Automating the Training Feedback Paradigm with Intelligent After Action Review

Barry Clinger, Christopher Sipes
Riptide Software
Orlando, FL
barry.clinger@riptidesoftware.com,
chris.sipes@riptidesoftware.com

Nate Blomberg, Robert Burch
Dignitas Technologies
Orlando, FL
bburch@dignitastechnologies.com,
nblomberg@dignitastechnologies.com

Jeremy Lanman, James Todd
U.S. Army PEO STRI
Orlando, FL
jeremy.t.lanman.civ@mail.mil,
james.a.todd28.civ@mail.mil

ABSTRACT

There are emerging trends to support an enterprise approach for providing training capabilities to the Warfighter at the point of need. As needs increase for easy operation, mobile training, and force reduction, an enterprise methodology focused on providing intelligent assistance and automation to After Action Review (AAR) artifact generation is needed to advance the training feedback paradigm. In addition, a wider variety of capabilities must be provided to the instructor, allowing for opportunities to provide more focused training specifically tailored to training tasks. An Intelligent AAR (IAAR) concept addresses future needs relative to providing adaptive training for the AAR. Intelligent Tutoring Systems (ITS) promise the ability to provide adaptive training focused on the specific needs of the trainee or team. ITS has provided that type of training in the past but usually at the cost of specialized ITS implementations each time. Many benefits can be realized through enabling an AAR to provide a common service oriented approach leveraging ITS as a reusable service. The approach allows a training system the ability to provide a cloud-based service that other actors can use to enable IAAR for any type of training. Such an approach must be capable of adapting to a wide variety of training systems and configurations to be a truly useful service in the larger enterprise. Our solution to this problem is the injection of optimized technology into the AAR process and toolset with focus on building an adaptive approach for representing training strategies and assessing trainee proficiency. We evaluated and prototyped various technologies and tools which may be appropriate for the development of an IAAR system. Specifically, we investigated technologies, tools, and algorithms to facilitate machine/adaptive learning, keyword spotting, object detection, intelligent tutoring, data analytics, and others for applicability to the use cases defined in an Army training domain analysis.

ABOUT THE AUTHORS

Mr. Barry Clinger is the Chief Technology Officer for Riptide Software, Inc. and the Software Product Line Architect for PM TRADE's Live Training Product Line. He has 30 years of experience in NASA, DoD, and commercial solutions specializing in the architecture and design of large-scale, distributed near real time systems.

Mr. Christopher Sipes is the Lead Artificial Intelligence Engineer for Riptide Software, Inc. He specializes in the end-to-end implementation of custom Machine Learning solutions and comes from a background of full stack development for enterprise applications.

Mr. Nate Blomberg is Principal Investigator for the IAAR project and is an Intelligent Tutor lead on GIFT programs.

Mr. Robert Burch is the Chief Technology Officer for Dignitas Technologies. He has over 30 years of development experience in the modeling and simulation industry working in both virtual and constructive domains.

Dr. Jeremy Lanman is the Chief Engineer of Digitized Training for the Project Manager Training Devices at the U.S. Army Program Executive Office for Simulation, Training and Instrumentation. He holds a Ph.D. in modeling and simulation from the University of Central Florida, an M.S. in software engineering from Embry-Riddle Aeronautical University, and a B.S. in computer science from Butler University.

Mr. James Todd is the Assistant Program Manager of Target Modernization for the Project Manager Training Devices at the Program Executive Office for Simulation, Training and Instrumentation. He has just under 30 years of experience in DOD acquisitions associated with Live Fire training systems. Mr. Todd holds a B.S in mechanical engineering and an M.S in industrial engineering.

Automating the Training Feedback Paradigm with Intelligent After Action Review

Barry Clinger, Christopher Sipes	Nate Blomberg, Robert Burch	Jeremy Lanman, James Todd
Riptide Software	Dignitas Technologies	U.S. Army PEO STRI
Orlando, FL	Orlando, FL	Orlando, FL
barry.clinger@riptidesoftware.com,	bburch@dignitastechnologies.com,	jeremy.t.lanman.civ@mail.mil,
chris.sipes@riptidesoftware.com	nblomberg@dignitastechnologies.com	james.a.todd28.civ@mail.mil

INTRODUCTION

Modern combat is complex and demanding. To fight and win, we must train our Soldiers to successfully execute their Techniques, Tactics, and Procedures (TTPs). Therefore, every training event is an opportunity to improve Soldier's and leader's readiness. Soldiers and leaders must know and understand what happened or did not happen during every training event to improve their individual and collective-task performances. After-Action Reviews (AARs) in the training domain are designed to provide Soldiers and units feedback on mission and task performances. After-Action Reviews identify how to correct deficiencies, sustain strengths, and focus on performance of specific Mission Essential Tasks List (METL) training objectives.

As training migrates from individual and crew to more collective based, the feedback to the individual Soldier about improvement opportunities and corrective actions for non-standard performance gets lost in the collective view. Every training event should provide both collective and individual feedback to allow for and support growth and improvement opportunities. A more automated process of collecting, consolidating, and analyzing the data from training events is needed, one which compares individual, crew, and collective training tasks against the standards and provides focused individual feedback for improvement. This individual feedback should look at those aspects of the standards with the greatest opportunity for improvement in readiness. The goal at the end of each training event, each Soldier should receive both crew and/or collective feedback, as well as individual feedback.

The primary objective of this effort is to prepare Intelligent AAR (IAAR) products and supporting processes that distribute the AAR products to the AAR participants. Currently, a live collective training event AAR occurs with only a copy of the AAR presentation provided to the participants. With the availability of the internet, secure data distribution, and cloud technology, a broad and deep AAR experience is possible. Ultimately, the training system will automatically create targeted AAR products and provide each to selected personnel in the training event.

DOMAIN ANALYSIS, USE CASE, AND IMPLEMENTATION PATH

As part of our research, we conducted a domain analysis of the AAR and Take-Home Package (THP) capabilities of the U.S. Army Force-on-Force (FoF) and Force-on-Target (FoT) live training systems. This analysis identified existing AAR gaps and limitations as well as opportunities for the application of advanced adaptive/machine learning techniques to aid in individual, crew, and collective training feedback. The analysis included the breadth of current US Army live training systems ranging from a simple platoon-based, land navigation exercise to a combat training center brigade level rotational event. The training systems were assessed on a scale of 1 to 9 (9 being best) against the following seven criteria: Quantity, quality, availability of data, competent / effective examples demonstrating what "right" is, availability of remedial training materials, availability of advanced training materials, and the ability to enhance the AAR experience in the future. The resultant summary scoring is shown in Figure 1 below.

Domain	Pros	Cons	Quantity	Quality	Availability	Effective Ex.	Remedial Training	Advanced Training	Enhance	Impact
Live Fire Urban	• Could provide quality examples for a SH	• Currently limited to mostly video data	4	2	2	6	5	5	7	7
Collective Urban	• Significant impact based on the number of soldiers	• Large training areas without well defined ideal performance examples	4	2	2	3	4	4	7	7
FoT Lane-based	• Well defined tasks and doctrine • LOMAH use case provides detailed data	• Limited number of LOMAH ranges • Access to data from multiple ranges	9	6	2	9	9	9	5	8
FoT Maneuver Infantry	• Outstanding potential use case	• Current state lacks adequate data collection	1	7	2	7	7	6	9	4
FoT Maneuver Stabilized	• Most varied data set collected • Well defined tasks and doctrine	• Access to data is local at range	7	9	3	9	9	9	7	5
FoT Maneuver Un-Stabilized	• Similar to stabilized gunnery	• Less data available than stabilized gunnery • Access to data is local at range	3	6	3	7	7	7	7	4
FoF Homestation	• Great potential to analyze squad to company performance	• Currently lacks defined scenarios, exercise objectives	3	7	2	8	9	6	8	6
FoF CTC	• Significant impact to the Army	• Massive training/amount of data	9	7	8	1	9	6	7	8
Land Navigation	• Cloud-based, no hindrance to data collection • Well defined tasks and doctrine	• Limited use case	2	4	9	9	9	9	5	8
MedSim	• Medical procedures are well defined • Large potential for impact	• Small number of LT2 fielded ranges	6	2	2	6	5	5	7	7

Figure 1. DAR of Existing Live Training Systems AAR & THP Capabilities

Data From Tactics, Techniques, and Procedures (TTPs)

Training Event: # of Day / Night Runs

Scenarios: # of Offensive Steps
# of Defensive Steps

Steps: # of Targets Exposed per Step
- Target Distance
- Target Type
- Exposure Time
- Exposure Sequence

Data From Raw Training Event

General
- In Tank Video
- Field Camera Video
- Tactical Audio
- Crew Audio
- Vehicle Position / Location
- Whiskers
  - Tank Commander
  - Gunner

Target Engagement
- Trigger Pull
- Hit / Miss
- Timing
- Weapon Type
- Ammo Type
- Target Exposure
- Position Info

Crew Evaluation
- Crew Cuts
- Scoring

Figure 2: DRTS Live Fire Exercise Data

Because of the domain analysis, we selected the FoT Maneuver Stabilized (gunnery crew training for tank, Bradley and Stryker platforms) use case mainly for its large variety of data, well-structured doctrine and established AAR procedures. The Army's Project Manager for Training Devices (PM TRADE) Digital Range Training System (DRTS) provides command and control of live fire stabilized gunnery exercises and the production of the resultant AAR artifacts. The data available from a DRTS live fire exercise is listed in Figure 2. In tank gunnery exercises the exercise was broken into **runs** which contain *n* number of **steps**, with each **step** containing a sequence of target exposures of different types, distances, and sequences; all under strict timing control. This highly structured event allowed a simple application of heuristic techniques with natural discrete start and stop times contained on the step boundaries to maintain automated AAR organization. In addition, the large amount of instrumented data from the tank and the surrounding range provided an outstanding first use case to explore machine learning on multiple data types with this research focusing primarily on video and audio sources.

A Decision Analysis Report (DAR) (Figure 1) utilizing a formal pairwise decision analysis process determined available technologies and tools appropriate for IAAR system development. The survey collected industry standard and leading-edge products in the broad categories of (1) machine learning frameworks, (2) data solutions, and (3) rules engines. Machine learning frameworks are computation libraries that provide abstractions over efficiently implemented and well-known algorithms. They provide an out-of-the-box solution to distribute the computations necessary to train a machine learning algorithm. Machine learning in this research provided us the ability to augment the existing exercise data by identification of teaching points that would traditionally be identified by the trainer/instructor. The second category of tools considered allowed for the non-relational storage of data housing any data type or format (e.g.; photos, video, audio, documents, etc.) and is designed for frequent access. The tool selected housed the repository of information ingested from the live fire range and supplemented by assertions from both the machine learning framework and rules engine. Lastly, the rules engine or ruleset management system allowed our system to separate the problem space on a set of defined rules or heuristic markers. In our design, the rules engine provided a structure from which to launch appropriate machine learning models, branching, and potentially apply alternate solution types based on the conditions presented.

In terms of an implementation path progression generally characterized as crawl, walk, run; we considered this use case selection paired with IAAR techniques as a "*walk*." A precursor "*crawl*" proof of concept entailed capturing data from a live fire marksmanship range controlled and instrumented with PM TRADE's Targetry Range Automated Control and Recording (TRACR) and augmented with an acoustic sensor system, Location of Miss and Hit (LOMAH), capable of transmitting X-Y coordinates of fired round locations in respect to the target silhouette. In particular, the work focused on capturing the firing data using xAPI, for the purposes of providing (1) individual feedback and access to personalized remedial interactive multi-media content based on shot grouping patterns, (2) aggregated data views for trainers and range operations personnel, (3) flexible data views for training researchers and resource analytics, and (4) automated sharing of qualification data with the Army Training Management System (ATMS) (Durlach & Washburn, 2015).

The challenge of this next step "*walk*" use case was to: (1) capture the data from completed DRTS exercises, (2) transform the data from a real time operational database to a data repository organized for efficient IAAR analysis and processing, (3) apply heuristics, machine learning, and other techniques to gain insightful AAR artifacts, (4) present those AAR artifacts in a Soldier accessible, web-based medium, and finally (5) identify meaningful trending and data analytics that can improve U.S. Army training doctrine.

RISE OF ANALYTICS THROUGH MACHINE LEARNING

Within the commercial market space there have been significant gains in big data and distributed processes that support enhanced data driven analytics (Najafabadi, et. al., 2015). This rise of capabilities aligns with the desire to enhance the AAR process through automated analytics of training data into AAR products. Also, vast improvements in video and audio data valuation, machine learning, and heuristics tools to 'assess' training proficiency beyond the data and potentially into unit and individual behaviors allows the enhanced IAAR application to be a near-term reality. Our focus early in this research was on the application and assessment of several machine learning models and architectures.

Applying Machine Learning to the Maneuver Stabilized Gunnery Use Case

For machine learning to successfully become an applicable solution, the following criteria needed to be addressed (Domingos, 2012): (1) an algorithmic approach that can be learned empirically using data, which requires the data to have features that can be extracted, and (2) there exists a quantifiable error/loss between expected and actual output that can be minimized during the learning process.

For the maneuver stabilized gunnery use case described earlier, the most applicable uses of machine learning were:

1. Assessing the accuracy, sequence, and effectiveness of verbal commands exchanged by the tank crew during the exercise—Army doctrine defines a strict vocabulary and sequence of verbal cues exchanged among tank crew members for enemy engagements. Incorrect, confused, or delayed verbal communication results in reductions in a crew's score (crew cuts) during evaluation and provides AAR teaching opportunities (U.S.

Army FM 17-12). An applied "audio recognition" model provided detection of spoken commands combined with a timeline of tank crew events, allowed assessment of crew cuts penalties.

2. Providing supplemental data to assist evaluators in determining appropriate scanning techniques and threat prioritization during the exercise—In each tank engagement step, the tank crew is presented with multiple threats at different distances and behaviors that the crew must successfully prioritize within tight time constraints (U.S. Army, FM 23-91). The application of an "object detection" model allows us to detect and classify range targets from the thru-site video feeds to populate doctrinally-based prioritization tables. Combined with live fire engagement data from the range, the crew is accurately assessed against prioritization doctrine for a given step and assessed crew cuts for mistakes.

3. Verifying correct ammunition loading protocol when operating the breech—An implementation of an "image classification" model classifies whether the breech is in the correct position combined with the "audio recognition" model that verifies the breech is up before the Loader crew member announces "Up", designating the round is ready to fire.

Applied Machine Learning Models for Verbal Tank Crew Commands

We adopted Google's implementation of the small-footprint "keyword spotting" model architecture that is presented in a Google report (Sainath & Parada, 2015). The "keyword spotting" model architecture is a small convolutional neural network that detects a specific set of verbal utterances that the model trains on. While traditional speech to text solutions are very complex and require a deep neural network and significant amounts of data to train, the "keyword spotting" problem is simpler and can be performant when trained on a modestly sized dataset. Due to the challenge of detecting verbal commands that are definitive with respect to the doctrine, "keyword spotting" is a good fit, and has proven practical to train and evaluate.

Initial attempts at utilizing Mozilla's open source implementation of the "deep speech" model architecture (Mozilla Research, 2018) presented in the paper from Baidu Research (Hannun, et. al.; 2014) were abandoned. The "deep speech" architecture is a robust deep neural network architecture that produces text transcripts from audio files, and trains on audio clips with unaligned text transcripts. At the time of our investigation, there were no pre-trained models available for inference or transfer learning, and thus we would have to train the model solely from our own custom datasets. The complexity of the model architecture that allows the deep learning of spelling, time alignment, and other complexities from unaligned text transcripts requires a significantly large dataset, of which we did not have the resources to purchase or build, and thus could not establish a performant model.

Applied Machine Learning Models for Assessing Appropriate Scanning Techniques and Threat Prioritization

Our initial approach was to use image classification, with performing transfer learning on Google's inception model architecture (Szegedy, et. al.; 2015) to perform a binary classification of whether or not a target was present in an image. This approach was not ideal, however, as the inference would not indicate where the target was located. An additional consequence of performing a binary classification is that it did not provide any information on the type of target in the image. Adding classes for each type of target would mitigate this, but would not properly indicate multiple targets within the field of view. To address these concerns, we instead adopted Google's object detection Application Programmer Interface (API). The object detection API is a pipeline for training object detection model architectures that have similar input and output specifications (Fergus, et. al.; 2005). There are many model architectures available in the API. We researched, trained, and evaluated the following four model architectures:

1. Faster R-CNN: This model is an industry standard region-based convolutional neural network or Faster R-CNN. Through research and testing we found that Faster R-CNN was a great option for accurate predictions, but it lacked the efficiency of other models in training and processing the predictions. Faster R-CNN was complex, the model contained large numbers of hidden, convolutional, and fully connected layers, which lead to tradeoffs.
    - Pros: (1) Faster R-CNN accurately predicted the location of the objects that were trained, (2) Faster R-CNN was easy to train. The configuration was straightforward and required a minimum time to configure and setup.

- Cons: (1) Faster R-CNN is comparatively slow. The time required to make a single prediction is usually less than a frame per second even with dedicated, advanced hardware ruling out this model for our application.

2. Mask R-CNN: Mask R-CNN is an architecture based on the Faster R-CNN. In parallel with the predictions made with R-CNN, Mask R-CNN placed a mask on the detected object along with a bounding box. The model slightly added to the computational cost of Faster R-CNN, but it gained the ability to see a full outline of the object for detection.
   - Pros: (1) Added masks provided additional capability for potential future applications. (2) Maintained the accuracy of Faster R-CNN.
   - Cons: (1) The slowest model we considered. (2) More difficult to set up due to the masks required in the annotations.

3. R-CFN: Due to the simple layout of the architecture, R-CFN executed with similar accuracy as Faster R-CNN in less time. R-CFN had many fully connected layers placed on top of each other that attribute to its simple design and efficiency. The model deduced position sensitive scores for each of the pixels and created a confidence score of the object's location based on the scores.
   - Pros: (1) A middle ground between speed and accuracy (moderate performance and moderate accuracy). (2) Straightforward to configure and setup. (3) Simple models lead to more efficient usage.
   - Cons: Not as accurate as Faster R-CNN for the detection of small objects.

4. Single Shot Multi-Box Detector (SSD): SSD was the highest performance model we assessed. The accuracy was acceptable, but the speed was exceptional. This model did, however, struggle with small object detection. The model relied on a small number of layers to make predictions. SSD eliminated a large amount of the layers contained in other models resulting in less accurate predictions. However, this model relied on a more sophisticated detection algorithm for identifying different features within the frames. The model reshapes, filters, and rotates every image to deduce the best feature maps, improving accuracy.
   - Pros: (1) SSD was highly efficient due to its simple architecture. (2) Straightforward to configure and setup. (3) Required less hardware.
   - Cons: (1) The lack of complexity made it hard for the model to generate accurate predictions on small objects.

Applied Machine Learning Models for Assessing Correct Ammunition Loader Protocol

Using Google's "inception" model was not an optimal choice for addressing the correctness of scanning technique and threat prioritization, as discussed above. However it was an optimal choice for addressing the correctness of the ammunition loader protocol. Performing a binary classification on images from the loader player unit video with the breech in view was sufficient in conjunction with the audio recognition on the recorded crew communications. The "inception" model architecture is widely accepted as a performant solution for image classification. There are many pre-trained versions available (the most popular being those trained on the ImageNet dataset) that are straightforward to perform transfer learning on with a custom dataset.

Application of a Heuristic-based Software Service to Organize the IAAR Analysis Process

Our research application, designed as a Service-oriented Architecture (SOA), contained a rules service built on the open source Java Drools Business Rules Management System (BRMS). Drools BRMS is a collection of tools that allow separation and reasoning over logic and data. The core rules engine was capable of both forward and backward chaining inferences. In general, the rules engine consisted of a knowledge base and a set of production rules. The system applied the production rules to data in the knowledge base creating additional data as output. This deduced data was added to the knowledge base where it contributed to future inferences.

During our research, we developed a set of doctrine driven production rules. Our rules were based on armor crew Table VI Gunnery and crew level Direct Fire Engagement processes. The rules service communicated with the machine learning service to detect crew communications and actions from the audio and video files in our knowledge base. The analysis process used multiple iterations of forward chaining logic to make a deductive closure on events in the knowledge base. This was done by applying production rules to extract previously uncaptured events. Uncaptured events may be appropriate actions, inappropriate actions or lack of required actions taken by the crew. These events were added to the knowledge base as deduced events or potential crew penalties. As a result, our system could not only add to the number and type of existing AAR artifacts, but also accurately deduce scoring penalties based on the combinatorial assessment of multiple artifacts much like a human instructor. Ultimately, this led to the reduction in the amount of time required by instructors to prepare AARs and enabling a completely automated IAAR.

Army training doctrine provides TTPs used by armor crews to achieve expertise while conducting their mission in any operational environment. Therefore, it is important that our IAAR rely on Army doctrine as the main reference in developing the intelligence associated with our system. Measures of Performance (MoP) and Measures of Effectiveness (MoE) were obtained from the doctrine listed in Table 1.

Table 1. U.S. Army Armor Crew Doctrinal Sources

Designation	Title	Description
TC 3-20.31 (March 2015)	Training And Qualification, Crew	Provides training principles and techniques for use by the crew to gain proficiency in engaging and destroying threats efficiently in any operational environment
TC 3-20.31-4 (July 2015)	Direct Fire Engagement Process (DIDEA)	Provides the standardized direct fire engagement process for crews, teams, squads, and small units. It provides principles coordinating the actions of these entities for establishing cohesion through a common, standardized engagement process.

Relating Causal Issues from Raw Training Data

Assessments were based on first order and second order conditions. First order conditions were comprised of simple questions that could be answered directly from the raw data. “Was the target hit?”, “Was the target killed?”, “How many rounds were fired?” and many other criteria could be evaluated upon as first order conditions. Evaluations were categorized as “At Expectation”, “Above Expectation”, or “Below Expectation.” These first order conditions were the basis for creating the “language” that defined second order conditions. Second order conditions were comprised of any combination of first order conditions that tried to answer more complex questions. Feedback could then be based off these more meaningful evaluations. For example, using the first order conditions mentioned above, evaluating “Target Killed” at “Below Expectation” and “Target Hit” at “Below Expectation” indicated that the target was not killed because it was never hit. Feedback could then be provided to improve accuracy. Another rudimentary example would be to evaluate “Target Killed” at “Below Expectation” and “Rounds Fired” at “Below Expectation” which indicated the target was not killed because the minimum number of rounds to kill it were not fired. Feedback can then be provided for the identification or engagement process. These were only a few simple examples, and as more data and different first order conditions are created, the vocabulary for more meaningful second order conditions will mature. Therefore, as the system becomes more robust, feedback can become more directed and precise.

APPLICATION OF AN INTELLIGENT TUTORING SYSTEM (ITS)

An Intelligent Tutoring System (ITS) is a computer system that aims to provide immediate and customized instruction or feedback to learners, usually without requiring intervention from a human teacher (Mislevy & Gitomer, 1995). ITSs have the common goal of enabling learning in a meaningful and effective manner by using a variety of computing technologies. There is a close relationship between intelligent tutoring, machine learning, cognitive learning theories and design; and there is ongoing research to improve the effectiveness of ITS. An ITS typically aims to replicate the demonstrated benefits of one-to-one, personalized tutoring, in contexts where students would otherwise have access to one-to-many instruction from a single instructor (i.e., classroom lectures), or no instructor at all (i.e., online lectures). In our research, we assessed the Army’s Generalized Intelligent Framework for Tutoring (GIFT) (Sottilare, et. al.; 2017), and applied our machine learning algorithms and heuristic models to develop a fully automated and IAAR capability.

Generalized Intelligent Framework for Tutoring (GIFT)

GIFT is a SOA based design that uses a message broker to route messages between the services. Figure 3 illustrates the current set of services in GIFT, contained in the yellow box, connected to external systems such as a Learning Management Systems (LMS) and training applications through the training gateway(s) (Sottilare, et. al.; 2017). In the figure, the common modules that make up GIFT instances are marked with a green star. Each of these are services and have a well-defined interface. In addition, since the GIFT tool supports both research and Intelligent Tutoring, the services and message structures are adaptable and can be extended as required for a transition. Figure 4 illustrates the minimum common reusable core of the GIFT system (Sottilare, et. al.; 2017). These services are the minimum to support ITS with the learner. The other aspects of GIFT are mainly used in support of ITS research and the specific implementation of connecting trainee devices. These services, referred to as the learning effect chain, are the "Learner Module", the "Pedagogical Module", and the "Domain Module." The "Learner Module" service manages the learner state which tracks the learner performance and levels of mastery as they progress through the training strategy. The "Pedagogical Module" service encodes the pedagogy, or the current method of providing training. This service can be adjusted to alter the manner of instruction and to provide both macro and micro adaptation recommendations for the learner. Finally, the "Domain Module" service supports the representation of the knowledge required for assessment of learner tasks and the training strategy for the specific domain knowledge.

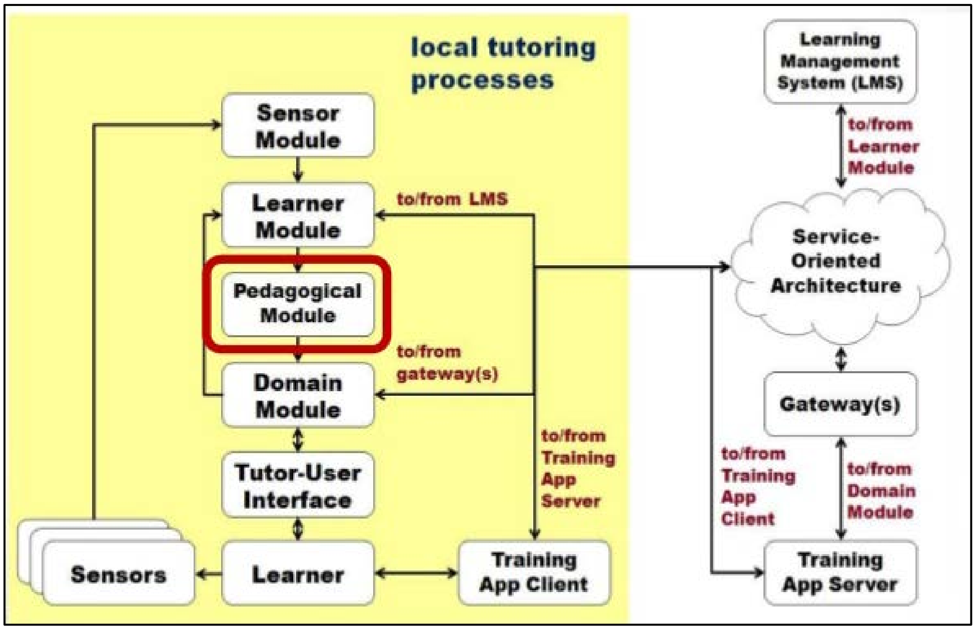

Figure 3: GIFT SOA System Decomposition

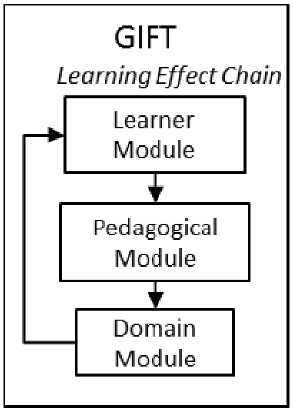

Figure 4: GIFT SOA Reusable Core Services

Our approach for the IAAR was to adapt GIFT to provide a set of software services that support connection to the DRTS system. The system provided interfaces for the instructors to specify the training strategy. The system also provided interfaces to specify metrics and trainee performance data for assessment. The system defined the approach for adaptation of the live data and provided “Data Collectors” to collect, filter, and/or classify exercise specific data (i.e.; target hit or miss) for assessment. The resulting services gave results and data on trainee proficiency based on the input training strategy specifications provided as AAR training doctrine artifacts.

Generating Intelligent Feedback from World State Data

GIFT generates feedback but currently is not tied to any type of world state data. For IAAR, we adapted this, and tied a feedback item directly back to the entity, location, and time data that occurred during a training exercise. This allowed for the implementation of many useful features. For example, time data could be used to synchronize the audio and video streams for the time that the feedback item occurred. World location data could later be used to map the feedback item on a 2-dimensional overhead map. Entity data could be used to tie feedback to a specific unit or crew member. The addition of this data enabled feedback to become more directed and precise, as well as made AAR review material more informative and intuitive.

Incorporating an Intelligent Training Strategy

To provide the interface, we developed a data model to represent domain knowledge and training strategy by leveraging the GIFT data model. This was represented as a set of XML data that decomposed what a learner must do to be proficient. For example, a list of tasks that must be accomplished with measures that tells the system how to determine when it’s done and how well the learner performed. The data also defined what the system should do next if the data is a specific value as part of the training strategy. For a simple example, assume the learner task resulted in a hit count of three as a task measure. The data could encode a strategy to say a hit count of three is low proficiency which prompts the system to provide feedback with something like “Adjust your aim.” In addition, the strategy might also specify that a hit count of three causes the scenario to get easier by making targets slow down or stop. On the other hand, the strategy might say that a hit count of ten has the system speed up targets or start shooting back—that is called micro adaptation. In the end, the training specialist encodes how the trainee should proceed by specifying the XML data for the assessment approach and the training strategy that the ITS interprets and acts on. At the end of the training session, the training strategy can also suggest the next course the student needs to take based on how well they did—that is called macro adaptation.

The advantage of this approach is it allowed the ITS to assess learner proficiency while using the training capability without tight coupling. There are several ITS solutions in the industry. However, most ITSs are tightly coupled, embedded in source code, and are specific to the training device. That tightly coupled approach makes it difficult to alter these ITSs for the next trainer or sometimes even for a new training task since the data is typically encoded into the system. The approach provided by GIFT results in systems being able to provide their own specific data driven descriptions and in return get back feedback/results and dynamic adaptation at runtime all based on their own training approach and data to measure.

RESULTS, LESSONS LEARNED, AND NEXT STEPS

This is an on-going research project where our initial results indicate that the combination of machine learning, heuristic techniques, and intelligent tutoring can successfully be applied to the existing AAR data captured from an instrumented live exercise to produce meaningful and insightful evaluation teaching points. Given our limited availability of data from live tank gunnery training events at Army ranges, our IAAR system successfully identified and detected crew cuts from audio and video data matched with U.S. Army doctrine and range system data.

The largest hindrance to the development of our models to date has been the availability of suitable live exercise data. Machine learning models crave large amounts of data in establishing “*what right is.*” Similarly, that same data can be used to create remedial training materials for Soldiers to review. Imagine presenting to the tank crew an example of “*what is right*” for a training task prior to execution, then conducting the live training, and immediately presenting that same crew with their performance overlaid with “*what is right*.” We believe the training benefits of such a strategy would improve comprehension of the tasks, retention, and future performance.

One lesson learned was how to navigate the classes of machine learning solutions and evaluate the appropriateness of a solution to the problem. For example, the initial approach taken for detecting targets was to use a binary image classification to indicate if there was a target in the picture. The purpose of detecting targets was to supplement the evaluator with information to assist in deducing incorrect scanning techniques or incorrect threat prioritization. Binary image classification was not a good fit as it did not indicate the type of target in the image, only if there was one present. Creating classes for each target type was not ideal either, as the image classification would produce its best guess of what the picture was, it could not indicate multiple instances of the same target. Image classification could indicate that there are different targets in the picture by providing multiple predictions with high confidence. However this ability is contingent on robustness of the training dataset, number of predictions produced from inferences, and the number of different targets containing classes. We also investigated an "Object Tracking" model, useful since it would maintain a history of which targets have already been in view and detected. This was not a practical solution, however, because the accuracy of "Object Tracking" models largely depend on smooth motion. The sudden differences in scale from changing magnification and complex translations were inhibitive to assuming smooth motion of the targets, especially at low frame rates. Finally, we implemented a working "Object Detection" solution, which allowed us to detect multiple targets, multiple target types, and their location, which can be a valuable supplemental tool to aiding evaluators deduce incorrect scanning techniques or incorrect target prioritization.

An additional lesson learned was the process of model discovery and evaluation to identify a good fit based on complexity, robustness, feature extraction process, online/offline training, and other model architecture characteristics. Our first selection for the audio recognition was a robust speech-to-text model architecture that trained on audio clips with their unaligned text transcripts. The model architecture was very deep and complex, and took massive amounts of data (2300+ hours of audio) and time to train (10-25 days). We had a limited dataset available to us, and the nature of the training data made it difficult to annotate new data. As a result, the model was in no way performant. Our next choice was the "Small-footprint Keyword Spotting," which was a significantly simpler model architecture that learns features using convolutional layers instead of the abstract feature learning from deep layers, and trains on one second samples of isolated utterances paired with the label. With as little as one hour of audio we could immediately see some performance improvements from the "Small-footprint Keyword Spotting" model architecture. The relationship between the complexities of the model architecture became an important consideration due to the amount of data and resources required to train it. This was an iterative pattern as we applied different and additional models to our data sets.

Existing AARs consist only of raw data for events that happened during an exercise. Giving context to that data in terms of what went well and what could be improved was integral for allowing the recipient of the AAR to digest and learn from it. One should be able to look at the AAR at any given point in time and quickly and effectively understand the lessons to be learned. There are still many data points and types of criteria that could be evaluated to add to the robustness of the IAAR system. Our research team was limited by the data set provided by DRTS. However, our next steps will include more data collection from other Army live training products and instrumentation systems such as the Combat Training Center-Instrumentation System (CTC-IS), Home-station Instrumentation Training System (HITS), Tactical Engagement Simulation (TES), and embedded training systems through new interfaces, sensors, telemetry, etc. that could further expand the IAAR system's capabilities.

REFERENCES

Domingos, P. (2012). "A few useful things to know about machine learning." *Communications of the ACM*. 55(10).

Durlach, Washburn, & Regan. (2015). *Putting Live Fire Range Data to Work Using the xAPI*. Proceeding of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2015, Orlando, Florida.

Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A. (2005). *Learning object categories from Google's image search*. Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 2005. Beijing, China.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., Ng, A.Y. (2014). Deep Speech: Scaling up end-to-end speech recognition. Cornell University Library. Available: https://arxiv.org/abs/1412.5567v2.

Mislevy, R., Gitomer, D. (1995). "The Role of Probability-Based Inference in an Intelligent Tutoring System." Wiley Online Library. ETS Research Report Series. Volume 1995, Issue 2, Pgs i-27.

https://doi.org/10.1002/j.2333-8504.1995.tb01676.x.
Mozilla Research. (2018). Expanding the Foundations of the Open Web. Accessed Online. Available: https://research.mozilla.org/machine-learning/.
Najafabadi, Villanustre, Khoshgoftaar, Seliya, Wald, & Muharemagic (2015). “Deep learning applications and challenges in big data analytics.” *Journal of Big Data*. Vol. 2:1. https://doi.org/10.1186/s40537-014-0007.
Sainath, T., Parada, C. (2015). *Convolutional Neural Networks for Small-footprint Keyword Spotting*. Proceedings of the International Speech (INTERSPEECH) Conference 2015, Dresden, Germany.
Sottilare, R., Brawner, K., Sinatra, A., Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). U.S. Army Research Laboratory—Human Research & Engineering Directorate. Accessed Online. Available: https://gifttutoring.org/.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). *Going Deeper with Convolutions*. Proceedings of the Computer Vision Foundation (CVF-CVPR) Conference 2015.
U.S. Army. (2000). Mortar Gunnery. FM 23-91. HQDA. Washington, D.C.
U.S. Army. (1964). Tank Gunnery. FM 17-12. HQDA. Washington, D.C.