AC 2009-2111: IMPROVING LEARNING OUTCOMES USING COGNITIVE MODELS IN SYSTEMS DESIGN

Joanna DeFranco, Pennsylvania State University
Lecturer of Information Science in the School of Graduate Professional Studies, Penn State University.

Colin Neill, Pennsylvania State University
Associate Professor of Software Engineering in the School of Graduate Professional Studies, Penn State University.

Improving Learning Outcomes using Cognitive Models in Systems Design

Abstract

System design courses typically incorporate team projects as both active learning components of courses and for student assessment. Research indicates, however, that actually working within a team generates a new set of problems, referred to as Problem B: managing the diversity of the problem solvers in contrast to Problem A: solving the actual problem the team is working on. Given the presence of Problem B, there is a risk that student learning will actually suffer because of the team. To mitigate this risk, we propose the use of the Cognitive Collaborative Model (CCM) in team system design exercises.

The CCM is a six-stage cognitive model that takes into consideration the cognitive and social activities that occur during collaborative problem solving by facilitating problem formulation, solution planning, and system design tasks during collaboration. The model was initially developed to improve the effectiveness of engineers performing such tasks in teams and this study posits that the model will be equally effective on student learning.

A detailed statistical experiment to study the effect of this model on subjects collaboratively solving an analysis and design problem was designed and executed. Randomly assigned teams of students were presented with one of four design problems. Half the teams were then exposed to the CCM while the remaining half was not. The effect on student learning was then measured using assessment of team deliverables.

1. Introduction

In a brief survey of the graduate engineering and technology courses offered at our campus it was determined that over 50% incorporated at least one student team assignment and for the two practicum capstones in the engineering programs the entire student assessment rested upon a collaborative effort. In a survey of instructors at eight engineering schools Felder[6] found that 24% always assigned a group project while another 52% assigned them in some courses. While there are pragmatic reasons for such teams (reduced grading load) their use is grounded in the reality of the engineering profession: the vast majority of graduates will spend their professional lives working in teams. Furthermore, there is evidence that if the team forms a cooperative learning group, the learning of the individual team members is enhanced[9,12].

Despite the obvious benefits, however, many students resist team projects citing concerns that: they have little influence and no control over their team-mates; they believe their grade will not reflect their contribution or competence; and the transaction cost of scheduling meetings, and working collaboratively are not worth the rewards, of which they see few[3]. These bad team experiences can have a profound impact on those students who are subsequently soured on teamwork far beyond their education studies and in to the workplace[2].

While there will always be group and interpersonal dynamic issues that we cannot overcome, we contend in this paper that a significant impediment to team success, team and individual learning, and ultimately student satisfaction is the diversity of problem solving styles that exists within any team, and the dissonance that this diversity creates with respect to the member's individual mental models. As Descartes said "we do not describe the world we see, we see the world we can describe." Since each team member has a different world view, different experiences, and a different way of thinking, each has a different mental model of the problem at hand and its solution. Without a consistent shared mental model team members struggle not only with the problem at hand, but also with the disparity between their individual perceptions of that problem. In order to overcome this impediment each individual's mental model must be challenged (their assumptions, misconceptions, etc.) in order for a new shared mental model to form - a process referred to as reflexive thinking, or double-loop learning[1].

In this paper we present a collaborative problem solving model, the Cognitive Collaborative Model (CCM), that takes into consideration the problem solving cognitive processes of a collaborative group and that addresses the psychological and sociological factors in teamwork. This model facilitates reflexive thinking by revealing the implicit mental models within the group thus mitigating the effects of Problem B. We tested the model in two engineering classes: a software development course and a system design course. In both cases the key objectives for the teams were to conceptualize the problem and determine a solution strategy. The experiments revealed that the CCM does indeed improve team effectiveness and student learning.

2. Background

Collaboration and problem solving are essential for designing, developing, and implementing effective system solutions. Controlled experimental studies indicate it is worthwhile to integrate collaborative activities even at the early stages of problem solving and system design/ development training[16]. An experiment with experienced software engineers[14] also demonstrated that collaboration improves the problem solving process.

The act of collaboration brings its own set of problems, however. Whenever a team of individuals comes together to tackle a problem, they are actually faced with two problems: problem A, the one they came together to solve; and problem B, the diversity in their mental models[13]. To be effective the team must expend more energy on problem A than problem B. The teams will typically manage this by introducing structure to their thinking, their interactions, and their communications. Unfortunately, this structure is simultaneously enabling (in that it provides a framework for cognition, for example in a taxonomy), and limiting (in that it constrains the available options, for example when we experience a paradigm shift). Kirton[13] calls this the paradox of structure and we must ensure that any collaborative model does not fall victim to it by ensuring that it is sufficiently flexible to allow each team member to express their thinking, while sufficiently predictable so that each member can structure their understanding.

Argyris examined problem solving, and more generally, organizational learning, in a similar vein. If an individual's mental model is too rigid, he will never challenge it, and thus will never learn in new situations. Argyris termed this "single-loop" learning in that each time a problem is encountered, the current mental model is unconsciously accepted and the same solutions are applied. In contrast, "double-loop" learning occurs when, in the face of new situation, the prevailing mental model is challenged (deep-rooted assumptions and norms are surfaced and questioned) and potentially altered before a solution is sought[1].

The CCM forces double-loop learning by systematically exposing each team member's own mental model. As the individual mental models are made explicit they can be interrogated and challenged until a shared mental model emerges.

In addition to sharing a mental model of the problem, collaborative groups appear able to deal with complex tasks more effectively than individuals, partly because groups automatically have a broader range of skills and abilities than individuals[7]. Despite this fact, studies indicate that group problem solving is intrinsically more complex than individual problem solving[7]. It can introduce difficulties that are specifically group-related, such as an interaction environment that inhibits the free expression of ideas[10], conflicts caused by interpersonal difficulties, or as mentioned above, complications arising from the paradox of structure and the cognitive diversity of the group. Overall, however, the benefits of collaboration in problem solving far outweigh its disadvantages[11]. One notable benefit is the improvement of human capital affected by collaboration, because the individuals involved in a group learn from the skills and abilities of the other group members[15]. The need to articulate designs, critiques, and arguments to other group members also sharpens an individual's technical, critical, and interpersonal skills[8]. If a group is properly coordinated, as when using a model or framework such as the CCM to design a system, the communication and collaboration of a group is enhanced[5].

3. Theory

The main goal of the CCM is to assist in facilitating critical thinking and effective problem solving among the collaborators. The CCM described briefly in this paper is made up of six stages: Problem Formulation, Solution Planning, Solution Design, Solution Translation, Solution Testing, and Solution Delivery. Each stage is further broken down into three phases. Each phase is a complete sub-process encompassing each of the major collaborative aspects of problem solving and implementation. Such aspects include: collaborative modality (the mode in which one collaborates) and group dynamics, where group dynamics breaks down into collaborative processes, side effects, and administration. Examples of each are shown in Table 1 below:

Table 1: Examples of Collaborative Aspects

Collaborative Aspect	Examples
Collaborative Modality	Asynchronous Discussion board
Collaborative Processes	Idea generation, negotiation, coordination
Collaborative Side Effects (negative and positive)	Eagerness, free riding, consensus building, conflict, cognitive synchronization
Collaborative Administration	Initiate vote for correctness

For the purposes of this paper we will only focus on the details of the first two stages of the CCM: problem formulation and solution planning. The three phases of the problem formulation stage (stage 1) are: *Preliminary Problem Description, Preliminary Mental Model, and Structured Problem Representation*. The goal of this stage is for the team to answer questions and gather information for each team member to understand the problem. In addition, the individuals need to communicate effectively and the group also needs to listen and make sure each member has the correct understanding of the problem.

For example, in the first stage the collaborators are to agree upon a preliminary problem description to make sure each team member has the same understanding of the problem. The model guides each team member to create a description in their own words and share it to each team member. Each team member discusses and votes to determine one problem description. Next, the team is charged with answering questions to develop a preliminary mental model. For example, the questions help the team to discus and determine givens, unknowns, conditions and constraints on the problem. The final part of this stage is for the team to identify and organize any relevant information to the problem thus creating a knowledge base from which the team will begin their solution plan (stage 2). The team organizes the information from the preliminary mental model by using the suggested format.

The three phases of the second stage of the CCM, solution planning, are: *Strategy Discovery, Goal Decomposition, and Data Modeling*. The goal for this stage of the CCM is for the team to answer questions and gather information to plan a solution. Specifically, they are going through the process of goal decomposition where they are refining goals into smaller sub-goals that are more easily solved. In addition, this stage provides a scheme to organize related data is discussed by the team.

For example, in the initial phase of stage two the team is beginning to strategize solution alternatives. The model provides questions for each team member to critically think about the solution alternatives. The team members then share their solution possibilities with each other. A vote commences to determine the solution that will be followed. Now that the team has agreed upon a solution it can be broken down into sub goals and distributed among the team members. The final phase of stage two is to yield a preliminary data model. This is accomplished by each team member associating their sub goal with the facts from the output of stage one. Specifically, they are integrating the givens and unknowns from stage one with the refined goals of stage two.

In the remaining four stages of the CCM the team would be translating the plan into a design, implementing the design, testing, and finally delivering the solution. Working through the first two stages of the CCM, the team is able to conceptualize the problem resulting in a more effective plan and in theory implementing a better solution.

4. Hypothesis, Methodology, and Results

This paper describes a multifaceted study that at this point consists of two complete experiments. Both experiments evaluated subjects as they conceptualized system design problems. The main difference in the two experiments was the subject types involved. The results from both experiments were similar thus the second experiment supported the results of the first experiment. In the first experiment, detailed in[4], the subjects were novice computer science students. The second experiment consisted of subjects who were experienced software engineers learning about system design.

The first experiment involved 44, three and four person teams. Half the teams were given the CCM and the other half were not. The students were all given the same assignment, a supermarket simulation, and their performance was assessed by two expert judges who were blind to the experimental conditions. The judges were evaluated to determine if they were trained properly and to determine the reliability and validity of their evaluation of the dependent variables. An inter-rater reliability check was performed with a bivariate Pearson 2-tailed test. It was found that there was a significant correlation at the .01 level between the two judges ($r=.932$, $P<.01$).   The judges were given two group output documents to evaluate: The Problem Understanding Document and the Solution Plan Document.

In assessing the Problem Formulation document the judges evaluated problem understanding on a 10-point scale, where 10 was the best. They were given a rubric that indicated groups were to score 6 points if they understood and explained the simulation where the user would input customer frequency, number of stockers, number of cashiers, and output statistics from the different objects defined (these were facts given in the initial problem description). If the judges found indication that the team accurately understood the given information, understood the solution goal with a clear and correctly stated problem description, and were able to extract facts out of the description given, they were to use their expert judgment and award a score of up to 10 points. With the Solution Plan document the judges evaluated the planning skills awarding a minimum of 7 points if the team created and distributed appropriate subgoals. They were to award up to 10 points depending on the detail of the plan, subgoals, and schedule.

The results of the study are summarized in Table 2. For both dependent variables the p value is at the .01 level for subjects having access to the model supporting the hypothesis that teams that had access to the CCM had higher problem understanding and planning skills.

Table 2: Results of CCM in experiment 1

Task	**Model**	**No Model**	**ALL**
Problem Understanding F= 6.196 p=0.017	Mean: 6.75 σ: 2.099	Mean: 4.81 σ: 3.00	Mean: 5.78
Solution Plan F=8.172 p=0.007	Mean: 6.82 σ: 1.83	Mean: 4.77 σ: 2.86	Mean: 5.795

In the second experiment, the model was tested with experienced engineers in a traditional graduate software engineering course at Anonymous University learning systems design. Our goal/hypothesis was to determine if skill development would be enhanced by teams using the CCM. The subjects participating in the study were 18 graduate software engineering students. The students were randomly assigned to three-person teams where they were given one of four equivalent assignments. Again, half of the teams were provided the CCM and the project was evaluated with five progress checkpoints. Their performance was assessed by one of the authors, who was the instructor for the course. Their score was calculated by assessing the completeness of the project solution as depicted by the design contracts, sequence diagrams, and the design class diagram of their project.

The results of the experiment are summarized in Table 3. With a p value at the .01 level for subjects having access to the model the hypothesis that teams that had access to the CCM performed better is supported.

Table 3: Results of CCM in experiment 2

Task	**Model**	**No Model**	**ALL**
Solution strategy F=19.98 P=0.00	Mean: 98.13 σ: 3.47	Mean: 89.0 σ: 5.16	Mean: 93.57

5. Conclusions and Future Work

Students are frequently expected to complete team assignments in systems design courses. The intent of the exercise is not to demonstrate the difficulties of teamwork, nor to assess the student's abilities in working collaboratively, but both issues arise and can leave a sour taste with the student and inhibit their learning of the course content. Several factors contribute to these unfavorable outcomes, but in particular it is the difficulty associated with disparate mental models of the problem at hand and the solution needed.

In this paper we have utilized a process, the Cognitive Collaborative Model, that seeks to make explicit the prevailing mental models of the team members such that those models can be interrogated, deep-rooted assumptions can be exposed, and a shared mental model across the team can be formed. To test the process two experiments were conducted where the team's ability to conceptualize the problem, plan system development, and form a solution strategy were assessed when exposed to the CCM. In each case the results indicate that there is a significant improvement when the model is used.

At this point we have confirmed that the model does facilitate greater competency in the assigned tasks, but we have yet to determine if overall improvement in learning has occurred. This is the objective of our future research where we intend to apply pedagogical theory in measuring learning outcomes through pre- and post-testing of students so that we can isolate the success of the team from the learning of the individual.

6. References

1. Argyris, C., "Teaching Smart People How to Learn," *Reflections: The SoL Journal*, Vol.4, No. 2, Dec. 2002, pp. 4-15.
2. Buckenmeyer, J.A. "Using teams for class activities: Making course/classroom teams work," *Journal of Education for Business*, Vol. 76, No. 2, Nov. 2000, pp. 98-108.
3. Caspersz, D.M., Wu, M., Skene, J. "Factors Influencing Effective Performance of University Student Teams," in *Proc. 26th Annual International HERDSA Conference*, Christchurch, NZ, pp. N/A CD Rom.
4. DeFranco, J., "A Study of Collaborative Software Development Using Groupware Tools," *Proc. Int. Conf. on Information Technology: Research and Education*, 2003 *(ITRE2003)*. Newark, NJ, 11-13 Aug. 2003, pp. 440-444.
5. Ellis, C., Gibbs, S., Rein, G., "Groupware Some Issues and Experiences," *Communications of the ACM*, Vol.34, No. 3, Jan. 1991, pp. 39-58.
6. Felder, R.M., Brent, R., Miller, T.K., Brawner, C.E. and Allen, R.H. "Faculty teaching practices and perceptions of institutional attitudes toward teaching at eight engineering schools," in *Proc. 1998 FIE Conf.*, Tempe, AZ, 1998, pp. 101–105.
7. Finnegan, P., O'Mahony, L., "Group Problem Solving and Decision Making: an Investigation of the Process and the Supporting Technology," *Journal of Information Technology*, Vol. 11, No. 3, Sept.1996, pp. 211-221,
8. Guzdial, M., Kolodner, J., Hmelo, C., Narayanan, H., Carlson, D., Rappin, N., Hubscher, R., Turns, J., Newstetter, W., "Computer Support for the Learning through Complex Problem Solving," *Communications of the ACM*, Vol. 39, No. 4, Apr. 1996, pp. 43-45.
9. Hilborn, R.B. "Team learning for engineering students," *IEEE Trans. Educ.*, vol. 37, no. 2, 1994, pp. 207–211.
10. Hoffman, L. R., "Group Problem Solving" in Advances in Experimental Social Psychology, Leonard Berkowitz (Ed.), Academic Press, New York, 1965.

11. Hohmann, L., Journey of the Software Professional, Prentice Hall PTR, New Jersey, 1997.

12. Johnson, D.W., Johnson R.T., and Smith, K.A. *Cooperative Learning: Increasing College Faculty Instructional Productivity: ASHEERIC Higher Education Report No. 4*. Washington, D.C.: The George Washington University, School of Education and Human Development, 1991.

13. Kirton, M., Adaption-Innovation: In the context of Diversity and Change, Routledge 2006.

14. Nosek, J. "The Case for Collaborative Programming," *Communications of the ACM*, Vol. 41, No. 3, March 1998, pp. 105-108.

15. Prey, J.C., "Cooperative learning and closed laboratories in an undergraduate Computer Science curriculum," *Proc. 1st Int. Conf. on Integrating Technology into Computer Science Education*, Barcelona, Spain, June 1996, pp. 23-24.

16. Sabin, R. E., Sabin, E., "Collaborative Learning in an Introductory Computer Science Course," ACM *SIGCSE Bulletin,* Vol. 26, Iss. 1, March 1994, pp. 304 – 308.