Bootstrapping a Socially Intelligent Tutoring Strategy

Jozef Tvarožek*
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
jtvarozek@fiit.stuba.sk

Abstract
We present an approach for computer supported education in the form of a socially intelligent learning environment that is available online. It integrates problem solving and instructional materials into individual and group learning scenarios. A Wizard-of-Oz-driven computer tutor accompanies students to maintain their motivation within the learning environment. The agent can hold off-task conversations and guide students to appropriate learning opportunities. Its tutoring strategy is devised by a reinforcement learning control method that operates on socially motivated state and action spaces induced by the human wizard whose interface facilitates rapid prototyping of relevant states and taking appropriate actions. To make the learning algorithm feasible, states are grouped into equivalence classes according to wizard selected state features, and contextual and linguistic reflection is employed to adjust the immediate action to the current learner's situation. The feasibility study of the socially intelligent agent demonstrated that students who engaged with the agent attained higher learning gains and liked the system more. The bootstrapping of the socially intelligent tutoring strategy was evaluated in simulated student scenarios. Evaluations suggest that our approach for using computers to support students in the learning process is technologically viable.

Categories and Subject Descriptors
K.3.1 [**Computer Uses in Education**]: Collaborative learning, Computer-assisted instruction (CAI); I.2.6 [**Learning**]: Knowledge acquisition

*Recommended by thesis supervisor:
Prof. Mária Bieliková
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on February 10, 2011.

Tvarožek, J. Bootstrapping a Socially Intelligent Tutoring Strategy. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) 33-41

Keywords
intelligent tutoring, social intelligence, reinforcement learning, tutoring strategy, learning environment, motivation

1. Introduction
Learning can be quite time consuming and unexciting. Even with some of today's best computer supported instructional technology, students engage in gaming behaviors associated with less learning [2]. Time spent on task and motivation as key factors of effective learning need to be sustained but contemporary tutoring systems seem to be failing in this respect; all too many students drop out due to low motivation. Can computer tutors build trust and respect with students that would motivate them to learn at all? When used properly, today's computer supported instructional technology can produce significant achievements [8]. How do we make children use them appropriately?

This work aims to provide at least partial answers to these questions through explorations in building and evaluating an online learning environment which features a socially intelligent tutoring agent that tries to sustain students' motivation and help them learn. The tutoring agent engages in off-task conversations with students before and after the instructional activities, manages relationships with students, monitors their (social) behavior, and recommends suitable learning activities. The approach was evaluated by conducting experiments in the domain of middle school mathematics. Evaluations suggest that our approach for using computers to support students in the learning process is useful and technologically viable.

2. Related Work
Tradeoffs between motivating students vs. providing them with actual learning experiences are still researched [5]. Various approaches for improving student's motivation and learning have been proposed: addressing emotional and affective states of students [9], narrative-centered environments with story-based learning [20, 17], and adaptive web-based systems [21, 3]. The affective support seems hard to realize in practice and currently remains limited [4], and since narrative-centered story-based approaches completely alter the way teaching occurs as compared to traditional classrooms or even a typical ITS interface, their use in traditional domains such as mathematics and computer science is not exactly straightforward.

The social context of individual students is important in learning, as putting students together in a collaborative group does not necessarily guarantee success [23]. Friends engage in more extensive conversations and have been found to be more supportive and critical that non-friends [11]. Additionally, expert human teachers in a tutoring session not only watch task-oriented performance indicators, but also closely monitor the motivational indicators [13].

People react to computer agents in fundamentally social ways. Several interesting effects in text-only human-computer interfaces have been demonstrated [18] such as computers which praise rather than criticize their users are preferred, users prefer the computer to match them in personality, and users prefer computers that become more like them over time rather than those which maintain a consistent level of similarity.

In order to develop socially intelligent agents, socially relevant interaction tactics for tutoring agents to accomplish motivational goals have been proposed [12]. Agents capable of improving motivation over longer periods of time have also been proposed [7]. Major research efforts also continue to explore politeness and its role in effective tutorial dialogue, motivating students and learning [15, 16].

In evaluation of affective gendered learning companions, it was found out that using artificial male tutors resulted in better attitudinal and emotional states and consequently better learning for both male and female students [1]. Yet another approach to supporting social interactions is a structural one, and suggests that complex natural language processing methods are not necessary to sufficiently support social interaction during collaboration [22].

An approach combining the Wizard-of-Oz design with reinforcement learning of dialog strategy overcomes the need of an initial corpus [19]. Dialog strategies are represented as Markov decision processes [14] in this approach. The method bootstraps a simulated environment using Wizard-of-Oz data and trains the optimal reinforcement learning policy in this environment; first a small representative corpus is collected (using a Wizard-of-Oz experiment) from which a simulated learning environment (the user model) is constructed, and finally the optimal reinforcement learning policy is devised through interaction with this simulated environment. In the evaluation, reinforcement learning policy has received significantly more rewards than the baseline policy learned by C4.5 decision trees and JRIP rule induction. In addition, human users rate the policy 10% better on average.

All in all, research suggests that intelligent tutors can maintain the appearance of being socially intelligent by carefully selecting the appropriate words at the appropriate time, not requiring the presumably unavoidable labor intensive language processing methods. In our approach we attempt to follow these observations.

In the last few years, the technology-enhanced learning landscape has been influenced by the emergence of social networking. Facebook, the most prominent social networking site, is actively used by more than 500 million users [29]. In common with other social sites, users can create their own personal profiles on Facebook by providing information that falls into predefined categories (e.g. contact information, work and education information). They can post pictures, participate in discussions, view other peoples profiles, and communicate with others using public or private messages, link with each other to become "friends," or create and join groups.

The potential for educational use of such sites seems tempting. Teachers can create their own profiles, a course page, and use Facebook's functions (e.g. discussion boards, instant messaging) to run the course. However, Facebook does not appear to diminish or eliminate barriers between teachers and students. Students use Facebook for student-to-student exchanges but are less likely to use it in teacher-student interactions [26]. Students using an on-line Facebook course tend to engage in passive activities such as viewing others' profiles and reading comments instead of active actions, as, for example, commenting and sending a message [25]. In addition, only 66% of the students sampled consider it acceptable to have teachers on Facebook. Acceptance has huge gender differences – 73% of men in the sample consider this acceptable as opposed to only 35% of the women. Issues arise as to what is appropriate for teacher-student interactions. For example, students feel uncomfortable poking or befriending teachers; they also feel uncomfortable when teachers poke or befriend them. Neither Facebook nor other social networking sites have been designed with an educational purpose in mind, and educators find it difficult to adjust to them [6].

3. Learning Environment

We propose a novel learning framework in which a socially intelligent agent (tutor) guides students through appropriate instructional activities. The tutoring strategy is improved continuously using a socially augmented reinforcement learning method. In addition to the ordinary exploratory part of reinforcement learning, human wizards provide improved guidance in the state and action spaces. In the evaluation, we observed that a reasonable number of human actions are sufficient to bootstrap the tutoring strategy that is followed by the wizard.

In order to enable socially intelligent instruction, the tutoring strategy operates on top of socially augmented components that constitute the learning environment: (1) problem solving, (2) course notes, and (3) off-task social dialogs. The components are, then, integrated by the tutoring strategy. A conceptual diagram of a student working in the proposed learning environment is provided in Figure 1.

3.1 Overview

The student begins with a dialog (with the tutoring agent) in which his immediate goals are determined, and combined with long-term goals and tendencies; the tutor recommends a learning activity to pursue. In order to do that, the user and student models are employed. Either study of course notes or problem solving is selected. For group mode learning activities, students can assemble collaborative groups from their available friends; anonymous introductions can be facilitated by the tutoring agent. Interactions between users are restricted to friends or tutor-recommended students only; hence a student does not come into contact with any entity to which his/her relationship cannot be predicted.

In problem solving, task templates are employed to enable novel approaches for the structuring of task solutions,

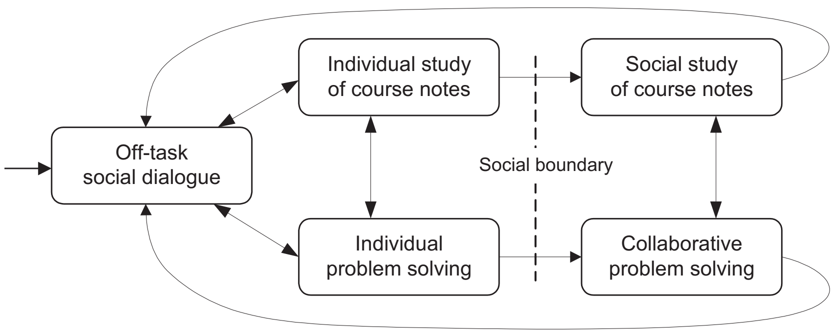

Figure 1: Learning activities with admissible transitions (arrows), which are facilitated by the tutoring agent.

and in personalization of task descriptions. Course notes are analogous to traditional reading materials augmented by a highlighting facility that works as social cues. The off-task dialog facility is primarily human wizard-driven and in theory, would work without automated support. Even so, we propose novel ideas for facilitating the reuse of knowledge (put in by human wizards) across different students: virtualized scaling-events timeline provides the baseline user model which is used by feature detectors to recognize patterns that constitute the states in the state space employed by reinforcement-learning tutoring. Contextual reflection can then detect parameters in human wizard actions and is capable of executing actions in different contexts in which they were originally used. Similarly at a lower level, linguistic reflection tailors the language produced to the context at hand. This design allows students to use their imaginations and use the social dialog facility to request almost anything from the tutor; at the same time they are able to utilize the generic learning facilities that are found in contemporary learning systems. The heart of our approach, then, is in the state-action view facilitated by the (1) off-task social capabilities of the tutoring agent which in turn are made possible via (2) novel user and student models of real-life social context, (3) careful Wizard-of-Oz design linked with reinforcement learning of human-like tutoring interaction strategy, and (4) robust underlying learning activities. All of this assumes that ”human-like” is the ”socially intelligent” we seek.

The proposed methods that constitute the learning environment work together as depicted in Figure 2. Individual and collaborative problem solving activities employ structured tasks; the basic tree-structured task model is an extension of the task model used in pseudo-tutor environments [10]; and the prioritized branches task model is an extension of the basic model in order to model more complex tasks effectively. Tasks are modeled in parametric templates that use on-the-fly generation of new task instances to discourage cheating, and enable personalization to a student and to a group of students. A method for evaluating collaboration can be used both as a metacognitive tool to support learning, and a way to detect suitable collaboration peers for group formation. The study of course notes can also take advantage of parametric descriptions – it can be tailer to the student's context.

The learning-related activities are complemented with non-learning (user) activities. The core model to represent user's social context, past events, future goals, etc is the virtualized scaling-events timeline along with the conceptual ontology, feature detectors, and contextual and linguistic reflections. The method for bootstrapping the tutoring strategy is used to induce a strategy followed by human wizards. Using this strategy, the computer tutor can automatically guide students to appropriate learning activities, and assemble suitable collaborative groups.

3.2 Problem solving

Tasks are problem solving exercises that can be used to acquire units of knowledge (e.g terms, concepts, procedures). Task descriptions consist of subtasks that are structured according to solution paths by the content author. The descriptions are parameterized and allow the content author to specify interdependencies between individual parameters. Students use primarily free-text to answer questions; these answers are judged according to categories for which the content author can specify scaffolding messages. If a student answers incorrectly, he/she may be prompted to correct the mistake. A student who answers correctly may be challenged with a more difficult question about the current task, or may continue with another task.

The basic model of tree-structured tasks does not require a fine grained cognitive problem solving model which is difficult to create, but rather, allows content authors to design simple structured exercises quickly. The improved model allows students to follow multiple solution branches prioritized according to their abilities and tutoring goals. Together with task parameters that allow for procedural solution, this is a shift toward a full tutoring system.

In the tree-structured model, solution paths for a particular task are structured in a tree $T = (V, E)$ with vertices $V = \{S_1, S_2, ..., S_N\}$ corresponding to subtasks that students may solve at one time or another during task solution, and directed edges E corresponding to possible student answers for respective subtasks. The tree's leaves represent terminal subtasks at which the solving of the task is finished. S_1 is the initial subtask in which the task administration begins. When a student reaches a subtask, the subtask description is displayed. Whenever the subtask is non-terminal the student is required to provide an answer. A default next subtask defaulti is designated for each non-terminal subtask S_i to enable continued solving of the task when an unanticipated student answer is received. These unexpected answers will be used by content authors later to improve the task tree.

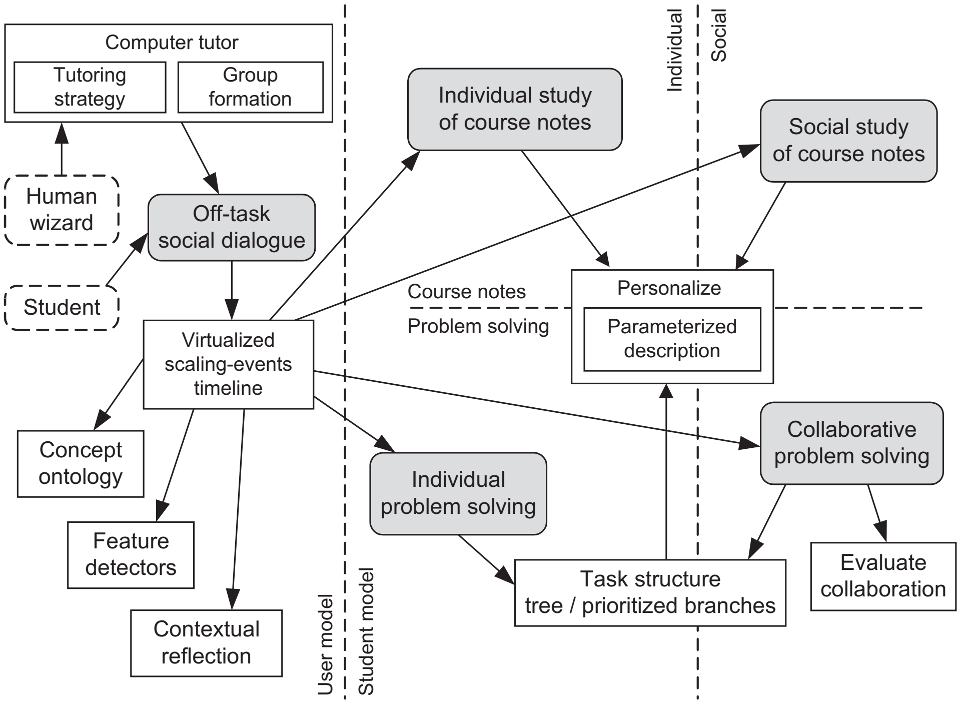

Figure 2: The components and methods proposed in the socially intelligent tutoring environment.

The tree-structured task model is relatively easy to use. The order of subtasks and hints (embedded in subtasks' descriptions) is set in advance by the content author and allows the student's experience during problem solving to be designed precisely. On the other hand, it does not allow any kind of adaptation until the task is completely solved. Analogous models are used in pseudo-tutor systems such as Assistments which uses a more simplified model that neither allows parameter specification nor answering judging scripts but instead, uses simple text descriptions, and matches student responses to predefined textual patterns [10].

The tree-structured task model may be impractical to use for more complex tasks that involve independent aspects of a single solution that will be accounted for in the solution path tree. The prioritized branches task model overcomes this limitation by traversing relevant tree's branches sequentially according to their priorities. In this improved model, each edge e is assigned a priority value $priority_e$ that describes the relative significance of the particular solution branch with respect to other branches. The branch (edge) that is matched as relevant to the student's solution path at the current node is put into a priority queue that selects next subtask as the subtask with the highest priority (for an example, see Figure 3). The order of traversal in this model is determined by branch priorities specified by the content author. During the task design process, the content author must decide on the priority values of the respective branches. Typically, the priority of a sub-branch will be given a higher designation than the priority of the parent branch when the sub-branch is necessary to solve the parent branch; a lower sub-branch's priority value signifies an optional part of the solution.

In our approach to collaborative problem solving, students first attempt to solve sub-tasks separately. They

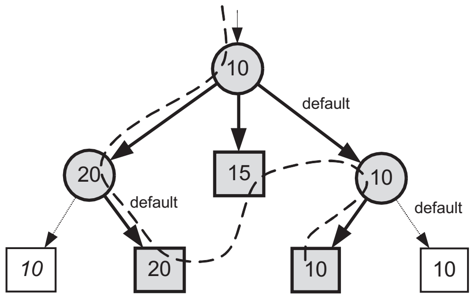

Figure 3: Example prioritized branches task, the order of subtasks is shown by the dashed line. Matched edges are displayed in bold, and subtasks administered with dark background.

consult each others' solutions only after they have worked out their own answers. After the work on the entire task has been completed, users are prompted to give both feedback on the effectiveness of the problem solving session and peer feedback concerning their teammates. In the dissertation, we present a method for evaluating collaboration that is used continuously during collaboration to monitor the process and is used to guide students as a meta-reflective tool, and a method for personalizing tasks to the group's/user's (social) context. These methods are not discussed here due to brevity.

3.3 Course Notes

Course notes are rich content documents describing concepts (knowledge units) susceptible to study. Knowledge organized in documents for students to read/study does not proclaim the need for a student model. Typically these documents are available from legacy repositories, and can be readily used within a learning environment without the need to create/use any model of the user.

Due to this advantage, most contemporary e-learning environments are organized around simple student models. In our approach to learning, we consider course notes as a fallback from problem solving activities. If a student is not capable to continue solving problems, an easier option is to read about what exactly the student should be doing during problem solving in the first place. Course notes are represented using the same custom rich content markup language as task descriptions. Moreover, existing documents can be readily used without lengthy formatting, while fancier formatting is available once the need to improve the material arises.

3.4 Off-Task Dialog

Students interact individually with the socially intelligent tutoring agent through an off-task social dialog facility. The dialogs are the primary source of data for the user model. Within the dialogs, either learning-related (task) or learning-unrelated (social) topics are discussed. In our approach, learning-related dialogs are reserved for the structured learning activities (problem solving and course notes) in which tutor's responses are pre-scripted in the activities' descriptions. Social topics require a degree of freshness in tutor's utterances that must be ensured in order to reflect the constantly changing social context of the student. We present a novel model of social life used to hold social conversations with students. In practice, conversations are not fully automated, and require the presence of human wizard operating in a Wizard-of-Oz design. Our model of social life is capable of reusing collected knowledge across students so that the manual load on the human wizards is kept at a manageable level even when multiple students are serviced.

The major aspect of the proposed model is a temporal representation of student's activities, plans, thoughts, etc in a timeline; timelines of different students may provide useful information for each other. See Figure 4 for an example timeline. Initially, the human wizard detects patterns in student's behavior, and plots the conceptual knowledge into the timeline of events for that particular student. An event can be anything that occurred at a time for the particular student, e.g. mouse click, submission of a subtask answer, or a school trip in the outside world. The timeline organizes events as intervals sorted by time. A group of nearby (in terms of time) events constitutes a state; different states can be compared with each other, on the basis of the conceptual similarity of the events occurring within the states. Let us consider two states A and B which are similar. The idea then is: whenever there is an event E which occurs in the state A, but is not yet plotted in the timeline near B, it may be considered as a new candidate event for addition to B's timeline at B's time. Similarly, for actions performed by human wizards: actions that wizards execute at a particular state suggest a recommended action to execute at similar states. Thus, similar groupings of events at different timelines are opportunities for reuse. Human wizard can perform an action of following three types:

1. user interface action (e.g. mouse click, window close);
2. learning action (e.g. recommend a learning activity, finish exam administration); and
3. social activity (e.g. plot an event to student's timeline, which may include a recommendation for an action to pursue).

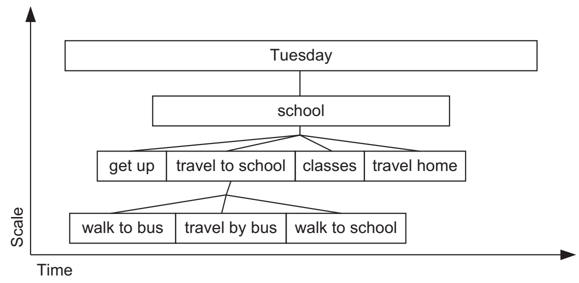

Figure 4: Example timeline with events at various scale and time, scaling rules are shown as hyperedges.

The virtualized scaling-events timeline is a model of user activity. The socially intelligent tutor makes decisions concerning a particular person at various points in time. The events in the person's timeline at a time point constitute the state of affairs concerning the particular person from the point of view of the tutor. In other words, in a given *state* (for the person) the tutor can choose to execute an *action*.

Given that a state denotes a set of features relevant to tutor's decision making process, how can we reuse the knowledge put in by human wizards (such as actions taken) in different states? Let *context* denote the other features detected at a particular situation, i.e. the *state* contains the relevant ones, and the *context* contains all the other features that were recognized by feature detectors. It is not that the contextual features are useless; on the contrary, they are important in tailoring the action to the particular context. Once the distinction between state and contextual features is established, a method to adjust actions (that were originally executed in another context) for executing in a different context can be employed – contextual reflection. The method is based on template actions. In effect, when wizards execute an action in a context they mark the action's contextual parameters (parameters that depend on contextual features). The next time the action is invoked, the parameter values are filled according to the values of the contextual features in the actual situation. Linguistic reflection is a special type of contextual reflection used for language generation.

3.5 Transitions

Transitioning between components (learning activities) is *student-directed* or *tutor-directed*. A student can change the learning activity rather easily; it is as simple as clicking an item in a menu. We describe the way in which the tutor facilitates the transitions and what needs to be done on the part of the learning environment for a transition to happen. We explore the goals of tutoring and how the tutor can plan to meet these goals. Then, we analyze individual transitions from off-task to on-task and back, from problem solving to course notes and back, and from individual mode to social mode and back.

A socially intelligent tutor has to provide both cognitive and motivational help. In order to do so, the goals of the tutoring strategy are multidimensional: task vs. nontask, individual vs. social. The tutor can operatively attribute a different importance to each dimension and hence, adjust the tutoring strategy to the actual requirements of the students situation. Consequently, we distinguish four basic goals: emotional, social, individual learning, and group learning. Obviously, different actions meet different types of goals. For example, a peaceful student-tutor dialog may satisfy the emotional needs of one student, while a week-long trip out of town with friends may be the only way to satisfy the social needs of another person. The role of the socially intelligent tutor is to recognize how these goals can be satisfied for a particular student since a prescribed amount of activities to meet the goals may not work for everyone to the same extent. The simplest representation of a goal for each goal type is a scalar value in R that expresses the current demand for a particular activity. Although a complex hierarchy/ontology of goals of each type could certainly be designed, at the end, we need to summarize the needs to a single scalar value in order for the tutor to optimize the student's activities by optimizing (minimizing) the value.

The issue of transitioning from off-task to on-task is usually an issue of motivation. When the student feels adequately motivated, he can start working on learning-related content. Otherwise, motivation must be improved so that the student can learn at all. The students current level of achievement and his/her learning profile are the primary factors used to determine transitions between problem solving and course notes. Some prefer more active learning (problem solving) and others prefer more passive (course notes) learning. However, this should not prevent the socially intelligent tutor from transitioning the student out of his comfort zone to teach him to be a more balanced learner. A change between individual and social modes is often due to student motivation and the student's level of ability; tutoring goals also play a small role, i.e. that we want to instruct and assess students for both the ability to work individually as well as in a team. Some people need to work with others to feel motivated; others are lone learners.

3.6 Evaluation Studies

A student begins working with the socially intelligent learning environment by engaging in a welcome conversation with the tutor. The tutor is personified by a person to whom the student can relate. Either a human wizard guides the computer tutor so it can select an action, or the computer tutor automatically follows the devised tutoring strategy. The proptotype environment offers various learning activities as described previously. A student can work both actively (problem solving) or passively (course notes), and individually or in collaboration with others. During problem solving, the student is presented with a subtask to solve in a linear manner. When an answer is submitted, its correctness is judged, and a new subtask for the student to solve is displayed allowing the student to scroll all the way back to the beginning and see the course of action he/she took. A free text prompt is typically used to answer subtasks; other elements such as drawing canvas, drop-down boxes, and radio buttons are used for structured interactions. In the social mode, students can work on problems in groups. Additional interactions such as synchronous messaging within the group, and a voting facility are utilized (Figure 5).

In both the individual and social modes, each students actions are transmitted to the server. They are retransmitted to other participants – students, wizards – who are involved with the particular student. Technically, the user interface widgets operate in a multi-user mode that enables multiple users to participate in a single user interface. For example, when the tutor instructs the student to press a button which the student is unable to find, the tutor may use either the student's mouse pointer or introduce a new mouse pointer and click the button on behalf of the user

In the feasibility study of the socially intelligent agent, when students were tasked to work with the system during an algebra class, we observed that students reveal on average 1.56 (st.dev 1.75) features about themselves in a social conversation with the tutoring agent, and only about 56% of the students engage with the tutoring agent at all. Students that engaged in a conversation with the tutoring agent exhibited higher learning gains. The not engaged group showed relatively low learning gain 3.7% vs. 12.3% exhibited by the engaged group. This effect however may also be due to their previously higher motivation, and cannot be attributed to the conversation with the intelligent tutor alone. We need to further investigate the motivational state of students before the experiment, and examine the role the tutor can play, if any, in motivating students that were not motivated before. In summary, students that engaged in social off-task dialogues with the tutor were more effective in solving problems correctly (57% vs. 37%), and liked the system more (4.22 vs. 2.86), suggesting that learning environments may produce higher learning gains by "being friends" with the students, providing them with socially relevant motivation.

4. Bootstrapping a Tutoring Strategy

The components of our learning environment can operate in individual and social modes and provide diverse opportunities for instruction. The role of the socially intelligent tutor, then, is to identify an appropriate learning activity for each student at a given time. In other words, the tutor recognizes the student's state and performs an action that guides him/her to the appropriate learning activity. In this chapter, we propose a method that integrates the individual components into a tutoring strategy, i.e. a policy that can select an appropriate action for the tutor to take in the learning environment. The proposed method for bootstrapping a socially intelligent tutoring strategy is based on reinforcement learning that is modified to be able to optimize the policy for a single student efficiently as well as when rewards for many students need to be considered, i.e. the method is suitable for social learning environments in which a large number of students participate. In the dissertation, we: (1) describe the state-action view of how the learning environment can be interpreted as a reinforcement learning problem; (2) present the reinforcement learning method for a solitary student; and (3) augment the method with a social graph in order to handle large number of students efficiently

Reinforcement learning is a machine learning approach that induces the optimal policy for an agent to follow in a stochastic environment while, at the same time maximizing cumulative rewards [24]. The agent repeats the following steps indefinitely: At each step:

- the agent observes the environment recognizing the current state;
- selects an action to perform;
- a reward (scalar in R) may be received from the environment; and

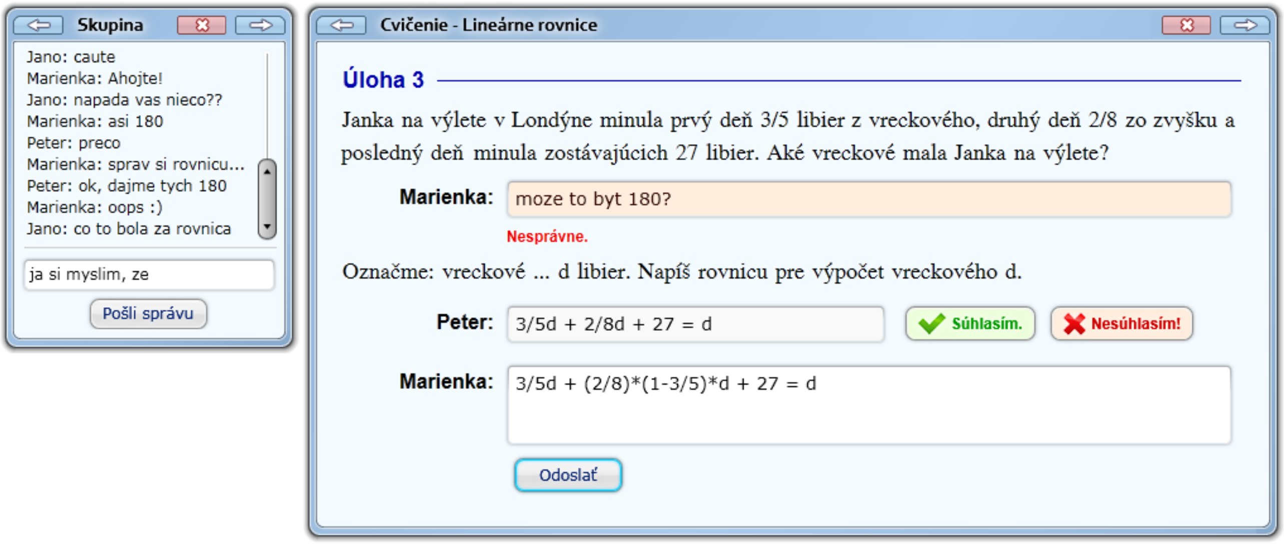

Figure 5: Collaborative problem solving free-text task answer with voting mechanism (in Slovak).

- the environment changes according to the action executed.

4.1 Overview

The idea is to observe which actions produce positive rewards, and progressively calculate better estimates of action utility values in each state. Ultimately, the policy should be able to select "good" actions in each possible state.

How do we translate our learning environment to this formalism? We cannot possibly hope to observe what students actually do at their computer desks; we can only observe the actions that students have performed within the learning environment. For that purpose, we have already proposed a detailed model of student activity, the virtualized scaling-events timeline. Feature detectors can recognize similar situations in the timeline; human wizards select the relevant features that make up the states which, in turn, constitute the state space for an individual student in our learning environment. Actions user interface actions, learning activity actions, or social actions on timeline that are executed in the learning environment are based on action templates that were originally performed by human wizards, i.e. there is no "chaotic" exploration of the state and action spaces. Finally, at times when an activity produces positive/negative results, multidimensional rewards (discussed in 3.4.1) are received. To find a globally optimal tutoring strategy, the computer tutor cannot follow two independent goals at once but needs to assign weights to the respective reward dimensions. For example, an individual learning-only strategy, a group learning-only strategy, an emotional-only, a social-only, or a weighted combination of any of these strategies can be followed, but in the end, a single scalar reward in R must be received after executing an action.

The expected utility values computed by the algorithm provide a way to look ahead so as to plan for future actions; after a reward has been received, the utility values at the previous state are updated. Importantly, this approach, Q-learning [28], does not require a model of the environment (obviously, we cannot predict outcomes of actions that are only recommendations for the student). In addition to traditional exploitation (selecting the best possible action at a state) and exploration (selecting a random action) we extend the reinforcement learning method with human wizard interventions (Figure 6). Wizard actions are welcome at any time, but the assumption is that because the wizard is a real person his/her actions are expensive. As a result, we are interested in minimizing the number of human wizard actions required to learn the optimal policy.

We propose a novel approach that integrates the individual components of the learning environment. It employs a socially-augmented reinforcement learning method to ascertain those actions that are appropriate for given situations. The learning policy is driven by human wizards that guide reinforcement learning into the exploration of relevant states and actions. A tutor following the induced policy is, in effect, simulating human decisions and becomes a socially intelligent computer tutor provided that the human wizards perform socially intelligent actions.

Due to excessive computational requirements, the generic reinforcement learning approach is not applicable to learning optimal policy in a large population of students. We propose a modified algorithm that in the case of a social scenario (when multiple students environments must be considered) aggregates the individual states of other students into a combined shared state, thus reducing the state space to a manageable size. Hence, the proposed method can run efficiently even when a large population of students is serviced.

The environment's state and action spaces are practically infinite. In a face-to-face tutoring session, the human tutor may employ any enlightened tutoring approach; obviously the computer tutor cannot tractably enumerate and explore all the options available in the computer setting. Therefore, guidance by human wizards in exploring applicable actions is indispensable for the computer tutor to obtain knowledge of the relevant states.

4.2 Evaluation

We evaluated the proposed method in a series of simulated scenarios: individual, social-only, and mixed sce-

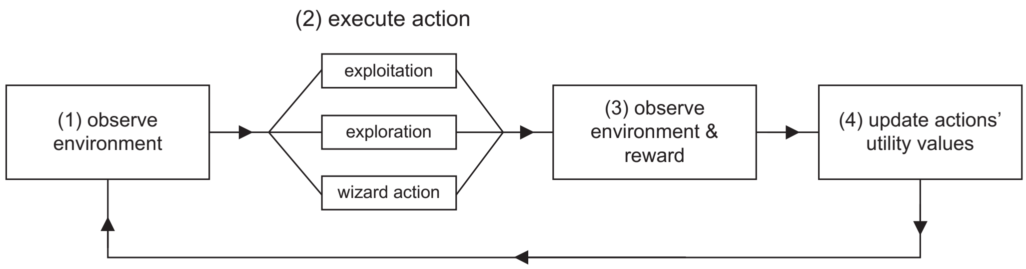

Figure 6: Reinforcement learning approach enhanced with Wizard-of-Oz action selection.

narios. The scenarios simulate a wizard's decisions using a hidden model of the environment allowing us to explore the way in which the number of wizard actions affects bootstrapping efficiency. Absent any wizard actions, the bootstrapping algorithm cannot find the optimal policy in a reasonable period of time. As little as 5% wizard participation is sufficient to find the optimal tutoring policy; greater wizard participation (in guiding exploration) results in faster learning of the optimal policy.

For a solitary student, to reach 90% of the optimal policy, the wizard must be used for at least 10% of the decisions. For an 80% policy only minimal wizard participation is necessary. In the social-only scenario (the tutor services M students), we observe, that with at least some (>0%) wizard participation, roughly 85% of the expected maximum policy can be learned. Interestingly, the sweet spot is again between 5% and 10%, with a 10% wizard rate being sufficient to induce a good tutoring strategy rapidly. The results suggest that the reinforcement learning method augmented with the social graph can indeed detect appropriate peers for social interactions. Moreover, 5% of wizard participation is sufficient to detect the optimal peers relatively quickly.

In the mixed scenario (where students can perform both individual and social actions, influencing each other), the interaction between individual and social actions caused the bootstrapping method to take longer. Wizard participation of 10% is sufficient to attain 85% to 90% of the theoretical maximum expected mixed (social and individual) policy. Higher wizard participation does not affect the quality of the policy identified but only improves the learning speed, and can produce a good policy very early in the bootstrapping process.

5. Conclusions

Todays learning environments do not engage students to the extent that is desirable. Two key aspects of effective learning environments are student motivation and the time that students spend learning; we explored approaches to support them.

We investigated components of effective learning environments, and proposed an approach to bootstrap a socially intelligent tutoring strategy that operates on top of these components. An automated computer tutor that follows the strategy can select appropriate learning activities for students. The method is designed to be able to balance individual and social, and cognitive (on-task) and affective (off-task) activities. Combined with the off-task dialog facility guided by human wizards, the proposed bootstrapping method is designed to provide novel interaction patterns to students for a relatively long period of time in order to maintain their motivation and increase the time students invest in study.

The learning environment presented was designed to provide students with socially relevant interventions that would motivate learning. The environment consists of a redesign of typical learning components so that diverse social interaction possibilities are available (to the students). Both problem solving and course note facilities are able to operate in individual (for a single student) or social (for group of students) modes.

The core contributions are:

- novel extensions of problem solving and course notes learning activities that enable student to work on personalized content in both individual and group/social mode;
- novel model of user's social context (past events, future goals, etc) that facilitates knowledge reuse across different students; and
- novel method for bootstrapping a socially intelligent tutoring strategy which can balance individual and group activities, and assemble suitable collaborative groups.

The proposed approach puts humans in the loop, and requires non-trivial human participation to sustain long-term operation; for example, student answers need to be judged, and the dialog facility requires human wizards to identify previously not encountered interaction patterns.

However, the evaluations suggest that the fruits of this approach may be worth it: (1) problem solving robustness discourages surface approaches to learning; (2) socially engaged students show higher learning gains; and (3) the bootstrapping of a tutoring strategy (followed by the human wizards) can find a balance between individual and group/social activities, and learning vs. non-learning activities for students.

Acknowledgements. This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 028-025STU-4/2010, the Scientific Grant Agency of SR, grant No. VG1/0508/09 and it is a partial result of the Research & Development Operational Program for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 26240120029, co-funded by ERDF.

References

[1] I. Arroyo, B. Woolf, J. Royer, and M. Tai. Affective gendered learning companions. In *14th International conference on Artificial Intelligence and Education (AIED 2009)*, 2009.

[2] R. Baker, A. Corbett, K. Koedinger, and A. Wagner. Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390. ACM, 2004.

[3] M. Barla, M. Bieliková, A. B. Ezzeddinne, T. Kramár, M. Šimko, and O. Vozár. On the impact of adaptive test question selection for learning efficiency. *Computers & Education*, 55:846–857, September 2010.

[4] E. Blanchard, B. Volfson, Y. Hong, and S. Lajoie. Affective Artificial Intelligence in Education: From Detection to Adaptation. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 81–88. IOS Press, 2009.

[5] K. Boyer, R. Phillips, M. Wallis, M. Vouk, and J. Lester. Balancing cognitive and motivational scaffolding in tutorial dialogue. In *Intelligent Tutoring Systems*, pages 239–249. Springer, 2008.

[6] C. Cain, C. Seals, and J. Nyagwencha. Social networking teaching tools: A computer supported collaborative interactive learning social networking environment for k-12. In J. Sanchez and K. Zhang, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2010*, pages 1612–1617, Orlando, Florida, USA, October 2010. AACE.

[7] J. Cassell and T. Bickmore. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and User-Adapted Interaction*, 13(1):89–132, 2003.

[8] A. Corbett. Cognitive computer tutors: Solving the two-sigma problem. *User Modeling 2001*, pages 137–147, 2001.

[9] S. Craig, A. Graesser, J. Sullins, and B. Gholson. Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Learning, Media and Technology*, 29(3):241–250, 2004.

[10] M. Feng, N. Heffernan, and K. Koedinger. Addressing the testing challenge with a web-based e-assessment system that tutors as it assesses. In *Proceedings of the 15th international conference on World Wide Web*, pages 307–316. ACM, 2006.

[11] W. Hartup. Cooperation, close relationships, and cognitive development. 1996.

[12] W. Johnson. Interaction tactics for socially intelligent pedagogical agents. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 251–253. ACM, 2003.

[13] M. Lepper and J. Henderlong. Turning" play" into" work" and" work" into" play": 25 years of research on intrinsic versus extrinsic motivation. 2000.

[14] E. Levin, R. Pieraccini, and W. Eckert. A stochastic model of computer-human interaction for learning dialogue strategies. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[15] R. Mayer, W. Johnson, E. Shaw, and S. Sandhu. Constructing computer-based tutors that are socially sensitive: Politeness in educational software. *International Journal of Human-Computer Studies*, 64(1):36–42, 2006.

[16] B. McLaren, S. Lim, D. Yaron, and K. Koedinger. Can a Polite Intelligent Tutoring System Lead to Improved Learning Outside of the Lab? In *Proceeding of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 433–440. IOS Press, 2007.

[17] S. McQuiggan, J. Rowe, S. Lee, and J. Lester. Story-based learning: the impact of narrative on learning experiences and outcomes. In *Intelligent Tutoring Systems*, pages 530–539. Springer, 2008.

[18] B. Reeves and C. Nass. *How people treat computers, television, and new media like real people and places*. CSLI Publications and Cambridge university press, 1996.

[19] V. Rieser and O. Lemon. Learning effective multimodal dialogue strategies from wizard-of-oz data: Bootstrapping and evaluation. In *Proceedings of ACL*, pages 638–646. ACL, 2008.

[20] J. P. Rowe, S. W. McQuiggan, J. L. Robison, and J. C. Lester. Off-task behavior in narrative-centered learning environments. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 99–106. IOS Press, 2009.

[21] M. Šimko, M. Barla, and M. Bieliková. ALEF: A Framework for Adaptive Web-based Learning 2.0. *Key Competencies in the Knowledge Society*, pages 367–378, 2010.

[22] A. Soller. Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12:40–62, 2001.

[23] A. Soller, B. Goodman, F. Linton, and R. Gaimari. Promoting effective peer interaction in an intelligent collaborative learning system. In *Intelligent Tutoring Systems*, pages 186–195. Springer, 1998.

[24] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. The MIT press, 1998.

[25] B. Teclehaimanot and T. Hickman. Student-teacher interaction on facebook: What students find appropriate. In T. Bastiaens, J. Dron, and C. Xin, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009*, pages 3181–3190, Vancouver, Canada, October 2009. AACE.

[26] T. L. Towner and C. L. Munoz. Letâs âŸfaceâ it: Facebook as an educational tool for college students. In J. Sanchez and K. Zhang, editors, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2010*, pages 1953–1958, Orlando, Florida, USA, October 2010. AACE.

[27] N. Wang, W. Johnson, R. Mayer, P. Rizzo, E. Shaw, and H. Collins. The politeness effect: Pedagogical agents and learning outcomes. *International Journal of Human-Computer Studies*, 66(2):98–112, 2008.

[28] C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[29] M. Zuckerberg. 500 million stories, the facebook blog, 2010.

Selected Papers by the Author

J. Tvarožek, M. Kravčík, M. Bieliková. Towards Computerized Adaptive Assessment Based on Structured Tasks. In *Adaptive Hypermedia and Adaptive Web-Based Systems – AH 2008*, LNCS 5149, pages 224–234, Hannover, Springer, 2008.

J. Tvarožek, M. Bieliková. The Friend: Socially-Intelligent Tutoring and Collaboration. In *Artificial Intelligence in Education – AIED 2009*, Frontiers in Artificial Intelligence and Applications, Vol. 200, pages 763–764, Brigthon, IOS Press, 2009.

J. Tvarožek, M. Bieliková. Feasibility of a Socially Intelligent Tutor. In *Intelligent Tutoring Systems – ITS 2010*, LNCS 6095, pages 423–425, Pittsburgh, Springer, 2010.

J. Tvarožek, M. Bieliková. Enhancing Learning with Off-Task Social Dialogues. In *European Conference of Technology-Enhanced Learning – EC-TEL 2010*, LNCS 6383, pages 445–450, Barcelona, Springer, 2010.

O. Ševce, J. Tvarožek, M. Bieliková. Enabling Content Recommendation and User Profiling upon Automatic Terms Retrieval, Rating and Categorization. In *Artificial Intelligence: Methods, Systems and Applications – AIMSA 2010*, LNCS 5149, pages 71–80, Varna, Springer, 2010. Springer, New York, 2010.

M. Jačala, J. Tvarožek. Identifying Entity Relationships in News Reports. In *Proceedings of SOFSEM 2010, Vol. 2*, pages 32–41, Špindlerův Mlýn, Institute of Computer Science AS CR, 2010.

J. Tvarožek, M. Bieliková. Social network-based evaluation of collaboration. In *Proceedings of ZNALOSTI 2010*, pages 154–164, Jindřichův Hradec, VSE Praha, Oeconomica, 2010.

J. Tvarožek, M. Bieliková. Peoplia - interaktívne vzdelávanie podporené sociálnymi väzbami. (In Slovak) In *Proceedings of ZNALOSTI 2010*, pages 279–282, Jindřichův Hradec, VSE Praha, Oeconomica, 2010.