In: Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013).
Approved for Public Release, Distribution Unlimited.

Drill Evaluation for Training Procedural Skills

Karen Myers, Melinda Gervasio, Christian Jones, Kyle McIntyre, Kellie Keifer

SRI International, Menlo Park, CA
{<firstname>.<lastname>@sri.com}

Abstract. The acquisition of procedural skills requires *learning by doing*. Ideally, a student would receive real-time assessment and feedback as he attempts practice problems designed to exercise the targeted skills. This paper describes an automated assessment and feedback capability that has been applied to training for a complex software system in widespread use throughout the U.S. Army. The automated assessment capability uses soft graph matching to align a trace of student actions to a predefined gold standard of allowed solutions, providing a flexible basis to evaluate student performance, identify problems, give hints, and suggest pointers to relevant tutorial documentation. Collectively, these capabilities facilitate self-directed learning of the training curriculum.

Keywords: procedural skills, automated assessment, relaxed graph matching

1 Introduction

Today's workers require a broad and growing set of *procedural skills,* which involve learning multistep procedures to accomplish a task. Procedural skills apply to both physical environments (e.g., how to repair a device, how to build a shed) and online environments (e.g., how to create a pivot table in Excel).

This paper reports on a system called Drill Evaluation for Training (DEFT) that was developed to facilitate the learning of procedural skills related to the use of a complex piece of software. More specifically, DEFT provides an automated assessment capability to evaluate students' performance as they learn how to use the Command Post of the Future (CPOF)—a collaborative geospatial visualization environment system used extensively by the U.S. Army to develop situational awareness and to plan military operations. Although a powerful tool, CPOF can be difficult to learn; furthermore, CPOF skills decay rapidly when not in regular use. Because soldiers have limited availability for formal training sessions, achieving and maintaining necessary skills presents a significant challenge.

DEFT addresses the training problem for CPOF through automated support for assessing learned skills and providing targeted feedback designed to further student understanding. An automated capability of this type would reduce the burden on instructors in classroom settings, thus enabling them to provide more personalized attention to individual students. It would also enable students to pursue independent supplemental training beyond a formal classroom setting.

We begin the paper with background on CPOF and its training curriculum, followed by a technical overview of DEFT. We then present results of a user study that assessed the usability and utility of DEFT for CPOF training. We close with a discussion of related work, a summary of contributions, and directions for future work.

2 Command Post of the Future (CPOF)

CPOF is a state-of-the-art command and control (C2) visualization and collaboration system. The CPOF software is part of the U.S. Army's Battle Command System, and as such is standard equipment for virtually every Army unit. Since its inception in 2004, thousands of CPOF systems have been deployed. Its usage spans organizational echelons from Corps to Battalion in functional areas that include intelligence, operations planning, civil affairs, and engineering. CPOF is used extensively to support C2 operations for tasks covering information collection and vetting, situation understanding, daily briefings, mission planning, and retrospective analysis [4].

CPOF uses geospatial, temporal, tabular, and quantitative visualizations specifically tailored to information in the C2 domain. Users can collaborate synchronously in CPOF by interacting with shared products. The ability to dynamically incorporate new information is critical to the success of any C2 operation; CPOF's "live" visualizations continually update in response to changes sourced from user interactions or underlying data feeds, thus ensuring that data updates flow rapidly to users.

The U.S. Army offers the Battle Staff Operations Course (BSOC) to provide instruction to students on basic CPOF interaction skills. Much of what is taught in the BSOC is procedural, i.e., determining what steps to perform and in what order to achieve a particular result. The following provides a portion of an exercise from the BSOC course materials: *Create a 2D map. Create a notional unit; name it A10 #X 1v2. Edit the size, type, and affiliation. Place the unit on the 2D map.*

An analysis of an examination used to test student mastery of BSOC material showed that 69% of the questions required demonstration of procedural skills; another 6% involved true/false or multiple-choice questions; the remaining 25% required short-answer responses. Similar exercises are used within the course itself to enable students to apply the classroom knowledge in a hands-on fashion. This predominance of procedural skills within the BSOC curriculum motivated the development of DEFT, as having an ability to automatically assess student performance could dramatically alter the manner in which CPOF training is conducted.

3 DEFT Technical Components

DEFT performs real-time monitoring of students as they attempt to complete exercises (see Fig. 1). While a student works on an exercise, DEFT logs a trace of the student's actions. That trace is compared to a representation of allowed solutions to the exercise (the *gold standard*) to create assessment information that identifies conceptual errors or mistakes, provides guidance in the form of hints to help the student complete a task, and suggests links to contextually relevant training materials.

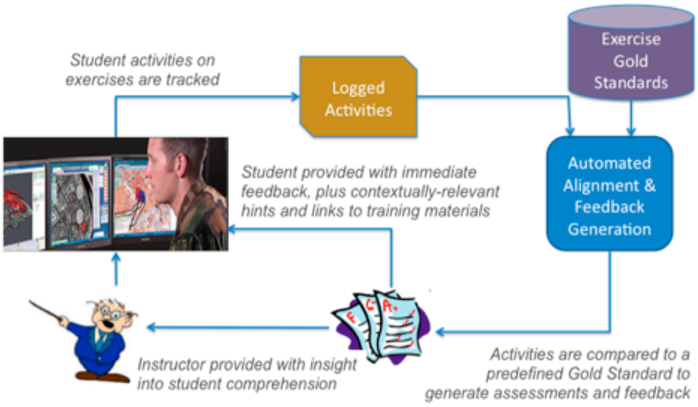

Student activities on exercises are tracked
Logged Activities
Exercise Gold Standards
Automated Alignment & Feedback Generation
Student provided with immediate feedback, plus contextually-relevant hints and links to training materials
Instructor provided with insight into student comprehension
Activities are compared to a predefined Gold Standard to generate assessments and feedback

Fig. 1. Automated assessment in DEFT

3.1 Gold Standard Representation

The *gold standard* defines the space of acceptable solutions to an exercise. For BSOC exercises, there can be numerous solutions that involve different actions and orderings between them, along with significant variability in the specific objects that are created or manipulated. This richness precludes an explicit enumeration of gold standard solutions as a collection of totally ordered actions.

Instead, we represent the gold standard as one or more traces obtained through demonstrations of correct solutions to an exercise, augmented with *annotations* that define allowable variations from the trace. A gold standard defines a partial ordering on the steps of a trace, where a step can be a (parameterized) CPOF action, a class of actions, or set of options, each of which is itself a partially ordered set of steps. The annotations take the form of constraints over steps or parameters. Currently, DEFT supports action ordering constraints, parameter equality constraints, parameter value constraints (between parameters and constant values), and a limited set of query constraints. Query constraints capture requirements on the application state or on object properties that cannot be determined from the arguments of the actions themselves. The abstractions provided by this scheme can yield compact representations of large solution spaces.

We anticipate that instructors will play a critical role in gold standard development by providing solution traces and annotating them. However, we can also leverage automated reasoning and machine learning techniques to facilitate the process. For example, we can apply heuristics to determine default annotations and generalize over parameters and actions from multiple examples.

3.2 Alignment

The automated assessment capability in DEFT centers on determining a mapping from the student's submitted response for an exercise to the predefined gold standard for that exercise. We have framed this alignment problem as a form of *inexact seman-*

tic graph matching in which a similarity metric based on graph edit distance is used to rate the quality of the mappings. Graph edit distance measures the number—more generally, the cumulative cost—of graph editing operations needed to transform the student response into an instance consistent with the gold standard. Intuitively, finding the lowest-cost alignment corresponds to DEFT finding the specific solution the student is most likely to have been attempting.

To use this graph matching approach in DEFT, we represent the gold standard as one or more *solution graphs*, with each graph representing a family of possible solutions to the exercise. Actions and their parameters are nodes; parameter roles within actions are links; and required conditions within the solution (e.g., action orderings, values of textual or numerical parameters) are constraints. The student response is represented similarly as a *response graph*.

Alignment involves finding the mapping between the response and a solution graph with the lowest edit distance cost. We associate costs that impose a penalty in the score for missing the respective action, parameter, constraint, and so on. Alignment to the closest solution allows DEFT to generate an assessment that identifies differences between the response and the gold standard, which translate both to specific errors the student has made (e.g., out-of-order actions, incorrect action parameter values, missing or extra actions) and to the corrections needed.

The alignment capability in DEFT builds on a pattern-matching algorithm that was developed originally for link analysis applications [10]. While this algorithm provided a reasonably good fit for solving the alignment problem, we developed a set of performance optimizations linked to the structure of our specific matching problem that significantly prune the overall search space.

3.3 Student Interface

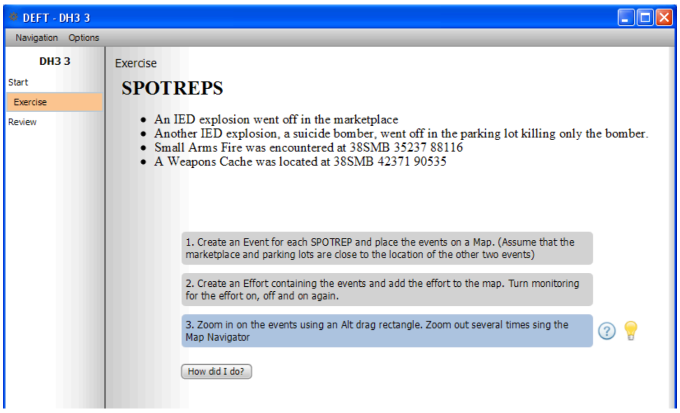
DEFT's student interface serves two functions. First, it provides a framework for exercise administration: presenting exercises for selection, supporting navigation through the exercises, and making available contextually relevant hints and documentation links. Second, it presents students with visual feedback on their solutions that shows problems detected by the automated assessment capability.

A user who selects an exercise is presented with background information from the BSOC training materials, including a statement of the learning objectives and links to relevant study materials. The user begins the exercise by clicking on a *Start* button on the bottom of the screen. The exercise is presented to the student incrementally as a sequence of numbered tasks. For example, Fig. 2 shows the three tasks that compose an exercise related to Spot Reports. The user interacts with CPOF to complete each task in turn, with instrumentation logging his actions. Upon completing a task, the user clicks on a *Next* button to proceed to the next task.

Users are presented with context-sensitive hints (accessed via the light bulb icon) and documentation links (accessed via the question mark icon) to facilitate their completion of tasks. DEFT uses hint sequences, with initial hints providing high-level guidance and subsequent hints progressively disclosing more complete directions for the task. Clicking on a documentation link displays the relevant section of the online

CPOF documentation in a Web browser. After completing all tasks, the user can click on the *'How did I do?'* button to view the DEFT assessment of his performance.

DEFT provides real-time feedback but at the level of exercises rather than individual steps. For the BSOC exercises, it is impossible to know whether a particular step is correct in isolation, as there can be multiple ways to complete subtasks within an exercise. In particular, it is important to interpret actions *in context*.



SPOTREPS

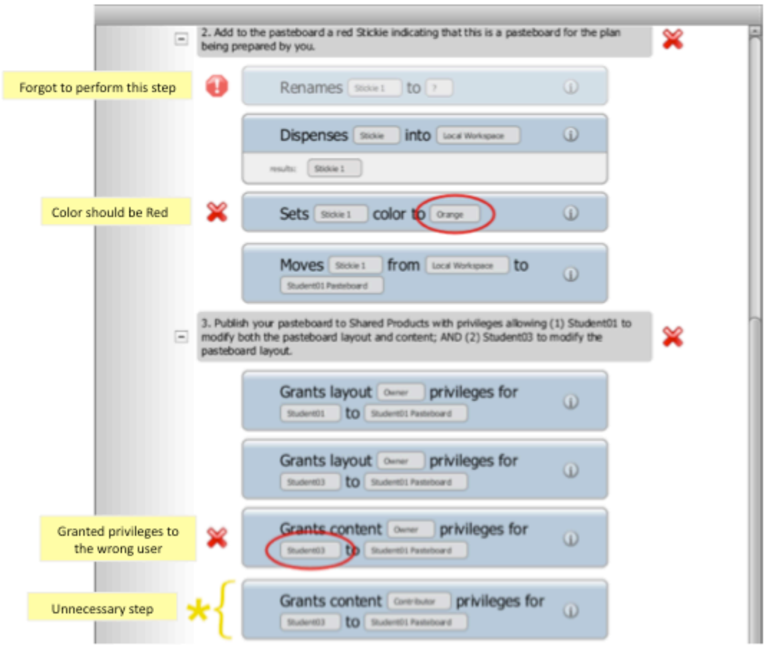
- An IED explosion went off in the marketplace
- Another IED explosion, a suicide bomber, went off in the parking lot killing only the bomber.
- Small Arms Fire was encountered at 38SMB 35237 88116
- A Weapons Cache was located at 38SMB 42371 90535

1. Create an Event for each SPOTREP and place the events on a Map. (Assume that the marketplace and parking lots are close to the location of the other two events)

2. Create an Effort containing the events and add the effort to the map. Turn monitoring for the effort on, off and on again.

3. Zoom in on the events using an Alt drag rectangle. Zoom out several times sing the Map Navigator

Fig. 2. Student interface: task structure for an exercise



2. Add to the pasteboard a red Stickie indicating that this is a pasteboard for the plan being prepared by you.

Forgot to perform this step

Renames Stickie 1 to ?

Dispenses Stickie into Local Workspace

results: Stickie 1

Color should be Red

Sets Stickie 1 color to Orange

Moves Stickie 1 from Local Workspace to Student01 Pasteboard

3. Publish your pasteboard to Shared Products with privileges allowing (1) Student01 to modify both the pasteboard layout and content; AND (2) Student03 to modify the pasteboard layout.

Grants layout Owner privileges for Student01 to Student01 Pasteboard

Grants layout Owner privileges for Student03 to Student01 Pasteboard

Granted privileges to the wrong user

Grants content Owner privileges for Student03 to Student01 Pasteboard

Unnecessary step

Grants content Contributor privileges for Student03 to Student01 Pasteboard

Fig. 3. Sample feedback from a fragment of a BSOC exercise

Fig. 3 shows sample feedback generated by DEFT. An icon to the right of each subtask indicates whether the subtask was completed successfully (green checkmark), contained mistakes (red x), or triggered warnings (yellow checkmark). An icon to the left of a step denotes a specific type of problem with that step. Hovering on the icon presents a textual description of the problem (the yellow boxes in the figure). Possible problem types include incorrect step values (red X and red circle on incorrect value), a missing step (red exclamation mark beside a grayed-out step), an unnecessary step (yellow asterisk), and incorrect ordering of steps (not shown here).

4 User Study

We conducted a user study to evaluate DEFT's ability to provide students with correct and comprehensible feedback regarding their performance on exercises derived from the BSOC training material. We had intended to conduct the study with active duty soldiers but, because of their limited availability, instead recruited ten participants without military backgrounds, spanning a variety of job roles including administrative assistants, technical editors, and project administrators. None had previous exposure to CPOF so they were given a two-hour hands-on CPOF training session the week before the study.

Typical BSOC students would have had minimal CPOF exposure; their facility with computers would vary, with most being comfortable using computers and a few having more advanced skills. Thus, other than their lack of military backgrounds, our subject pool was reasonably representative of the target population. Because BSOC training concentrates on the use of software rather than on operational content, the lack of a military background was not a significant concern.

4.1 Methodology

The user study comprised ten individual participant sessions, each lasting two hours. Each session involved the participant, a facilitator, and a note-taker; and was conducted in three parts. First was a 15-minute introduction to the use of DEFT to perform exercises in CPOF. The participant was guided by the facilitator in performing an exercise and introduced to the hints and online help mechanisms. Second was a 75-minute session during which participants were asked to think aloud as they performed exercises on their own and viewed DEFT's assessments of their solutions. They were also presented with assessments of erroneous solutions handcrafted to include various types of errors. Finally was a 30-minute debrief where the participant was asked to complete two brief questionnaires and then engaged in an open discussion. First was a standard questionnaire for calculating System Usability Scale (SUS) scores [3]; second was a compilation of questions regarding computer usage. The open discussion was structured around "product response cards" [2], a set of 55 adjectives (positive and negative) from which the participants were asked to select five that best described what they thought of DEFT and then to elaborate on their selections.

4.2 Results

Demographics. All ten participants self-reported being "comfortable" or "very comfortable" with the use of computers. On the questions regarding computer and software use, on a scale of 0 to 4 (0 = never, 4 = very often), they averaged 3.22 on online activities, 2.73 on office applications (e.g., word processing, spreadsheets), 1.67 on games, and 0.56 on advanced computer use (e.g., programming, sound/video editing). Six reported having taken a programming class at some point, but none were active programmers. All reported having taken a computer-based training or online course.

Automated Assessment. Each participant completed two to three exercises and viewed two to three additional assessments within the time allotted. Performance on the exercises varied greatly, with some completing exercises with few errors or none at all, while others struggled on all exercises. The instructions in the exercises were intentionally designed to elicit some errors and all the participants committed at least a few errors. DEFT's automated assessment module correctly identified all the errors during the study except in two situations where the system crashed due to CPOF instrumentation issues in the prototype system. All the participants were able to correctly interpret the error feedback on their solutions and, in the cases where they were asked to repeat an exercise, to correct their mistakes. Everyone was also able to interpret assessments of the handcrafted erroneous solutions but required more effort to do so because of the additional need to interpret someone else's solution.

However, based on the results of the think-aloud sessions and the discussions afterwards, it was apparent that most participants found the assessment visualizations too busy or too long. Several stated that they would prefer a simple textual rendering, with a few suggesting just a summary of the results. One participant found DEFT's focus on error feedback (i.e., only errors were pointed out) to be particularly harsh and suggested providing positive feedback as well. Many also wanted not just to be told what they had done wrong but also to be directed on how to fix it.

The perceived deficiencies of the assessment visualization were surprising, given that we had designed them in close collaboration with CPOF instructors. However, we realized that instructors and students have distinct needs. For an instructor, who needs to see the performance of an entire classroom, seeing individual user responses and high-level assessments in the form of markups (checkmarks, Xs, and circled elements) is especially valuable. In contrast, students already know what they did and are more interested in the assessment along with guidance on how to fix identified errors.

Exercise Administration. The study provided the opportunity to evaluate DEFT's exercise administration functionality. Participants found the DEFT workflow of loading an exercise, performing a sequence of tasks, and getting an assessment to be straightforward. However, a few expressed a desire for more immediate feedback to guide them through an exercise. In a number of situations, a participant started floundering and was then unable to make progress without intervention from the facilitator.

DEFT's task-specific hints and links to online help were perceived by all participants to be valuable and everyone relied on them at some point. Although a few tasks involved CPOF concepts that the participants had not been or were only briefly exposed to during their CPOF training, most were able to use the hints and help to accomplish the tasks anyway. Most participants preferred the brevity and directness of hints, often finding the online CPOF documentation to be overwhelming.

Usability and Usefulness. The SUS scores ranged from 35 to 90, with a mean of 61.25 and a median of 62.5 (scores that can be interpreted to mean roughly "average"). There are too few participants to draw statistically significant conclusions. However, together with our observations during the think-aloud sessions and the open discussions with the participants, these results indicate that although the participants found DEFT easy to use, gaps remain in its exercise administration and automated assessment capabilities.

In the product response cards exercise, participants were asked to choose the five words best describing what they thought of DEFT. The results (Fig. 4) reveal that participants had a predominantly positive response to DEFT, with several describing it as "useful", "straightforward", "relevant", and "valuable". A few participants found DEFT "frustrating"; further probing revealed that their reaction was at least partly due to their lack of familiarity with CPOF and with military terminology in the exercises.

Across the board participants expressed their belief that DEFT was a valuable training tool. They appreciated its tight integration with the training application (CPOF, in this case). All the participants readily suggested examples where they thought a tool like DEFT could be useful for training. These included various procedures they had encountered in their work, such as accounting processes, website navigation, webpage creation, and timecard management; as well as more unusual suggestions such as learning a new language or how to play an instrument.

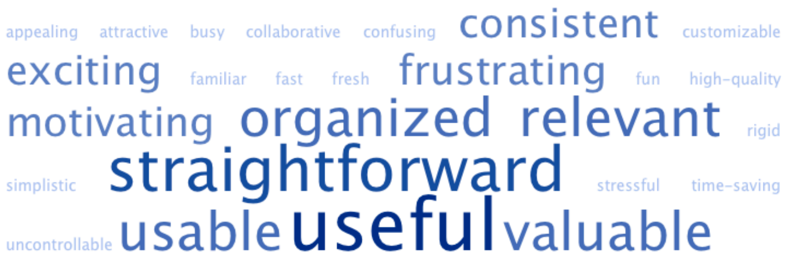


Fig. 4. Tag cloud depicting subjective participant response to DEFT, with word size reflecting the number of times it appeared in participants' Top 5 lists.

4.3 Discussion

The user study provided valuable feedback and encouraging results regarding DEFT as a training tool for procedural tasks. It is notable that although the participants in the study were complete novices in both the application (CPOF) and the domain (military operations), they were able to use DEFT to complete real training exercises in CPOF. And in spite of the difficulty in performing a task (encountered by most of the participants at some point during the study), the participant response to DEFT was predominantly positive. However, as a prototype system whose primary focus has been on automated assessment, DEFT has room for improvement. In particular, to be an effective tool for self-directed learning, it needs to provide more student-focused interactions, including a tighter integration between performance, assessment, and correction, as well as more comprehensive and focused explanatory feedback.

5 Related Work

Example-tracing tutors [1] assess procedural skills by comparing student actions against a *behavior graph* that represents all acceptable ways of achieving a task, much like DEFT compares student solutions against a *gold standard*. Both behavior graphs and our gold standards capture a range of solutions by allowing alternative actions, ranges of values used in actions, and alternative action orderings. However, because an example-tracing tutor's primary task is to *teach* a procedural skill, its assessment is focused on recognizing what the student is trying to do and ensuring that the student remains on track to successfully accomplishing a task. In contrast, DEFT is designed primarily to *assess* how well a student has performed a skill and is thus focused on identifying key mistakes in the student solution.

This distinction also applies when comparing DEFT to model-tracing [6,9] and constraint-based tutors [7]. In addition, model-tracing tutors are designed for domains such as math and physics where automated problem-solvers can be developed; they are less applicable to open-ended domains like CPOF. Meanwhile, constraint-based tutors are designed for tasks where the challenge is not in the selection of actions and parameter values but in the selection of values that satisfy potentially complex constraints. Although CPOF requires capturing such constraints as well, the variety of actions available to accomplish a task requires evaluating the procedures themselves.

In *programming*, assessment can be performed entirely on the end product (the program): whether it produces the correct results, meets complexity and style criteria, is efficient, and so on [5] To some extent, such assessment can be performed on the final information products in CPOF but the real-world need for efficient operation and adherence to best practices further demands assessment of how products are created.

6 Conclusion and Future Work

Several CPOF instructors enthusiastically endorsed our automated assessment and feedback capability, noting benefits of the technology on several levels. In a classroom setting, it would enable high achievers to progress more rapidly, potentially exploring challenge concepts beyond the baseline skills required for the entire cohort; for weaker students, the technology would provide real-time, personalized feedback. The instructors were also excited by the prospect of being able to track individual and aggregate student performance to help them identify concepts that are problematic for students and to adjust their instruction accordingly. Finally, the technology opens the door to supporting student-directed acquisition of skills outside of the classroom.

DEFT is currently a research prototype. Given the encouraging results from the user study and the strong desires expressed by CPOF trainers for a capability of this type, we believe that it would be valuable to continue this line of work with the objective of generating a fully operational assessment and feedback capability that could be deployed to facilitate self-directed CPOF training.

To date, gold standards for the BSOC exercises have been hand-coded by members of our research team. Ideally, curriculum developers would be able to construct gold standards on their own. For this, we envision a tool that would enable an instructor to demonstrate the procedural structure of an exercise solution, augmented with an annotation mechanism for specifying the companion constraints that define allowed variations from the demonstration. We believe that it would be feasible to develop such an authoring tool, leveraging learning by demonstration technology we have deployed previously within CPOF to enable automation of routine tasks [8].

Although our focus was on CPOF skills, the assessment capabilities in DEFT are not CPOF-specific and could be readily applied to other procedural training tasks.

Acknowledgments. This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-09-D-0183. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force and DARPA. The authors thank Chris Bonheim and the training staff from the U.S. Army Tactical Mission Command, who provided valuable input into the design of DEFT.

7 References

1. Aleven, V., McLaren, B., Sewall, J., Koedinger, K.: A New Paradigm for Intelligent Tutoring Systems: Example-tracing Tutors. Intl. J. of AI in Education. 19(2), 105-154 (2009)
2. Benedek, J., Miner, T.: Measuring Desirability: New Methods for Evaluating Desirability in a Usability Lab Setting. In: 2nd Conf. of the Usability Professionals Assoc. (2002)
3. Brooke, J.: SUS—a quick and dirty usability scale. In: Jordan, P.W., Thomas, B. Weerdmeester, B.A., McClelland, A.L. (eds.) Usability Evaluation in Industry. 188-194. Taylor & Francis, London (1996)
4. Croser, C.: Commanding the Future: Command and Control in a Networked Environment, Defense & Security Analysis. 22(2), 197-202 (2006)
5. Douce, C., Livingstone, D., Orwell, J.: Automatic Test-based Assessment of Programming: A Review. ACM Journal of Educational Resources in Computing. 5(3) (2005)
6. Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A.: Intelligent Tutoring Goes to School in the Big City. Intl. J. of AI in Education. 8, 30-43 (1997)
7. Mitrovic, A.: NORMIT: A Web-enabled Tutor for Database Normalization. In: Intl. Conf. on Computers in Education, 1276-1280 (2002)
8. Myers, K., Kolojejchick, J., Angiolillo, C., Cummings, T., Garvey, T., Gaston, M., Gervasio, M., Haines, W., Jones, C., Keifer, K., Knittel, J., Morley, D., Ommert, W., Potter, S.: Learning by Demonstration for Collaborative Planning. AI Magazine, 33(2), 15-27 (2012)
9. VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treay, D., Weinstein, A., Wintersgill, M.: The Andes Physics Tutoring System: Lessons Learned. Intl. J. of AI in Education, 15(3), 678-685 (2005)
10. Wolverton, M., Berry, P., Harrison, I., Lowrance, J., Morley, D., Rodriguez, A., Ruspini, E., Thomere, J.: LAW: A Workbench for Approximate Pattern Matching in Relational Data. In: 15th Conf. on Innovative Applications of AI, 143-150 (2003)