The SEREBRO Project: Fostering Creativity through Collaboration and Rewards

Rose Gamble
gamble@utulsa.edu

Sandip Sen
sandip-sen@utulsa.edu
University of Tulsa
800 South Tucker Drive
Tulsa, OK 74104

Bradley Brummel
bradley-brummel@utulsa.edu

Frank Grove
dean-grove@utulsa.edu

Noah Jorgenson
noah-jorgenson@utulsa.edu
University of Tulsa
800 South Tucker Drive
Tulsa, OK 74104

David Baker
david-baker@utulsa.edu

Dan Guernsey
dan-guernsey@utulsa.edu

Daniel Hampton
daniel-hampton@utulsa.edu
University of Tulsa
800 South Tucker Drive
Tulsa, OK 74104

Jordan Hughes
jordan-hughes@utulsa.edu

ABSTRACT

Software Engineering is a highly creative endeavor that challenges Computer Science (CS) students to establish an innovative vision and to craft an outstanding product. Curriculum standards for CS education typically lack creative approaches to Software Engineering, focusing on technological solutions rather than innovative design. Accountability for and contribution to creative initiatives are therefore not part of grading methods in typical Software Engineering courses. In this paper, we introduce a unique framework to foster creativity within an asynchronous, interactive, and graphical environment that tracks the team's product understanding through the phases of the Rational Unified Process for software engineering. We incorporate a layered, multiagent system to apportion rewards for creative contributions that correspond to theories of creativity based on external motivation.

Keywords

Multiagent Systems, education, learning

1. INTRODUCTION

Creativity is defined as the ability to produce work that is novel, high quality, and appropriate [34]. Researchers generally agree that creativity is influenced by a number of factors, such as personality, motivational, environment, skills, and knowledge [37]. Recent theories posit that creativity is a decision. According to the investment theory of creativity [36], individuals decide whether to use their resources to generate new ideas, evaluate those ideas, and then sell those ideas to others. The SEREBRO project targets the malleable factors that affect creativity. Incentives, motivation and rewards for creativity are typically lacking in undergraduate Computer Science (CS) education due to existing curriculum guidelines that focus on technological solutions rather than innovative designs. The SEREBRO project has three objectives for fostering and rewarding creativity within the pedagogical environment of a year-long Software Engineering Projects class at the University of Tulsa:

- Build a graphical, asynchronously communicated, idea network to interconnect ideas and support creativity via a social networking platform.
- Tailor idea network usage to explicit phases of software development to encourage creativity toward project milestones.
- Investigate mechanisms that assess idea proliferation and propagation to reward creativity.

This project synergizes expert knowledge and experience in software engineering, multiagent systems and learning, and industrial/organizational psychology. In this paper, we present our novel framework for developing a multi-faceted educational software system called SEREBRO (Software Engineering REwards for BRainstorming Online). This system forms the foundation of the project to understand how creativity in software development can be enhanced through technology and reward.

2. BACKGROUND

In this section, we overview the different aspects related to the SEREBRO project.

Cite as: The SEREBRO Project: Fostering Creativity through Collaboration and Rewards, R. Gamble, S. Sen, et. al., *Presented at EDUMas Workshop, part of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary

2.1 The Creative Process

Creativity has been studied from a variety of perspectives. One of the most common distinctions has been between the creative process and individual differences in creative behavior. Our project focuses on enhancing the creative process by applying various aspects of two theories of creativity. Sternberg and Lubart's [36] investment theory of creativity states that in order to be creative, individuals must be willing and able to "buy low and sell high." In other words, creative individuals pursue unpopular or unknown ideas with potential and then persist until they convince others of the idea's value. The underlying idea of this theory is that "creativity is in large part a decision that anyone can make but that few people actually do make because they find the costs to be too high" [35]. In this theory, creativity requires a confluence among six critical resources: intellectual skills, domain knowledge, thinking styles, personality, motivation, and environment.

Amabile's componential model of creativity [3, 4, 5] states that there are three critical components necessary for creativity: domain-relevant skills, creativity-relevant processes, and intrinsic task motivation. Domain-relevant skills include a minimal level of domain-specific talent and expertise. Creativity-relevant processes involve the ability to deal with complexity and break through one's mental set when problem solving, an understanding of methods for producing novel ideas, and a high-energy, focused work style. Intrinsic task motivation refers to "the motivation to engage in an activity primarily for its own sake, because the individual perceives the activity as interesting, involving, satisfying, or personally challenging" [7]. Individuals will be intrinsically motivated to the extent that the task is interesting, they have autonomy, they feel competent to perform the task, and they feel a sense of self-determination and control. On the other hand, external motivation is "the motivation to engage in an activity primarily in order to meet some goal external to the work itself, such as attaining an expected reward, winning a competition, or meeting some requirement" [7].

This model posits that intrinsic motivation enhances creativity, while extrinsic motivation can either undermine or support creativity. Specifically, research has found that creativity suffers when task autonomy is reduced [14], competition is introduced [2], or a performance evaluation is expected [1]. Amabile and colleagues term these "non-synergistic extrinsic motivations" because they undermine one's self-determination and competence. However, extrinsic motivation that reinforces one's self-determination and competence may support creativity. Rewards and recognition for creative ideas, clear task goals, and performance feedback that confirms competence and provides guidance on how to improve should enhance creativity. Amabile and colleagues refer to these external motivators as "synergistic extrinsic motivations" because they can combine with intrinsic motivation to foster creativity. The creative process involves problem presentation, idea generation, preparation, idea validation, and idea communication, and is generally thought to occur in a non-linear fashion [3]. As previously mentioned, creativity requires both novelty and appropriateness [3, 34]. Novelty is critical during the early stages of the creative process, while appropriateness is more important during the later stages [4, 5]. Intrinsic motivation is especially important for fostering novelty, and synergistic extrinsic motivation helps individuals persist and focus on the upcoming evaluation. Thus, intrinsic motivation is especially important in the early stages of the creative process, while synergistic extrinsic motivation is particularly important during the later stages.

Because many creative activities take place in a social context [11], it is essential to consider how the group or team can influence the creative process. One model of team innovation states that team creativity and innovation will be influenced by group task characteristics and group knowledge diversity and skills [41]. Intrinsic motivation should be highest when the group task is a whole task, creates varied task demands, and allows autonomy, learning opportunities, and social interaction opportunities. These task characteristics are consistent with socio-technical systems theory, which emphasizes the need for autonomous teams to optimize the social subsystem with the technical subsystem [8]. Both group task characteristics and group knowledge, diversity, and skills should affect group creativity and innovation via integrating group processes that support these goals. Specifically, group processes should clarify and ensure commitment to the group's goal, allow participation when making decisions, manage conflict effectively (i.e., constructive task-related controversy), support innovation, and create a sense of safety for creativity [41].

2.2 Creativity and Software Engineering

It is generally agreed that in order to truly support creativity through the use of a software tool, the tool should accommodate all skill levels and seamlessly interface with other tools the user might employ in their creative endeavor [26, 29]. Interface usability is an important aspect of creativity support tools (CSTs) because the tool should not hamper creativity [18]. Production of these tools has not reached the quality needed for use in the classroom. We believe, however, that collaborative problem solving activities for course projects can benefit from CSTs. These same activities are also ideal for studying creativity processes as students with overlapping knowledge levels, expertise, and capabilities collaborate in a supportive environment [9]. While creativity has been widely studied in the social science literature, CSTs are not yet mature for IT applications. Some key desirable features for CSTs as identified by researchers are [6, 29]:

- exploration and collaboration with open interchange
- simple, attractive, and non-intrusive, yet powerful
- flexible, with support for different learning, designing, and thinking styles
- allow asynchronous and non-linear work flow.

Researchers studying Computer Supported Collaborative Work [13, 27] have focused on the distributed, social aspects of creative activity [11, 19] resulting in more discourse on CSTs. Several brainstorming tools exist for eliciting an improved return-on-investment. Three categories of tools are prevalent: idea prompt generators for associational thinking [28], mind mappers for visual thinking [42, 25, 16], and information organizers for organizational thinking [15, 24]. Few of these tools combine the visual and organizational aspects of brainstorming. None include a concept of rewards or personnel evaluation.

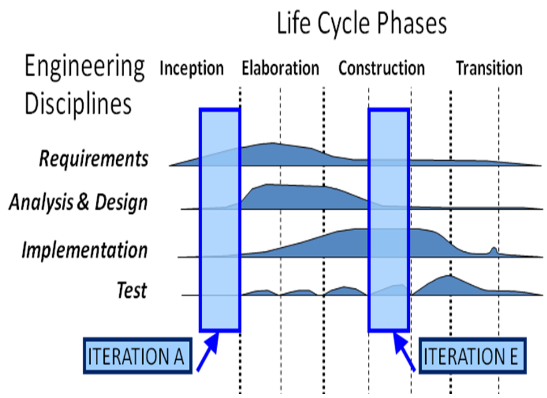

Figure 1: The Software Engineering Disciplines in the Rational Unified Process.

2.3 Software Development Process

A standard development process structures the development of non-trivial software from the first set of requirements to product delivery. For software engineering classes, we use the Rational Unified Process (RUP) [30] which provides relevant content to empower students with immediate process knowledge though they have limited experience. Figure 1 is a modified version of RUP from [30].

The RUP is an iterative process with four phases: Inception, Elaboration, Construction, and Transition. An iteration achieves particular tasks and milestones incrementally, breaking the software development across smaller, more manageable disciplines for each phase. Thus, each iteration contains a portion of the development effort by gathering and clarifying requirements, establishing design and architecture models, implementing executable portions of the product, and testing implemented code along with code integration. The Inception Phase clarifies the interpretation of the customer's requirements, explores technology solutions to reduce product development time and increase quality and performance, and generates an initial set of artifacts, such as a vision document, an architecture, and a throw-away prototype. If done properly, vast amounts of intellect and creativity can be communicated, discussed, and decided upon before leaving the inception phase. Milestones are reached upon receiving positive feedback from the customer and, in the case of a class, a good grade from the instructor. Though mistakes can lead to the wrong path, the iterative approach embedded within the RUP keeps them contained and identified early in the development process. The Elaboration phase refines initial artifacts and moves the team toward the software design. Milestones of Elaboration include creating the experimental arrangement of architecture entities and the building of an executable prototype. The Construction phase has multiple milestones for coding and testing, though some requirements gathering and design interpretations occur. In this phase, creativity is often associated with the use of technology, efficient coding and reuse, and repeated contact with the customer. Transition is the process of delivering the system in alpha, beta, and final versions to the customer site. Creative development of the user's manual is part of this phase. The RUP is a feed forward approach because considerations are given to all disciplines of the product in all phases. Thus, creative activity can be restarted and refocused within each iteration. Though versioning is included in tools that organize software projects according to the RUP, creative brainstorming and idea organization are not available within these tools.

2.4 Problem Structures in MultiAgent Systems

Various network models have been proposed to model agent interactions in a multiagent system [17, 33, 40]. We are particularly interested in a distributed goal representation framework that allows effective coordination between multiple agents [23]. The Generalized Partial Global Planning (GPGP) and its associated TAEMS task network representation have been developed as a domain-independent on-line coordination framework to facilitate cooperation by small teams of intelligent agents. The framework develops a distributed solution to a global optimization problem by decomposing the global problem into a set of dynamically evolving local optimization problems assigned to individual agents. Initially agents possess only local views of their assigned subtasks and their interrelationships stored in the TAEMS representation, which adds quantitative information and sequencing constraints to AND/OR goal trees. GPGP provides information gathering and coordination to develop more global views and make commitments to facilitate coordination. While GPGP/TAEMS is a complex, elaborate framework, for our purpose, the relevant components is the use of a set of hard and soft relationships between tasks to represent various non-local effects. An example hard constraint is *enables* which denote that completion of a sub-task is necessary before another sub-task can be processed. *Facilitates* is a soft relationship signifying results from a sub-task may be useful for processing another task. We plan to adapt and expand these features from the GPGP/TAEMS framework to provide the set of activities available for creative collaboration in SEREBRO to codify the links between ideas.

A key component of the SEREBRO framework is the feedback of external reinforcement over various internal and intermediate actions, proposals, etc. The postings, suggestions, comments of the users are used to develop a multi-connected, feed-forward network structure. While various learning schemes can be used to feedback external reward over internal nodes in a network, e.g., backpropagation scheme in neural network structures [31], we believe reinforcement learning techniques are most appropriate for addressing the challenge of effective reward distribution [20, 38].

Payoff distribution among agents in a coalition has received significant attention in multiagent systems research [21, 32, 39]. Solution concepts like the *core*, the *Kernel*, etc., have been proposed to generate payoff sharing that guarantees stable coalition structures. The current paper does not directly use these techniques because we assume that a stable group of users are using the system. However, we plan to use aspects of fair payoff distributions from the abovementioned research to create more fair reward structures based on creative participation and contribution by the users towards effective design solutions.

3. THE SEREBRO TOOL

The main challenge behind recognizing and rewarding creative contribution by team members in a Software Engineering project is to devise a framework that captures and relates ideas as they are generated, guides the creative design process toward the project artifacts and milestones, and rewards those who contribute to the project's creative elements. Our core notion of the creative process is an idea. An idea can be a belief statement, a problem solving approach, a solution to a problem, or a discussion related to any idea type. Idea-related activities include:

Brainstorming: asserting initial ideas, vision, and designs without past support;

Spinning: connecting supporting ideas to support progress towards milestone;

Pruning: disallowing further spinning of an idea and forcing the pursuit of other paths;

Finalizing: declaring an idea as a decision-action node to signify a definite team agreement to complete an iteration goal.

The partitioning of nodes into the above roles aids the association of creativity metrics, and reward distribution. Brainstorming allows for asynchronous, spontaneous input. Spinning produces pathways for creative expression from initial concept to ideas that agree and extend the concept or ideas that disagree and offer contrasting input. Pruning is a joint decision of contributors to stop (perhaps temporarily) a non-productive pathway so that focus can be put toward other ideas. Finalizing is the collective decision or result that stems from multiple ideas. Finalized nodes move the team toward achieving a milestone associated with a RUP iteration or completed phase.

These nodes are linked to each other in a feed-forward manner capturing the time-line of activities to form an *idea-network*. Since our approach applies links to ideas instead of users, it allows feedback to be given on a per-idea basis. It also allows measurements to be taken based on the links between ideas. The social activity that centers on ideas is associated with actual users for the purpose of rewarding the correct person. Thus, analysis can be performed on the ideas much the same way that it is performed on the social network of people [22, 10] to determine how creativity enhances the team software development experience.

Our web-based application, called SEREBRO 1.0, is an idea generation tool that incorporates graphical and textual idea expression, idea connectivity, and agree/disagree capability. It is used as part of a 2-semester course sequence in software engineering. Its pedagogical goal for undergraduate software development teams is to enhance students' knowledge of software engineering principles and techniques through participation in creative group activities. Personal responsibility and satisfaction combined with goal achievement are stressed. The plan is to couple motivational methods with reward systems that overlay dynamic idea generation dedicated to fulfilling the requirements of a software project.

Figure 2 provides an overview of the SEREBRO tool. The upper right section shows the Forum in which topics are posted across the RUP. The lower right section shows the Graph View of an idea network. Agree nodes are green triangle. Orange triangles are disagree nodes. Yellow pentagons are finalized ideas. Each node has a corresponding post entered in a thread-based environment that reflects idea connectivity, as seen on the left half of the figure. The left portion of the figure labeled as Post View reflects a sample of that thread-based environment. Within the Post View section, the lower left portion of this section reflects the finalization of a node in which tags indicate those ideas that contributed to the final content, while the lower right portion of this section shows the stop procedure used for pruning a thread of discussion.

4. MULTIAGENT REWARD FRAMEWORK

To properly facilitate and allocate the internal and external reward distribution, we developed a multiagent framework. The framework is designed to allow a fair and responsive reward distribution by assigning a user-agent to each individual participating in a SEREBRO project, an agent for each phase of the RUP, and an arbiter agent to finalize and distribute rewards. These agents must negotiate a mutually acceptable distribution of internally or externally generated rewards. Examples of external reward includes those received from instructor evaluation, user-satisfaction of delivered product, etc. Examples of internal reward includes those received based on contribution and influence of individual participants in the group problem solving process.

In order to determine the importance of each node within the SEREBRO framework, these agents will distribute points to other agents that the user has responded to. At prespecified times in the classroom agenda, the agent's aggregate points are passed back over the network to the phase agents. Each phase agents then computes a weight W_p vector with weights w_a for each user participating in that phase. After each phase in the RUP is completed, the phase-agent submit W_p to the arbiter agent, who distributes the rewards. In the final phase, the phase agents negotiate over the final weight vector W_f. Next we present further details of the multiagent framework, negotiation protocol, and the point distribution system.

The agent based reward distribution framework is designed to achieve a fair and accurate distribution of rewards by the SEREBRO system. In the future we expect the reward system to handle both external feedbacks, from users of the product or service generated by using the system, as well as internal feedback, based on activities of the participants involved in generating the product using the SEREBRO system. In this paper, we focus on the generation and distribution of rewards based only on internal feedback.

Each user is assigned a user-agent i. Each user-agent i has a point repository and must adhere to the point distribution protocol. The protocol distributes points based on the participation of the users and the estimated overall utility of an idea. Our goal in designing the reward distribution scheme was to handsomely reward ideas that spawn large discussions while being less enthusiastic in rewarding ideas that do not generate fruitful responses. Hence, the topology and links in the idea network play a pivotal role in the reward amount distributed to an idea node. The goals identified above are supported as follows:

Agree with a Node: When a user agrees to a node, x points are created and passed to the node being replied to. If the user is agreeing with himself/herself, no points are distributed.

Disagree with a Node: When a user disagrees with a node, the agent that created the node being replied with is charged $\frac{x}{2}$ points.

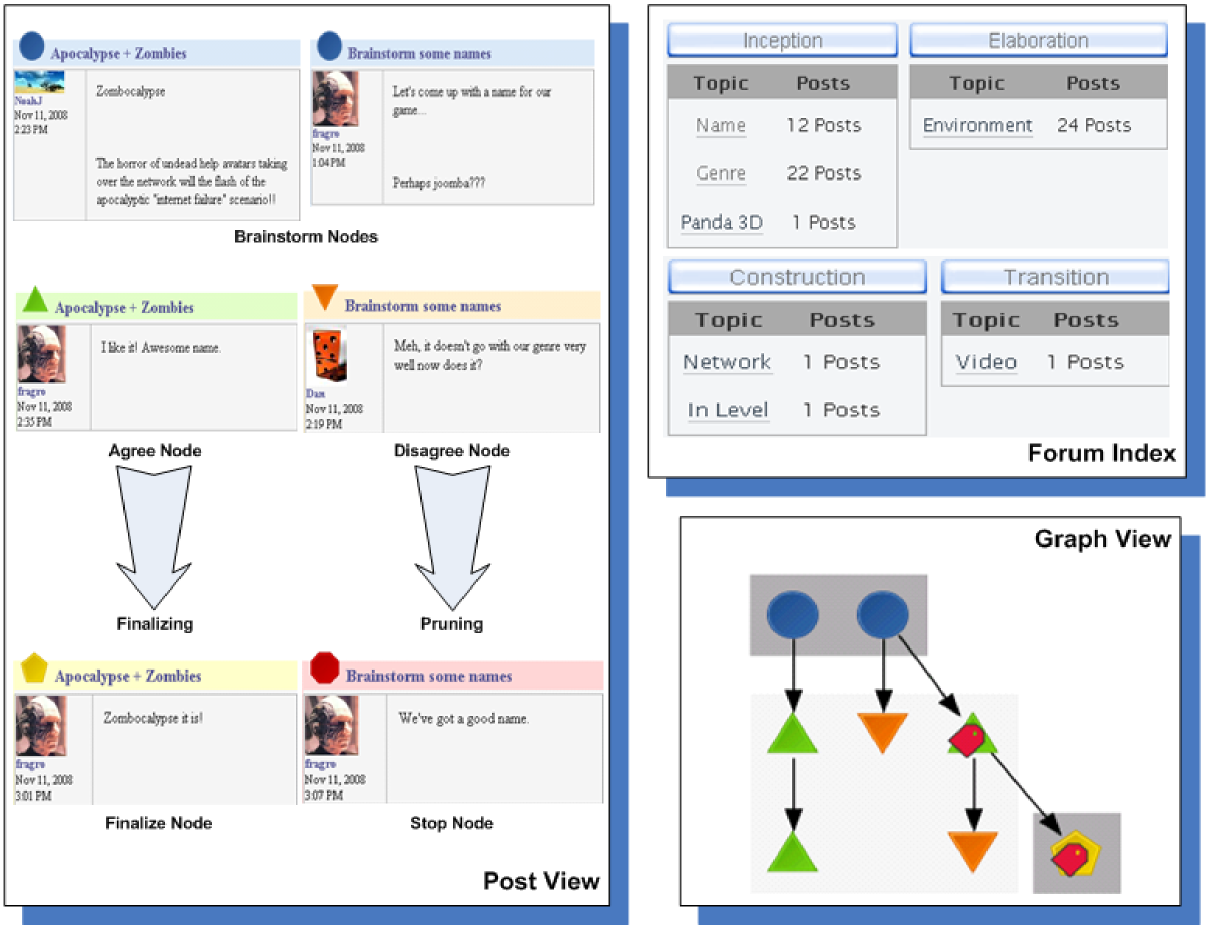

Figure 2: The Software Engineering Disciplines in the Rational Unified Process

Agent Receives x Points from Reply: When a user's node a is replied to with an agree node b, the user-agent who posted a will receive x points. If the node a is also a reply, the user-agent must pass $\frac{x}{2}$ to the node c to which it replied.

Finalized Node: The finalized node allocates $k * x$ points to the node it is replying, and x points to each node that is tagged with it.

This protocol efficiently allocates points to the users that participate the most and whose participation is most appreciated. Over time the points will accumulate which also allows instructors to observe the progress of threads and individual users over time. At an arbitrary time selected by the instructor, rewards will be allocated. At this time, each user-agent submits their points to the phase agents where the points originated. The phase agents then simply pass along the points, so that the arbiter may distribute N rewards to the top N users. These rewards may occur during the end of a RUP phase, as employed by the instructor, or at the demonstration of a prototype. The process that determines the final reward must take into account the points accumulated in all four phases.

Based on the communication between users, each phase agent p computes a point distribution, W_p, containing the points each student accumulated within that agent's respective RUP phase. These values are weighted by the satisfaction vector S_i. In order to properly determine the reward distribution, the phase-agents must negotiate an outcome that maximizes their satisfaction vector S_i for each user i. This satisfaction vector magnitude is a number in the range [0,1].

S_i ensures the relevance of outliers in the point distribution. Since point distributions from each phase vary, no phase-agent's S_i magnitude is 1. In the improbable case when these point distributions are equal there is no need for negotiation. These outliers created by the difference in distributions are addressed through a bidding process. This process is important because these outliers can be extreme. For instance if a user had low points in an early phase but exceeded all others in later phases, the user's contributions must be considered carefully. This bidding process begins when each agent submits a bid to the arbiter agent equal to it's point distribution W_p.

The bids are received by the arbiter which takes the average w_a and returns this point distribution to each phase agent. The phase-agents then make incremental changes to the point values of the averaged distribution to maximize the sum of S_i and resubmits the new point distribution as a bid to the arbiter agent. This process continues until the arbiter agents detects convergence, i.e., the difference between the bids and the average falls below a threshold, or after a maximum number of bidding sessions. Once the arbiter has ceased bidding rewards are allocated based on the final average point distribution.

5. SEREBRO USE

5.1 Software Projects I

Software Projects I is the first class in the two-course software engineering sequence that is required for seniors in Computer Science. Multiple software challenges are explored by change teams of students throughout the semester. Toward the end of the semester, final teams are chosen and assigned projects [12]. Initial ideas for product name, vision, and functional requirements are needed for the team to complete the semester final. To test the functionality and usage outcomes, we evaluated the first prototype, SEREBRO 1.0, against SEREBRO Lite, a thread-based tool that only showed connected textual posts without the ability to agree or disagree with a post. Student teams were randomly assigned to either SEREBRO 1.0 or SEREBRO Lite 1.0. We focused on evaluating:

- Satisfaction with the tool
- Their feeling toward creative expression
- Their perceived benefits to the team and to the progress of the project

Among those students using SEREBRO 1.0, the response to a series of questions indicated the overall feelings for SEREBRO were positive. Those surveyed were satisfied with the usefulness of SEREBRO for idea communication and organization. Most indicated a positive feeling toward creative expression. These teams had the visual benefit of seeing a response to their ideas, as well as having an understanding of the teams perception of an idea according to its graphical connectivity. Many of the features currently implemented in SEREBRO were expressed as being highly desirable by the reviewed users. The thread based connectivity of ideas, the agree/disagree options for idea connectivity, graphical display of idea nodes and the tag-based system to meta-connect ideas were all preferred by nearly 80 percent of those users. In particular, the functionality for uploading files for sharing and email notification upon node creation were extremely well received, with a nearly unanimous 90 percent indicating the usefulness of these features.

Percent rated functionality as useful	
Idea communication	78 %
Idea organization	72 %
Thread based connectivity of ideas	78 %
Upload of files for sharing	94 %
Version control for uploaded files	83 %
Agree/disagree options for idea connectivity	67 %
Graphical display of idea nodes	78 %
Tag-based system to meta-connect ideas	78 %
Email upon node creation	89 %

Table 1

Examination of the SEREBRO graphs and threads from the fall experiment correlated to the results as we saw a clear distinction between the users on SEREBRO 1.0 and SEREBRO Lite 1.0. While the students on SEREBRO 1.0 were utilizing the agree and disagree options and properly creating new brainstorm nodes when a new idea was being formed, the groups on SEREBRO Lite were noticeably more inactive than the users on SEREBRO 1.0. This was so much the case that one group on SEREBRO Lite hardly used the tool. The groups using the SEREBRO Lite didn't display as good idea organization, and the spinning of ideas was nearly nonexistent with no real development if an idea was posted. In contrast, the users on SEREBRO 1.0 were more active and in total had more posts with more content, most likely the result of a highly visual graphical view of the nodes and the ability to specifically agree and disagree when discussing ideas. This is supported by the survey results indicating that nearly all the students suggested the features unique to SEREBRO 1.0 were more useful overall. Grading was done prior to and without consulting SEREBRO activity. Thus, grading was solely based on traditional assessment of artifacts turned by teams. Once grading was complete, those teams that used SEREBRO 1.0 had an average of 10 percentage points higher on their final than those that used (or neglected to use) SEREBRO Lite. More assessment is needed to determine if usage correlates to higher quality artifacts because of SEREBRO or if the teams were actively working on their final and SEREBRO was another part of that activity.

5.2 Broader Evaluation and Use

Currently, SEREBRO is undergoing extensive testing as part of the second half of the Software Engineering Projects class sequence. Three non-trivial software projects are under development:

- A chemistry lab creator with automatic scoring and database history
- A game aimed at recruiting junior and senior high school students to computer science
- A Web services management system for responding to crises

Students are separated across six teams with two teams per project. Standard rubrics developed for accreditation purposes and from previous years' classes are used to grade core artifacts (e.g., documentation, code, and demonstration) for the four 'builds' during the semester. The structure of the class coupled with the traditional grading approach allows us to directly compare and contrast quality of the project and productivity of the team with the creativity metrics and reward system that are part of SEREBRO.

In the Inception phase, teams are using SEREBRO to identify additional functional requirements, technology for use in the product, and programming platform. They also develop a prototype to present and be judged by the class for its sufficiency in demonstrating a risky or complex requirement and the flawlessness with which it is scripted and shown. At the end of the phase, SEREBRO will be used to finalize their ideas to move to the Elaboration Phase. The Elaboration Phase should see creative ideas for generating design artifacts such architecture, class, and sequence diagrams. Additional prototype development will take place where the teams can discuss the implementation of various requirements. Another integral part of Elaboration is the understanding of project risks related to technology, experience, and customer interpretation that can cause a project to fail. Idea threads will be created per identified risk for

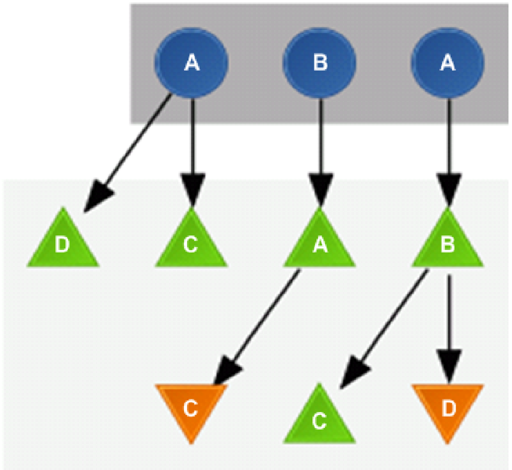

Figure 3: Sample Graph from current Experimentation

generation methods for mitigation. The Construction Phase will continue the risk discussion as well as examine different aspects related to programming and the problems therein. As the customer interacts with various prototype demonstrations, ideas will be circulated as to how to creatively answer the questions and requests by the customer within the product. The Transition phase will encompass the creation of the User's Manual and delivery of the system. This phase is planned to have the least input unless problems with the system delivery arise. After semester grades have been completed, we will review the SEREBRO use in terms of creativity metrics, (e.g. fluency and originality of ideas) and team effectiveness (quality of project and productivity of members) that we will correlate with how the reward system designated its results.

5.3 Sample reward distribution

As mentioned above, the SEREBRO framework is currently being tested in a Senior Software Project class. We now present a small sample graph from this class to illustrate the point distribution process. This example is demonstrated in Figure 3. In this example users A, B, C, and D are taking part in a discussion related to the project. We use a point value of $x = 1$ for this example. The results from the point distribution scoring on this example can be found in Table 2. While this is only a small sample from existing discussions, we can see that the individual scores have already begun to diverge.

Scores	
A	3
B	1.5
C	0
D	0

Table 2.

From this example, we can see that users who take part in the discussion may not necessarily score points. Other users must respond positively to a user's comments in order for points to be accumulated. Initial analysis shows that this will bias our results towards users whose comments are replied to most frequently and most positively. We believe our protocol will operate well under the assumption that the creative and productive users are most likely to initiate these threads of discussion.

Although our analysis is preliminary, we believe that the point distribution protocol will perform well in allocation of rewards. As the class moves into the next phases of the RUP, more data will become available to test our protocol against the performance perceived by those taking part in the experiment. At this point we will proceed with more in depth analysis of the point distribution protocols described here and continue with the implementation of the phase-agent's negotiation protocol.

6. CONCLUSION

In this paper, we introduce the framework for the SEREBRO project to codify creativity theories and metrics within a type of social networking system where ideas are the basis for expression and reward. We show actual screen shots from SEREBRO's graph and post views, along with the types of ideas expressible and where in the software development process they can be housed. Our multiagent reward system is described with initial reward assignments to project teams. The SEREBRO system brings together complementary expertise of research groups from software engineering, psychology, and multiagent systems to promote creative collaboration and problem solving by CS students in group projects. While initial testing shows performance improvement of students of SEREBRO 1.0 over those that use SEREBRO Lite, much remains to be done to harness the potential of cross-fertilizing ideas, concepts, methodologies, and processes from these diverse research areas. In particular, the interplay of external and internal reward metrics and their redistribution over the idea network produced based on asynchronous group problem solving poses exciting opportunities and challenges for identifying and fostering fruitful, creative user contributions.

Acknowledgement.

This work is supported in part by a National Science Foundation grant IIS-075743.

7. ADDITIONAL AUTHORS

8. REFERENCES

[1] T. Amabile. Effects of external evaluation on artisitic creativity. *Journal of Personality and Social Psychology*, 37:221–233, 1979.

[2] T. Amabile. Children's artistic creativity: Detrimental effects of competition in a field setting. *Personality and social Psychology Bulletin*, 8:573–578, 1982.

[3] T. Amabile. *The Social Psychology of Creativity.* Springer-Verlag, New York, 1983.

[4] T. Amabile. Motivational synergy: Toward new conceptualizations of intrinsic and extrinsic motivation in the workplace. *Human Resource Management Review*, 3(3):185–201, 1993.

[5] T. Amabile. *Creativity in Context: Update to The Social Psychology of Creativity.* Westview, Boulder, CO, 1996.

[6] L. Candy and E. Edmonds. Supporting the creative user: A criteria based approach to interaction design. *Design Studies*, 18:185–194, 1997.

[7] Collins and T. Amabile. *Motivation and Creativity.* Handbook of Creativity. Cambridge University Press, New York, 1999.

[8] R. Cooper and M. Foster. Sociotechnical systems. *American Psychologist*, 26:467–474, 1971.

[9] M. Csikszentmihalyi. *Creativity: Flow and the Psychology of Discovery and Invention.* Harper Perennial, 1997.

[10] J. Dietrich and N. Jones. Using social networking and semantic web technology in software engineering - use cases, patterns, and a case study. *Journal of Systems and Software*, 81:2183–2193, 2008.

[11] G. Fischer. Creativity and distributed intelligence, 2005.

[12] R. Gamble and M. Smith. Moving toward "reality" in team selection for software engineering. *Frontiers in Education Conference*, pages F3H–21–F3H–26, 2008.

[13] I. Greif. *Computer Supported Cooperative Work: A Book of Readings.* Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[14] B. Hennessey. The effect of extrinsic constraints on children's creativity while using a computer. *Creativity Research Journal*, 2:151–168, 1989.

[15] InfoSelect. http://www.miclog.com/.

[16] Inspiration. http://www.inspiration.com/.

[17] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[18] P. Jennings. Creativity support tools for and by the new media arts community. *NSF Workshop on Creativity Support Tools*, 2005.

[19] V. John-Steiner. *Creative Collaboration.* Oxford University Press, Oxford, 2000.

[20] L. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of AI Research*, 4:237–285, 1996.

[21] S. Ketchpel. Forming coalitions in the face of uncertain rewards. In *Twelfth National Conference on Artificial Intelligence*, pages 414–419. MIT Press/AAAI Press, 1994.

[22] I. Kwan, D. Damian, and M.-A. Storey. Visualizing a requirements-centered social network to maintain awareness within development teams. In *1st Int'l Workshop on Requiremetns Engineering Visualization (REV'06)*, 2006.

[23] V. Lesser, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X. Zhang. Evolution of the gpgp/taems domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems*, 9(1):87–143, 2004.

[24] Microsoft OneNote. http://office.microsoft.com/en-us/onenote/.

[25] MindManager. http://www.mindjet.com/products/mindmanager/.

[26] K. Nakakoji. Seven issues for creativity support tool researchers, 2005.

[27] J. S. Olson and G. M. Olson. Culture surprises in remote software development teams. *ACM Queue*, 1(9):52–59, 2003.

[28] PodBrainer. http://www.solutionpeople.com/podbrainer/.

[29] M. Resnick. Design principles for tools to support creative thinking, 2005.

[30] P. N. Robillard and P. Kruchten. *Software Engineering Processes: With the UPEDU.* Addison-Wesley, 2002.

[31] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, MA, 1986.

[32] O. Shehory and S. Kraus. A kernel-oriented model for coalition-formation in general environments: Implementation and results. In *Proceedings of the Thirteent National Conference on Artificial Intelligence*, pages 133–140, August 1996.

[33] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, Cambridge, MA, 2009.

[34] R. Sternberg. *Handbook of Creativity.* Cambridge University Press, New York, 1999.

[35] R. Sternberg. The nature of creativity. *Creativity Research Journal*, 18(1):87–98, 2006.

[36] R. Sternberg and T. Lubart. *Defying the Crowd: Cultivating Creativity in a Culture of Conformity.* Free Press, New York, 1995.

[37] R. Sternberg and T. Lubart. The concept of creativity: Prospects and paradigms. In *Handbook of Creativity.* Cambridge University Press, New York, 1999.

[38] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[39] F. Tohme and T. W. Sandholm. Coalition formation processes with belief revision among bounded rational self-interested agents. *Journal of Logic and Computation*, 9:1–23, 1999.

[40] G. Weiß. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.* The MIT Press, Cambridge, MA, 1999.

[41] M. West. Sparkling fountains of stagnant ponds: An integrative model of creativity and innovation implementation in work groups. *Applied Psychology: An International Review*, 51(3):355–424, 2002.

[42] Curio. http://www.zengobi.com/products/curio/.