Developing a Pattern Recognition Structure to Tailor Mid-Lesson Feedback

Jeremiah T. Folsom-Kovarik[1], Michael W. Boyce[2]
[1]Soar Technology, Inc., [2]U.S. Army Research Laboratory

INTRODUCTION

The Generalized Intelligent Framework for Training (GIFT) has the potential to increase the micro-adaptive individualization of many training systems by overlaying adaptive feedback to learners during training sessions. For example, GIFT can augment a particular scenario in a first-person infantry simulation without needing to change the scenario itself, by displaying feedback messages in the tutor user interface (TUI) when particular learner experiences are observed. Feedback that GIFT delivers in the TUI can be as effective as feedback embedded directly in the system (Goldberg & Cannon-Bowers, 2015).

Expected observations (such as learner inputs or actions) that should trigger a response in GIFT are typically defined via a domain knowledge file (DKF). Importantly, feedback that responds to domain observations is best tailored to individual learners' needs when GIFT can select and deliver it on the basis of a rich collection of actionable information about learner experiences and characteristics. To this end, the DKF is enhanced with a new ontology of *patterns* that draw information from the relationships between single observations. Examples of patterns include order, timing, and repetition relations between observations.

A powerful existing tool to author and identify patterns is the Student Information Model for Intelligent Learning Environments (SIMILE) (Mall & Goldberg, 2014). The present work is compatible with SIMILE to the extent that SIMILE generates conditions which can be processed as input. Relative to SIMILE, the present research adds domain-general reasoning about features extracted from the domain-specific patterns. Furthermore, because it is native GIFT code, the present contribution is possible to use with GIFT Cloud. Interpreting patterns within GIFT's learner module and pedagogical module can increase their power to recognize and respond to proper performance in the training domain, learners' skill and knowledge, and inferences about learners' cognitive states and traits.

An initial demonstration of the work is being constructed for a military cognitive-perceptual training task that combines social and tactical challenges within each scenario. The demonstration uses patterns to define expected timing and order of responses in the domain, infer the latent mental processing steps of individual learners, and respond to learners with immediate formative feedback.

COGNITIVE-PERCEPTUAL TRAINING DOMAIN

Initial experimentation is grounded in a software system for tailored training and assessment previously created by SoarTech under DARPA funding (Hubal, van Lent, Marinier, Kawatsu, & Bechtel, 2015) known as Adaptive Perceptual and Cognitive Training System (APACTS). During the present research and development, APACTS is being modified to work with GIFT as an external training application and will be made available to GIFT users.

APACTS contains challenging, realistic decision-making scenarios developed in conjunction with experienced operators from Army and other training domains. The target training audience is an Army small-unit leader. The military battlespace where these leaders operate is characterized by uncertainty due to missing information, time pressure from the need to take advantage of tactical situations quickly, and high complexity with many interacting factors to consider (Thunholm, 2005). At the same time, the leaders' quick decisions can have far-ranging impacts on the larger U.S. mission (Malone, 1983).

APACTS scenarios test learners' decision-making ability in scenarios that draw on both tactical and social skill in the same scenario. APACTS sequences video, images, and two types of assessments: multiple-choice decisions and a perceptual task (Figure 1) that lets learners annotate images with specific visual cues that occur in the scene. Feedback is delivered via an after-action review (Figure 2). Key to the present work is that GIFT adds tailored mid-lesson feedback to APACTS via the TUI. GIFT selects feedback by recognizing and interpreting patterns in the learner performance during APACTS scenarios. The approach is general and may also be used to find and respond to patterns in other GIFT training tools.

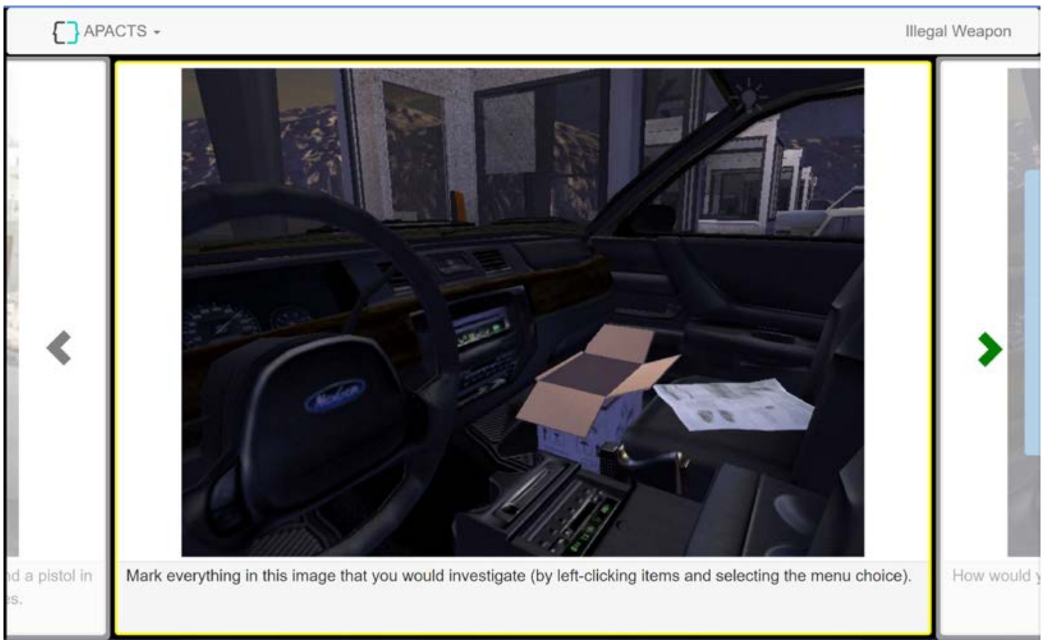

Figure 1. The APACTS cognitive-perceptual assessment tasks assess how learners process visual cues.

Terminology

For the purposes of this research, an *observation* is defined as a provable fact about what one learner has done within a learning tool. Examples of observations are *the learner opened a door* or *the learner scored 15 out of 20*. An observation happens at a single point in time, does not have duration, and has always either happened or not happened. An observation does not have a likelihood, does not need to be inferred, and cannot be incorrect. Any uncertainty surrounding an observation is assumed to be resolved by the training tool where the observation originated.

Within GIFT, individual observations are assembled into *patterns* via new additions to the domain module. Patterns are groups of observations that take on meaning in relation to each other. For example, clearing a room might require two observations within some period: *the learner opened a particular door* and then *the learner moved to the right*. The two observations might be related by ordering (one before the other) and by timing (one immediately after the other). Patterns may also be grouped and nested to arbitrary depth.

While occurrences of observations and patterns are both considered incontrovertible facts in the GIFT point of view, inference comes into play when the DKF defines *constraints* on observations and patterns. The satisfaction or violation of these constraints lets GIFT detect learner errors (constraint violations) and infer what *misconceptions* might underlie the observed performance. Misconceptions encode predictable but incorrect cognitive processing (Koedinger, Corbett, & Perfetti, 2012; Sleeman, Ward, Kelly, Martinak, & Moore, 1991). Misconceptions in GIFT extend the domain concept objects with new information that can help tailor feedback and provide appropriate pedagogical strategy.

During the present work, APACTS was instrumented with a typical interop plugin that communicates learner performance to GIFT through the gateway module and domain module. Because of these changes, GIFT has visibility into observations of individual learners as they progress through APACTS. This provides a testbed for demonstration and evaluation of the new GIFT capabilities to observe patterns in learner performance, infer errors and misconceptions, and tailor mid-lesson feedback.

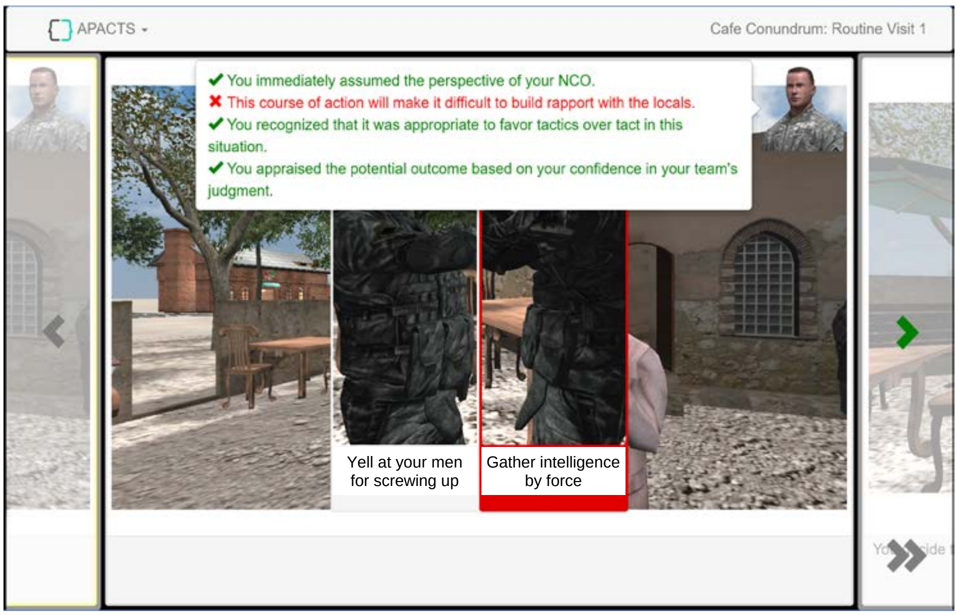

Figure 2. APACTS combines tactical / social decision-making assessments and AAR feedback.

OBSERVABLE PATTERNS

Patterns relate multiple observations to each other in time. Formal temporal logic is well studied in the context of, for example, characterizing software synchronization and timing (Clarke & Emerson, 1981) or reasoning about plans (McDermott, 1982). Patterns of observations that are implemented in GIFT represent a subset of temporal logic operators defined by Allen and Ferguson (1994). The patterns chosen for implementation reflect those hypothesized to be valuable for instructors or instructional designers to describe learner performance in a variety of modern training tools, to include APACTS and more sophisticated open-world simulations.

The GIFT team has made authoring tools a strong emphasis of the adaptive training program. Existing research has identified lessons learned for authoring tools in GIFT explaining the importance of empowering the user, which will build trust and confidence in the authoring process (Ososky, 2016). Authorability of patterns by nontechnical personnel is a key design consideration when choosing and defining patterns.

When future work adds the new patterns into GIFT authoring tools, it is vital that they align with how instructors think about learner performance. Instructional alignment may be more valuable than complete expressivity of the language when it can prevent errors, reduce cost of authoring, and increase technology acceptance among instructors and instructional designers (Folsom-Kovarik, Wray, & Hamel, 2013). For this reason, limiting the patterns that an author can express may actually improve the utility of the new constructs more than making it possible to express many more patterns but also requiring an engineering or mathematical background to get the patterns right. Furthermore, even the simplified syntax used in this paper may be hidden from nontechnical authors by presenting a graphical interface such as the draggable box and line diagrams described in (Woods, Stensrud, Wray, Haley, & Jones, 2015).

Although actions of other learners (or constructive characters) may cause observations and otherwise affect learners, the present work focuses on the individual learner use case. At present the GIFT patterns do not include a full definition of patterns that could be expected in a team environment.

Required, Forbidden, and Optional

The basic elements of patterns provide building blocks that instructors can assemble to describe learner performance in a training tool. The definitions of the basic elements in GIFT are based on constraint logic built for the Dynamic Tailoring System (DTS) (Wray & Woods, 2013).

First, observations may be *required*, *forbidden*, or *optional* (Table 1). These generalized basic ideas are already present in GIFT and implemented by individual domain module conditions. They help define types of errors that can let GIFT differentiate the cognitive processing in this learner, such as insufficient automaticity or presence of specific misconceptions, that led to the incorrect performance (Woods et al., 2015).

Table 1. The basic building blocks of patterns are single observations.

Notation	Meaning
A	*Required*. GIFT must observe condition A. The learner must carry out step A.
~A	*Forbidden*. GIFT must not observe condition A to be true. The learner must not do A.
A*	*Optional*. GIFT may or may not observe condition A; it is not required or forbidden.

In the absence of other constraints defined below, failure to observe a required element constitutes an *error of omission* that GIFT detects when the required element goes out of scope without being observed. The observation of a forbidden element constitutes an immediate *error of commission*. In the formal study of errors – for example in adverse event analysis (Donchin, Gopher, Olin, Badihi, Biesky et al., 1995) or in the human factors design of a system (Boyce, Sottilare, Goldberg, & Amburn, 2015) – errors of omission and errors of commission are typically considered to arise from different cognitive pathways and demand different diagnosis. An error of omission reflects a failure to carry out a required step, perhaps forgetting or not recognizing the need to do so. An error of commission is an incorrect action or actively doing what should not be done. Other error categories, such as sequence errors or context errors, are discussed below.

Optional elements can never cause a constraint violation, but when they are observed they can cause other processing in conjunction with the below patterns. Required and optional elements may be observed more than once without causing an error. To limit the repetitions allowed, see *repetitions* below.

Clusters, Dependency, and Strict Ordering

Clusters of observations are unordered sets that group elements together for checking. Clusters may have any number of members and may be nested to arbitrary depth. Importantly, checking a cluster can imply either a logical *AND* or a logical *OR* relation between members, depending on where the cluster appears.

This helps move away from a strict temporal logic and toward a language of patterns that should match the intuitions of nontechnical users for teaching and training. Examples of the difference appear in Table 2: compare the logical processing implied by (A B) → C as contrasted with A → (B C).

A *dependency* relation between two elements (observations or groups of observations) indicates that the second element should not be observed before the first. For example, a learner on a patrol mission should not proceed outside the wire before completing a mission briefing. By contrast, a *strict ordering* relation indicates that not only must the second observation come after the first, but also the first observation becomes forbidden, and may no longer be repeated, after the second is observed. It is still permissible to observe A or B multiple times each, as long as they are not out of order.

Both dependency relations and strict ordering relations may be chained to arbitrary length, as shown in Table 2. When several elements participate in a strict ordering, there is no way to exempt a member element from being strictly ordered (no partial ordering).

Table 2. Clustering and ordering multiple observations.

Notation	Meaning
(A B)	*Cluster*. A and B are separate observations, but are checked together. They are required.
~(A B)	Both A and B are forbidden.
(A B)*	Both A and B are optional.
A → B	*Dependency*. B depends on A. The learner cannot do step B without first doing step A.
A → B*	B does not need to be observed, but if it is observed then it triggers an error of omission unless A is observed first.
A* → B	Equivalent to just B, because A can either happen or not happen before B is observed. Note that this is a change from interpretation in the SoarTech DTS.
(A B) → C	Both A *and* B must be observed before C may be observed.
A → (B C)	A must be observed before either of B *or* C may be observed.
(A, B)	*Strict order.* A and B must be observed in order. It is not allowed to do B until A is done, and also it is not allowed to do A after B.

Note that the strict ordering (A, B) is simply equivalent to A → (B ~A). The definition of strict ordering as another first-order constraint becomes useful for authors when there are several elements that need to be ordered. For example, (A, B, C, D) is easier to encode than A → (B ~A) → (C ~A ~B) → (D ~A ~B ~C). Furthermore, if a graphical UI is used to author these relations, fewer nodes and edges will be required.

Relevance and Exclusivity

Relevance refers to the notion that required or forbidden elements (usually clusters) will not be checked under certain conditions when they are not instructionally relevant. Relevance is similar to the concept of scope in computer programming. For example, checking whether a learner clears a room correctly is (at first) not relevant in the context of a patrol scenario, but may become relevant if the patrol comes into contact with the enemy and must conduct a tactical engagement.

Controlling the relevance of an element is not simply a matter of saving computational resources. It is also important instructionally. Relevance can be used to make assessment tractable in ill-defined domains (Nye, Boyce, & Sottilare, 2016; Woods et al., 2015). For example, if a learner on patrol enters a village elder's home for tea, it would be inappropriate for GIFT to state he made an error by not clearing the room first.

Exclusivity refers to the requirement that all unmentioned observations are considered forbidden. Exclusivity can be specified at the same level of granularity as any cluster, including a top-level luster that contains all others. Otherwise, if a cluster is not marked as exclusive, any unmentioned observations are considered optional. When a cluster is not relevant, its exclusivity constraint is not checked.

Repetition, Pause, and Duration

Elements may be *repeated* a number of times that instructors specify. For example, in a patrol scenario the learner might need to greet between two and four civilians in the local language. Any number of occurrences in the range satisfies the constraint. An exact value can also be specified. The repetition constraint does not rule out other observations between the repetitions or after them.

A *pause* is an interval between two observations. The time starts counting every time the left-hand side of the constraint is satisfied. If the right-hand side becomes satisfied before the minimum specified delay, then the delay constraint is violated. If the maximum specified delay expires and the right-hand side is not satisfied, the delay constraint is violated.

Finally, *duration* describes how long it should take the learner to complete one or more observations. The entire cluster must be satisfied within the timespan specified by max. If the time specified by max elapses and the cluster is not satisfied, the constraint is violated. Like all constraints, durations may be nested, enabling a series of observations that have a total time for completion and duration for each individual item. This method is valuable when studying speed-accuracy tradeoffs (Goldhammer, 2015).

Table 3. Patterns of repeating and timed observations.

Notation	Meaning
A r[min..max]	*Repetition.* Element A must be observed at least *min* times and at most *max* times.
A r[exact]	Element A must be observed exactly *exact* times.
A r[2]	The learner must do A twice. If the learner does A three times, no error happens.
A r[2] → B	The learner must do A twice before doing B.
A r[2] → ~A	The learner must do A twice, after which the learner may not do A again.
→ p[min..max]	*Pause.* The time between these two observations must be between *min* and *max*.
A → p[30 sec ..] B	B must occur after A and also at least 30 seconds must separate them.
A → p[.. 30 sec] B	B must occur after A and also within 30 seconds after A is observed.
A d[max]	*Duration.* Element A will be relevant for up to *max* seconds.
(A B) d[30 sec]	The learner has 30 seconds to complete A and B.
(~A) d[30 sec]	The learner may not do A for the first 30 seconds that the constraint is relevant.

In the present work, patterns that express order, repetition, and timing form the basis for infering general insights about learners and improving feedback tailoring.

INTERPRETING PATTERNS AND TAILORING FEEDBACK

The first modification in GIFT to take advantage of information from observable patterns is the idea of a *misconception*. Misconceptions modify concepts within the domain module. They give GIFT additional information about learner performance – not just whether a concept has been mastered or not, but also inferences about why a concept may not be mastered and what specific feedback may be needed.

Misconceptions have been well studied elsewhere and evidence exists that they are valuable to providing tailored feedback. A few example benefits are listed. Detecting and addressing specific misconceptions can challenge learners' incorrect mental models when untailored feedback would otherwise allow them to gloss over the differences (Swan, 1983). Feedback focusing on misconceptions is also more directive, when GIFT detects that such feedback is more appropriate for an early stage of learning (Moreno, 2004) than an alternative facilitative feedback or an exploration experience during later stages. Inferring the presence of misconceptions can also support increased specificity in feedback which is appropriate when learners are more performance oriented (Davis, Carson, Ammeter, & Treadway, 2005). In conjunction with GIFT's *active* and *constructive* feedback mechanisms, the addition of misconceptions will help to provide feedback that aligns with many guidelines for delivering formative feedback (Shute, 2008).

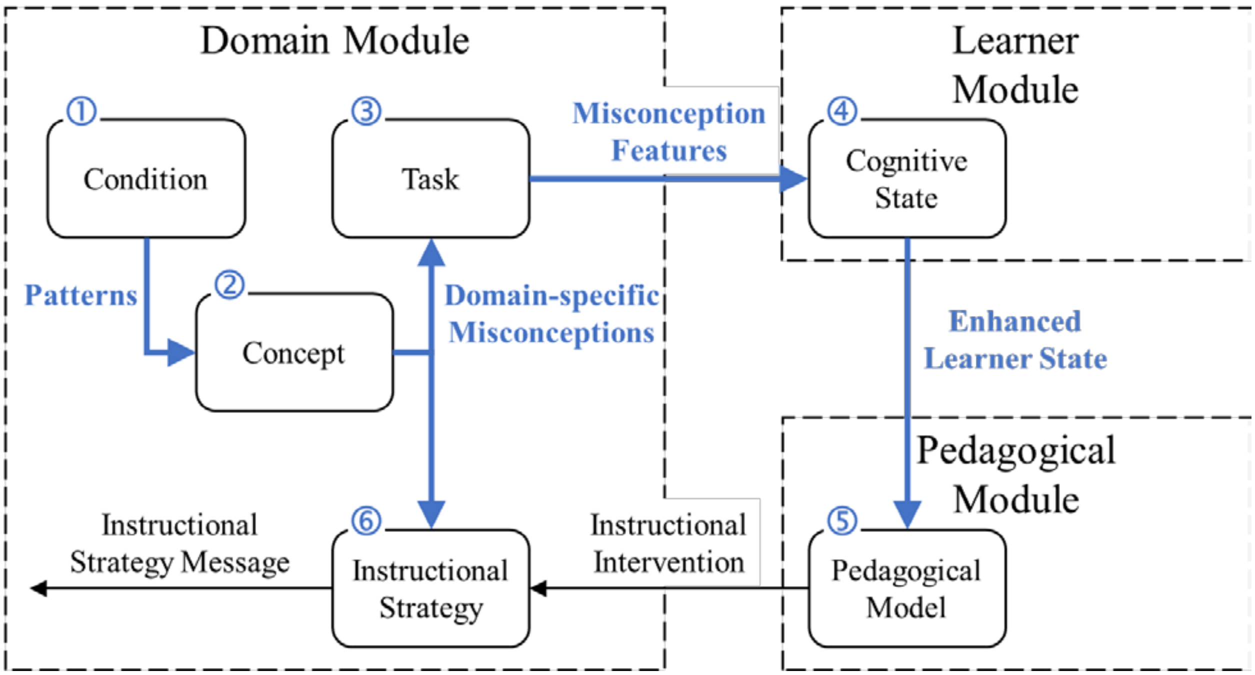

Figure 3. High-level data flow for inferring misconceptions and using them to tailor feedback.

Figure 3 depicts a high-level data flow for observing patterns of learner behavior and inferring the presence of misconceptions. Selected relevant classes within each GIFT module are shown with blue (bolded) lines indicating changes to the standard GIFT classes and messages.

First, GIFT patterns implement new kinds of conditions (1). Like conditions, the patterns can modify the state of domain-specific concepts (2). Concepts are similarly modified to contain an arbitrary number of misconceptions, each of which is tied to its parent concept. So, it becomes possible to differentiate between specific ways that a learner may act or know incorrectly. The different misconceptions may require different levels of urgency or different modes of feedback.

While misconceptions within the domain module are domain-specific, GIFT needs to reason about misconceptions in a domain-general manner within the learner module and pedagogical module. For this reason, dimension reduction in the domain module passes along only a subset of features for each detected misconception. The extracted misconception features are domain-general and include the importance, urgency, and certainty of each misconception. Determination of these values is task-specific (3). For example, in a VBS scenario the domain module might detect a pattern of learners walking around with weapon in the wrong ready state. In a squad tactics setting this might be an unimportant error, while the error would be more important if the scenario is targeting intercultural communication with civilians while on a patrol.

Once the learner module has domain-general information about misconceptions as they are detected, stored in the cognitive state class (4), the pedagogical module gains new information within the learner state message set on which to base real-time tailoring decisions. While performance assessment messages are domain-specific, the misconception features are generalized and thus update the learner cognitive state. Within the scope of the present research, a simple algorithm will be added to an existing pedagogical model (5) that acts on the domain-general features of misconceptions to direct instructional interventions. For example, the initial pedagogical algorithm might indicate that some number of unimportant misconceptions may be addressed through AAR or reteaching, while any misconception with importance above some threshold must be addressed through immediate feedback.

Finally, when the pedagogical module requests an instructional intervention, the domain module contains the full misconception information that is required to deliver needed feedback with high specificity (6).

APACTS Examples

Two examples suggest the value of leveraging observable patterns in GIFT.

First, Figure 1 above depicts an example of a visual scan task in APACTS. The learner is stationed at an entry control point and must respond to a civilian vehicle as shown in a static image. Optionally, there is a time limit on the learner's response.The correct response is to mark two objects in the Figure 1 image: the box on the passenger side floorboard, and a pistol grip that is visible between the center console and the passenger seat (Figure 4). However, by using a new observation ordering pattern, GIFT can now add specificity to the APACTS assessment of correct behavior.

Figure 4. Detail of Fig 1, highlighting threat item.

This image is designed so that the more threatening object, the pistol, is less visually salient (less noticeable) compared to the box, which is easier to see because it is larger and a lighter color. Since APACTS communicates each click the learner makes to GIFT, it is easy for GIFT to define domain-specific constraints that not only require clicking on both objects, but also differentiate between which object was clicked on first. If the learner clicks the box before the gun, that ordering may be caused by a more reactive cognitive processing of the scene (Schatz, Colombo, Dolletski-Lazar, Carrizales, & Taylor, 2011), and can be associated with the inference this is a more novice learner. If the learner clicks the gun first, that observation provides evidence that the learner is more expert in visual scene assessment.

As a second example, Figure 2 above depicts a typical multiple-choice assessment in APACTS (although overlaid with the built-in AAR feedback). Multiple-choice assessments provide an opportunity to gather information via observation timing patterns. With these, GIFT can make use of observations that a human instructor might value such as the amount of time the learner considered the question before making a choice. Fast choices might be associated with a more expert learner. GIFT can also make use of information such as whether the learner changed between choices before submitting, or simply hovered the mouse over one option or the other, to differentiate hesitation from other reasons for delay such as inattention.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In conclusion, GIFT is being enhanced with new domain-specific and domain-general representations of learner performance and underlying cognitive state that will make tailored feedback specific and impactful.

The implementation status of the work described includes initial changes to the GIFT source code in a development branch. The changes will be made available to the GIFT community in the future, after appropriate code review. APACTS software and scenarios will be published and made available.

A demonstration is planned via a human-participants study of APACTS. The demonstration is expected to compare training efficacy using new, tailored feedback against the baseline of APACTS alone. In addition, the implementation work supporting the study may be reused as a publicly available reference or showcase of the new capabilities and how to use them.

Future development work will include adding the new patterns into GIFT authoring tools. Finally, the patterns will be demonstrated on a second domain besides APACTS. That work will demonstrate the generality of the approach and utility to enhance widely used tools such as VBS or other training systems.

Finally, interesting directions for future funded research might include machine learning of patterns such as time limits that differentiate different cognitive processing pathways, helping to assess automaticity of skill performance. GIFT research efforts such as metacognition assessment, or active and constructive interventions, should also be combined with this work in order to improve the simple tailoring algorithms.

ACKNOWLEDGEMENT

This material is based upon work supported by the Human Research & Engineering Directorate - Advanced Training and Simulation Division (HRED - ATSD) and the Army Contracting Command (ACC) - Aberdeen Proving Ground (APG) - Research Triangle Park Division (ACC-APG-RTP), Orlando Branch under Contract No. W911NF-16-C-0063**.** Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ATSD.

REFERENCES

Allen, J.F., & Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of logic and computation, 4*(5), 531-579.

Boyce, M.W., Sottilare, R.A., Goldberg, B., & Amburn, C.R. (2015). *Work Domain Analysis for Ecological Interface Design of Tangible Interfaces*. Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Clarke, E.M., & Emerson, E.A. (1981). *Design and synthesis of synchronization skeletons using branching time temporal logic.* Paper presented at the Workshop on Logic of Programs.

Davis, W.D., Carson, C.M., Ammeter, A.P., & Treadway, D.C. (2005). The interactive effects of goal orientation and feedback specificity on task performance. *Human Performance, 18*(4), 409-426.

Donchin, Y., Gopher, D., Olin, M., Badihi, Y., Biesky, M.R., Sprung, C.L., et al. (1995). A look into the nature and causes of human errors in the intensive care unit. *Critical care medicine, 23*(2), 294-300.

Folsom-Kovarik, J.T., Wray, R.E., & Hamel, L. (2013, July 9-13). *Adaptive assessment in an instructor-mediated system.* Paper presented at the 16th International Conference on Artificial Intelligence in Education, Memphis, TN.

Goldberg, B., & Cannon-Bowers, J. (2015). Feedback source modality effects on training outcomes in a serious game: Pedagogical agents make a difference. *Computers in Human Behavior, 52*, 1-11.

Goldhammer, F. (2015). Measuring ability, speed, or both? Challenges, psychometric solutions, and what can be gained from experimental control. *Measurement: interdisciplinary research and perspectives, 13*(3-4), 133-164.

Hubal, R., van Lent, M., Marinier, B., Kawatsu, C., & Bechtel, B. (2015). *Enhancing Good Stranger Skills: A Method and Study*. Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Koedinger, K.R., Corbett, A.T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science, 36*(5), 757-798.

Mall, H., & Goldberg, B. (2014). *SIMILE: an authoring and reasoning system for GIFT.* Paper presented at the 2nd Annual GIFT Users Symposium.

Malone, D.M. (1983). *Small unit leadership: A commonsense approach*: Random House Digital, Inc.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive science, 6*(2), 101-155.

Moreno, R. (2004). Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science, 32*(1-2), 99-113.

Nye, B.D., Boyce, M.W., & Sottilare, R.A. (2016). Defining the Ill-Defined: From Abstract Principles to Applied Pedagogy *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling*.

Ososky, S. (2016). *Practical requirements for ITS authoring tools from a user experience perspective.* Paper presented at the International Conference on Augmented Cognition.

Schatz, S., Colombo, G., Dolletski-Lazar, R., Carrizales, M., & Taylor, A.H. (2011). *A Proposed Taxonomy for Perceptual—Cognitive Skills* Orlando, FL: MESH, a DSCI Company.

Shute, V.J. (2008). Focus on formative feedback. *Review of educational research, 78*(1), 153-189.

Sleeman, D., Ward, R.D., Kelly, E., Martinak, R., & Moore, J. (1991). An Overview of Recent Studies with PIXIE. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring* (pp. 173–186). Norwood, NJ: Ablex Pub. Corp.

Swan, M. (1983). *Teaching decimal place value: A comparative study of "conflict" and "positive only" approaches*: Nottingham University Shell Centre for Mathematical Education.

Thunholm, P. (2005). Planning under time pressure: An attempt toward a prescriptive model of military tactical decision making *How professionals make decisions* (pp. 43-56).

Woods, A., Stensrud, B., Wray, R., Haley, J., & Jones, R. (2015). *A Constraint-Based Expert Modeling Approach for Ill-Defined Tutoring Domains.* Paper presented at the The Twenty-Eighth International Flairs Conference.

Wray, R.E., & Woods, A. (2013). *A Cognitive Systems Approach to Tailoring Learner Practice.* Paper presented at the 2nd Advances in Cognitive Systems Conference, Baltimore, MD.

ABOUT THE AUTHORS

Dr. Jeremiah T. Folsom-Kovarik *is a lead scientist with Soar Technology, Inc. He earned a Ph.D. in Computer Science from the University of Central Florida in 2012. Dr. Folsom-Kovarik's research focuses on novel methods, algorithms, and knowledge representations that help automate (1) capturing and assessing sensor inputs with an emphasis on monitoring trainee performance, (2) modeling underlying reasons for observations and inferring individual trainee needs, and (3) automating intelligent systems that can reason about the contexts, goals, and tradeoffs required to carry out effective, individualized skill training.*

Dr. Michael W. Boyce *is a research psychologist at the Army Research Laboratory, Human Research and Engineering Directorate (ARL-HRED) supporting the Adaptive Training Research Program and the Advanced Modeling and Simulation Branch. His research focuses on the integration between the Generalized Intelligent Framework for Tutoring (GIFT) and other training applications such as the Augmented REality Sandtable (ARES). He earned his doctorate in Applied / Experimental Human Factors Psychology from the University of Central Florida in 2014.*