Moo. – A Partner Recommendation System for Effective Collaborative Learning

Tathagata Chakraborti
tchakra2@asu.edu

CSE 591 Technologies for Online Learning
Project Proposal - 10/10/2017

ABSTRACT

In this project, we present Moo. – a partner recommendation system for online learning communities. The software is aimed at identifying potential groups of students who can participate in a joint curriculum, based on their existing skills sets and their target skills. This is useful both from the point of view of promoting social learning as a community but also in reducing the overhead of forming such communities.

One of the key technologies required to facilitate the social learning paradigm is Intelligent Tutoring Systems (ITSs) that can support such dynamic community formation based on the needs of the individual students in a class. To this end, we aim to integrate techniques from the AI planning community towards making ITSs more personalized towards the needs of individual students and more responsive to the contents of a particular course, especially geared towards making the learning experience more social. This is based on the idea of generalized models of the student and the course, which can be used to reason over by an AI engine for optimizing curriculum dynamically based on the skill sets and performance of the students in a class. We will demonstrate how this can help in adapting a course curriculum based on the skills of an individual student as well as in bringing together students with differing skills to facilitate collaborative learning.

Author Keywords

Intelligent Tutoring Systems; Learning 2.0, Collaborative Learning; Automated Planning; Explanations as Model Reconciliation, Required Concurrency

MOTIVATION

While the last decade has seen massive advances in technologies aimed at creation and dissemination of knowledge across a variety of platforms, concerns remain as to how effectively this knowledge is absorbed at the user (student) end. This is especially true for both massive open online courses (MOOCs) or for (rapidly growing sizes of) physical classrooms where targeted attention towards individual students is often hard to attain. One of the many advantages of these social platforms for learning is peer feedback and participation that can allow for knowledge advancement as a community.

Too much Collaboration, Too little Learning

However, forming study partners remains an arduous task, especially in large classrooms such as in online learning communities where students usually do not know most of their classmates (or their skill sets). Without principled drivers for building in-class communities that can promote learning, effective collaborations are hard to achieve. As such, the issue of forming useful teams of collaborative study can become a problem by itself rather than a facilitator for learning to the extend that students are either spending too much effort in just building or maintaining teams, or are just reduced to studying by themselves, thus leaving the potential benefits of being in a social learning environment mostly untapped.

ITS as a Critical Component of Online Learning

The prospect of intelligent tutoring systems (ITS) that can provide such personalized support, and bring in expert (human) tutors in the loop where necessary, is especially fascinating. However, much of the existing work on ITS has focused on specific learning platforms or courses without any coherent discussion of the general principles of design and implementation of the roles usually attributed to ITSs. The aim of this project is to introduce techniques from the planning community that can formalize some of these concepts and provide a framework for building such systems from the ground up. This has useful implications for both the planning as well as the educational technologies communities – i.e. well-explored areas of planning can provide solutions to existing problems in ITSs (as we will demonstrate in this project) while the perspective of the learning community can provide useful feedback towards the refinement of said techniques, including defining new areas of research of mutual interest.

Specifically, for this project, we will explore the following four concepts as it relates to the social learning paradigm –

(C1) How the designers of coursework (e.g. the instructor) can use an interface to define the high level features of a course using a declarative language that can be utilized internally by the AI engine to perform tasks (2) and (3) below;

(C2) How given a particular problem / assignment, and the ITS's estimated model of the student, the planner can auto-generate a personalized and *optimized* curriculum (e.g. sequence of tutorials) for the student (which might involve suitable study partners, c.f. C4);

(C3) How the planner can participate actively as a student solves an assignment by providing helpful guidance via tips and hint (without solving the problem itself); and finally

(C4) How techniques (2) and (3) come together in bringing together students in the class with diverse skills into *collaborative* problem solving (information processing) and learning (knowledge advancement).

The biggest advantage of such an approach is that the techniques are defined at the procedural level and can be grounded with the description of a particular course as needed.[1]

Proposal Outline

Moo. as a platform thus has two major components – the frontend that makes the social learning paradigm easily accessible to students and the backend which makes the collaborative learning process effective. We will describe this in more detail after a brief overview of related work and basic taxonomy.

RELATED WORK

In the following section, we provide a very brief overview of the related work in the field of learning technologies and provide quick pointers to techniques in the automated planning community that are relevant to the project.

Learning 2.0

The world of learning is changing fast - information can now be provided across a variety of platforms to large groups of people who can access *on demand* knowledge and participate in the learning process as a *community*. This is the Learning 2.0 paradigm [35], and requires a rethink of the affordances [25] expected off current learning tools.

Collaborative Learning.

This involves two critical aspects – *knowledge building as a community* [33] and *information processing* [41] on the part of the individual student as a member of that community. We will see how different parts of the project aim to adapt ITS functionalities to these Learning 2.0 principles.

Intelligent Tutoring Systems and AI

ITSs are extremely useful for providing directed feedback to students and can be as effective as a human teacher when designed properly [39]. A thorough description of the different components of ITSs can be found in [38]. Existing applications of such systems range from solving numerical problems like Andes [11] which can help in teaching basic laws of physics [34], Dragoon [40], Q&A type problems as in Autotutor [13] or for an SQL tutor [26]. ITSs, of course, go beyond individual information processing stage and find uses in knowledge building as a community [21] as well, thereby embracing the principles of the Learning 2.0 paradigm.

Student Assessment Models

One of the most important capabilities an ITS needs to have is to be able to estimate the (mental) model or capabilities of the student. This has been explored in the context of the (1) *item response theory (IRT)* [14] which treats learning and testing as separate processes and the (2) *Bayesian knowledge tracing (BKT) theory* [9] which considers a more dynamic model of the student state. The latter becomes more relevant in the context of ITSs that can provide more dynamic feedback and hints as discussed next. Indeed this is an issue where AI techniques have been deployed before for dynamic modeling of the evolution of the student model in terms of knowledge components, concentration / focus levels, etc. [28]. This includes different techniques such as *decision theoretic approaches* (i.e. Markov Decision Processes or MDPs) [28, 27], and *reinforcement learning* [7, 23, 22]. The project assumes for the most part[2] that these techniques are available and builds on top of that assumption, i.e. being able to estimate the student model is necessary for ITS techniques and we want to demonstrate, from the perspective of automated planning[3] how this can be exploited to provide a better learning experience to a student.

Feedbacks and Hints

Once the ITS has estimated a model of the student, it can provide targeted feedback to improve the learning process. Existing work in this area [1, 37, 31, 32] has largely focused on ITSs operating as *recommender systems*. The project is largely situated in this space but aimed at providing much more sophisticated feedback in both the inner and outer loops [38] of an ITS which requires longer-term sequential reasoning at higher levels of abstraction, and can be generalized across different learning tasks at the procedural level.

Automated Planning

Automated planning, as a field, has been around ever since the inception of AI, and is considered a necessary ability of any autonomous system - the ability to reason about and decide on a course of action (CoA) given the current state of the world. However, in the context of operating with humans in the loop, traditional methods of AI planning are not sufficient. A planner in the course of it's deliberations must be able to take into account the (mental) model of the user. Much of my thesis topic [4] has revolved around this theme of how monolithic planning techniques evolve to accommodate the human in the loop in the context of different applications. Presently, we will briefly introduce three specific concepts that we are going to use in the project.

Forward Planning

Forward planning, or just planning [15] in the traditional sense of the word, involves computing a sequence of steps or a plan given a domain (physics or rules of the domain) and a problem (instance of a particular situation in that domain). However, planners in the context of ITSs must go beyond traditional tasks of *plan generation* towards helping with action or plan *recommendations*, plan *completion*, plan *validation*, *excuse and explanation generation*, and so on for proactive support directed at the student. I have explored some of these principles before in [36]. Section on technical contributions later will describe briefly how these techniques may be adopted for the smart design of ITSs.

[1] Note that the scope of the project only includes the actual learning and interaction phase and does not address post-hoc reflection and evaluations as in [19, 18, 8].

[2] In fact, the explanation generation technique discussed later in the paper can be modified to function as an estimator for the student model but this is outside the scope of the proposal.

[3] It is possible, albeit computationally infeasible, to model the whole environment in decision theoretic terms. This discussion is, again, out of scope for the proposal, and will be included in the final report.

Explanations as Model Reconciliation

I have recently worked on an algorithm aimed at generating *explanations* [5] for solutions (or plans) to planning problems in the event when the description of the planning problems available to the planner and the human in the loop differ. The process of explanations in such situations become one of *model reconciliation* whereby the planner, by providing updates or corrections to the human's mental model, brings the human on the same page with regards to a particular planning problem in question. We will see later how we can use this technique to generate an optimal curriculum for a given student model.

Multi-Agent Planning and Required Cooperation

Multi-agent planning (MAP) [3], as the name suggests, is a variation of the classical planning problem but with multiple agents involved in the execution process. In [42] the authors demonstrated the concept of *required cooperation (RC)* that defines under what circumstances the solution of a multi-agent planning problem demands the participation of more than one agent as a necessary component of the final plan (e.g. agents having complementary capabilities necessary to achieve a goal). We will draw upon these concepts to generate curricula that can leverage diverse skills of different students in the class so as to encourage collaborative learning as a community.

BASIC TERMINOLOGY

In the following section, we will introduce informal descriptions of some basic terms that we will be reusing in the course of the rest of the discussion. The exact formulation of the ITS setting (and proposed approaches in the next section) in the planning framework will be provided in Deliverable 1.

Assignments

A particular class is defined in terms of a set of assignments $\{A_i\}$ that test the understanding of the student at regular intervals during the learning process. Technically, these are sensing actions for the ITS in determining the knowledge state of the student. Thus, an assignment may be used both as a way of estimating the student model as well as a technique for indirectly imparting knowledge to the student.

Knowledge Components

Knowledge acquisition by a student can be decomposed into smaller components that are referred to as knowledge components (KCs) [20]. KCs can be anything from a production rule [24], to a facet, misconception, fact or even a skill [2]. The aim of the learning process is to make a student acquire different KCs based on their and their classmates already existing ones.

Tutorials

The class also constitutes of a set of tutorials $\{T_i\}$ that directly modify the student's knowledge state by providing information on specific topics or on how certain problems or (parts of) assignments may be solved. An ITS can use these to define a curriculum for solving a set of assignments.

Student Model

The student's knowledge state or *model* is defined in terms of KCs. As described before, the tutorials may be used to affect or advance the student model while the assignments may be used to sense the current state of the student model.

Moo. – THE FRONTEND

The interface for the proposed platform is shown in Figure 1. A preliminary implementation of the interface can be accessed at[4] – `https://goo.gl/cYzg6S`. The interface is build on Bootstrap (so as to provide seamless responsive design on mobile devices as well) with support from standard JS. Software modules to connections at the backend are yet to be determined.

The interface has the following primary components –

Quicklinks

These allow the student to quickly browse through the interface as well as to external resources like the ASU Blackboard (where the study materials like assignments and tutorials will be hosted) as well as other links associated with the topic of study. This panel also includes an embedded chat window (Hangouts) for a group to talk among themselves directly. The chat functionality is yet to be implemented.

Your TODOs

The TODO Panel (see Figure 2) includes information pertaining to an individual student's curriculum. This may include information related to particular assignments or tutorials (e.g. due date, estimated duration, prerequisites) as well as information pertaining to the social learning paradigm (e.g. prerequisites and study partners). Prerequisites may be used externally by the student to look for potential study partners or internally by Moo. to recommend the same based on its knowledge of the classroom. Notice that the personalized curriculum generated for a student has other students participating in it, in what will hopefully provide directed support and incentive for knowledge assimilation at the individual level through participation at the community level. Further, the TODO panel also provides external endpoints to *Blackboard where the actual tutorials or assignments are hosted.*

Your Team

The Team Panel (see Figure 3) provides quick access to the status / progress of fellow students partners as well as a chat feature to engage with them directly. The assignments allocated to the individual may involve participation of one or more students from this group. Note that the progress information is only available to other students based on whether that particular student chose to make it publicly viewable or only viewable to their group members or neither.

Your Class

The Class Panel (see Figure 4) provides a view of the rest of the class (and thus help a student identify potential study partners). As before, the information regarding the status / progress of a classmate is shown to help a student make this decision. An invite button also lets the student reach out to a potential partner. Again, the information is displayed only if a student in the class allows it to be externalized. Internally, Moo. uses student information from across the class to make recommendations of potential study partners, as well as use this information to dynamically generate the curriculum of an individual student as seen in the TODO Panel.

[4]This is meant to serve as an illustration only, has not been stress tested on different screen sizes or browsers.

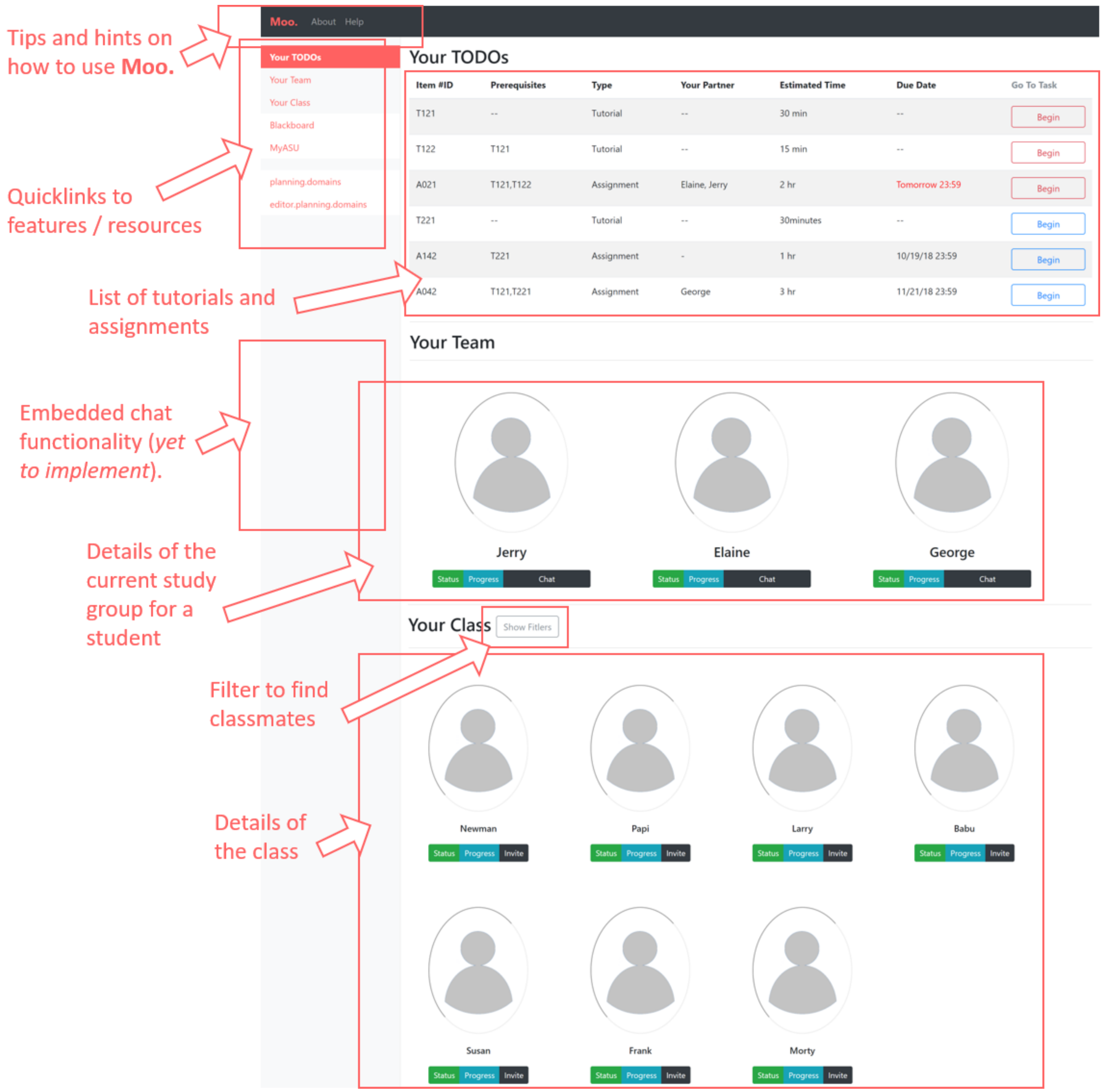

Figure 1. **The** Moo. **interface. A preliminary implementation can be accessed at** `https://goo.gl/cYzg6S`.

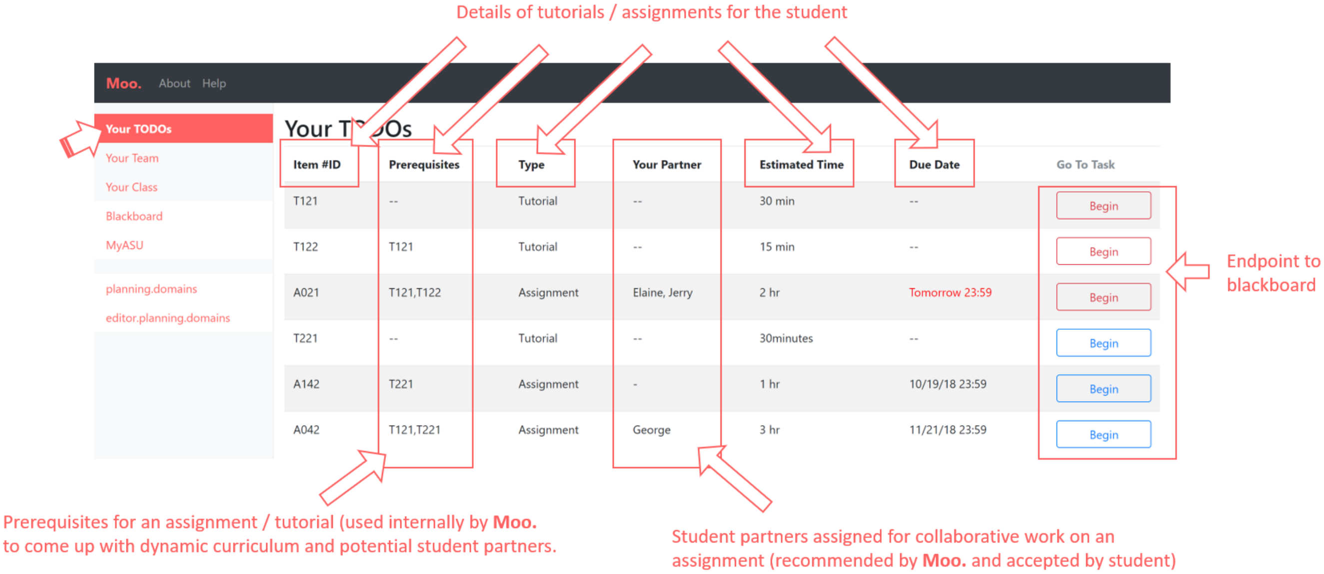

Figure 2. TODO panel outlining a dynamic personalized curriculum to a student based on progress / status of their classmates.

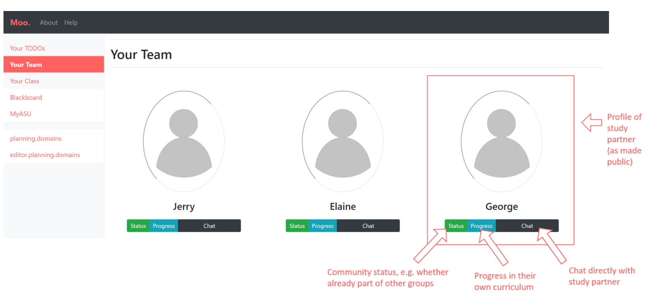

Figure 3. Information Panel allowing individual students to measure their progress with respect to other students in their group as well as correspond directly with them with regards to their joint curriculum.

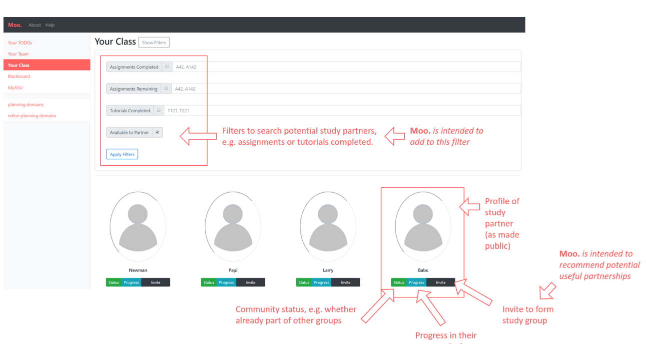

Figure 4. Information Panel for other classmates (who might be potential study partners). A set of filters provide ways for the student to search for other students with *a particular set of skills*.

Moo. – THE BACKEND

In the planning framework of an ITS, each assignment is represented as a planning problem, while each tutorial serves as a model change action. The student model, constructed out of KCs, is an instance of the domain description (thus, each assignment is an instance of a planning problem in this domain). Formal definitions are excluded due to lack of space. Instead we will provide an informal outline of proposed approaches below. Let us now go back to the contributions outlined before in the introduction and see how these come together in light of the concepts introduced above.

(C1) Planning problems are inherently described in terms of declarative rules, which will be internalized. The instructor creating the contents for the class will use an interface to describe $\{A_i\}$ and $\{T_i\}$ in terms of KCs. This will be automatically compiled into rules usable by the planner in the backend, similar to [6].

(C2) The student-tutor interaction provides an ideal venue for model reconciliation as introduced in [5]. Here the tutor's model includes the correct KCs and is thus the ground truth, while the student's model is an incomplete or incorrect version of this model. Thus the process of multi-model explanations for planning problems [5] can be compiled into an approach for automated curriculum generation (given an estimation of the student model) where the tutorials act as model change.

(C3) As mentioned before, a planner's role evolves when operating as a decision support for a *human planner* (here, the student). For example, plan validation and failure prediction [10, 17] requires computation of *landmarks* [16] that are essential to achieve a goal (thereby providing a concise summarization of the task at hand as well as identify potential pitfalls), validate plans under construction and alert [12, 5] on possible failures, bottlenecks or oversights on the user's part which might jeopardize the plan going forward. Further, before the agent can assist the user, it must interpret their actions and recognize their intentions. This makes the planning and plan recognition [29, 30] processes interdependent, and a crucial component of a planning backend for an ITS. In [36] I have explored these techniques in the context of decision support, and many of the same techniques carry over for providing tips and hints (feedback) to a student solving a particular assignment (represented in the form of a planning problem).

(C4) Finally, once the ITS setting has been formalized in the planning setting, it makes itself amenable to established techniques in the planning community to realize a spectrum of capabilities of an ITS. Specifically, we exploit notions of required cooperation or RC [42] to bring together students in the class for collaborative learning. This is done by reasoning over individual student models, and how these can be potentially be aggregated to solve joint assignments. We will demonstrate how the computation of such effective pairings is itself a MAP problem.

EVALUATION

The evaluation of the proposed platform is planned in two stages (respective timelines are mentioned below) –

(1) Ablation studies using simulated agents to establish internal properties of the algorithms in the backend; and

(2) User studies measuring the learning experience of participating students in different conditions with or without the planner in the backend (see below).

Task (2) is going to use the **Moo.** interface to establish different classroom conditions for the students to evaluate the effectiveness of the proposed community building techniques.

C1. Strawman – Here students work on their own;

C2. Baseline – Here students are allowed to seek out fellow classmates and form groups by themselves; and

C3. Proposed – The final condition allows active recommendations of partners and dynamic curriculum generation for individual students using the proposed techniques.

The **Moo.** interface to the individual students, including all relevant information regarding the progress and skill sets of their classmates (as shown in Figure 1) *remains the same across all three conditions* in order to remove latent factors during the evaluation process.

The evaluations are aimed towards quantifying the effectiveness of the proposed approach not towards facilitating formation of ad-hoc collaborations in an online classroom, but rather to show that the learning process has been improved as a result of it. This can be done through a set of objective as well as subjective measures, such as

(1) Objective Measures – Time taken to finish the tasks, acquired expertise of individual students, etc.;

(2) Subjective measures – This will be in the form of a Likert-scale evaluation of how students felt about the social learning process as presented via **Moo.**; and

The acquired expertise of individual students is an especially important marker for the success of the entire curriculum (not just as a means for knowledge assimilation as a group but rather knowledge accommodation at the individual level as well). The exact metrics will become clearer as we move closer to the evaluation phase.

The learning task is likely to be *"how to use a planner"* as a sort of a meta domain since it is likely to be unfamiliar to most participants (hoping to recruit fellow students in the class for the initial tests :-).

EXPECTED TIMELINE

Oct. 9	Deliverable 1 due – final project outline.
Oct. 31	Finish implementation of backend algorithms.
Nov. 6	Deliverable 2 due – report simulation results.
Nov. 30	Finish interface design and set up user studies.
Dec. 4	Project demos.
Dec. 6	Final projects due.

REFERENCES

1. Tiffany Barnes and John Stamper. 2010. Automatic hint generation for logic proof tutoring using historical data. *Journal of Educational Technology & Society* 13, 1 (2010), 3.
2. Benjamin Samuel Bloom, Committee of College, and University Examiners. 1964. *Taxonomy of educational objectives*. Vol. 2. Longmans, Green New York.
3. Ronen I Brafman and Carmel Domshlak. 2008. From One to Many: Planning for Loosely Coupled Multi-Agent Systems.. In *ICAPS*. 28–35.
4. Tathagata Chakraborti, Subbarao Kambhampati, Matthias Scheutz, and Yu Zhang. 2017a. AI Challenges in Human-Robot Cognitive Teaming. *CoRR* abs/1707.04775 (2017).
5. Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. 2017b. Plan Explanations as Model Reconciliation: Moving Beyond Explanation as Soliloquy. In *IJCAI*.
6. Tathagata Chakraborti, Kartik Talamadupula, Mishal Dholakia, Biplav Srivastava, Jeffrey O. Kephart, and Rachel K.E. Bellamy. 2017c. Mr.Jones – Towards a Proactive Smart Room Orchestrator. In *AAAI Fall Symposium on Human-Agent Groups: Studies, Algorithms and Challenges*.
7. Min Chi, Kurt VanLehn, and Diane Litman. 2010. Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In *Intelligent Tutoring Systems*. Springer, 224–234.
8. John Connelly and Sandra Katz. 2009. Toward more robust learning of physics via reflective dialogue extensions. In *EdMedia: World Conference on Educational Media and Technology*. Association for the Advancement of Computing in Education (AACE), 1946–1951.
9. Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.
10. Maria Fox, Richard Howey, and Derek Long. 2005. Validating Plans in the Context of Processes and Exogenous Events. In *AAAI*.
11. Abigail S Gertner and Kurt VanLehn. 2000. Andes: A coached problem solving environment for physics. In *International conference on intelligent tutoring systems*. Springer, 133–142.
12. Moritz Göbelbecker, Thomas Keller, Patrick Eyerich, Michael Brenner, and Bernhard Nebel. 2010. Coming Up With Good Excuses: What to do When no Plan Can be Found. *Cognitive Robotics* (2010).
13. Arthur C Graesser, Patrick Chipman, Brian C Haynes, and Andrew Olney. 2005. AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education* 48, 4 (2005), 612–618.

14. Ronald K Hambleton, Hariharan Swaminathan, and H Jane Rogers. 1991. *Fundamentals of item response theory*. Vol. 2. Sage.

15. Malte Helmert. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26 (2006), 191–246.

16. Jörg Hoffmann, Julie Porteous, and Laura Sebastia. 2004. Ordered Landmarks in Planning. *JAIR* (2004). http://dl.acm.org/citation.cfm?id=1622487.1622495

17. R. Howey, D. Long, and M. Fox. 2004. VAL: automatic plan validation, continuous effects and mixed initiative planning using PDDL, In 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004). *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)* (2004), 294–301. http://strathprints.strath.ac.uk/2550/

18. Sandra Katz, David Allbritton, and John Connelly. 2003. Going beyond the problem given: How human tutors use post-solution discussions to support transfer. *International Journal of Artificial Intelligence in Education* 13, 1 (2003), 79–116.

19. Sandra Katz, Gabriel O'Donnell, and Heather Kay. 2000. An approach to analyzing the role and structure of reflective dialogue. *International Journal of Artificial Intelligence in Education (IJAIED)* 11 (2000), 320–343.

20. Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. 2010. The knowledge-learning-instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. *Cognitive Science* (2010).

21. Ioannis Magnisalis, Stavros Demetriadis, and Anastasios Karakostas. 2011. Adaptive and intelligent systems for collaborative learning support: A review of the field. *IEEE transactions on Learning Technologies* 4, 1 (2011), 5–20.

22. Travis Mandel. 2017. Refraction: Teaching Fractions through Gameplay. https://goo.gl/BrPpmV. (2017).

23. Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. 2014. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. IFAAMAS, 1077–1084.

24. Richard E Mayer. 1981. Frequency norms and structural analysis of algebra story problems into families, categories, and templates. *Instructional Science* 10, 2 (1981), 135–175.

25. Catherine McLoughlin and Mark JW Lee. 2007. Social software and participatory learning: Pedagogical choices with technology affordances in the Web 2.0 era. In *Proceedings of ICT: Providing choices for learners and learning*. 664–675.

26. Antonija Mitrovic. 2003. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education* 13, 2-4 (2003), 173–197.

27. R Murray and Kurt VanLehn. 2006. A comparison of decision-theoretic, fixed-policy and random tutorial action selection. In *Intelligent Tutoring Systems*. Springer, 114–123.

28. R Charles Murray, Kurt VanLehn, and Jack Mostow. 2004. Looking ahead to select tutorial actions: A decision-theoretic approach. *International Journal of Artificial Intelligence in Education* 14, 3, 4 (2004), 235–278.

29. Miquel Ramírez and Hector Geffner. 2009. Plan Recognition As Planning. In *IJCAI*.

30. Miquel Ramírez and Hector Geffner. 2010. Probabilistic Plan Recognition Using Off-the-shelf Classical Planners. In *AAAI*.

31. Kelly Rivers and Kenneth R Koedinger. 2013. Automatic generation of programming feedback: A data-driven approach. In *The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)*, Vol. 50.

32. Kelly Rivers and Kenneth R Koedinger. 2017. Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *International Journal of Artificial Intelligence in Education* 27, 1 (2017), 37–64.

33. Marlene Scardamalia and Carl Bereiter. 2006. Knowledge Building: Theory, Pedagogy, and Technology. *The Cambridge Handbook of the Learning Sciences* (2006), 97–118.

34. Kay G Schulze, Robert N Shelby, Donald J Treacy, MC Wintersgill, Kurt VanLehn, and A Gertner. 2000. Andes: An active learning, intelligent tutoring system for Newtonian Physics. *THEMES in Education* 1, 2 (2000), 115–136.

35. John Seely Brown and RP Adler. 2008. Open education, the long tail, and learning 2.0. *Educause review* 43, 1 (2008), 16–20.

36. Sailik Sengupta, Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. 2017. RADAR - A Proactive Decision Support System for Human-in-the-Loop Planning. In *AAAI 2017 Fall Symposium on Human-Agent Groups: Studies, Algorithms and Challenges*.

37. John Stamper, Michael Eagle, Tiffany Barnes, and Marvin Croy. 2013. Experimental evaluation of automatic hint generation for a logic tutor. *International Journal of Artificial Intelligence in Education* 22, 1-2 (2013), 3–17.

38. Kurt Vanlehn. 2006. The behavior of tutoring systems. *International journal of artificial intelligence in education* 16, 3 (2006), 227–265.

39. Kurt VanLehn. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist* 46, 4 (2011), 197–221.

40. Kurt VanLehn, Jon Wetzel, Sachin Grover, and Brett van de Sande. 2017. Learning how to construct models of dynamic systems: An initial evaluation of the Dragoon intelligent tutoring system. *IEEE Transactions on Learning Technologies* (2017).

41. Noreen M Webb. 2013. Information processing approaches to collaborative learning. *Routledge Handbooks Online* (2013).

42. Yu Zhang, Sarath Sreedharan, and Subbarao Kambhampati. 2016. A Formal Analysis of Required Cooperation in Multi-Agent Planning. In *ICAPS*.