Raj CHAUDHURY
Norfolk State University,
700 Park Avenue
Norfolk, VA, USA

Abstract. Innovators need a common, reusable framework for implementing patterns of collaborative learning on wireless handheld devices. Our project explores the application of an architecture from distributed system research, "tuple spaces," to a wide variety of collaborative learning scenarios. We are testing this approach with documented scenarios from prior work published by our team and others. We are also designing new scenarios that leverage the architecture to engage learners optimally in cognitive, collaborative work. Our research explores the conceptual alignment, programming complexity, and classroom usability of this approach.

Keywords. Collaborative and group learning; Learning systems platforms and architectures; wireless and mobile technologies

Introduction

Collaborative learning has a well-established research base in education and social sciences [1, 2, 3, 4]. In classroom practice, coordination is fundamental to modern pedagogy. Groups of learners and their teachers routinely work in more complex configurations than in traditional lecture-based classes. They take roles, contribute ideas, critique each other's work, and together solve aspects of larger problems, all to good effect [5, 6, 7, 8]. Managed flow of information and control is essential to the structure of many of these successful educational activities [9]. With the advent of mobile, wirelessly connected technologies in education [10, 11], innovators have begun to design applications to coordinate students' work to produce better science, technology, engineering, and math (STEM) education outcomes. Yet there has been little work on formal computational foundations to support this core function of coordination [12].

Consider as an example, classroom response systems, which are an increasingly popular application in K-12 schools and universities across many subjects. These systems rapidly aggregate the input of all students anonymously and present a histogram that reveals the students' understandings [13]. While research suggests that these systems can have powerful benefits in classrooms [14]—for example, by enabling student groups to focus on improving their conceptual understanding [15]—this technology is limited to a very narrow range of representational types and information-sharing topologies; generally speaking, the main data type is a response to a multiple-choice question and the only topology is many-students-to-one-teacher. In contrast, researchers in the subfield of computer supported collaborative learning (CSCL) have documented how shared dynamic representations can support participation and discourse [16, 17, 18]. Researchers are calling for more powerful classroom network technology that can expand to richer representations and interaction topologies [19, 20, 21] and have assembled prototypes of advanced uses, such as participatory simulations [22, 23], structured group reasoning tools [24, 25], comparison of scientific data sets [26], and ways to harvest students' problem solving to highlight emergent mathematical and scientific structure [21]. Research has shown these prototypes to be educationally promising, but they have been hard to build with available technologies, and the resulting applications have been slow and unstable.

Robust and responsive architectural support for coordinating multiple learners is thus an increasingly important topic [27, 28, 29]. Despite growing awareness of how the detailed structure of learners' participation in activity affects learning [30, 31], there has been little progress directly linking the computational foundations of coordination (e.g., as found within distributed system research) with the educational fundamentals of participation.

1. Tuples Spaces for Collaborative Learning

1.1 Using Techniques from Distributed Computing

Computer scientists working in the area of distributed computing have established a foundational abstraction called "Tuple Spaces" as a means for coordinating the work of multiple computers. Tuple Spaces provide a coordination language [32] that elegantly addresses the problem of organizing distributed scientific computations. Tuple spaces have grown beyond their original scientific niche. It has been claimed that they provide the best available technique for coordinating the participation of multiple mobile agents in collective activities [33]. They have been used to solve both technical and social coordination among these agents [34, 35].

A tuple space coordinates the computations of different processes by allowing the processes to read and write structured data to a common space [36]. Tuples provide, in essence, an associative shared memory. The word "tuple" refers to the use of vectors of primitive data types (text, numbers), which constitute the primary data type. In the classic "Linda" model [32], processes coordinate by *writing* data in the form of tuples to represent the state of computation and *reading* data by providing a template to be matched. The read operation blocks such that if no match is immediately found, the request will wait until one is available and then return immediately. Tuple spaces provide one critical additional operation, *take*, which allows exactly one process to read a match and take the tuple out of the space simultaneously. This last operation allows processes to coordinate such that one process uniquely works on each task. The combination of blocking reads and takes allows efficient, just-in-time execution of parallel processes without excessive central management.

1.2 Students as Distributed Processors: A Provocative Analogy

Tuple spaces excel at optimizing the engagement of processors with different abilities (e.g speed, memory capacity, etc.) working towards a group goal. If we consider students, not their computers, to be the distributed processor, we can pose the question of whether, by analogy, a tuple space can be utilized to engage and coordinate students with different abilities and preferences (e.g to solve problems, communicate, critique, etc.) in working towards the goal of mastering classroom content.

Our project seeks to address this question. We conjecture that this architecture can provide new levels of cognitive and communicative engagement in classroom activities, yielding new levels of group performance. This socio-technical system can be conceived of as *cybernetic* in the tight coupling between "processors" giving and receiving information.

1.3 An Architectural Vision for Classrooms of Wireless Handhelds

Wireless handheld devices, connected in a network, along with a projected display space, present an affordable and attractive hardware platform for classroom learning. We are proposing tuple spaces as an alternative to peer-to-peer, client-server, or web-based services for collaborative learning. We expect that a simple, general framework for coordinating the work of multiple learners will ease rapid prototyping, enable new types of activities, and make these systems easier to support in the field.

2. Using a Tuple Spaces Architecture in Collaboration Tools

To test the feasibility of our vision, we plan to implement and test a variety of collaborative learning scenarios using Tuple Spaces. For all implementations, we are using the TSpaces server [36] and client programs written in Java. We have been first testing the tools on laptops and later will test them on handhelds, with applications that will inform us about the possibilities of both the technology and the pedagogy.

2.1 Scenarios

We began the project by collecting collaborative learning scenarios from the literature. The group identified 41 scenarios. Of these, a set of 13 were selected as high priority because they (a) were deemed most important for use in STEM classrooms, (b) had been tried in classrooms and worked well in the past, and (c) clearly leveraged technology (e.g, would not be easy to do with paper and pencil). With this set, we also strove to span a wide variety of patterns (e.g, working in pairs, small groups, or an entire class; different workflows such as a pipeline or very distributed) and highlight a variety of issues (e.g, sharing, exchanging, publishing information) that would exercise the technology and pedagogy space and stimulate the imaginations of developers. So far we have implemented five activities representing these scenarios. Below we describe three of these activities and their Tuple Space implementations.

2.2 Question Posing

The question-posing tool [37] is a twist on the conventional response system. In the conventional response system, the teacher asks a question and all students answer it. Using our tool, students determine what questions are important, either for the purposes of examining a new topic or for composing a set of review questions. The tool allows students to *submit* new questions to the question space, *browse* and rank other's submitted questions, and *cluster* related questions. *Submit* is a simple *write* operation, *browse* is a *read* of a tuple that fits the "question" template. *Rank* involves *taking* a question tuple and *writing* a new version of the tuple with the applied rank as part of its data. *Clustering* here involves the *writing* of "cluster" tuples that describe a relationship between one or more questions. Students can perform any of the tasks in any order they wish, and to whatever depth required. Thus, students more adept or interested in posing questions are free to do so as much as they wish, and students more interested in sorting and evaluating questions may do so, all without explicit central coordination. The result is a ranked set of important questions, with related items gathered in clusters.

2.3 A Novel Critical Reading Tool

This text markup tool provides an application that can display a body of text and allow selections of that text at some specific grain size (e.g., lines, paragraphs, words). For example, students are presented with a poem, and asked to highlight all of the near-rhymes in the text. Students select relevant lines of text one-by-one, by checking a box beside ech line that illustrates a near-rhyme. When they are finished, they submit (*write*) their selections to the space. They can then browse and view (*read*) the selections made by others, which appear highlighted in gray on their text display. Their own selections appear highlighted in yellow. Bar graph displays to the right of each line indicate the number of students who highlighted the particular line, and the font size of more popular line is also increased to indicate frequency of selection. In this way, students' markup is aggregated on each individual display, and discussion may then ensue about disagreement amongst the highlighted lines.

2.4. Draw My Molecule

The molecule drawing tool provides a game-like environment to help students become familiar with a multiple scientific representations for various molecules and compounds. This tool provides an example of integrating exisiting software––in this case, ChemSense [38]—with the TSpaces architecture. Using this tool, students first submit (*write*) names of molecules to the space. Once several molecule names have been generated, students grab (*take*) a random molecule name from the space. The tools asks them to draw the molecule in one of three chemical representations: molecular formula, electron dot, or ball and stick. Students use the ChemSense drawing tool to draw the molecule, and then submit (*write*) their representation back to the space. If a student is unable to generate the representation, she can give up on that molecule-representation pair to release (*write*) it back to the space, and grab a different random molecule-representation assignment. Once a student submits a particular representation (e.g., electron dot) for a particular molecule (e.g., water), no other students will get this particular pair to draw. As a result, after some minutes of this distributed drawing activity, the matrix of molecule-representation types becomes mostly or even fully complete. In a final activity, students

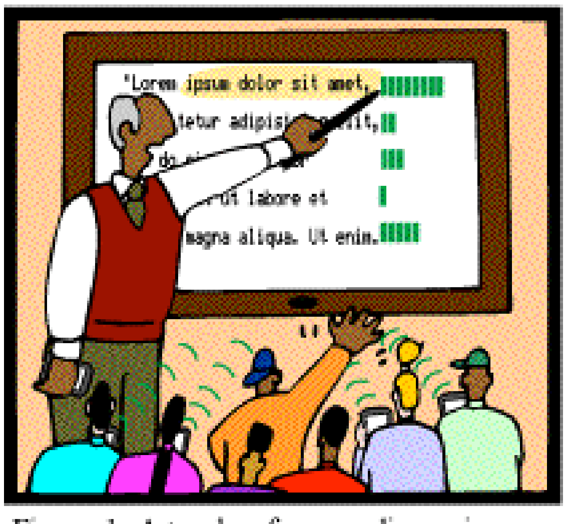

Figure 1: A teacher focuses discussion on the line of text that many students selected

use the tool to play a "slot machine" game [39] in which they get (read) three random representations from the space, and indicate, through checkboxes, which (if any) of the three representations are for the same molecule. If they indicate correctly, they "win" that slot round. Wins and losses are tallied for both individual and group scores.

3. Research and Evaluation

Our overall objective is to develop methods, knowledge, and tools that better align social needs and distributed computing techniques with respect to the opportunity to improve collaborative learning. Below we describe three key research questions and our current progress towards answering them.

3.1 Is there a sound alignment of the tuple space formalism to the patterns and principles of collaborative learning?

We are investigating this question by implementing a wide range of collaborative learning activities either previously described in the literature or of our own invention. So far, we have implemented five quite different activities and have found good alignment between the Tuple Space formalism and collaborative learning. We have found the general formalism to be fairly easy to use. The data management concepts are straightforward and primitive by design, and we haven't needed to wrap the primitive operations. One area of difficulty is reconceptualizing pre-existing patterns of connectivity, such as peer-to-peer interaction, in the distributed Tuple Spaces perspective. We plan to further explore this question by implementing a high-level collaboration framework, such as that specified by the Collaboration Modeling Language, CML [40]. CML is a general framework for describing workflow and interaction in a networked classroom environment, such as group formation and the playing of specific roles,.

3.2 Can a wide variety of programmers adequately understand how to build designs for learning using these techniques?

We are investigating this question by involving programmers from different settings and of different abilities in our work. So far, four different programmers have developed activities using the TSpaces environment. In most cases, implementing an initial prototype activity took about three days, including time spent on issues unrelated to TSpaces, such as user interface development (with Swing, a graphics library for Java) or content development. Database-like transactions were easily implemented using the transactions built into the TSpaces architecture. One advanced developer noted some difficulty giving up features built-in to more sophisticated databases, such as referential integrity and type protection. In the summer of 2005, we will host a 3-week International workshop in which students will code new collaborative learning tools using our framework. Further, in the fall of 2005, we will teach a course at Virginia Tech and study undergraduates learning and using the Tuple Spaces framework. These activities will further inform our understanding of the nature and development of expertise in programming using tuple spaces for social coordination.

3.3 Do tuple space servers work well in realistic classroom settings?

Evaluation of Tuple Spaces in a classroom is scheduled to start in September 2005. Three challenges we anticipate are server overload, basic wireless infrastructure issues and set-up/installation problems. For example, if a teacher instructs students to follow her in lockstep, and everyone in a class clicks on the same button at the same time, the server may overload. So far, our tests suggest that network traffic is fairly light, but we will need to track this. While our system does nothing to improve the wireless link layer, such improvements do not seem necessary. The system appears to tolerate network dropouts, partly because data packet transmissions are typically small and few with small retransmission cost. We are currently working on handling the set-up and installation problems that typically dog CSCW and CSCL systems, and which prove so difficult for teachers to handle.

4. Conclusion

Innovators need a common, reusable framework for implementing patterns of collaborative learning on wireless handheld devices. Our project explores the application of an established architecture from distributed systems research, "tuple spaces," to a wide variety of collaborative learning scenarios. Results to date indicate that this approach can both implement existing scenarios from the literature and enable new scenarios that may optimally engage learners in cognitive, collaborative, and cybernetic work. Our research is exploring the conceptual alignment, programming complexity, and classroom usability of this approach. We invite international partnerships in this research.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Number #0427783. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Koschman, T. (Ed.). (1996). *CSCL: Theory and practice of an emerging paradigm*. Mahwah, NJ: Lawrence Erlbaum Associates

[2] Koschman, T., Hall, R., & Miyake, N. (Eds.). (2002). *CSCL 2: Carrying forward the conversation*. Mahwah, NJ: Lawrence Erlbaum Associates.

[3] Wasson, B., Ludvigsen, S., & Hoppe, U. (Eds.). (2003). *Designing for change in networked computing environments*. Boston, MA: Kluwer.

[4] Webb, N. M., & Palincsar, A. S. (1996). Group processes in the classroom. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of educational psychology* (pp. 841-873). New York: Macmillan.

[5] Angelo, A. A., & Cross, C. K. (1993). *Classroom assessment techniques*. San Francisco: Jossey-Bass.

[6] Hake, R. R. (1998). Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics, 66*, 64-74.

[7] Hatano, G., & Inagaki, K. (1991). Sharing cognition through collective comprehension activity. In L. B. Resnick & J. M. Levine (Eds.), *Perspectives on socially shared cognition* (pp. 331-348). Washington, DC: American Psychological Association.

[8] Palincsar, A. S., & Herrenkohl, L. R. (1999). Designing collaborative contexts: Lessons from three research programs. In A. M. O'Donnell & A. King (Eds.),

Cognitive perspectives on peer learning (pp. 151-177). Mahwah, NJ: Lawrence Erlbaum Associates.

[9] Guribye, F., Andreassen, E. F., & Wasson, B. (2003). The organisation of interaction in distributed collaborative learning. In B. Wasson, S. Ludvigsen & U. Hoppe (Eds.), *Designing for change in networked learning environments* (pp. 385-394). Boston: Kluwer.

[10] Norris, C., & Soloway, E. (2003, April). *The viable alternative: handhelds*. Retrieved May 12, 2003, from http://www.aasa.org/publications/sa/2003_04/soloway.htm

[11] Soloway, E., Norris, C., Blumenfeld, P., Fishman, B., J., K., & Marx, R. (2001). Devices are ready-at-hand. *Communications of the ACM, 44*(6), 15-20.

[12] Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys, 26*(1), 87-119.

[13] Abrahamson, L. A. (1999). *Teaching with a classroom communications system: What it involves and why it works*. Retrieved February 20, 2004, from http://www.bedu.com/Publications/PueblaFinal2.html

[14] Roschelle, J., Penuel, W. R., & Abrahamson, L. A. (2004). The networked classroom. *Educational Leadership, 61*(5), 50-54.

[15] Crouch, C. H., & Mazur, E. (2001). Peer instruction: Ten years of experience and results. *The Physics Teacher, 69*(9), 970-977.

[16] Cobb, P., Yackel, E., & McClain, K. (Eds.). (2002). *Communicating and symbolizing in mathematics: Perspectives on discourse, tools, and instructional design*. Mahwah, NJ: Lawrence Erlbaum Associates.

[17] Roschelle, J., Penuel, W. R., & Abrahamson, L. A. (2004). The networked classroom. *Educational Leadership, 61*(5), 50-54.

[18] Suthers, D. D. (2001). Towards a systematic study of representational guidance for collaborative learning discourse. *Journal of Universal Computer Science, 7*(3), 254-277.

[19] Abrahamson, L. A., Davidian, A., & Lippai, A. (2000). *Wireless calculator networks: Where they came from, why they work, and where they're going.* Paper presented at the 13th Annual International Conference on Technology in Collegiate Mathematics, Atlanta, GA.

[20] Hegedus, S. J., & Kaput, J. (2001). *New activity structures exploiting wirelessly connected graphing calculators.* Paper presented at the 23rd Conference for the North American Chapter of the Psychology of Mathematics Education, Snowbird, UT.

[21] Stroup, W. M., Kaput, J., Ares, N., Wilensky, U., Hegedus, S. J., Roschelle, J., et al. (2002). *The nature and future of classroom connectivity: The dialectics of mathematics in the social space.* Paper presented at the Psychology and Mathematics Education North America conference, Athens, GA.

[22] Colella, V. (2000). Participatory simulations: Building collaborative understanding through immersive dynamic modeling. *Journal of the Learning Sciences, 9*(4), 471-500.

[23] Stroup, W. M. (1999). *Participatory simulations: Network-based design for systems learning in classrooms.* Paper presented at the Computer Supported Collaborative Learning Conference, Stanford, CA.

[24] Pinkwart, N., Hoppe, U., Bollen, L., & Fuhlrott, E. (2002). *Group-oriented modeling tools with hetrogeneous semantics.* Paper presented at the ITS 2002, San Sebastian, Spain & Biarritz, France.
[25] Suthers, D. D., & Hundhausen, C. D. (2003). An experimental study of the effects of representational guidance on collaborative learning processes. *Journal of the Learning Sciences, 12*(2), 183-219.
[26] Davis, S. (2003). Observations in classrooms using a network of handheld devices. *Journal of Computer Assisted Learning, 19*(3), 298-307.
[27] Chang, C.-Y., Sheu, J.-P., & Chan, T.-W. (2003). Concept and design of ad hoc and mobile classrooms. *Journal of Computer Assisted Learning, 19*(3), 336-346.
[28] Pinkwart, N., Hoppe, H. U., Milrad, M., & Perez, J. (2003). Architectures and educational scenarios for cooperative use of PDAs in wireless networks. *Journal of Computer Assisted Learning, 19*(3), 383-391.
[29] Roschelle, J. (2003). Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning, 19*(3), 260-272.
[30] Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. New York: Cambridge University Press.
[31] Rogoff, B., Baker-Sennett, J., Lacasa, P., & Goldsmith, D. (1995). Development Through Participation in Sociocultural Activity. In J. Goodnow, P. Miller & F. Kessel (Eds.), *Cultural practices as contexts for development* (pp. 45-65). San Francisco: Jossey-Bass.
[32] Gelernter, D., & Carriero, N. (1992). Coordination languages and their significance. *Communications of the ACM, 35*(2), 97-107.
[33] Cabri, G., Leonardi, L., & Zambonelli, F. (2002). Mobile agent coordination models for Internet Applications. *IEEE Computer, 33*(2), 82-89.
[34] Murphy, A. L., Picco, G. P., & Roman, G. C. (2003). *LIME: A coordination middleware supporting mobility of hosts and agents* (No. WUCSE-03-21). St. Louis: Washington University, Department of Computer Science.
[35] Nichols, J. R., Yechezkal, S. G., Paschetto, J., Rush, K. S., & Martin, C. (1999). How the internet helps build collaborative multimedia applications. *Communications of the ACM, 42*(1), 79-85.
[36] Lehman, T. J. (2001). Hitting the distributed computing sweet spot with TSpaces. *Computer Networks, 35*(4), 457-472.
[37] Penuel, W. & Yarnall, L. (2005) Designing handheld software to support classroom assessment: Analysis of conditions for teacher adoption, *Journal of Technology, Learning, and Assessment, 3*(5).
[38] Schank, P., & Kozma, R. (2002). Learning chemistry through the use of a representation-based knowledge building environment. *Journal of Computers in Mathematics and Science Teaching, 21*(3), 253-279.
[39] Vahey, P., Tatar, D. & Roschelle, J. (2004) Leveraging Handhelds to Increase Student Learning: Engaging Middle School Students with the Mathematics of Change. *Proceedings of the Sixth International Conference of the Learning Sciences* (pp. 553-560). NJ: Lawrence Erlbaum Associates.
[40] Brecht, J., Pea, R., & Chung, M. (2002). *CML – The ClassSync Modeling Language.* Paper presented at the Computer Supported Collaborative Learning Conference, Boulder, CO.