

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/281210032>

Development of a Smart Tutor for a Visual-Aircraft Recognition Task

Conference Paper · September 2015

DOI: 10.1007/978-3-319-20816-9_56

CITATIONS

0

READS

79

6 authors, including:



Priya Ganapathy

Utopia Compression

16 PUBLICATIONS 23 CITATIONS

SEE PROFILE



Ion Jovina

Wright State University

64 PUBLICATIONS 600 CITATIONS

SEE PROFILE



Tejaswi Tamminedi

Utopia Compression

12 PUBLICATIONS 34 CITATIONS

SEE PROFILE



Gautam Kunapuli

Utopia Compression

29 PUBLICATIONS 227 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



traumatic brain injury assessment [View project](#)



Value-based decision-making [View project](#)

Development of a Smart Tutor for a Visual-Aircraft Recognition Task

Priya Ganapathy¹, Ion Juvina², Tejaswi Tamminedi¹, Gautam Kunapuli¹
Matt Sherwood², Saif Usmani²

¹Utopia Compression Corporation, USA
²Wright State University, USA
{priya@utopiacompression.com}

Abstract. The goal of this project is to design an intelligent tutor to teach visual aircraft recognition (VACR) skills to military population. Under extreme cognitive demand, soldiers must learn to rapidly recognize and identify the aircraft prior to engagement. The goal of the smart tutor is to train the trainees to look at specific features (called Wings, Engine, Fuselage and Tail- WEFT) of the aircraft that will help them proceduralize the skill of aircraft recognition.

We have conducted an empirical (fMRI, eye-tracking, behavior) study and developed a cognitive model based on ACT-R architecture that emulates trainee performance. In this paper, we present insights gained from ACT-R modeling which, in turn, will be used to develop a VACR smart tutor.

Keywords: Visual aircraft recognition, smart tutor, ACT-R models, dynamic Bayesian networks, intelligent tutoring systems

1 Introduction

1.1 Current limitation of VACR training

Visual Aircraft Recognition (VACR) task is a military-relevant training procedure (Pliler, 1996) to visually recognize aircraft and categorize them as friendly, neutral, or hostile prior to engagement. Typically, VACR training involves a two step-procedure: Step 1) involves an instructor-led classroom session where VACR and the concept of breaking down an aircraft in terms of wings, engine, fuselage and tail (WEFT) components are introduced; Step 2) is a computer-based training (CBT) where trainees learn to recognize 50-75 aircraft in 2 days. Research on VACR training efficacy was performed in the early 80s where the technique resulted in a poor outcome with only 30% of 900 trainees achieving a minimum pass accuracy of 90% (Tubbs, Deason, Evertt, & Hansen, 1981).

1.2 Motivation to use ITS for VACR

Research on ways to improve VACR training is once again gaining prominence with increase in unmanned air systems (UASs) and the rate of fratricide reported in recent

wars. With the increased rate of adoption of learning management systems (LMS) and personalized tutoring across all three DoD agencies there is a strong emphasis on building an upgraded VACR training system with a smart tutor. The goal of this smart tutor is in general to minimize the training time by accelerating learning, facilitating better retention and transfer. Intelligent tutoring systems (ITS) developed based on insights gained from learning science and cognitive theory have shown significant learning gains over traditional CBT methods (Anderson, Corbett, Koedinger, & Pelletier, 1995).

1.3 Structure of the tutor

An ITS mainly consists of the following modules: A student assessment model, a domain (or an expert) module, a pedagogic (tutor) module and the front-end interface that the trainee interacts with (Corbett & Anderson, 2004). The student model infers student's skill level and knowledge gaps using knowledge-tracing approaches (Corbett & Anderson, 2004). The domain module contains the domain knowledge and the expert model to solve the problems the tutor module presents to the students. The tutor module controls the interaction with the student, based on its teaching knowledge and comparisons between the student and the expert models. Depending on the domain, complexity involved in modeling of the expert and student models may vary. For example, in this particular application the expert model would simply be the correct answer to the aircraft. The tutor model is essentially to teach the subjects to learn around 50-75 aircraft as is typically done for military trainees. The tutoring (or pedagogic) material includes the WEFT procedure to help trainees recognize aircraft and distinguish them from confounding ones. Due to the relative simplicity of the pedagogic content generation compared to more complex domains such as Math problem solving (Koedinger & Anderson, 1993) or dermatology-based problem diagnosis (Crowley & Medvedeva, 2006) the central component of our ITS is the student model.

1.4 Development of student model

There are various methods to model the student behavior in an ITS. Once modeled, the tutor can track the student behavior and compare with that of the expert. Popular tutors are generally based in cognitive science of learning and leverage adaptive control of thought-rational (ACT-R) architecture to model and track student behavior. These rule-based cognitive models form the core component of the tutor and simulate student thinking and how students can solve a given problem in multiple ways. These models also contain rules that represent incorrect behavior enabling the tutor to track errors and provide timely hints by means of a knowledge tracing algorithm. By tracking individual student behavior over time it is possible to infer the mastery of students across different skills.

Skill mastery tracking is generally performed using a Bayesian knowledge tracing (BKT) algorithm, developed originally by Corbett and Anderson (Corbett & Anderson, 2004). The BKT algorithm has been used to predict the likelihood that an individual student will get the next question correct and the updated probability that the student

has mastered the skill based on the actual answer. A traditional ITS, such as the Cognitive Tutors for Algebra, have both the cognitive models and the BKT algorithms to fully infer the student state. In this proposed effort, we use cognitive-models based on ACT-R architecture to model the VACR task. These models were then used to predict student behavior and thereby aid in modeling the attributes of the tutor. Our motivation to build an ACT-R student model was to overcome the limitation of collecting large amounts of data for testing various attributes of the tutor that can cause or accelerate learning. The ACT-R model allowed us to test the different ways to present the training/tutoring material and optimize the tutor to maximally benefit the student. We present the ACT-R model development effort, validation of these models based on a small study and the simulation results of the models under different tutoring scenarios.

The preliminary data to refine the ACT-R based model was collected during a 10-week behavior and interspersed fMRI study of subjects undergoing VACR training at Wright State University (WSU), Dayton, OH. The data collection protocol and analysis of results is presented in Juvina et al (Juvina, et al., 2015).

2 METHODS

We begin this section by briefly describing a study that was performed at WSU. The 10-week study was designed to collect behavior (accuracy and response time and eye tracking measures) and neuroimaging data from $n = 15$ participants (age: 18-35) learning to recognize 75 military aircraft. In **Fig. 1**, a pictorial description of the sequence in which the data gathering sessions (in lab OR in the fMRI scanner) were performed is shown. Each in-lab session consisted of 3 training (TR) rounds and 1 test (TE) round (sequence: TR, TR, TR, and TE). In each round, 75 aircraft were randomly shown just once to the user. The participants had to choose the correct aircraft name from three other random aircraft names from the list. In TR rounds, after participants provide an answer, they were given feedback if they answered correctly and what the correct aircraft name is. Only in TR rounds the participants were expected to learn from the feedback provided (**Fig. 2**). In TE rounds, no feedback was provided. This was a fixed-paced study with subjects given 4 seconds to give an answer and learn from feedback. Each fMRI session had around 2 TE rounds. To refine the ACT-R model only the in-lab session data (behavior measures only) was used.

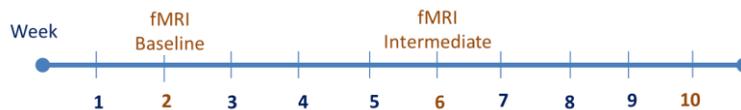


Fig. 1. The study was designed for 10 weeks with 7 in-lab sessions and 3 fMRI sessions. Each in-lab sessions included 3 training (TR) and 1 test (TE) round.

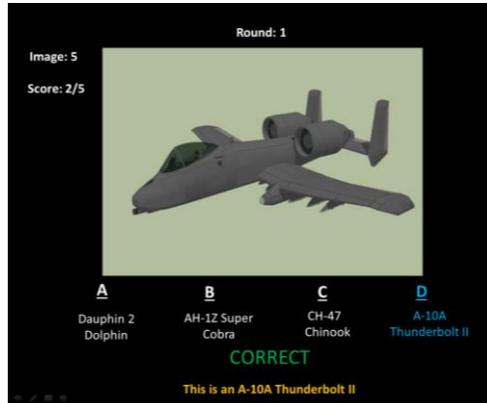


Fig. 2. The interface used for collecting data from $n=15$ participants. Same interface was used for fMRI data collection.

2.1 ACT-R architecture

Adaptive control of thought-rational (ACT-R) is a cognitive architecture based on the rational analysis theory developed by Anderson et al. (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004). ACT-R provides us a mechanism to study how different aspects of cognition take place in humans. The architecture provides us with two main memory modules: 1) declarative memory (DM) – to code knowledge of the world, facts and domain-specific information and 2) procedural memory contains knowledge of how to do things (skill-based). There are perceptual modules (visual, auditory) and motor modules to take inputs from or send response to environment. The other modules are the goal, retrieval and imaginal modules which co-ordinate with other modules via a production system to complete or learn a given task. The production system fires a list of productions (series of condition-action, “if-then”, rules) to complete a given goal. The knowledge, current state of the modules (via corresponding buffers), and actions of the model are represented symbolically. However, rather than performing as a deterministic system given the pre-coded productions and DM, the ACT-R performs more human-like due to underlying sub-symbolic knowledge which come into play when productions are fired and the model learns from the outcome of those productions. Thus it is possible for us to simulate real-world outcomes by modeling incomplete knowledge of the world, introducing random noise, decay of given fact if not used often and proceduralization (i.e., forming new implicit rules).

2.2 ACT-R model for VACR

In this effort, we have modeled the ACT-R environment to match the experimental conditions of the study. Similar to the study, the model is presented with 75 aircraft (symbolic representation and not an image, see Fig. 3) in random order each round. For each of the corresponding aircraft image four corresponding options (one option is the

correct aircraft name) is provided to the model. The model learns to recognize the correct option based on reinforcement learning. ACT-R implements a special category of reinforcement learning where the estimated reward received by the model depends on the behavior reward (provided by the environment) and the time when the correct or wrong rules were fired (temporal difference) during the whole trial (Gray, Schoelles, & Sims, 2005). As a result, a good ACT-R model can emulate human learning within almost comparable number of training sessions (Janssen & Gray, 2012). The modeling of the positive and negative reinforcement (reward) may not match in magnitude with the reward given in the actual environment. The reward (modeled in ACT-R as “Trigger Reward”) is largely determined by the number of rules that were fired during a single production cycle. With smaller reward, not all the correct rules at beginning of production cycle will receive enough weight to propagate further as good rules.

Here, our focus in building the model is to capture higher-order cognitive processes associated with learning and retention and therefore, there is less emphasis at this point to develop an environment where actual visual aircraft features are provided to the model (Vinokurov, Lebiere, Herd, & O'Reilly, 2011) (Hiatt & Trafton, 2013). We therefore use generalized functions in the ACT-R framework for retrieval/chunk activation and production utility compilation to show the proceduralization of the VACR skill (Gray, Schoelles, & Sims, 2005) (Bothell, 2007) (Janssen & Gray, 2012).

We have built this model based on our assumption of how VACR task will be learnt by subjects and then explore various teaching/ pedagogic strategies of presenting these aircraft stimuli that can potentially accelerate learning.

To model the TR rounds similar to the experimental interface (**Fig. 2** and **Fig. 3**) the model is given the actual correct answer after trial is complete. Additionally, the model is allowed to rehearse the recent trial with the correct answer put in the goal buffer (to increase strength of the correct chunk). This simulates the two-level (correct/incorrect and the correct aircraft name) learning in subject during training rounds. Consequently, in the TE rounds, there is no reward provided to the model for giving correct answers (i.e., there is no feedback of any kind). This is similar to the experimental UI where TE rounds are only to test how much of learning has actually occurred in subjects.

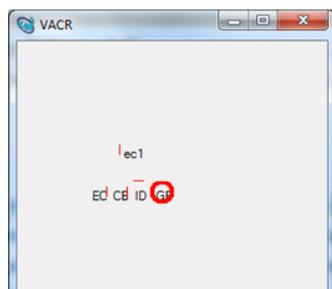


Fig. 3. Aircraft (EC)’s notional image representation (as ‘ec1’) is shown to the model. The four options similar to VACR UI in **Fig. 2**. Here, aircraft names EC, CB, ID and GF) are attended by the visual module. The model here is currently attending to option GF.

The primary set of production rules developed by the modeler is as follows:

- FindStimulus: Find the stimulus ('notional aircraft image') on the screen and place it in the visual buffer,
- RetrieveAircraftContent: retrieve the aircraft content (i.e., an internal representation of the aircraft image),
- RetrieveAircraftName: retrieve the appropriate aircraft name
- FindOption1, FindOption2, FindOption3, FindOption4: find the options shown in the user interface and place the options in the imaginal buffer
- CompareWithOption1, CompareWithOption2, CompareWithOption3, CompareWithOption4: compare the retrieved aircraft name with existing options list in the imaginal buffer.

Parameter Optimization: Once developed, we try to match the model performance trace (for same number of TR and TE rounds) with subject performance by adjusting relevant ACT-R parameters (discussed below). Optimization is crucial as the teaching strategies will be derived by testing them in the performance-matched ACT-R model.

Trigger Reward: Combines the objective feedback provided by the environment and the reward required by the model to discount utilities.

Alpha: Controls the rate of learning the rule utilities. With very high learning rate, the model reaches a local optimum solution faster. A slow learning rate will result in slower elimination of bad rules and strengthening of good rules thereby converging at a slower rate.

Egs: Accounts for variation in noise associated with utility values. A low Egs would mean less noise in utilities and therefore restricting the model to exploit as opposed to explore and find new strategies. A high Egs value will result in too many explorations and never settling on any strategies.

Set-Similarities: Allows controlling the degree of similarities between chunks placed in the declarative knowledge. The objective knowledge of aircrafts "knowledge of world" can be encoded using set-similarities command.

Initial utility (iu) and new utility (nu): The initial rules encoded by the modeler are given more chances to fire as opposed to new utilities created by ACT-R's production compilation module. The new utility parameter is set to zero to allow for new rules to prove themselves before they fire.

Goal activation (GA) and Maximum associative strength (MAS): The parameters control how quickly declarative chunks can be retrieved and increase the activation of retrieved chunks placed in the goal buffer.

Mismatch penalty (mp): Controls whether the partial matching of the DM chunks is enabled. To model the errors in aircraft recognition due to confusion between similar aircraft, "mp" was not set to nil (i.e., partial matching is enabled).

The initial values for these parameters were set based on literature (Hiatt & Trafton, 2013) (Vinokurov, Lebiere, Herd, & O'Reilly, 2011) and further fine-tuned in a trial and error fashion. Once the model behavior matches reasonably well with the subject traces we then use the model to predict student performance when number of training aircraft are increased, increase in level of confusion between aircraft, changes in the aircraft presentation method.

3 Results

In this section, we present the results of the ACT-R model behavior with parameter optimization and the model performance on changes to aircraft presentation method. The performance data from the study was clustered to identify two groups of learners, named “fast” and “slow” learners. By varying the following parameters, Alpha, ϵ , GA and mp values we could obtain a good match for the “fast” and “slow” learners, respectively (See Section 3.1). We choose the “Fast” learner-based model to run the additional simulations on testing different teaching strategies (See Section 3.2). The “Fast” learner category reaches a minimum acceptable accuracy of 90-92% for VACR by the 6th session (end of 24 rounds including TE rounds) (**Fig.4**). Therefore, the goal is to identify which aircraft presentation strategy can result in even faster learning to minimize overall training time from 2 hours (75 aircraft x 24 rounds x 4 seconds presentation time = 1.9 hours). Subsequently, once identified, the next step (not covered in this paper) is to develop the second piece of the tutoring component which is presenting aircraft-specific hints in terms of WEFT. We hypothesize that presentation of hints to help recognize aircraft will benefit the “Slow” learners compared to “Fast” learners. Consequently, a simulation to see if further reinforcement of aircraft knowledge results in improving learning in the “Slow” learner-based model (See Section 3.2) was performed.

3.1 ACT-R parameter-optimized results

In **Fig.4** we show the model-optimized result for both “Slow” and “Fast” learner category.

3.2 Optimized ACT-R model simulation results

Changing aircraft presentation rate The goal was to manipulate the presentation of aircraft images to reduce training time and/or improve learning performance. The true learning accuracy was assessed by showing all 75 aircraft once during the test rounds (TE). Every 4th round is a test round. It was observed (**Fig.5**) that by showing only incorrect aircraft after Round 16 yielded similar performance to showing all 75 aircraft once during each round (for total 24 rounds) (refers to default case- **Fig.4**). This result shows that a sufficient degree of proceduralization occurs by Round 16 and consequently, showing only those aircraft that were incorrect previously in the next TR rounds (Rounds 17-24) does not impact the accuracy of learnt aircraft (TE round accuracy remains at 90%).

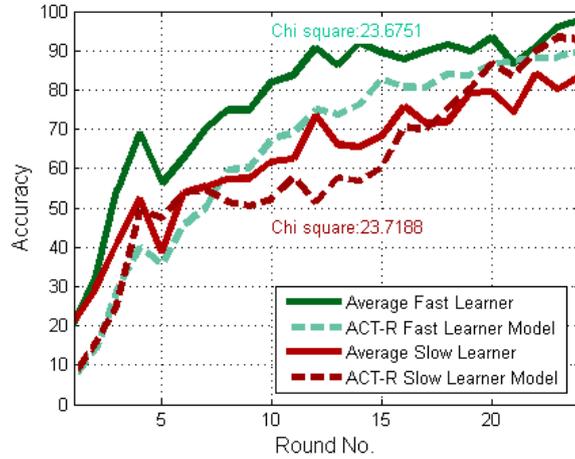


Fig. 4. ACT-R model predictions for ‘fast learner’ and ‘slow learner’ groups

Subset training. We group subsets of aircraft based on their level of similarity. We created 3 different clusters with 25 aircraft. We presented the first cluster in the 1-10 rounds and then sequentially added the second and third cluster for rounds 10-20 and then 20-30. We observe a clear dip in performance when model training is performed in subsets than a random presentation (**Fig. 6**).

Presentation of additional hints. We ran a simulation where additional hint was presented to the model to reinforce the aircraft. The simulation shows that by presenting additional hint in terms of aircraft feature the model learning rate is accelerated. The simulation was performed for the slow learner model (**Fig. 7**).

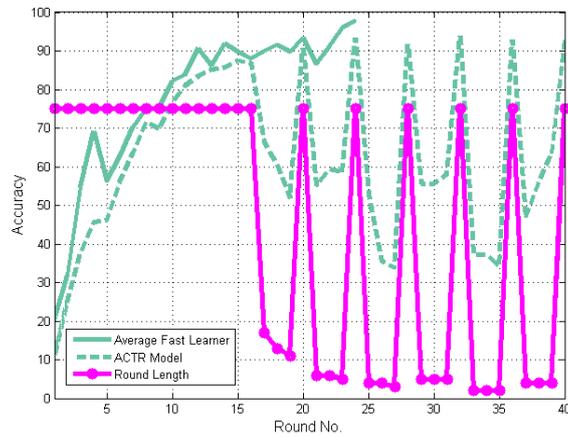


Fig. 5. After Round 16, there is sufficient proceduralization as a result we can increase presentation of aircraft that are harder to learn and decrease presentation of learnt aircraft. Here, after 16 training rounds, each round only consisted of 7-8 aircraft that subjects had to learn.

4 Discussion

4.1 ACT-R model's performance optimization

By adjusting the learning rate parameter 'alpha' the initial increase in learning (from round 1 to 3) seen in the study data could not be achieved in the model (**Fig.4**). The increase in learning during the initial few rounds (1-3 training rounds) could be achieved only by updating the mismatch penalty parameter (MP) incrementally after each round up to round 3 and then keeping it constant. This in real-world context would mean that as the trainees receive feedback they quickly resolve the aircraft with obvious discriminant features. The model does however underperform compared to subjects, mainly the fast learners (**Fig.4**). From an ITS standpoint, this would mean the model-predicted performance end point (may occur sooner in trainees). We did not do an exhaustive search of the entire parameter space as we know for a fact that trainees were shown actual aircraft images thereby the degree of learning due to visual stimuli and feedback will always result in higher learning compared to that in our model.

The training sessions in the study were set up a week apart. However, no significant degrading of performance was observed when participants played the first round of a new week's session (Juvina, et al., 2015). As a result, there was no decay introduced in the model also to account for gaps in actual training sessions.

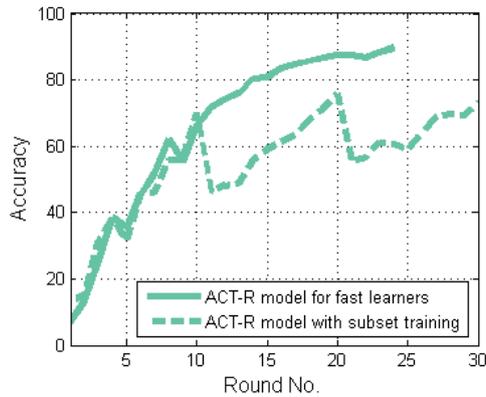


Fig. 6. Introducing aircraft in subsets (grouped by their level of similarity) for the model to learn ('dashed' plot).

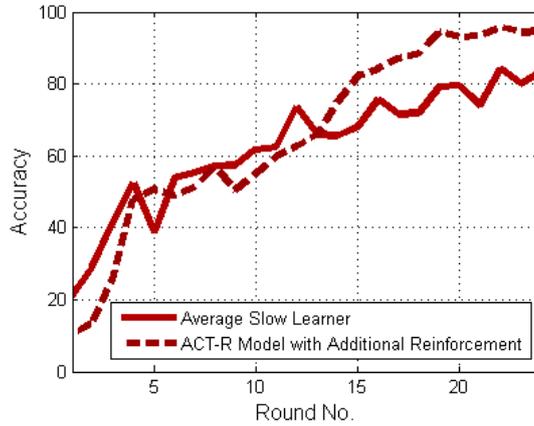


Fig. 7. Improvement in ACT-R model ('dashed' line) after additional reinforcement (20% increase in accuracy post-round 16).

4.2 Proceduralization of aircraft knowledge

The impact of introducing the objective knowledge of the aircraft and their differences with respect to one another slows down the proceduralization of aircraft. This is reflected in the fact that only after around 16-20 rounds, does the learning curve start to plateau. The fMRI results (Juvina, et al., 2015) further confirms our finding that proceduralization of the aircraft starts to occur only after 20 training rounds (approximately 2 hours of training). However, we could reasonably start manipulating the aircraft presentation stimuli after about 16 rounds (i.e., show only incorrect aircraft tracked in the past 4 consecutive rounds) without significant loss of performance (**Fig.5**). This will further bring down the training time from a total of 2 hours to 1.4 hours without loss in accuracy.

4.3 Other changes to aircraft presentation strategy

We introduced clusters of aircraft as part of the training sequence (a total of 3 clusters with 25 aircraft each – total 75 aircraft) (**Fig. 6**). The goal was to allow the model to only learn 1st cluster for the initial rounds (rounds: 1-10) and then add the 2nd cluster (rounds: 11-20) and 3rd cluster (rounds: 21-30). However, due to reduced presentation time of each aircraft from newly introduced cluster the corresponding correct rules were not given sufficient time to be reinforced in the model. Rather the random presentation of aircraft from all the 3 clusters such that all 75 aircraft were shown at least once in each round (although slightly longer overall time ~ 20 minutes) showed higher end accuracy.

4.4 Performance boost with additional reinforcement

The eye tracking results from the behavior study shows that slow learners spent very less time looking at the aircraft (both before response and after they receive feedback) (Juvina, et al., 2015). In general, the behavior profile shows that slow learners are low on motivation compared to fast learners. The subjects seem to be less mindful of the task and need extra help/hints to remember the aircraft. We obtained a performance boost (about 20% increase post round 15) in the slow learner-based model (**Fig. 7**) by providing an additional ‘notional hint’ (a distinct aircraft feature) to remember the aircraft. Therefore, in the actual ITS as well we will introduce the pedagogic component in the form of hints based on student performance.

The simulations presented here are not exhaustive; as we continue to develop the ITS design, we will test different training hypotheses using this ACT-R model.

5 Conclusion and future work

Our choice of VACR as the candidate task is strongly tied to its utility in operation and its wide-spread requirement for operation of Stinger missiles by ground troops. The lessons learnt from the ACT-R modeling experiment will be implemented as part of the ITS tutor design. We have also developed a BKT based algorithm (integrated with ITS) for real-time tracking of student performance. Our next study will use the Amazon Mechanical Turk platform to recruit participants to validate the findings from the model simulation and also test the overall efficacy of our ITS.

Acknowledgment. The authors thank the Office of Naval Research (SBIR Phase II, Navy Contract # N00014-13-C-0138) for funding this project.

Bibliography

1. Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, 1036–60.
2. Anderson, J., Corbett, A., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: lessons learned. *J Learn Sci*, 4(2):167–207.
3. Bothell, D. (2007). *ACT-R 6.0 Reference Manual*. Retrieved February 20, 2015, from ACT-R Web site: <http://act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr6/reference-manual.pdf>
4. Corbett, A., & Anderson, J. (2004). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4 (4): 253–78.
5. Crowley, R., & Medvedeva, O. (2006). An intelligent tutoring system for visual classification problem solving. *Artif Intell Med.*, 36(1):85-117.
6. Gray, W., Schoelles, J., & Sims, R. (2005). Adapting to the task environment: Explorations in expected value. *Cognitive Systems Research*, 27-40.
7. Hiatt, L., & Trafton, J. (2013). The Role of Familiarity, Priming and Perception in Similarity Judgments. *Proceedings of the 35th Annual Meeting of the Cognitive Science Society* (pp. 573-578). Berlin, Germany: Cognitive Science Society.

8. Janssen, C., & Gray, W. (2012). When, What, and How Much to Reward in Reinforcement Learning based Models of Cognition. *Cognitive Science*, 36(2), 333-58.
9. Juvina, I., Ganapathy, P., Sherwood, M., Usmani, S., Kunapuli, G., Tamminedi, T., et al. (2015). Neurocognitive correlates of learning in a visual object recognition task. *Human Computer Interaction*. Los Angeles.
10. Koedinger, K., & Anderson, J. (1993). Effective use of intelligent soft-ware in high school math classrooms. In: Brna P, Ohlsson S, Pan H, editors. Proceedings of the world conference on artificial intelligence in education. *AACE*;, 241-8.
11. Pfliler, J. (1996). *Aircraft Recognition Training: A Continuing Process*. Air Defense Artillery.
12. Tubbs, J., Deason, P., Evertt, E., & Hansen, A. (1981). CHAPARRAL Training Subsystem Effectiveness Analysis (TSEA). *Proceedings 23rd Annual Conference of the Military Testing Association*.
13. Vinokurov, Y., Lebiere, C., Herd, S., & O'Reilly, R. (2011). A Metacognitive Classifier Using a Hybrid ACT-R/Leabra Architecture. *AAAI Workshop*, (pp. 50-55). San Francisco