Providing Adaptive Support in an Interactive Simulation for Learning: an Experimental Evaluation

Samad Kardan, Cristina Conati
Department of Computer Science
University of British Columbia, Vancouver, Canada
{skardan, conati}@cs.ubc.ca

ABSTRACT

Recent rise of Massive Open Online Courses (MOOCs) with unlimited participants, makes employing learning tools such as interactive simulations all but inevitable. Interactive simulations give students the opportunity to experiment with concrete examples and develop better understanding of concepts they have learned. However, some students do not learn well from this relatively unstructured form of interaction, suggesting the provision of adaptive support as a way to address this issue.

This paper presents a formal evaluation of this approach. We describe the process of designing an intervention delivery mechanism for adding adaptive support to an exploratory interactive simulation. The experimental evaluation of the adaptive version of the simulation indicates that the adaptive support provided to students significantly improved their learning performance. Quantitative and qualitative evaluations of users' acceptance of the system are generally positive but pinpoint areas for improvement.

INTRODUCTION

Advances in HCI continuously aid the creation of novel interfaces to support education and training. For example, interactive simulations are increasingly used as exploratory/open-ended environments for learning different subjects (e.g., physics and math [18,21]). These exploratory educational simulations (ESims hereafter) are designed to foster exploratory and active learning, by giving students the opportunity to proactively experiment with concrete examples of concepts and processes they have learned in theory. However, it has been shown that some students may not learn well from this relatively unstructured and open-ended form of interaction, because they lack skills needed to explore effectively [9,10]. Hence, students may benefit from having additional guidance when they interact with an ESim (e.g., [9,19]).

In this paper, we evaluate an approach to provide this guidance adaptively, based on the real-time evaluation of whether a user's interaction behavior with an ESim is conducive to learning or not. The test-bed for our research is the AIspace CSP applet, an interactive simulation that illustrates the workings of the AC3 algorithm for constraint satisfaction problems. This applet is part of AIspace, a publicly available suite of interactive simulations for a variety of AI algorithms that is routinely used in AI courses both in our university and elsewhere. There is extensive positive feedback from students when these applets are used for in-class demonstrations, and formal studies on the CSP applet have shown that it both fosters learning and student motivation [1]. These studies, however, also showed that not all students leverage at best the functionalities available to explore the AC3 algorithm mechanics. We hypothesize that the CSP's applet effectiveness could be further enhanced by providing guidance on how to use these functionalities to those students who need it. To investigate this hypothesis, we devised a new version of the CSP applet that provides *adaptive interventions* designed to provide such guidance.

The adaptive version of the CSP applet (adaptive-CSP henceforth) leverages a user modeling approach proposed in [11,12] to determine in real time if/when a user needs help in using the applet and why. In this paper we introduce a mechanism that, by relying on this user model, selects which interventions to deliver at any given point of the interaction. We illustrate the design process we used to understand how to deliver the interventions in an effective and unobtrusive manner, and present a user study aimed at evaluating the resulting adaptive interventions by answering the following research questions:

- Do the adaptive interventions added to the CSP applet increase the applet's effectiveness in fostering student learning?
- Do users have a positive impression of the interventions, in terms of relevance, usefulness, noticeability, and unobtrusiveness?

Our results provide positive answers to both questions. This work contributes to the general area of intelligent learning environments. In this area, there is extensive evidence that

CHI 2015, April 18 - 23 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04…$15.00.
http://dx.doi.org/10.1145/2702123.2702424

real-time personalized support can increase user learning during structure problem solving tasks, where feedback can be provided based on a clear definition of correctness for the student behaviors [22,25]. There is much more limited evidence for providing support for *open-ended exploration* in ESims, due to the more unstructured nature of the interaction which makes it difficult to define a priory how student behavior relates to learning. Our work contributes to fill this gap, and shows encouraging evidence that personalized support to open exploration with Esims, provided by helping students understand how to use the simulation's functionalities at best, can increase the pedagogical effectiveness of these educational tools.

In the rest of this paper, we first discuss related work. Next, we introduce the original CSP applet, followed by a brief description of the user modeling approach leveraged by the adaptive version. Then we describe the process of adding adaptive interventions to the applet, the study we conducted to evaluate their effectiveness, and its results. We conclude with a summary of findings and future work.

RELATED WORK

Providing adaptive support to students is an important part of Intelligent Tutoring Systems (ITS), where this support is intended to partially replace the guidance a student would receive in a one-to-one tutoring setting [24]. There is ample evidence that adaptive support can be beneficial in ITSs that target problem solving and other pedagogical activities where there is a well-defined set of solutions/behaviors that the ITS can target. For example, Westerfield et al., showed the effectiveness of personalized feedback on errors the students make in an ITS that provides step-by-step training on assembling a computer mother board [25]. Stamper et al., [22] used a data mining approach to generate automatic solicited hints for an algebra tutor, and showed that the tutor performed better compared to a non-adaptive version in terms of both number of problems solved and overall learning. AutoTutor, an ITS that provides adaptive feedback on answers to physics questions, was successfully evaluated in [8]. It should be noted that this ITS includes a simulation that helps find the answers but no support is given on how to use it.

On the other hand, providing adaptive support for exploratory learning environments such as ESims, is in early stages of research [16]. One defining characteristics of these environments is that there is usually no definition of correct behaviors. Students can explore the environment as they like. This makes it difficult to judge a priori and track which ensemble of user interaction behaviors should be the target of adaptive support [23]. Most of the work done so far on designing and evaluating adaptive feedback for learning environments that include interactive simulations has dealt with the challenge by limiting the exploratory nature of the interaction. For instance, the simulations developed by Johnson [7] provide feedback on how to behave in pre-defined cultural/language-related scenarios with clear definition of correct answers/behaviors. CTAT-VLab, [2] provides help on well-defined steps required to run a scientific experiment. Science Assistments, [6] provides feedback on the specific problem of controlling for variables in experimental design. Thus, although [7], [2] and [6] add some form of adaptive feedback to interactive simulations, the feedback targets pre-defined behaviors that help students find correct answer/solutions in the instructional domain. In contrast, in our work we designed and evaluated adaptive support for a more open-ended exploratory interaction, with no prior definition for correct actions/solutions. The support aims to help students use the simulation to explore the underlying algorithm effectively, not find a correct answer/solution.

The work closest in nature to ours is eXpresser, a simulation environment for learning algebra, which leverages a set of predefined feedback strategies to provide both unsolicited and solicited feedback to students based on the student's interactions [16]. A preliminary qualitative assessment of this feedback resulted in modest positive ratings for student perception, but no quantitative study has been published so far [16]. Thus, our work is the first (to the best of our knowledge) to provide a rigorous evaluation of an adaptive support mechanism for open-ended exploration in an ESim. Moreover, while in eXpresser a significant amount of manpower has been spent on defining the behaviors to be tracked in student model and the feedback strategies, in our work both adaptive interventions and the user model activating them are generated by mining the user interaction data with minimal human involvement.

THE AI SPACE CSP APPLET: ORIGINAL VERSION

A CSP consists of a set of variables, variable domains and a set of constraints on legal variable-value assignments. Solving a CSP requires finding an assignment that satisfies all constraints. The CSP applet illustrates the Arc Consistency 3 (AC-3 [20]) algorithm for solving CSPs represented as networks of variable nodes and constraint arcs (see Figure 1). AC-3 iteratively makes individual arcs consistent by removing variable domain values inconsistent with a given constraint, until all arcs have been considered and the network is consistent. Then, if there remains a variable with more than one domain value, a procedure called domain splitting is applied to that variable in order to split the CSP into disjoint cases so that AC-3 can recursively solve each case.

The CSP applet demonstrates the AC-3 algorithm dynamics via interactive visualizations on graphs using color and highlighting (Figure 1). The applet provides several mechanisms for the interactive execution of the AC-3 algorithm on a set of available CSPs. These mechanisms are accessible through the applet's toolbar (see top of Figure 1), or through direct manipulation of graph elements. The user can perform seven different actions: (1) Fine Step: use the fine step button to see how AC-3 goes through its three basic steps (selecting an arc, testing it for consistency, removing

domain values to make the arc consistent); (2) Direct Arc Click: directly click on an arc to apply all these steps at once; (3) Auto AC: automatically fine step on all arcs one by one using the auto arc consistency button; (4) Stop: pause auto arc consistency; (5) Domain Split: select a variable to split on, and specify a subset of its values for further application of AC-3 (see pop-up box in the left side of Figure 1); (6) Backtrack: recover alternative sub-networks during domain splitting; (7) Reset: return the graph to its initial status.

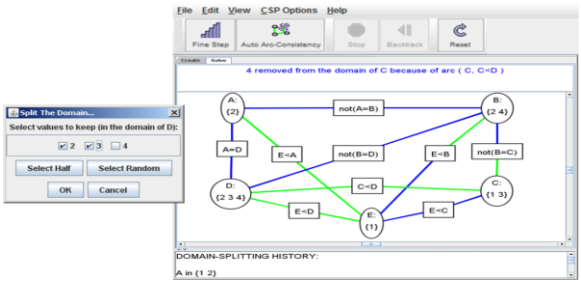

Figure 1. The CSP applet with an example CSP problem

The CSP applet is the product of an extensive process of iterative design and evaluation, and formal studies have shown that it can foster learning [1]. However, because the available affordances make it relatively easy to progress through a given task (e.g., finding a solution for a CSP problem), some students might not learn at best with this tool if they fail to use the available affordances to understand the underlying AC-3 processes. To address this issue, we have developed a new version of the applet that provides adaptive interventions designed to help these students use the simulation's functionalities more effectively.

In the next section, we briefly summarize the user modeling approach we leverage to determine in real-time during interaction if and why a user needs an intervention, proposed in [12]. The two following sections introduce the mechanism we designed to generate and deliver these interventions in the CSP applet.

MODELING STUDENT LEARNING IN THE CSP APPLET

The user modeling approach used in the adaptive-CSP applet consists of two phases: *Behavior Discovery* and *User Classification* (Figure 2). In *Behavior Discovery* (Figure 2-top), data from existing interaction logs is preprocessed into feature vectors where features consist of statistical measures that summarize the user's actions in the interfaces (e.g., action frequencies, time interval between actions). Each vector summarizes the behaviors of one user. A clustering algorithm then groups these vectors according to their similarities, thus identifying users who interact similarly with the interface. Next, association rule mining is applied to each cluster to extract its common behavior patterns, i.e. rules in the form of X→ *c*, where X is a set of feature-value pairs and *c* is the predicted cluster for the data points where X applies (Table 1 shows samples of these rules, which will be further explained in the next section).

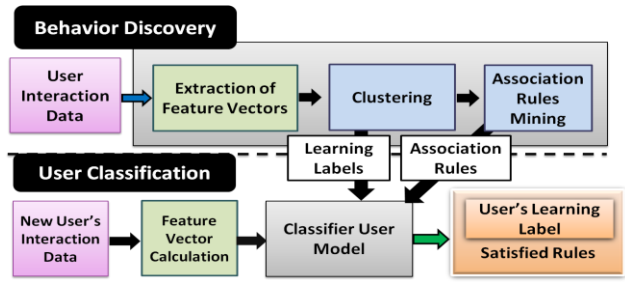

Figure 2. The user modeling framework

Clusters are then analyzed to identify how they relate to student learning performance. Thus, the *Behavior Discovery* phase generates groups of users who are associated with different levels of learning performance, as well as sets of interaction behaviors typical of each group.

The *User Classification* (Figure 2-bottom) phase uses the clusters and class association rules extracted in the *Behavior Discovery* phase to build an online classifier user model. This classifier assesses in real-time the (possibly evolving) learning performance of a new user by (i) incrementally building a feature vector based on the interface actions seen so far; (ii) classifying this vector in one of the available clusters. Note that the classification can change over time depending on the evolution of the user's interaction behaviors.

We built the user model for the adaptive CSP applet by applying Behaviors Discovery to a dataset of 110 users obtained from two previous studies on the this simulation [12,13]. From this dataset, the Behavior Discovery generated two clusters of users that achieved significantly different learning levels, labeled as High Learning Gain (HLG) and Low Learning Gain (LLG) groups from now on. Learning level was measured using pretest and posttest scores collected in the original studies, and a standard measure of learning performance (Percentage Learning Gains, or PLG), calculated as follows:

$$PLG = \begin{cases} \frac{PostT - PreT}{Max - PreT} \times 100 & if\ PostT - PreT > 0 \\ 0 & otheweise \end{cases} \quad (1)$$

where PreT and PostT are the student's test scores and Max is the maximum possible score.

Table 1 shows a subset of behavior patterns (association rules) discovered for each cluster from this dataset. A total of four and fifteen rules were found for the HLG and LLG respectively. We evaluated the accuracy of the classifier built from the HLG and LLG clusters and the corresponding association rules via 10-fold cross-validation on the available dataset. The classifier achieved an accuracy of 78.2% after observing only the first 25 percent of their interactions, indicating that this user model can be reliably used to trigger adaptive interventions from the early stages of the interaction.

Rules for HLG cluster:
Rule1: Auto AC frequency = Lowest
└ **Rule2:** Auto AC frequency = Lowest **and** Fine Step frequency = Lowest
Rule3: Direct Arc Click pause average = Highest
Rules for LLG cluster:
Rule4: Backtrack frequency = Highest **and** Direct Arc Click pause average = Lowest
Rule8: Auto AC frequency = Highest **and** Direct Arc Click frequency = Lowest
Rule 10: Reset frequency = Highest

Table 1. A selection of representative rules for HLG and LLG clusters in the CSP dataset

ADAPTIVE INTERVENTIONS FOR THE CSP APPLET

In addition to assigning a "label" to the user, the CSP's user model also returns the satisfied association rules causing that classification decision (Figure 3). These rules represent the distinctive interaction behaviors of a specific user so far, including a subset of behaviors associated with high learning (effective), as well as a subset of those associated with low learning (suboptimal). The adaptive interventions evaluated in this paper are designed specifically to target these behaviors (i.e., reinforcing the effective behaviors and discouraging the suboptimal ones).

For instance, rule 8 in Table 1 indicates that low learners show very high frequency of *Auto AC* actions, while rarely using the *Direct Arc Click* (DAC) action (values for behavior's frequency and latency are binned based on a mechanism described in [12]). A possible explanation of why both these behaviors are associated with limited learning is that they identify users who are not very engaged in the exploration process because they prefer to (i) run the algorithm to completion instead of stepping through it; (ii) leave to the applet the selection of the next arc to work on, rather than being proactive in choosing it.

Thus, this rule identifies two possible interventions (*intervention items* from now on) that address these suboptimal behaviors exhibited by low learners: (i) discourage excessive use of the Auto AC; (ii) encourage higher usage of DAC. Table 2 summarizes all behavior patterns (i.e., the intervention items) identified by the association rules that were generated in the *Behavior Discover* phase.

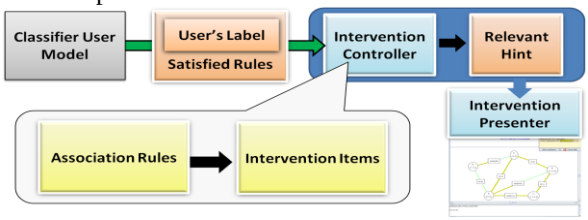

Figure 3. The intervention mechanism

The intervention mechanism in the CSP applet consists of two components: intervention controller and intervention presenter (Figure 3). Intervention controller selects the next intervention to be given and intervention presenter delivers the respective hint. The process of providing adaptive interventions starts by identifying which of the items in Table 2 are relevant at any given point of a user's interaction with the applet (intervention controller in Figure 3). More specifically, when a user is classified as LLG, the intervention controller identifies which detrimental behaviors they should stop performing and/or which beneficial behaviors they should adopt, based on the association rules that caused the classification.[1] For instance, if the user is being classified as LLG, and one of the association rules activated says that the frequency of the *Direct Arc Click* action is lower than a threshold (e.g. Rule 8 in Table 1), then the intervention controller marks the *DAC_fr* intervention in Table 2 as relevant. This intervention entails prompting the user to perform more Direct Arc Click actions.

Intervention Code	Intervention Description
DAC_fr	Using Direct Arc Click more often
DAC_PA	Spending more time after performing Direct Arc Clicks
Reset_fr	Using Reset less frequently
AAC_fr	Using Auto Arc-consistency less frequently
DS_fr	Using Domain Splitting less frequently (only when appropriate)
FS_PA	Spending more time after performing Fine Steps
BT_fr	Using Back Track less frequently (only when appropriate)
FS_fr	Using Fine Step less frequently
Reset_PA	Spending more time after performing after resetting for planning

Table 2. Description of hints

There may be several intervention items that are relevant given a user's behaviors at any given point of their interaction with the CSP applet (*active* intervention items from now on). To avoid confusing or overwhelming the user, in this version of the adaptive applet we decided to deliver one intervention at a time, chosen based on a ranking that reflects the relevance of the behaviors being targeted for learning. More specifically, the intervention controller gives each intervention item a score calculated as the sum of the weights of the association rules which triggered that item. The weight of a rule in turns indicates its importance in classifying a user as a high or low learner. At each hinting opportunity, the intervention controller chooses the intervention item with the highest score among the active items. The next section illustrates the design and rationale behind the mechanism that delivers interventions (intervention presenter in Figure 3) while trying to maximize effectiveness and minimize intrusiveness of the process.

[1] Potential causes of the suboptimal interaction (e.g., lack of needed skills, cognitive overload, etc.) are not revealed by our approach. In this respect, our model and resulting hints are rather shallow. They attempt to correct the behaviors, but cannot do anything to address their causes.

DELIVERING THE ADAPTIVE INTERVENTIONS

Our first design principle is to deliver adaptive interventions incrementally, following well established practices in the Intelligent Tutoring Systems literature [15]. Thus, each selected intervention item is first delivered with a textual hint that prompts or discourages a target behavior, followed when needed by a textual hint that reiterates the same advice, accompanied by a related interface adaptation that can help the user follow the advice (e.g., highlighting relevant interface items).

Delivering adaptive interventions also requires deciding whether the interventions should be subtle or forceful. Subtle interventions are in the form of suggestions that can be easily ignored by the user. Forceful interventions make the user follow the related advice by reducing or eliminating user's options for the next action. We decided to try the subtle approach for the adaptive-CSP applet because, although with this approach the user can easily decide to ignore the system's suggestions (even when they are appropriate), it has the very desirable advantage of being less intrusive than the forceful approach. Therefore, from a usability point of view, it makes sense to try and see whether subtle adaptive interventions can already improve the effectiveness of the CSP applet. The general mechanism to deliver subtle incremental adaptive interventions in the CSP applet works as follows:

(i) each intervention item selected for delivery (*target item* in the rest of this section) is first presented as a textual hint message shown in a box at the upper left corner of the applet, as shown in Figure 2a (level-1 hint). This message is phrased as a suggestion for behaviors to be adopted or avoided. For instance, a level-1 textual hint for the DAC_fr intervention item in Table 2 is "*Do you know that you can tell AC-3 which arc to make consistent by clicking on that arc?*" which aims to promote the *Direct Arc Click* action. In order to draw user's attention to the new message, the hint box first appears in the center of the screen, and then quickly moves to the upper left corner of the applet.

(***ii***) After receiving a level-1 hint on the *target item*, the student is given some time to change her behavior accordingly (a *reaction window* equal to 20 actions). During this time, the user model will keep updating the feature vector describing the user interaction behavior, its classification and the ranked list of active intervention items (excluding the *target item*), delivering any of them as needed. To avoid overwhelming the user with hints, a maximum frequency of one hint per ten actions is enforced.

(iii) At the end of the time window, based on the updated feature vector the user model determines whether the user has followed the hint for the *target item* or not. If at this point the preconditions for the adaptation rule that generated the level-1 hint for the *target item* are still satisfied, then the user has not followed the hint and the *target item* is selected for delivery again.

(***iv***) In this case, a level-2 hint is delivered, consisting of both a text message as well as the highlighting of relevant interface elements that can help the user follow the hint. Figure 2b shows an example of level-2 hint for the DAC_fr intervention item. The text messages in level-2 hints are more detailed than for level-1 hints. The first part of a level-2 hint reiterates the level-1 hint with a slightly different wording, while the second part provides a justification for the suggestion and, if relevant, mentions the highlighted elements of the interface. For instance, in Figure 2b, the arcs of the displayed CSP graph that can be selected via Direct Arc Click are highlighted in yellow. The phrase "relevant arcs" in the hint message is highlighted similarly to create a visual link between the text and the change in the graph. The highlight effect is automatically removed as soon as the user performs a relevant action (e.g., user clicks on an arc for DAC_fr). The highlights can also be manually toggled using the "hide/show highlights" button at the bottom of the hint box (see Figure 2b).

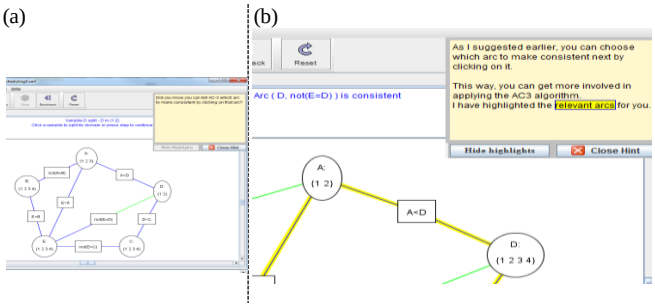

Figure 4: Sample Level-1 and Level-2 hints

(***iv***) If the user's behavior during the reaction window following a level-2 hint shows that the user still did not follow the system's suggestion, the corresponding intervention item will be put back on the list of active items and, if selected, it will be delivered starting again from a level-1 hint.

The hint delivery mechanism described above is the result of an iterative design and evaluation process based on three different pilot studies. All these studies had a similar set up: participants first studied a textbook chapter on the AC-3 algorithm, and then used the CSP applet to solve two CSPs, while their gaze was tracked with a Tobii T120 eye-tracker. A follow-up questionnaire and interviews were administered to gain a qualitative evaluation of relevant aspects of the interaction.

The first pilot was a Wizard-of-Oz during which an experimenter triggered the interventions based on a set of adaptation rules similar to those described earlier. The goal of the study was to pilot test the general two-level hints approach in terms of visibility, intrusiveness and follow rate of the interventions, and involved 6 second and third year computer science students.

Participants generally did not find the hint messages intrusive, and they rated them as useful. The interviews however, uncovered various issues with visibility: for the

textual hints, subjects reported that sometimes it was hard to notice when a new message appeared. For the highlights provided in the level-2 hints (which in that version were based on bolding the relevant interface elements), some participants did not notice them at all; others saw them but failed to make the connection with the textual hint. These observations were supported by the gaze data collected with the eye-tracker.

The final design presented in this section is the result of changes we made to the original design to address these issues, tested with two subsequent pilot studies. More specifically:

- Visibility of new textual hints was increased by adding the motion and change in initial positioning of the hint box as describe earlier.
- Visibility of the highlights for level-2 hints was increased by testing two alternative ways to emphasize the target elements, namely yellow highlighting vs. Blinking (increase in attentional draw [5]). Yellow highlighting proved to be the preferred and more effective method based both on user's feedback as well as gaze data.
- The connection between the level-2 hint messages and the related interface changes was emphasised by both mentioning in the hint text that the relevant elements are highlighted, and by using the same highlight effect on keywords in the text.

The last two pilots were also used to (i) evaluate ways to give user control over displaying the interface changes by adding the "hide/show highlights" buttons (Figure 2b); (ii) adjust the thresholds controlling the hint delivery rate.

USER STUDY

To investigate the effectiveness of the intervention mechanism, we conducted a user study that tested the following hypothesis:

H1: Under similar conditions, participants who work with the adaptive-CSP applet (i.e., receive adaptive support) will have a higher learning performance compared to ones that work with the original CSP applet (control condition).

Procedure

The experiment was a between-subjects design, fitting in a single session lasting at most 2 hours. There were two randomized experimental conditions: in the *control* condition participants worked with the original CSP applet; in the *adaptive* condition participants used the adaptive-CSP applet.[2]

During the study each participant: (1) studied a textbook chapter on the AC-3 algorithm[3]; (2) wrote a pre-test on the concepts covered in the chapter; (3) watched an introductory video on how to use the main functionalities of the CSP applet; (4) used the CSP applet to solve three CSPs. (5) took a post-test analogous to the pre-test; and (6) took a questionnaire on aspects relevant to both groups (described below). Participant in the Adaptive condition took an additional questionnaire on aspects specific to the adaptive-CSP applet. The study ended with a follow up interview that solicited explanations for user ratings.

Study Material

The study tests involved items selected from a standard bank of homework questions used in the introductory AI course at a university in North America. They required students to apply knowledge of different aspects of the AC-3 algorithm on selected CSP networks. The maximum possible score for both tests was 25, based on existing marking schemes.

All study participants took a post-questionnaire (*general questionnaire* from now on) in which they rated the following three statements (using a Likert scale from 1: strongly disagree to 5: strongly agree):

- I felt more confident answering the post-test questions compared to pre-test questions (*improved confidence*)
- I think I did better in the post-test compared to pre-test (*perceived learning*)
- I found the applet helpful in understanding the AC-3 algorithm (*helpfulness of the applet*)

The additional questionnaire administered to participants in the Adaptive condition (referred to as *intervention questionnaire*) included 18 items on different elements of the intervention mechanism, also ranked on a 5-point Likert scale. Pairs of similar statements were provided to evaluate hint messages and highlights in terms of *relevance*, *usefulness*, *noticeability*, *intrusiveness* and *annoyance*:

- In general, the [hint messages | highlighting of the interface items] were appropriate given my behavior (*relevance*).
- The [messages displayed in the hint box| highlighting of the interface items] were useful for me (*usefulness*).
- I easily noticed the [new messages displayed in the hint box| highlighting of the interface items] (*noticeability*).
- I found the [hint box| highlighting of the interface items] intrusive.(*intrusiveness*)
- I found the [hint box| highlighting of the interface items] annoying (*annoyance*).

It should be noted that, in our first pilot, we did not have items for annoyance, because we thought intrusiveness would capture a user's negative perception of how the interventions interfere with their work. However, some

[2] It should be noted that this design does not fully isolate the contribution of the hints from the adaptive delivery component. We did plan to have a third condition with hints triggered randomly, but we could not recruit enough users with the right background for a 3-way study. We chose not to use this random condition as control because we used it in our pilots and user feedback indicated irritation with it. Thus we felt that it would not be a good control due to its intrusiveness.

[3] This is all the instruction participants get on the AC-3 algorithm. Hints in the adaptive condition did not provide additional instruction on the algorithm.

participants who rated the hints as intrusive commented in follow-up interviews that they did not perceive these interruptions as negative; therefore a new set of items targeting annoyance was added.

The remaining 8 items focused on specific attributes of each element of the intervention mechanism, not discussed in this paper due to space constraints. Screen shots of all the relevant intervention elements were provided in the questionnaire to facilitate recollection.

Study tasks

The 3 CSP problems used in the study task were selected to provide incremental coverage of all relevant aspects of the AC-3 algorithm, as well as different outcomes. The first CSP had one solution that could be found without domain splitting. The second CSP had no solution and required one domain splitting to reach this conclusion. The third problem had 3 solutions, which required multiple domains splitting and backtracking to be found.

Participants were given an instruction sheet indicating that they needed to report the outcome of using AC-3 on each of the problems. They were given a minimum of 15 minutes for working on this activity, but no maximum time limit.

Study Participants

Thirty eight university students (11 female) participated in the experiment. We selected the number of participants by performing a power analysis [14] a priori on the parameters of our experimental design, defined to detect large effect size of $d = 1.0$ in terms of performance with 0.8 power.

Participants were selected such that they did not have any knowledge on solving constraint satisfaction problems beforehand (a new concept); however they had the computer science pre-requisite knowledge necessary to learn this concept (e.g., basic graph theory and algebra). We used flyers, email and social media to recruit participants with these prerequisites at a North American university.

RESULTS FOR PERFORMANCE MEASURES

We looked at the impact of study condition on three different measures of performance: *learning performance* from pretest to posttest, as measured by the Percentage Learning Gains (PLG) defined in equation 1; *task performance*, which is a score out of 10, given based on the number of correct solutions found for the 3 CSP problems using the applet (including identifying the CSP with no solution); *task time*, i.e., the time in minutes spent working on the 3 CSP problems using the applet; Summary statistics for the three performance measures and pre-test for each condition are shown in Table 3.

An independent samples t-test on the pretest scores of the two conditions found no significant difference ($t(36) = .189$, $p = .851$, $\eta^2 = .001$), indicating that the random assignment of participants to conditions was successful in generating two groups with comparable initial knowledge.

In addition to studying the impact of experimental conditions on our performance measures, we are also interested in verifying whether the impact is moderated by a student's initial knowledge, as measured by pretest. Thus, we ran a moderated multiple regression analysis [4] for each of our performance measures, with that measure as the dependent variable. Independent variables (IV) are condition (coded as -1 for Control and 1 for Adaptive), pretest and the interaction between condition and pretest. This interaction is calculated by first centering pretest (by subtracting the sample mean) and then multiplying pretest and condition together. Alpha level is set to 0.05, and Pearson r is reported to show the effect size (small for .1, medium for .3, and large for .5 [3]). Models are adjusted for family-wise error by applying the Bonferroni correction. Data screening did not suggest problems with assumptions of normality and linearity. The multicollinearity assumption was not violated in any of the models (based on Variance Inflation Factors [17]). For each independent variable, we report both standardized regression coefficient (β) and squared semipartial correlation (sr^2).

Measure	Pretest		PLG		Task Performance		Task Time	
Condition	Cont.	Adapt.	Cont.	Adapt.	Cont.	Adapt.	Contr.	Adapt.
Mean	9.66	9.37	28.59	54.29	8.58	8.42	18.32	18.37
Std. Dev.	5.553	3.696	19.045	24.784	2.434	2.364	3.198	5.756

Table 3. Descriptive Statistics for the performance measures

Results on learning performance

The overall regression model significantly predicted PLG, $F(3, 34) = 7.435$, $p = .001$, $R^2 = .396$, adjusted $R^2 = .343$. There was a significant Pretest×Condition interaction, $\beta = -.296$, $t(34) = -2.048$, $p = .048$, $sr^2 = .075$, with a medium effect size ($r = .331$). The interaction accounts for an additional 7.5% of the variance in PLG. There was also a significant main effect for condition $\beta = .517$, $t(34) = 3.878$, $p < .001$, $sr^2 = .267$, with a large effect size ($r = .554$), however the main effect for pretest was not significant ($\beta = .128$, $t(34) = .886$, $p = .382$, $sr^2 = .014$).

The main effect for condition shows that the difference in learning performance of the two groups in the study is significant; with the Adaptive condition having greater learning performance compared to the Control condition (the average standardized difference is 1.034 which is slightly more than one standard deviation). This confirms our hypothesis H1.

The interaction effect indicates the effect of condition on PLG is moderated by pretest. To better qualify this finding, we plotted the effect of condition on PLG at three different levels of standardized pretest scores labeled (z-scores) as high (z=1), average (z=0), and low (z=−1); (Figure 4). The line equations are derived from the standardized β values. As shown in Figure 4 the effect of condition on PLG is highest for the low pretest group and decreases as pretest performance improves, i.e., the interventions are most effective on the students with lower initial knowledge. This is

very encouraging for us because these are the students who need the most help.

Results on task performance and completion time

The regression models with the aforementioned IVs, did not significantly predict task performance, $F(3, 34) = 1.650$, $p = .196$, $R^2 = .127$, adjusted $R^2 = .050$ or task time, $F(3, 34) = .554$, $p = .649$, $R^2 = .047$, adjusted $R^2 = .038$. These results indicate that condition did not significantly explain the variance of task performance or task time, i.e., the two conditions are not significantly different in terms these two measures. The fact that participants in the Adaptive and Control conditions performed equally well in the task with the CSP applet, while achieving very different results for learning gain, confirms that unaided interaction with the CSP applet can lead students to find solutions without obtaining a clear understanding of the process.

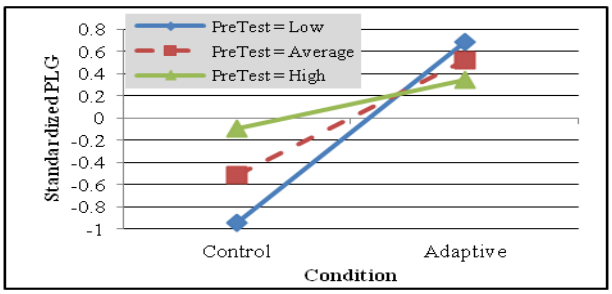

Figure 5. Interaction between Pretest and Condition

Given the system design and study results, we think the students in the adaptive condition learned more because the hints help them correct some of the suboptimal ways in which they use the applet to explore the AC-3 algorithm. Although we do not have hard data to support this intuition, post-hoc interviews provide some insights on how the hints affected user behaviors. Students reported that hints helped them to "control their pace thus understand the process better", "explore more actions", "direct their focus".

RESULTS ON INTERVENTIONS ACCEPTANCE

Success of any adaptive support mechanism highly depends on the users' perception of its quality. To gauge this perception, in this section we will look at both objective and subjective indicators: respectively, the actual follow rate for the hints and post-questionnaires ratings.

Actual hint follow rate

For every hint received by a user in the Adaptive condition, we check the user model after the reaction window to see if the user is still showing the behavior that triggered the hint. If the user was not showing that behavior, the hint is marked as followed and vice versa. Table 4 shows the descriptive statistics for number of hints given in total, at level 1 and level 2, along with the follow rate for each. One participant did not receive any hints, and thus it is not included in this analysis as well as in the analysis of subjective measures in the next section.

	N	Mini	Max	Mean	Std. Dev.
Number of Hints	18	1	15	7.06	4.65
Level-2 Hints	18	0	8	1.56	2.09
Follow rate Level-1 (%)	18	16.67	100	62.96	32.95
Follow rate Level-2 (%)	10	0.0	100	73.33	37.84
Follow rate overall (%)	18	16.67	100	64.44	31.65

Table 4. Descriptive Statistics for Hints

As shown in Table 4, the overall follow rate is 64.4 percent, indicating a relatively good acceptance of the adaptive suggestions. Notably, the average follow rate for hints increases from 63 percent for level-1 hints to 73.3 percent for level-2 hints. This validates the 2-level design of our interventions.

Analysis of post-questionnaire ratings

We first compare the user ratings on the general post questionnaire items, which were filled by both the control and adaptive conditions. Then, we discuss the ratings provided by participants in the Adaptive condition for a selected set of items from the interventions questionnaire.

User Ratings for the General Questionnaire

We used Mann-Whitney U tests to compare the ratings of the three items in the general post questionnaire. The distributions in the two groups did not differ significantly for Improved Confidence (Mann–Whitney $U = 150.50$, $n_1 = n_2 = 19$, $P = .322$ two-tailed) , Perceived Learning (Mann–Whitney $U = 142.50$, $n_1 = n_2 = 19$, $P = .223$ two-tailed) and for Helpfulness of the applet (Mann–Whitney $U = 171.50$, $n_1 = n_2 = 19$, $P = .766$ two-tailed). The ratings were generally positive for all three items across conditions (see Figure 5). The positive ratings reflect the value of the CSP applet for visualizing the steps of the AC-3 algorithm compared to studying examples in the textbook, as revealed by the follow up interviews (e.g., It is "much easier to learn" from the applet, I could "see the outcomes in real time" and "step-by-step", I easily "learned from my mistakes").

Notably, however, there is a mismatch between the relatively high perceived learning for the Control condition, and their actual learning performance. Possibly, perceived learning is positively influenced by their task performance, which as illustrated in the previous section was as good as that of the students in the Adaptive condition. The mismatch between perceived and actual learning is concerning because these students will not feel the need to explore more problems to improve their learning, if left to their own devices.

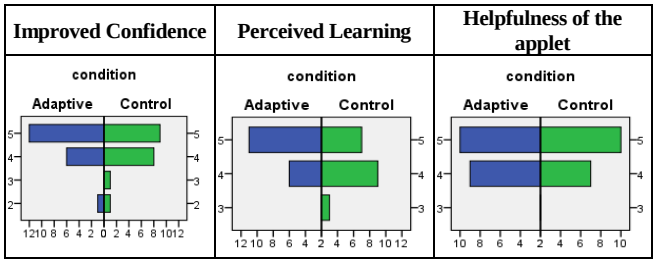

Figure 6. User ratings of the general items (5: most positive)

User Ratings for the Interventions Questionnaire

In this section we report user ratings for items on the Intervention questionnaire that evaluated hint messages and highlights in terms of *relevance*, *usefulness*, *noticeability*, *intrusiveness* and *annoyance* (as described in the *Study Materials* section). Due to the fact that highlights were only present for the level-2 hints, and only 8 of the 18 participants received a level-2 hint with highlights, we report ratings for hint messages and highlights separately.

To simplify the analysis in the following discussion we combined the "agree" and "strongly agree" ratings as *agree-all* and "disagree" and "strongly disagree" ratings as *disagree-all*. Using this aggregation, the rating distribution of the mentioned items for the hint box were: *relevance* (67% agree-all, median = "agree"), *usefulness* (44% agree-all, median = "neutral"), *noticeability* (100% agree-all, median = "agree"), *intrusiveness* (27% agree-all, median = "disagree") and *annoyance* (39% agree-all, median = "neutral"). The majority of the users (over 67%) found the hints relevant to their behavior, whereas 28% found them irrelevant. Usefulness received fewer negative ratings, with only 22% reporting that they did not find the hint useful, however there were more neutral ratings, resulting in fewer users agreeing over usefulness (44%). Based on post interviews, one prominent reason given for negative rating of usefulness and relevance was the nature of the textual hints. Some users reported expecting the hints to give more explicit help on how to solve the problem at hand, rather than telling them how to use the applet, possibly because they focused on getting the job done (e.g., "I wanted to find the solution and the hints kept telling me to slow down") as opposed to trying to improve their understanding of the AC-3 algorithm, which is the task that is supported by the intervention mechanism.

Our design decision to increase the prominence of the appearance of the hint box (to ensure new hint messages are not missed), succeeded in improving noticeability of these hints, reaching 100% of agree-all rating, at the expense of only a small increase in ratings for intrusiveness (28% agree-all, which is 6% higher than the analogous ratings from the first pilot). Compared to results from our pilot studies, a higher number of users rated the hint box as annoying. In the follow-up interviews, the majority of who reported annoyance indicated as the reason receiving hints that they did not agree with repeatedly (i.e., beyond a level-2 hint). This happens when a user insists on not following a hint, and this hint stays ranked high in the list of active intervention items, indicating high relevance for learning. These negative ratings are mirrored by the actual follow rate of repeated hints, for which we found a trend to be lower than for hints given up to level 2. We cannot judge from the available data to what extent hints given repeatedly were actually justified (recall that the user model is not 100% accurate), but regardless our results indicates that we should consider reducing the number of repeated hints in future versions of the applet. In the interview, none of the users attributed their annoyance to the animation used for appearance of the hint box, which provides further positive feedback on the current design of the hint box.

Ratings for the corresponding questionnaire items for highlights were generally more positive than for the hint box, with the exception of noticeability: *relevance* (88% agree-all, median = "agree"), *usefulness* (75% agree-all, median = "agree"), *noticeability* (88% agree-all, median = "agree"), *intrusiveness* and *annoyance* (0% agree-all, median = "disagree" for both items). Positive ratings for highlights show an effective balance between noticeability and intrusiveness/annoyance.

DISCUSSION AND FUTURE WORK

In this paper, we present the formal evaluation of user-adaptive interventions designed to guide students in learning at best through open-ended exploration of an educational interactive simulation called AIspace CSP applet. There is extensive evidence that not all students can learn well from interactive simulations, due to the rather unstructured and open-ended nature of the interaction that they support. Our work is the first to show formal results on the effectiveness of providing support for open-ended exploration in an interactive simulation. An additional contribution of our work is that support is provided via strategies that are automatically learned from previous learner interaction data.

We described a mechanism for selecting adaptive suggestions based on the output of a user model that identifies in real-time if a student is learning well from the CSP applet and, if not, why. We also described the design process that generated a 2-level subtle method of delivering interventions using both text messages and interface highlights. A formal evaluation of these interventions against the original version of the CSP applet shows very encouraging results for both users learning and their perception of the system, but also provides valuable insights for further improvements.

Students in the two experimental conditions performed similarly in terms of time on task and task performance, but students in the adaptive condition learned significantly more than students in the control condition. Furthermore, the effect of interventions on learning is even more pronounced for students with lower levels of initial knowledge. In terms of users' acceptance of the intervention mechanism, there was a positive attitude towards the applet in general and users followed 64.4% of the recommendations given to them. The users' ratings of the interventions were generally positive; however they indicate that the number of repeated hints should be reduced, since that was a source of annoyance for users. Thus, we are planning to further investigate how to strike the right balance between providing hints when they are needed and not overwhelming the user. We also plan to investigate having a third condition in a new study where hints are given in a fixed manner to better isolate the effect of getting any hints from receiving them adaptively. We will also work on adding adaptive interventions to other applets in

the AIspace suite of adaptive simulations, to test the generalizability of our approach. Finally, we are aware of the limitations of a lab setting study and plan to make the applet available online and use remote logging for further analysis of its performance.

REFERENCES

1. Amershi, S., Carenini, G., Conati, C., Mackworth, A.K., and Poole, D. Pedagogy and usability in interactive algorithm visualizations: Designing and evaluating CIspace. *Interacting with Computers 20*, 1 (2008), 64–96.
2. Borek, A., McLaren, B.M., Karabinos, M., and Yaron, D. How Much Assistance Is Helpful to Students in Discovery Learning? In U. Cress, V. Dimitrova and M. Specht, eds., *Learning in the Synergy of Multiple Disciplines*. Springer Berlin Heidelberg, 2009, 391–404.
3. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences.* Lawrence Erlbaum, 1988.
4. Cohen, J., West, S.G., Aiken, L.S., and Cohen, P. *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences.* Lawrence Erlbaum, Mahwah, N.J, 2002.
5. Gluck, J., Bunt, A., and McGrenere, J. Matching attentional draw with utility in interruption. In *Proc. CHI '07*, (2007), 41.
6. Gobert, J.D., Montalvo, O., Toto, E., Sao Pedro, M., and Baker, R.S.J. d. The Science Assistments Project: Scaffolding Scientific Inquiry Skills. In *Proc. Intelligent Tutoring Systems*, (2010), 445–445.
7. Hussain, T.S., Roberts, B., Menaker, E.S., et al. Designing and developing effective training games for the US Navy. *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, NTSA (2009).
8. Jackson, G.T., Olney, A., Graesser, A.C., and Kim, H.J. AutoTutor 3-D Simulations: Analyzing user's actions and learning trends. In *Proc. the 28th annual meeting of the Cognitive Science Society*, (2006), 1557–1562.
9. De Jong, T. Technological Advances in Inquiry Learning. *Science 312*, 5773 (2006), 532–533.
10. Van Joolingen, W.R., De Jong, T., and Dimitrakopoulou, A. Issues in computer supported inquiry learning in science. *J. of Computer Assisted Learning 23*, 2 (2007), 111–119.
11. Kardan, S. Data Mining for Adding Adaptive Interventions to Exploratory and Open-Ended Environments. In *Proc. UMAP 2012*, Springer Berlin / Heidelberg, 2012, 365–368.
12. Kardan, S. and Conati, C. A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. In *Proc. the 4th Int. Conf. on Educational Data Mining*, (2011), 159–168.
13. Kardan, S. and Conati, C. Exploring Gaze Data for Determining User Learning with an Interactive Simulation. In *Proc. UMAP 2012*, Springer Berlin / Heidelberg, 2012, 126–138.
14. Lenth, R.V. *Java Applets for Power and Sample Size.* 2006. http://www.stat.uiowa.edu/~rlenth/Power
15. Maloy, R.W., Verock-O'Loughlin, R.-E.A., Edwards, S.A., and Woolf, B.P. *Transforming Learning with New Technologies.* Pearson, Boston, 2013.
16. Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., and Noss, R. Design requirements, student perception indicators and validation metrics for intelligent exploratory learning environments. *Personal and Ubiquitous Computing*, , 1–16.
17. O'brien, R.M. A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity 41*, 5 (2007), 673–690.
18. Perkins, K., Adams, W., Dubson, M., et al. PhET: Interactive Simulations for Teaching and Learning Physics. *The Physics Teacher 44*, 1 (2006), 18–23.
19. Ploetzner, R., Lippitsch, S., Galmbacher, M., Heuer, D., and Scherrer, S. Students' difficulties in learning from dynamic visualisations and how they may be overcome. *Computers in Human Behavior 25*, 1 (2009), 56–65.
20. Poole, D.L. and Mackworth, A.K. *Artificial Intelligence: Foundations of Computational Agents.* Cambridge University Press, 2010.
21. Salajan, F.D., Perschbacher, S., Cash, M., Talwar, R., El-Badrawy, W., and Mount, G.J. Learning with web-based interactive objects: An investigation into student perceptions of effectiveness. *Computers & Education 53*, 3 (2009), 632–643.
22. Stamper, J., Eagle, M., Barnes, T., and Croy, M. Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. *Int. J. Artif. Intell. Ed.22*, 1 (2013), 3–17.
23. Ting, C.Y., Zadeh, M., and Chong, Y.K. A decision-theoretic approach to scientific inquiry exploratory learning environment. In *Proc. Intelligent Tutoring Systems*, (2006), 85–94.
24. VanLehn, K. The Behavior of Tutoring Systems. *Int. J. Artif. Intell. Ed. 16*, 3 (2006), 227–265.
25. Westerfield, G., Mitrovic, A., and Billinghurst, M. Intelligent Augmented Reality Training for Assembly Tasks. In *Proc. Artificial Intelligence in Education*,(2013), 542–551.