Interaction Synchronicity in Web-based Collaborative Learning Systems[*]

Ari Bader-Natal
Grockit, Inc.
United States
ari@grockit.com

Abstract: While many web-based learning systems connect students asynchronously, fewer systems focus on facilitating synchronous interactions among learners. Given the value of real-time communication – the social and motivational benefits of having a cohort of peers and the ability for a student to get immediate answers to pressing questions – it is perhaps surprising that more systems do not support interaction synchronicity. We suggest that this is due, in part, to a mismatch between the hypertext document-oriented nature of the web and the social activity-oriented nature of learning, and we explore how several systems address this discrepancy. We discuss Grockit, a web-based learning environment that we designed to support both synchronous and asynchronous interactions, and share lessons learned from grappling with the choices enabled by this flexibility: Which interactions should to be synchronous? Which should be asynchronous? Which should be a mix? What should that mix be?

Introduction

Web applications designed to support learning are becoming increasingly popular, offering a compelling set of advantages over their desktop-based counterparts. Most web applications – including those designed specifically for learning – are platform-independent, accessible from any network-connected computer, and require no additional software to be downloaded or installed. The "Web 2.0" moniker, popularized by Tim O'Reilly, describes an additional set of application characteristics, including the trend towards developing "rich" desktop-like user interactions, the opportunity afforded by easy software deployment to make the design-feedback loop into an ongoing, near-continuous process, and the practice of leveraging collective user participation to improve the individual user experience (O'Reilly 2005). Downes describes the new trend in co-opting these social web applications for educational purposes (and the new wave of e-learning software built on these design principles) as "e-learning 2.0" (Downes 2005). But as Dohn has recently argued, fundamental differences between the goals of educational experiences and the goals of Web 2.0 participation may complicate this partnership (Dohn 2009). In this work, we explore how incorporating interaction synchronicity – a core component of face-to-face educational experiences – into web-based participatory learning environments can add complexity to the architecture but flexibility and power to the design.

We attribute the relative scarcity of support for synchronous interactions in "e-learning 2.0" applications to added complexity stemming from the differences between the social activity-oriented nature of classroom interactions among learners and the hypertext document-oriented nature of the web. We present a set of responses to this issue, drawn from currently deployed systems, ranging from (a) ignoring synchronous interactions, to (b) designing around the synchronicity limitations of the web, to (c) engineering a solution enabling synchronicity on the web. We describe the third response in greater detail, using Grockit[1] – a web application that we launched in 2008 to provide live collaborative learning activities in a virtual study group format – as an example. Notably, after developing a system able to support various combinations of synchronous and asynchronous interactions among learners, we found ourselves faced with a new set of questions: Which interactions in our system should be synchronous? We address some of these questions, and conclude by sharing some of the principles that we developed to guide our thinking about interaction synchronicity in learning systems.

[*] A version of this paper appeared in Theo Bastiaens, Jon Dron, and Cindy Xin, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009*, pages 1121-1129, Vancouver, Canada, October 2009. AACE. http://editlib.org/p/32603

[1] http://grockit.com

Why support synchronous interactions among learners?

A recent meta-analysis published by the U.S. Department of Education consolidates research findings on the relative effectiveness of online versus face-to-face learning environments, and reports that while a lack of sufficient work has been done with K-12 students, the available studies (in higher education, medical training, corporate and military education, among others) suggest that students in the online or "blended" learning groups exhibited higher learning rates than those in the traditional classroom environments (Means et al. 2009). While this is a promising sign for the e-learning field, this study, like earlier meta-analyses in distance education (Zhao et al. 2005, Bernard et al. 2004), identifies a variety of factors that influence the relative effectiveness of online/blended/distance learning as compared to traditional face-to-face classroom learning. One such factor that Zhao et al. determined to be a significant predictor of the difference in effectiveness was the type of interactions among participants. Studies on distance learning programs that incorporated *both* synchronous and asynchronous interactions found that the distance learning was significantly better than face-to-face learning (d=0.22, p<0.001), while neither of the mean effect sizes of studies involving only synchronous *or* asynchronous interactions were significant at the $\alpha = 0.05$ level.

Bernard et al. directly compared the effectiveness of synchronous and asynchronous distance education (DE) programs to that of face-to-face environments. They found that while the mean effect size of neither was higher than face-to-face classroom groups, the mean effect size for asynchronous DE was higher than that of synchronous DE.[2] One interesting observation noted in this work was that the retention rate of learners in the synchronous DE groups was higher than that of those in the asynchronous DE groups, suggesting that synchronicity may offer social or motivational benefits. The authors cite group affiliation, social pressure, and the need for spontaneous guidance and feedback among the possible explanations for the lower dropout rate (Bernard et al. 2004). These influences are particularly important in situations for which student participation is elective. We cited many of the same reasons for building Grockit's web-based learning system on synchronous collaboration, social networking, and game-like dynamics (Bader-Natal 2009). Means et al. notes that the increased effectiveness of online learning over classroom learning may likely be due to the additional time that students spent in the online learning environments, and that rather than statistically controlling for this extra time investment, we can view this as a new opportunity: Additional learning gains may be achieved simply by creating a more engaging learning environment in which students choose to spend more time, and real-time social interactions seem to provide a promising path to achieving this.

How do existing learning systems address interaction synchronicity?

One challenge in supporting synchronous interactions among participants in a web-based learning system is that the underlying application-level protocol, HTTP (Fielding et al. 1999), is not naturally conducive to supporting a peer-to-peer social interaction pattern. HTTP exchanges traditionally consists of a client (such as a web browser) requesting a resource from a web server, the server responding by (generating and) returning that resource, and the client receiving (and visually rendering) the response. Social interaction-based learning exchanges do not follow a strict information transmission structure. In classroom environments, for example, teachers may explain a concept, they may elicit questions on that topic, they may try to identify and correct student misunderstandings, or they may organize students into small groups for team-based activities. Students, too, may initiate interactions, by raising questions during class or seeking additional help after class. In each case, these interactions are far less structured than the basic client-server request-response model.

---

[2] It is worth noting that the definition of synchronicity used by Bernard et al. does not distinguish between what Means et al. identifies as the learner *experience* dimension, including transmission-based expository instruction, inquiry-based active learning, and collaboration-based interactive learning (Means et al., 2009). Of these, the mean effect size for studies with expository instruction (+0.36) and that of interactive learning (+0.28) were significantly positive, while that of active learning (+0.15) was not. Furthermore, as Means et al. notes, synchronous DE traditionally involved a (centralized) instructor's classroom connected to several distance classrooms, whereas synchronous interactions in web-based learning generally involves a (distributed) network of learners.

Given this discrepancy, designing software to support synchronous discussion-like interactions poses a challenge. We identify three classes of responses to this challenge. System designers may (a) opt to focus on supporting asynchronous interactions that are better suited to the medium, (b) focus on support synchronous interactions, but avoid the HTTP protocol, or (c) engineer a solution to support synchronous interactions over the web. We briefly describe a few examples of each of these responses to the challenge.

Web-based systems that do not support synchronous interactions

Many web-based learning systems provide no affordance for synchronous interactions among learners. This may not be a response, per se, to the challenge inherent in doing so. Instead, these systems may be based on interactions centered on some digital content. The recent rise of open educational resources such as those stemming from the open courseware initiatives at various universities[3] is an example of a one-way asynchronous interaction between a teacher and a student. Course materials such as readings, assignments, lecture recordings, tests, and even full textbooks may all be freely available on the site. The primary basis for learning remains one of knowledge *transmission*, focused on providing or delivering learning objects, modules, or content to the student.

Some asynchronous systems provide for learner interactivity. Blogs and wikis are often used in educational settings. The authoring and commenting on blog posts and the engagement in wiki document authorship[4] are two examples of web-based asynchronous student-to-student interactions.

Non-web-based systems that do support synchronous interactions

A wide variety of synchronous communications technologies exist that are based on application layer protocols other than HTTP, and several of these have been adopted by the educational community in support of learning. Peer-to-peer video conferencing systems, instant text messaging systems, and immersive virtual worlds are a few examples of technologies that have been adapted for the classroom or to connect students and teachers (including student-student, student-tutor, and teacher-teacher). It is worth noting that these technologies were not developed with the intent of support learning processes, but rather were adopted by the educational community as tools to support synchronicity.

While these systems offer a high degree of interaction synchronicity each requires a separate application to be downloaded, installed, and maintained.[5] Doing this in a school environment often requires technical assistance and/or institutional approval, creating an additional barrier to adoption. This barrier is one of the motivations for the third solution class.

Web-based systems that do support synchronous interactions

Various solutions have been developed to work around the page-at-a-time structure of the web in order to support live peer-to-peer interactions, drawing on a variety of techniques described using the umbrella terms “Ajax” (Garrett 2005) and “Comet” (Russell 2006). Ajax techniques seek to let a user action modify some part of a web page, without reloading the page in its entirety. Comet techniques seek to let the server pro-actively push such changes to the client, without first receiving a user-initiated request. Using these techniques, the stateless HTTP client-server protocol can be coerced into supporting persistent client-client interactions, via a shared server. In doing so, synchronous communications (such as those similar to instant messaging applications) can be supported within any modern web browser, without requiring a separate desktop application. These techniques provide a technical solution that enables various modes of real-time interactions in a web-based learning system, which may include student-teacher, student-student, and student-group interactions, among others. A growing list of web

---

[3] The OpenCourseWare Consortium (http://www.ocwconsortium.org) includes several university initiatives, such as MIT’s OpenCourseWare project (http://ocw.mit.edu).
[4] When adopted by a learning community, wikis can provide a focal point for learner interactions and consensus, ideally requiring students to negotiate meaning and arrive at consensus.
[5] Additionally, these networked applications often require modifications to firewall settings.

applications fall into this category, including DimDim, WiZiQ, EduFire, and Grockit[6], among others. In each of these applications, some combination of synchronous and asynchronous learning interactions is supported. In the following section, we share some of our own experiences incorporating such interactions into Grockit's web application.

What modes of study does Grockit support? For whom?

Grockit, first launched in September 2008, offers a web-based live collaborative learning platform through which students can learn primarily through working practice problems, engaging in synchronous interactions with peers and with instructors, and by reading and asynchronously discussing expert-authored explanations. While the platform is currently being piloted with several schools and districts, most students use the system on their own time, such as those studying for standardized tests when applying to college or graduate school. The GRE and GMAT student networks in Grockit consist primarily of individuals who are working — often in isolation — towards a common, well-defined goal. Since the learning goal is shared by a large number of students, Grockit's live collaborative learning networks offer a venue for peer-assisted study and real-time assistance that is active around-the-clock.

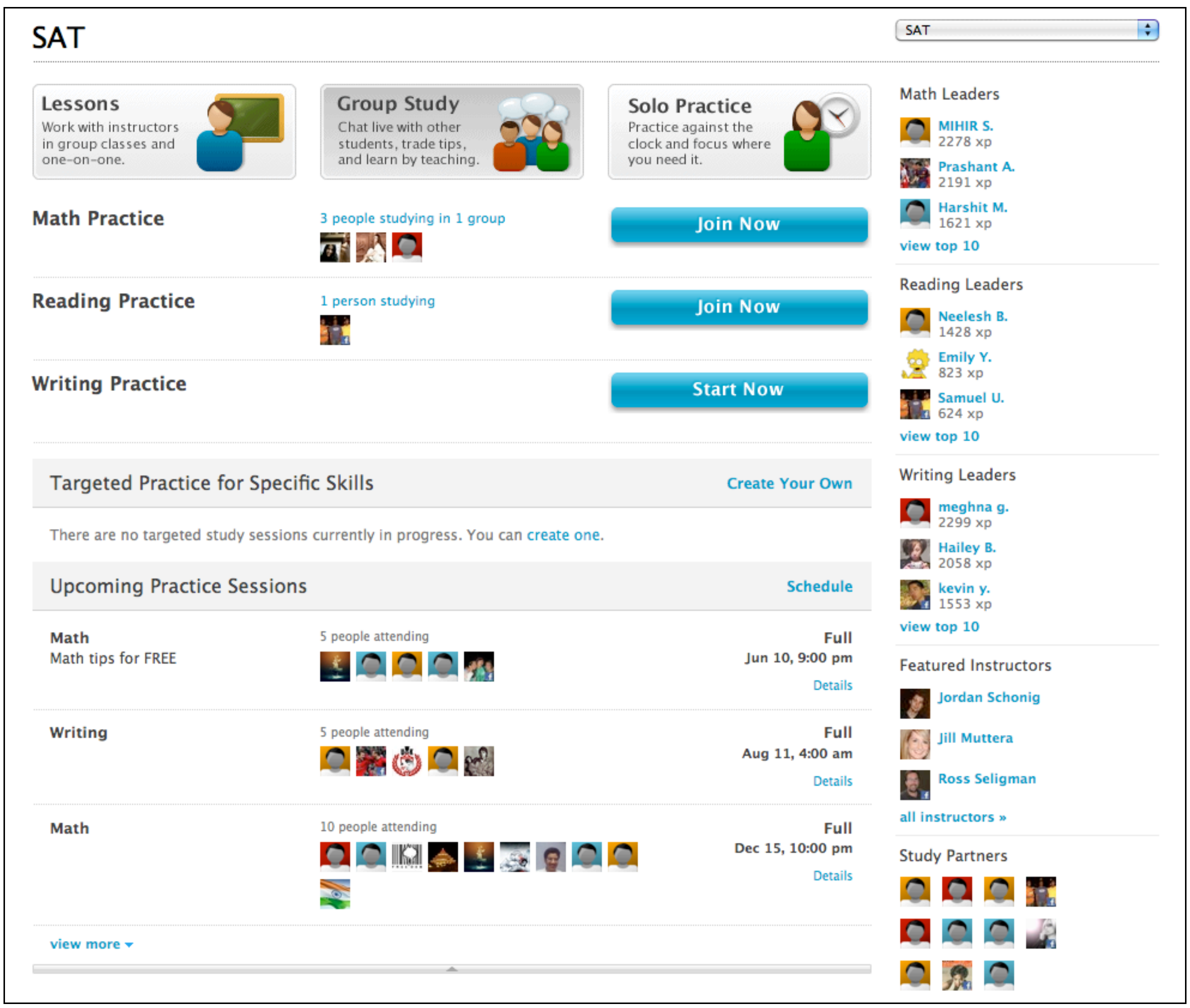

Figure 1: Lobby. Three modes of study are supported on Grockit (individual practice, peer-group study, and instructor-led lessons), and students may choose freely among these available learning contexts.

---

[6] http://dimdim.com, http://wiziq.com, http://edufire.com, http://grockit.com

Grockit is also being tested within a virtual school environment. Virtual schools can offer a student the ability to complete a course on their own schedule, from any location. One challenge in providing a flexible, individualized learning environment is that students may feel disconnected from each other and may miss the opportunity to learn from interactions with their peers. Grockit attempts to fill this gap by extending the benefits of social and collaborative learning to the geographically dispersed students enrolled in a virtual school. We are currently piloting usage of Grockit within the Algebra I curriculum at Florida Virtual School.[7]

Three distinct modes of study are supported on Grockit: (a.) individual practice, (b.) small peer-group study, and (c.) instructor-led lessons. The algorithms and affordances used in these three modes draw on three corresponding areas of research: (a.) Individual practice draws on work in the Intelligent Tutoring Systems field, including techniques for adaptively choosing challenges based on statistical models of the likelihood of response accuracy. (b.) Peer-group study draws on work in Computer-Supported Collaborative Learning, such as techniques for discussion scripting and group formation, and (c.) Instructor-led lessons draw on collaboration tools common in the E-Learning field, such as shared slides, whiteboards, and real-time document editing. These three modes of study serve to organize the site itself, visible in the top-level tabs in the "Lobby" in Figure 1.

What learner interactions does Grockit support?

When students log into Grockit, they see a list of in-progress and upcoming study sessions (some limited to a particular test section, skill set, or difficulty level), and generally proceed to join one of the currently in-progress sessions.

After joining the session, the student sees a screen similar to Figure 2. In the header are instructions for the screen and a timer counting down remaining time. On the left-hand portion of the screen (with the white

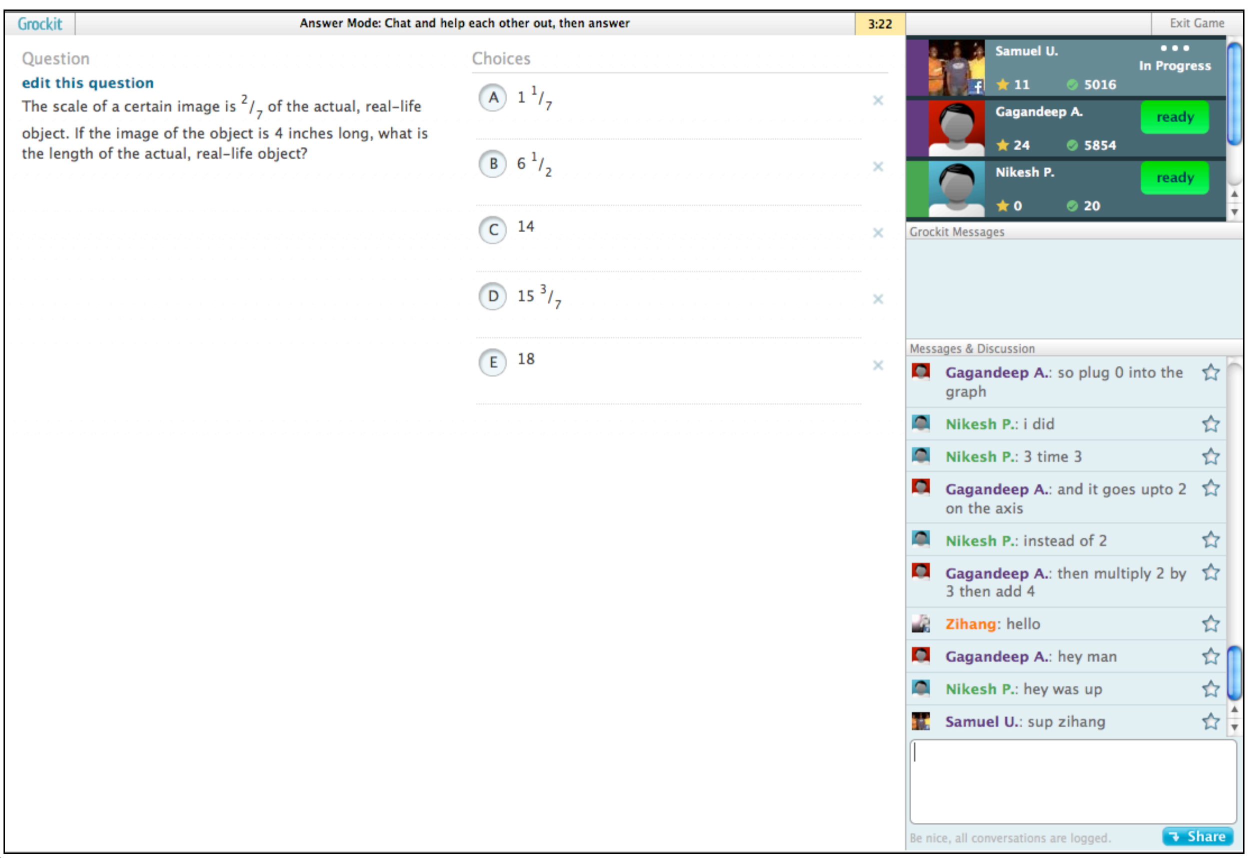

Figure 2: Answer Round. The user interface when a student in the group first sees a new question.

background), a question and a set of answer choices are displayed. On the right-hand side of the screen is a set of tools for synchronous interactions: The upper portion of this includes a list of students in the session. This list changes when students enter or exit, whenever a student's status changes (e.g. whether a response is "in progress" or the student is "ready" to continue, whether that student is currently typing a discussion message), and when points are gained or lost (based on question difficulty and response accuracy, or on points awarded by others to acknowledge helpfulness). Below this is a set of short messages, some authored by students in the group (e.g. asking questions, discussing answers, social chatting), and some generated by the system (e.g. feedback on response accuracy, points earned or lost, discussion prompts). The stars visible to the right of each discussion message provide students with an affordance to acknowledge when another student's comment was particularly helpful. When one student clicks a star, all students in the group are notified of the award.

Once each student in the group selects an answer, the system displays a second screen (Fig. 3), slightly different from the first. Since all students have already answered, the accuracy of each person's response is made visible to all participants.[8] Synchronous chatting remains available, and students are encouraged to ask each other questions, help identify any misunderstandings, and generally attempt to work through difficult parts of the problem together. Once each student indicates that they are satisfied with the question, the screen updates with a new question in the Answer Round mode.

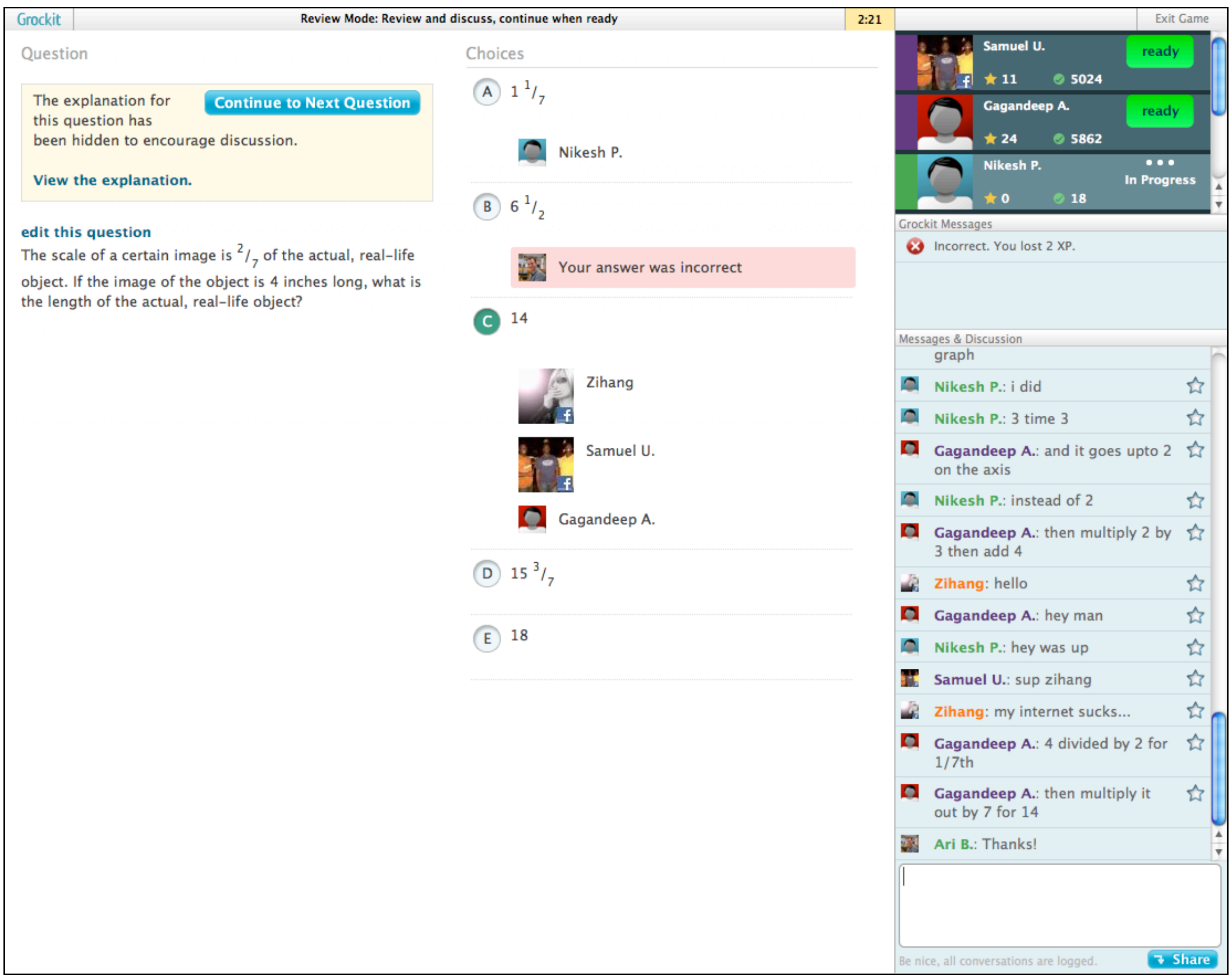

Figure 3: Discussion Round. The user interface after all students in the group have answered the question.

[8] This delayed sharing of all student responses could, arguably, be described as an asynchronous interaction.

After a student exits the study session, they may later revisit it, in the form of an asynchronous Review (Fig. 4). While the screenshot may look similar, there are a number of important differences. First and foremost, each question is annotated with expert-written explanations. This asynchronous teacher-student interaction allows the explanation authors whatever time is required to provide a comprehensive treatment of the question. Reviews include both a snapshot of the original synchronous interaction that occurred around the question and also include additional infrastructure (at bottom) to support asynchronously commenting on the question explanations. While the discussion messages (at right) are only visible to individuals who were present in the original study session, the question comments (at bottom) are visible to anyone who subsequently reviews the question. Students typically use the discussion messages to discuss the questions whereas they generally use question comments to discuss the explanations.

What have we learned about mixing synchronous and asynchronous learner interactions?

One interesting challenge in building a system that supports both synchronous and asynchronous interactions has been in deciding what portions of the experience should be delegated to which type of interaction. Initially, the application was designed to rely exclusively on synchronous interactions. When a student selected an answer, others immediately saw the selection. After all students answered a question, the full expert-written explanation was visible in the discussion round. The Review simply showed a snapshot of the synchronous

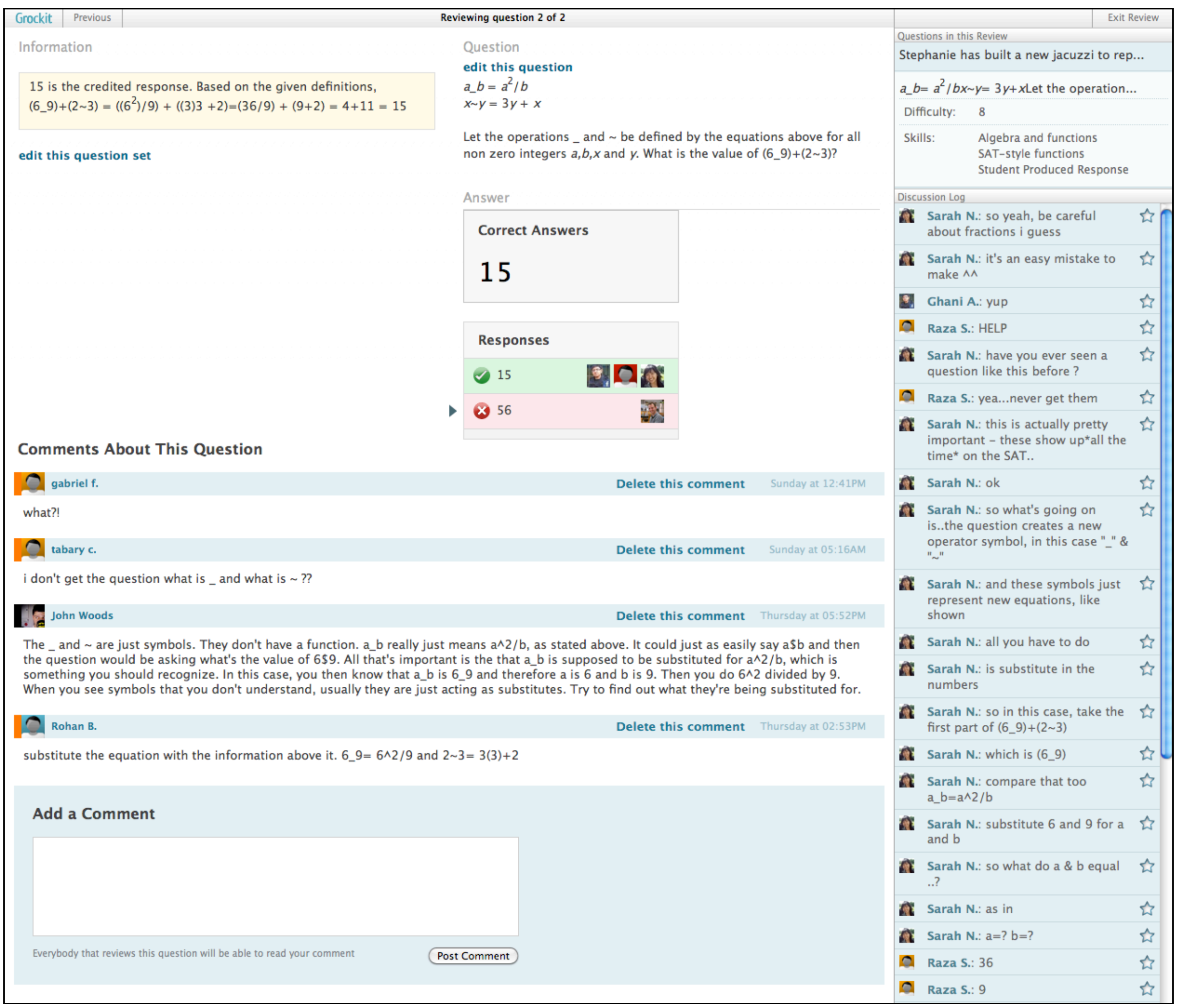

Figure 4: Reviews. The user interface when a student revisits a past study session in which they participated.

interaction (without providing for any sort of asynchronous discussion). Over the past year of development, several portions of the application have shifted back and forth between the synchronous question-answering activity and the asynchronous question-reviewing activity. Our discussions about these changes, often motivated by student feedback and requests, often led us back to the issue of activity-appropriate synchronicity.[9] We raise a few of these questions below, and attempt to generalize lessons learned from each:

Should long reading passages (such as explanations) be visible during synchronous interactions? Initially we felt that including the full explanations during the collaborative portion of the activity would provide an opportunity for students to help each other learn from them. We found that this could frustrate participants when reading speeds widely varied, as the length of the text exacerbated the differences in time required and desired pace among students in a group.[10] One lesson to draw here is that when the length of time necessary to complete an activity varies significantly from student to student, asynchronous interactions are likely to be more appropriate. When the length of time needed is either consistent or short (e.g. writing chat messages), a synchronous interaction may be more beneficial.

Should students see their peers' answers immediately, or only see them after everyone has answered? Again, we initially took the route of providing real-time feedback, but students generally felt that seeing this cut short their time to think independently about the question. But while this type of synchronous interaction may make students feel rushed, others real-time indicators may help alleviate time concerns. Specifically, we found that some students were confused by the lack of information about why they were waiting to continue to the next round. We found that by visually indicating to all participants *when* each student answers without reveal *what* they answered, we were able to provide sufficient information to account for any delays without interfering with each student's ability to independently arrive at their response. The green "Ready" oval next to the participant's name (see Fig. 3) is one such indicator. We subsequently incorporated other types of indicators next to names to indicate other states, such as one to indicate that a participant is current typing a discussion message. In general, introducing translucent state indicators offer a simple way to strike a balance between sharing too much information and not enough information about the stream of events that occur during synchronous interactions.

Should students be able to chat while answering questions? Answering this question requires weighing the risk of students potentially giving each other the answer against that of students feeling frustrated by an inconsistent user interface (i.e. one that alternatively allows and disallows chatting). Since students voluntarily use Grockit to study a set of skills that are ultimately tested in a different environment, they do not have a significant motivation to cheat during participation. As such, we have currently opted for the improved user experience offered by a consistent graphical user interface. The concern about synchronous interaction enabling cheating may be more relevant for other applications, but we urge system designers to think carefully before restricting interactions on this basis: students who wish to game the system may simply use a different channel to communicate, such as an out-of-band instant messaging client.

If a student identifies a comment as "useful," can the comment be reused later, in a different context? One initial goal of the discussion message "starring" action was to identify those comments that students found most helpful in understanding a particular question, and use those comments to help other students in the future. We found that the style of writing in synchronous environments was quite different than that of asynchronous environments: Students often broke their thoughts into a series of short messages, presumably to reduce the waiting time for those reading the comments. Unfortunately, this makes it difficult to identify a single discussion message that captures a self-contained idea. In response to this, we introduced a new type of student interaction, a thread of Question Comments (at the bottom of Reviews in Fig. 4), as a venue for asynchronous discussions without time pressures. While these comments have been more cohesive and less context-dependent than the real-time chat messages, they have been a less popular type of interaction. We believe that this speaks the value of immediacy in question answering. Particularly when a question may require reiteration, further clarification, or follow-up questions, the quick feedback loop of the synchronous interaction seems better suited for finding answers to questions (and when the student most cares about it!) Perhaps it is worth making a distinction between *asking* a question and *formulating* a question, recognizing that the appropriate mode of synchronicity may differ. Similarly,

---

[9] Perhaps a classroom equivalent of this is to determine what is best taught directly by the teacher, what is best done in the context of small group work in class, and what is best assigned as homework.
[10] This is of particular concern for an application designed for use by a combination of native and non-native English speakers.

one might argue that *answering* a question may best be done synchronously, while *explaining* an answer may be better done asynchronously.

Should the number of students participating in a synchronous learning activity be limited? Just as when an (offline) conversation between two people (e.g. study partners) changes when a few more people join in (e.g. study groups), and changes again when a dozen more join (e.g. discussion sections), a few dozen join (e.g. classroom), or a few hundred join (e.g. lecture hall), online conversation dynamics also change with scale. While Dron has pointed to this shift as an inability for computer mediated communications to scale (Dron 2008), we suggest an alternative approach: Choose the group size that offers the collaboration dynamics desired, and scale *how many* such groups are in session concurrently. We opted for a virtual study group paradigm – which allows for active discussion participation by all participants without overloading the conversation – and we create as many study groups as are needed. Beyond addressing concerns about scalability, this approach offers a new way for a learning system to improve as the number of students increases: groups can become increasingly specialized. With sufficient participation, students are able to form study groups around specific question types, skill sets, and/or difficulty levels. As the number of concurrently active groups increases, a student's options for collaborative study on particular areas of interest increase.

How can a learning environment reach the critical mass necessary to sustain synchronous activities? A simple yet non-trivial definition of the critical mass needed to support synchronous interaction is as follows: Regardless of time or day, there is always two or more learners participating. This poses a significant hurdle in launching any new real-time collaborative learning activity. A variety of techniques can be used to provide functionality before reaching this level of participation: encouraging students to coordinate with their friends (via email, telephone, etc.), encouraging group-at-a-time participation (e.g. a classroom at once), raising awareness of the system through PR or marketing efforts, or some combination of the above. One component of the application that we chose to build prior to the initial beta release was the ability to pre-schedule (and RSVP for) a study session. The intent was to provide a mechanism such that even with a small number of beta users, students could asynchronously coordinate on a time to meet synchronously.[11] Perhaps designers of new systems that seek to support both synchronous and asynchronous interactions might consider first focusing on supporting asynchronous interactions until student participation approaches critical mass, and then shift resources towards supporting and enhancing synchronous interactions.

Discussion

The web itself may not be the best infrastructure for real-time social activity-oriented learning, but given the various advantages of designing applications that run on the web instead of on the desktop – no restrictive operating system or hardware requirements, no software download necessary, no software installation or maintenance necessary, no firewall modifications required, the opportunity to deploy frequent updates to the software, the ability to use data collected from all participants to improve the experience of each individual participant – the misfit seems worth working around.

While the motivation for supporting synchronous interactions among peers was primarily intended to provide a compelling social experience that motivated students to engage in learning collaboratively, it also had the effect of opening an interesting new set of questions. We shared our responses to a few of these to provide a notion of how future system designers might identify and approach issues of determining interaction synchronicity. The particular decisions that we made in response to these questions are frequently revisited and occasionally changed, but we believe that the lessons learned from grappling with the notion of appropriate synchronicity remain valuable.

[11] It is worth noting that time zone differences, and the task of handling and communicating them, do complicate the process of supporting pre-scheduled sessions.

Acknowledgements

This paper is due, in large part, to an ongoing series of (synchronous) verbal conversations and (asynchronous) email exchanges on this topic with Farbood Nivi. Many thanks to the entire Grockit team for continually experimenting with how to best support students in achieving their learning goals.

References

Bader-Natal, A. (2009). Incorporating game mechanics into a network of online study groups. In Craig, S. D. and Dicheva, D., editors, *Supplementary Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED-2009)*, volume 3, Intelligent Educational Games workshop, 109-112, Brighton, UK. IOS Press.

Bernard, R., Abrami, P., Lou, Y., Borokhovski, E., Wade, A., Wozney, L., Wallet, P., Fiset, M., and Huang, B. (2004). How does distance education compare with classroom instruction? A meta-analysis of the empirical literature, *Review of Educational Research*, 74(3):379.

Dohn, N. (2009). Web 2.0: Inherent tensions and evident challenges for education. *International Journal of Computer-Supported Collaborative Learning*, 4(3):343-363.

Downes, S. (2005). E-learning 2.0, *eLearn Magazine*, Volume 10: ACM, New York, NY, USA.

Dron, J. (2008). Designing the undesignable: Social software and control, *Educational Technology and Society*, 10(3):60-71.

Fielding, R et al. (1999). Hypertext Transfer Protocol – HTTP/1.1, http://tools.ietf.org/html/rfc2616#section-9, Retrieved: 2009-09-06.

Garrett, J. J. (2005). Ajax: A new approach to web applications. http://www.adaptivepath.com/ideas/essays/archives/000385.php, Retrieved: 2009-09-06.

Means, B., Toyama, Y., Murphy, R., Bakia, M., and Jones, K. (2009). Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies. Technical report, *U.S. Department of Education, Office of Planning, Evaluation, and Policy Development*, Washington, D.C.

O'Reilly, T. (2005). What Is Web 2.0, http://oreilly.com/web2/archive/what-is-web-20.html, Retrieved: 2009-09-06.

Russell, A. (2006). Comet: Low latency data for the browser. http://alex.dojotoolkit.org/2006/03/comet-low-latency-data-for-the-browser/, Retrieved: 2009-09-06.

Zhao, Y., Lei, J., Yan, B., Lai, C., and Tan, S. (2005). What makes the difference? A practical analysis of research on the effectiveness of distance education. *The Teachers College Record*, 107(8):1836-1884.