Evaluating the Effectiveness of AI in Training Simulations

Robert Sottilare, Ph.D., Christopher Ballinger, Ph.D.
Soar Technology, Inc.
Orlando, Florida
bob.sottilare@soartech.com, cballinger@soartech.com

Keith W. Brawner, Ph.D.
US Army CCDC
Orlando, Florida
keith.w.brawner.civ@mail.mil

ABSTRACT

Artificial intelligence (AI) has become an essential element of the modeling and simulation industry and particularly military training and education. AI concepts take many forms (e.g., machine learning, intelligent agents, computer vision, and natural language understanding and generation). AI is used to create computer-based augmentations for training and education that include intelligent forces, virtual characters, instructional guides and coaches, and methods to automate processes, reduce labor, and assess trainee performance. AI tools and methods are used to model trainees and make optimal decisions about recommendations, feedback and support provided by computer-based tutors or adjustments to scenario difficulty based on learner performance and predicted task success. Military acquisition agencies regularly solicit for and receive AI solutions as part of deliverable training systems, and it is often tedious to validate the effect of these delivered solutions on military learning, performance, and readiness. Evaluations often take the form of formal learning effectiveness studies involving human participants, extensive institution reviews, data collections and finally the application of analysis methods to determine their impact on training outcomes. This paper examines intelligent agent-based methods to accelerate the effectiveness evaluations of deliverable AI concepts within training simulations. The goal of this research is to systematically consider AI design and evaluation processes to discover innovative methods to reduce the time and skill required to understand AI impact on training effectiveness, learning, and readiness. The driving motivation is to understand what methods, services and processes are needed in an AI testbed methodology to rapidly evaluate the effectiveness of various types AI using in training systems today. The output of this research is a set of recommended practices for designing and developing AI testbeds for training.

ABOUT THE AUTHORS

Dr. Robert Sottilare is the Director of Learning Sciences at Soar Technology, Inc., and the founding Chairman of the Board & Executive Director for the not-for-profit Adaptive Instructional Systems (AIS) Consortium, an industry group focused on enhancing the effectiveness of tailored training and educational solutions. He has nearly 40 years of experience as a researcher, designer, developer, and evaluator of instructional technology and training systems. He received his doctorate in modeling & simulation with a focus on intelligent systems at UCF. His experience spans government, industry, and academia. His research focuses on adaptive instruction including learner/team modeling, automated authoring, instructional management, and evaluation methods for intelligent tutoring systems (ITSs). He is the father of the award-winning Generalized Intelligent Framework for Tutoring, an adaptive instructional architecture. He is widely published with over 250 technical papers and over 2500 citations. He is a senior member of IEEE and founding Chair of the IEEE AIS Working Group, the AIS Conference, and the GIFT Symposia.

Dr. Keith Brawner is a senior researcher for the U. S. Army Combat Capability Development Command Soldier Center at the Simulation and Training Technology Center (CCDC-SC-STTC), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 14 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current efforts are on artificial intelligence for the Synthetic Training Environment Simulation and the Combat Capabilities Development Command Soldier Center.

Dr. Christopher Ballinger graduated from the University of Nevada, Reno with his PhD in Computer Science and Engineering. His thesis focused on using generative adversarial algorithms to solve production pipeline problems in a strategic real-time combat simulator. Since joining SoarTech, Christopher has continued to apply his expertise as an AI Engineer and has demonstrated his proficiency with a multitude of AI and machine learning methodologies, including novelty search, SOAR, and neural networks.

Evaluating the Effectiveness of AI in Training Simulations

Robert Sottilare, Ph.D., Christopher Ballinger, Ph.D.
Soar Technology, Inc.
Orlando, Florida
bob.sottilare@soartech.com, cballinger@soartech.com

Keith W. Brawner, Ph.D.
US Army CCDC
Orlando, Florida
keith.w.brawner.civ@mail.mil

INTRODUCTION

Military acquisition agencies regularly solicit and receive AI solutions as part of deliverable training systems, and the process to validate the effect of these delivered AI solutions on military learning, performance and readiness is often difficult and tedious (Toubman, 2019; Ovalle, 2019; Campbell & Bolton, 2005; Wallace & Laird, 2003). Currently, the evaluation of AI capabilities usually consists of a monotonous experimentation process (Wray, Woods, Haley & Folsom-Kovarik, 2016; Wray & Stowers, 2017) that is usually handcrafted and frequently produces inconsistent results. Further, a comparison of competing products is frequently not performed, or performed on a single standardized task.

Evaluation methodologies often include formal learning effectiveness studies involving human participants, extensive institution reviews, data collections and finally the application of analysis methods. This paper examines AI in the form of an intelligent agent testbed methodology (Hanks, Pollack & Cohen, 1993) as a tool to evaluate impact of various AI concepts integrated within training system processes. The goal of this research is to discover innovative methods that are accurate measures of effectiveness without the need for extensive research studies or training effectiveness evaluations. This paper seeks to answer the question – what methods, services and processes are needed in an AI testbed to rapidly model and evaluate the effectiveness of various types AI in training?

Artificial intelligence (AI) is now an essential element of the modeling and simulation industry and particularly military training and education. AI comes in many forms. Machine learning is the study of computer algorithms that improve without human intervention through their own experiences and through the use of data (Mitchell, 1997), and its algorithms and neural networks are used to explore relationships in data and use this information to classify current states and predict future events. Intelligent agents are autonomous entities which observe using sensors and act on their environment to achieve their goals (Anderson & Leigh, 2007). The discipline of computer vision seeks to understand and develop computers that sense and gain understanding from digital images or videos in much the same way that human use vision to understand their environment and master tasks like reading, writing, and drawing. Natural language programs use AI to understand and accurately process human language, and in some cases generate appropriate responses (Winograd, 1972). An understanding of different types of AI and their processes set the stage to define AI and to understand its intended role in computer-based training applications (e.g., simulations).

While the nature of AI is to be autonomous, AI capabilities are a human designed and produced (as opposed to a naturally occurring ability – e.g., animal instinct). AI capabilities share several salient characteristics. An AI agent senses or takes in information and judge the credibility and importance of that information. An AI agent may also think abstractly and apply knowledge and skill to favorably manipulate its environment and achieve its goals (van Lent, 2019). Some AI agents learns from their experiences and can quantify that learning (e.g., production rules, neural network weight updates) to improve future performance. In other words, AI agent capabilities often seek to mimic and automate human skills and processes. The goal within simulation is to reduce the need for human intervention and thereby reduce the human skills and the time/cost required to complete assigned tasks (e.g., author courses, augment forces with virtual entities, or otherwise perform a job previously done manually). A motivating factor for the research described in this paper is to understand how to optimally exploit a testbed methodology to improve the evaluation and validation processes for AI capabilities in training. To optimize our desired outcome, we begin by describing the current uses of AI capabilities in military training systems.

AI in Military Training Systems

AI capabilities are used widely to enable computer-based augmentations within military training and education systems, and these include intelligent forces, virtual characters, instructional guides and coaches, automated processes, recommendation engines and methods of performance assessment. AI capabilities are also used to model trainee states and traits. Using trainee models, AI-based instructional systems (e.g., intelligent tutoring systems – ITSs) attempt to optimize recommendations, feedback, and other instructional interventions. AI in training systems may also adjust scenario difficulty based on individual trainee or team performance and predicted task success to maintain a balance between trainee capabilities and the relevance of training experiences (Vygotsky, 1993).

AI in the form of computer vision supports methods for acquiring, processing, analyzing, and understanding images and other visual data. In military simulations, AI is used to detect, and recognize trainee states in immersive simulations. Trainee states may involve the use of motion tracking to model the physical state and movement of people and objects. Eye tracking sensors collect eye movement and pupillary data to understand trainee focus during visual tasks. Virtual characters or personalized assistants for learning (PALs; Somers, Oltramari, & Lebiere (2020) may include image recognition software to detect trainee facial markers and classify their emotional states (Hemalatha & Sumathi, 2014). The ability to accurately classify trainee emotional states in real-time enables the opportunity for more social and empathetic agents (Bartneck, 2002; Van Dyke Parunak, 2006). Natural language interaction (NLI) is used to understand trainee verbal communications and generate appropriate responses, but it can also be used to detect and classify trainee emotions through voice inflection or language content (Ezhilarasi & Minu, 2012). Table 1 provides exemplars of machine learning, perception and action methods, and knowledge-based applications that are used in military training.

Table 1. Examples of AI Used in Military Technology-Based Training Applications

AI Type	Method	Goal	Example Use in Military Training
Machine Learning	Supervised Learning (labeled data)	Classification (predict correct label for new input data) or regression (predict outcome based on input)	• Assessing trainee progress toward goals • Understanding the context and influencers of skill acquisition
	Unsupervised Learning (unlabeled data)	Discover patterns and detect outliers in data	• Clustering or grouping of trainees with common attributes (e.g. low performers)
	Reinforcement Learning	Improve performance over time by maximizing reward	• Optimizing learning by rewarding behaviors that lead to progress
Perception and Action	Intelligent Agents	Goal achievement	• Perception and action of autonomous entities (e.g., virtual humans) reinforced by rewards
	Natural Language Processing	Comprehension of verbal and written language	• Processing trainee speech and responding appropriately
	Computer Vision	Detection, segmentation, localization and recognition of objects within images	• Image recognition by virtual humans
Knowledge-Based Systems	Expert Systems	Solve complex problems by reasoning within a body of knowledge	• Deterministic logic used to support decision-making of virtual entities with complete understanding of context
	Intelligent Decision Support	Support decisions regarding determinations, judgments, and courses of action	• Probabilistic logic used to support decision-making of virtual entities with incomplete understanding of context

AI Testbed Design Goals and Principles

In this section, we introduce the basis for design goals and principles as they have evolved under the US Army's Learning and Readiness Intelligent Agent Testbed (LARIAT) research and development project. The LARIAT project is exploring opportunities to use the power of intelligent agents as observers and actors to automate part or all of the effectiveness evaluation process. The primary goal for LARIAT and any AI testbed capability is to *reduce the time and skill required to conduct AI effectiveness evaluations and determine the influence of an AI capability on trainee learning, retention, and transfer of training*. This goal may be achieved by guiding users through a repeatable evaluation process and by automating steps in the process to ensure consistency and fairness in the evaluation methods, and the ability to evaluate AI methods under a variety of scenarios and conditions. An AI testbed should also adhere to proven design principles and provide a methodology to improve the understanding of both the functional requirements and the operational behavior of the AI under evaluation.

The design of an agent-based testbed is intended to automate the processes of perception and action, and maximize the number of scenarios and conditions under which the AI will operate. Within the AI testbed, intelligent agents are designed to both observe and act on the AI under evaluation and in some cases act as augmentations in testbed scenarios to support realistic interactions and behaviors. This is like the participation of human observer controllers in live training exercises and the participation of other human who fill needed roles as training aids to ensure the training experience meets its intended objectives.

According to Hanks, Pollack & Cohen (1993), testbeds are integral tools in the exploration, confirmation, and generalization of research. During exploration, the AI testbed provides an environment in which the AI under evaluation will behave under varying conditions. While behaviors in the exploratory phase of research may be more loosely defined, the testbed is expected to drive, observe, and report behaviors. During confirmatory evaluations, behaviors are defined more explicitly and the evaluator tests specific hypotheses. The testbed should incorporate targeted measures of AI performance assessment from which quantitative results about the effectiveness of the AI method can be derived.

The ability to author scenarios in which specific behaviors can occur, be observed, and be reported is essential to the testbed design. For example, if we are evaluating agent behaviors with respect to lethality, then lethality is defined to a degree that the testbed can observe the agent's behavior and classify its level of lethality. Confirmatory evaluations require the testbed to provide a high degree of experimental control and facilitate experimental setup, execution, data collection, and data analysis.

Finally, in the generalization phase of research we seek to define the degree to which results can be replicated by others. The ability to exercise the AI under a variety of parameterized conditions within the testbed environment aid achievement of generality or the observation that different agents in different test beds exhibit common behaviors under common sets of conditions. Now that we have identified the high-level design goals and principles for an AI testbed, the next step is to identify specific processes and services required to support objective evaluations of various types of AI capabilities.

AI TESTBED PROCESSES

In this section, our discussion focuses on design recommendations for the AI testbed processes. The LARIAT architecture and its associated design process will be used to highlight considerations for a scalable, extensible, easy-to-use evaluation testbed. The authors realize that the design process for LARIAT is not the only approach to designing an evaluation testbed and the LARIAT project information is shared as an exemplar. It is intended to be a source for discussing testbed design tradeoffs. To support this discussion, the following taxonomy is provided to identify the relationship of items within the LARIAT architecture.

- ***Projects*** – provides an overarching method to save all the resources (e.g., scenarios, agents, AI under evaluation) related to a particular evaluation effort as a single heading for storage and retrieval
- ***LARIAT scenarios*** – a collection of configured elements (simulation, simulation scenario, agents, measures, assessments, algorithms) needed to run an evaluation; multiple LARIAT scenarios evaluating different conditions can be saved under a single LARIAT project

- ***Simulation scenarios*** – a native version of the simulated environment (terrain), entities and associated entity data (e.g., location, state, position)
- ***Agents*** – configured entities used to support observations and interactions in the evaluation environment
- ***AI under evaluation*** – configured capabilities that are subject of the evaluation effort and containerized based on AI type (e.g., machine learning algorithms, agents, computer vision, natural language interaction)
- ***Reports*** – configured information that is automatically populated based on the selection of measures and outcomes

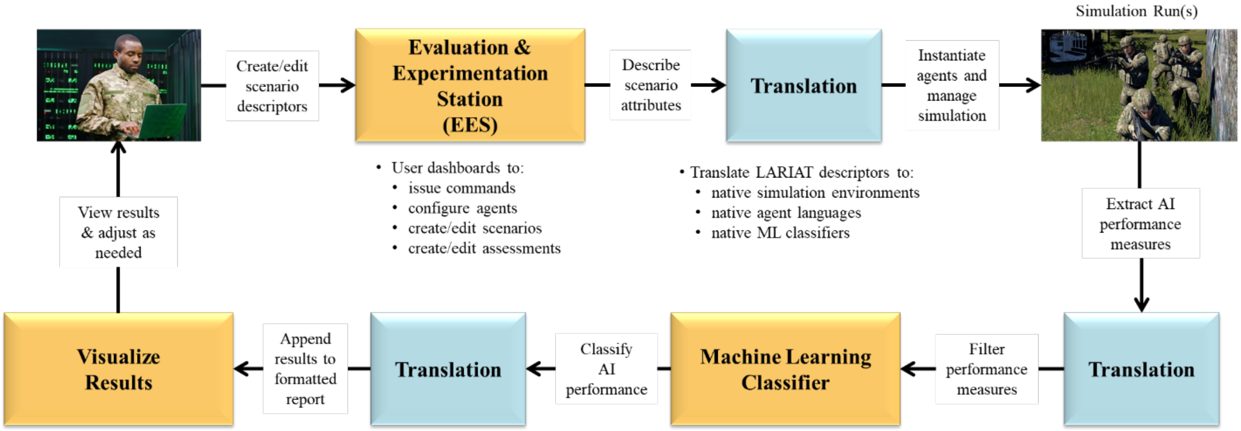
Create/edit scenario descriptors
Evaluation & Experimentation Station (EES)
Describe scenario attributes
Translation
Instantiate agents and manage simulation
Simulation Run(s)
• User dashboards to:
• issue commands
• configure agents
• create/edit scenarios
• create/edit assessments
• Translate LARIAT descriptors to:
• native simulation environments
• native agent languages
• native ML classifiers
View results & adjust as needed
Extract AI performance measures
Visualize Results
Append results to formatted report
Translation
Classify AI performance
Machine Learning Classifier
Filter performance measures
Translation

Figure 1. Simplified View of an AI Testbed Process Flow

To support the development of an AI evaluation scenario, a browser-based operating station known as the Evaluation & Experimentation Station (EES; Figure 1) is designed with a series of dashboards or user interfaces to facilitate user interaction and the management of each of the six fundamental LARIAT processes below:

- Authoring evaluation scenarios
- Integration of external systems (e.g., simulations, serious games) and their associated scenario editors
- Integration and configuration of machine learning algorithms and other AI capabilities
- Creation and configuration of agents
- Storage and retrieval of projects, scenarios, models, AI under evaluation, agents, and reports
- Project and simulation initialization

Authoring Evaluation Scenarios

An AI testbed process is required to author (create/edit) scenarios and save one or more scenarios along with other resources as projects. Authoring dashboards include user interfaces to create and configure *AI under evaluation*. The authoring process should also support: 1) methods to configure autonomous agents that serve roles in evaluation scenarios (e.g., opposing forces, observers or representative urban populations), 2) simple methods to link configured agents/populations to entities in the external simulation that will be acting as a sandbox for the evaluation, 3) methods to identify measures of performance for the agent(s) under evaluation, 4) classifiers to assess agent performance, 5) a simulation scenario to provide the conditions needed for the evaluation, and 6) a method to visualize the evaluation results.

Integration of External Systems

To provide a sandbox in which to evaluate AI capabilities, there are two primary design considerations: 1) native authoring of scenarios, and 2) real-time interaction between the AI testbed architecture and the simulation. The design team selected the Rapid Integrated Development Environment (RIDE) Unity-based simulation (USC ICT, 2021; https://ride.ict.usc.edu/; Ustun, Kumar, Reilly, Saijadi & Miller, 2021) to serve as the initial LARIAT sandbox. Our intent is to expand to include other simulations in a distributed sandbox, and the above architecture supports such functionality. Further, being built with interchangeable parts allows the AI game technologies in game environments (such as RIDE) to be ported as intelligent agents into militarily relevant environments such as OneSAF, a constructive simulation that represents the behaviors of individual and aggregate level entities.

To address scenario authoring within RIDE, LARIAT launches the Unity scenario editor in its authoring dashboard. To address real-time interaction, LARIAT directs agent actions and receives agent observations through the RIDE API and runtime environment. Initialization and simulation management is also facilitated through the RIDE API. Future integration of external systems will be facilitated by a standardized gateway specification. Machine learning classifiers will be integrated through the Waikato Environment for Knowledge Assessment (WEKA; Hall et al, 2009) which provides an open source machine learning suite that can be accessed through either a graphical user interface, standard terminal applications, or a Java API. The EES will provide dashboard selections to integrate compatible (tested and validated) external systems.

Integration of AI Capabilities for Evaluation

The initial AI container will support the integration of intelligent agents that will include both agents under evaluation and agents in various roles within the scenario. It is envisioned that additional containers will be constructed to facilitate the integration of various AI categories (e.g., machine learning, computer vision, natural language interaction). The EES will provide dashboard selections to integrate compatible (tested and validated) AI types.

Creation and Configuration of Agents

The initial LARIAT baseline models the behaviors of intelligent agents as finite state machines (FSMs) using a lightweight agent editor and a dashboard to allow evaluators to configure FSMs using a graphical interface. More complex agent architectures like the Soar Cognitive Architecture (Laird, 2019) and Sigma (Rosenbloom, 2013) are candidates for future integration within LARIAT. Various agent types and authoring dashboards are envisioned: simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents, and learning agents.

Storage & Retrieval Process

The ability to create and edit AI testbed resources also requires the ability to store, search and retrieve those same project resources. Defining the hierarchical relationship between projects, scenarios, agents, AI under evaluation and other project resources is critical to this process. Currently, resources can be located through simple name searches or by associating resources with a specific evaluation project. However, there are plans to provide data fields to support metadata searches in near-term versions of LARIAT. Project and scenario data, agent configurations, reports, messages, and other communications are stored in a resource repository in defined formats.

Project and Simulation Initialization

Once a project is retrieved from the resource repository, the scenario attributes, agent definitions, machine learning classifiers, assessment criteria and other pertinent information must be translated for use by the external simulation (sandbox) and the rest of the AI testbed architecture. Scenarios are initialized (e.g., entity types, position, location and state data) are conveyed to the simulation and instantiated so they can be visualized (as needed). The machine learning classifier is initialized with a trained model and begins to classify performance as soon as the scenario begins.

AI TESTBED SERVICES

In addition to processes, an AI testbed architecture should be designed to provide a set of services to enable steps in the various processes. For example, referring to Figure 1, there exists a need for translator services (e.g., LARIAT storage format to the appropriate intelligent agent language) to format information for future operations later in the initialization process. This section describes a minimal set of services that provide both user interfaces in the form of dashboards and the necessary structure to enable the testbed processes described in the section above.

Intelligent Agent Service

To enable the evaluation of intelligent agents in a testbed architecture, it is necessary to either import or create intelligent agents. The LARIAT project is leveraging the DARPA Deep Agent framework (Garibay et al, 2020) to create agents as finite state machines that represent military entity states and behaviors (observations and actions). Future development will include methods to import common agent types (e.g., rule-based agents, utility agents or learning agents) into the LARIAT architecture.

Communication Services

The AI testbed architecture provides two distinct communication pipelines, the LARIAT bus and the Generic Event Representation (GER) bus. Both buses based on Apache Kafka and Apache Flink. Kafka is a framework implementation of a software bus using stream-processing and Flink is an open-source, scalable, unified stream-processing and batch-processing framework used for data analytics. The LARIAT bus provides smart data exchange for the various LARIAT architectural elements (e.g., agents, authoring tools, results viewer). The GER bus provides a continuous flow of simulation data to the machine learning classifier and then shares the results with the visualizer. Together the assessments provided by WEKA along with the events and context identified by the GER process provide the data necessary to evaluate intelligent agent performance of tasks under conditions assessed by standards.

EES Support Services

The user interfaces within the EES support creation, editing, training, evaluation, and experimentation. The EES also requires additional services to support simulation management, AI testbed performance and load balancing, data management, export and visualization, model training, agent lifecycle management, and hypothesis development and testing.

Translator Services

Translator services are required to filter and format data for consumption by various processes in the AI testbed pipeline. Translators are needed to translate data descriptors into native formats of simulation environments, agent languages, and machine learning classifiers. The methods, timing, and frequency of translation influences the data management load of the AI testbed architecture and should be carefully considered during the design process.

Performance Assessment Service

The performance assessment of the *AI under evaluation* is another critical service where the data management workload of the AI testbed architecture can be influenced by the scenario complexity (e.g., number of entities), the assessment criteria selected (e.g., direct vs. inferred measures), and the classification methods, timing, and frequency. A series of benchmarking experiments will likely drive the final implementation for the LARIAT performance assessment service and design decisions about assessment frequency and methods.

After Evaluation Report Service

Depending on the type of results to be visualized, the real-time vs. stored final report format of each evaluation will also impact the workload associated with data management. The EES provides a simple dashboard format for detailing the required output for results (e.g., text or CSV), but additional agent observations may also be captured and detailed in this report. Future versions may also include multi-media fragments (e.g., pictures or video clips) to support agent observations.

AI TESTBED DESIGN RECOMMENDATIONS

The following AI testbed design recommendations are provided to drive the efficiency, control, usability, and extensibility of processes for evaluating AI commonly used in training systems. These recommendations are largely based on the processes and services that have evolved to support the vision for LARIAT system architecture shown in Figure 2. The authors of this paper make no claim that our technical approach is the only approach. Our intent is to share the lessons learned from our design and development experience and provide recommendations to help others understand the necessary elements of a testbed architecture. The design goals of future testbeds will drive the processes and services that are needed. As in all development projects, budget is a major consideration and we took every opportunity to leverage existing capabilities (e.g., Deep Agent, RIDE, WEKA, and the GER pipeline) to support rapid prototyping of LARIAT. Figure 2 illustrates how each of these capabilities support needed processes and services.

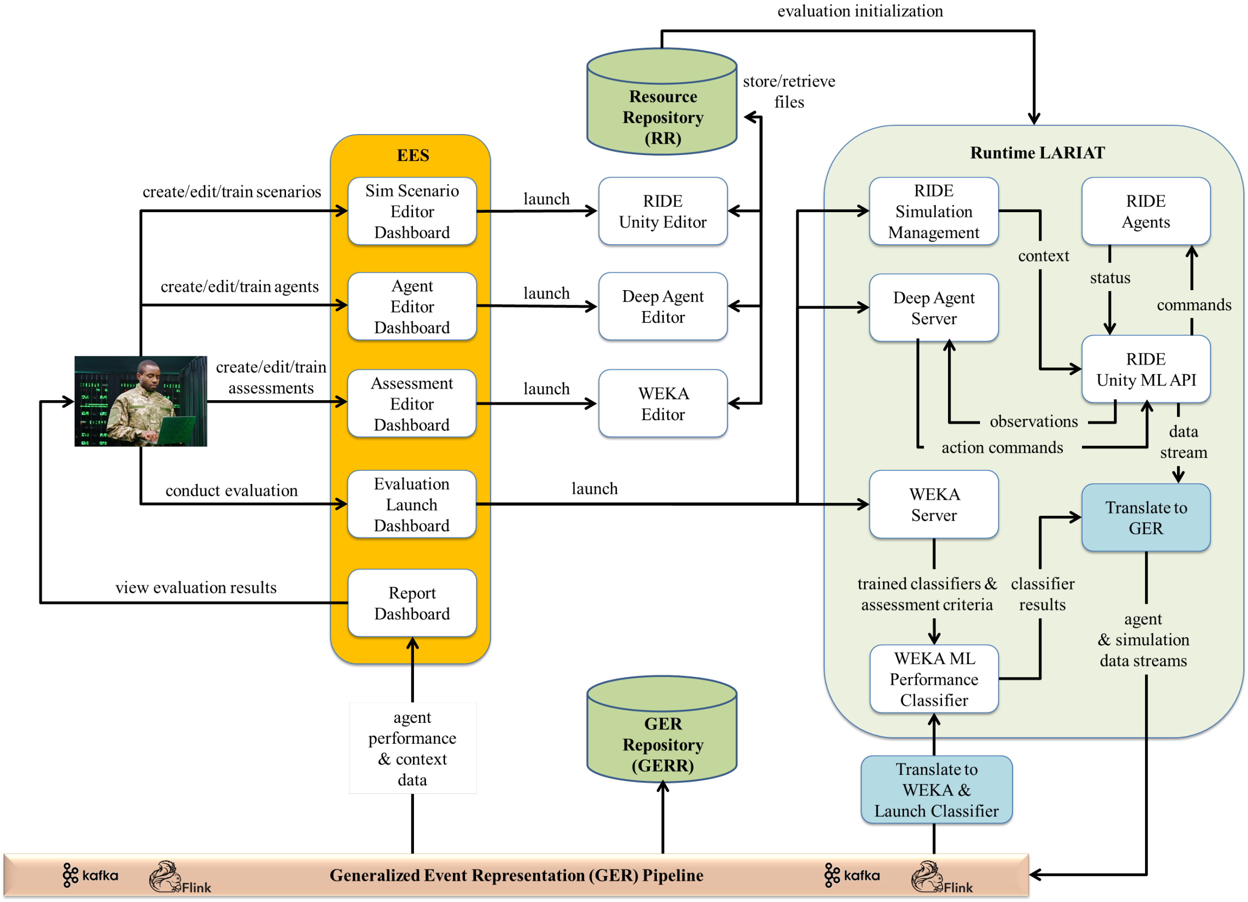

Figure 2. LARIAT Architecture

Now that we have describe the testbed concept of operations, defined its processes and services, we are ready to make recommendations that describe not just what the intelligent agent testbed is, but also considers its uses or missions. These recommendations center on design considerations for AI roles and goals, distinguishing between evaluations and experiments, optimizing automation in testbed processes, and understanding load management and opportunities to scale both vertically and horizontally.

Consider the Role and Goals of the AI in Formulating Evaluation Criteria

Unlike many of the AI testbeds used to train and apply intelligent agents in context (applied to the conditions under which the agent under evaluation will operate), it is critical to consider the impact on trainee performance. For example, an academic testbed may examine an intelligent agent is instantiated as an automobile. The agent is placed in an urban environment and through trial and error learns to identify an empty parking space and park without running into anything. The result of training this agent is interesting, but since this is a closed, agent-only system, it has no impact on a trainee. In contrast, an agent instantiated as an opposing force that is seeking to optimize its decisions and actions is different. It's learning objective is to provide a credible military interaction with human trainees and therefore, the agent's performance has direct impact on the trainee's learning and even its transfer of training to future operational environments.

When considering how to assess the impact of AI on desired training outcomes (e.g., knowledge and skill development), we should consider the role of that specific AI in the training process and its performance goals as shown in Table 2. If the AI performance goals are not directly aligned with trainee performance, then the evaluator should consider measures of assessment that aligned with the role of the AI and is goals. For example, an intelligent agent may serve the role of an opposing force in a training scenario where its goal is to survive and inflict casualties. To serve credibly in that role, the agent's competency may be adapted to the trainees' competency (ala Vygotsky's (1993) zone of proximal development), but its performance measures for the evaluation are survivability and lethality.

Table 2. Goals for the Evaluation of AI-based Training Capabilities

Goals	Method(s)	Measure(s)	Source(s)
Measure AI impact on trainee knowledge and skill	• Formative evaluation • Summative evaluation	• Learning (Kirkpatrick Level 2; See Bates, 2004) • Learning effect size (derived measure) • Key performance indicators (KPIs – Kirkpatrick Level 3)	• Formal and informal assessments of learning and performance (application of learning)
Measure AI impact on trainee competence	• Longitudinal summative evaluation of learning across the target population	• Learning effect size • Trainee knowledge, skills and abilities	• Formal and informal assessment • Context-based problem solving and decision-making experiences
Measure ability to manage trainee cognitive states	• Impact evaluation focused on the influence of interventions on cognitive states	• Attention • Engagement • Workload	• Impact assessment of AI ability to enhance attention and engagement, and optimize workload
Measure ability to manage trainee emotions	• Impact evaluation focused on the influence of interventions on positive and negative emotions	• Joy • Confusion • Anger/Frustration • Fear/Anxiety • Boredom	• Informal and impact assessments of AI ability to manage trainee emotions
Measure validity and reliability	• Summative evaluations focused on completion of training objectives	• Percentage of objectives met	• Formal assessment of the ability of AI to support objectives
Measure adaptivity	• Impact evaluation focused on the influence of AI decision-making	• Variability of interventions • Learning effect across the target population	• Impact assessment of the ability of AI to tailor training

Consider Distinct Processes for Evaluations and Experiments

It is strongly recommended that the design of your AI testbed architecture provides distinct processes and services for evaluations and experiments. As noted earlier in this paper, testbeds are research tools to explore, confirm and generalize (Hanks, Pollack & Cohen, 1993). The goal of evaluations usually a mix of exploration, confirmation, and generalization, and experiments are typically more exploratory. Evaluations seek to confirm the performance of a particular AI in context, but may also explore methods to optimize AI performance. The preparation and execution of evaluations and experiments may involve different requirements to control testbed processes, and therefore require different services.

Consider Automation

As noted earlier, developing intelligent agent evaluations or experiments is tedious, monotonous and requires significant skills to earn consistent results (Wray, Woods, Haley & Folsom-Kovarik, 2016; Wray & Stowers, 2017). One sure-fire method to eliminate onerous processes is to leverage automation. We highly recommend breaking down processes into small steps where automation might be possible. For example, adopting standards for defining and importing finite state machine states, observations, actions, and transitions can go a long way to reducing the time and effort required to feed your intelligent agent testbed. Figure 3 illustrates a very simple 5-state machine with only six possible observations. This finite state machine took over 30 minutes to construct in a very well understood mission space.

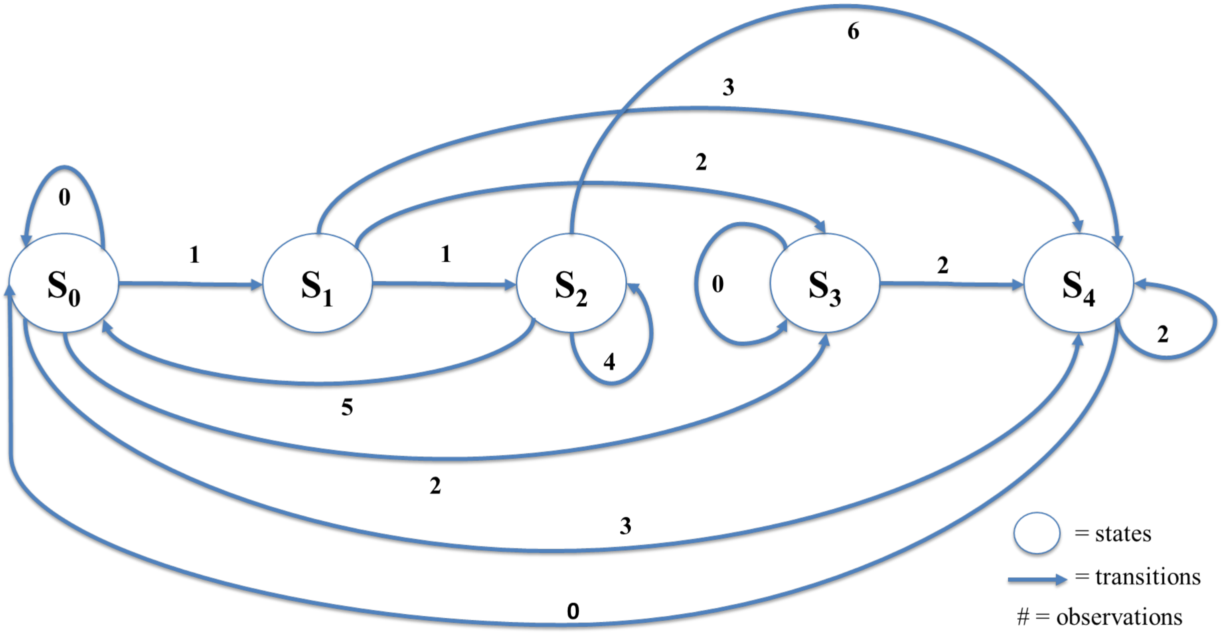

Figure 3. Five state finite state machine transition diagram

Consideration should also be given to automation of the testbed reporting processes. Using your testbed will lead to an understanding of common outputs and formats to support additional analysis or reconfiguration of agents to optimize their performance.

Consider Load Management & Scalability

It may be desirable to run multiple scenarios within an evaluation project where the scenario conditions differ. This process may be automated or allow for human observation. To scale appropriately, the testbed design should consider the human observation in the evaluation creation process. Automated evaluations may be more efficient given the variability of operating conditions. Consider the ability to run evaluation scenarios more efficiently in faster than real-time by eliminating the visualization of the environment.

Finally, your AI testbed design should consider whether processes and services can be executed offline or online. In a complex testbed framework, it may provide more flexibility to execute services during the evaluation, but the processing capability of the host may not be sufficient to support real-time or faster than real-time evaluation goals.

ACKNOWLEDGEMENTS

The design, research, and development described in this paper is sponsored by, or in part by, the U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC SC) under contract # W912CG21C0002. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the US Army CCDC. Software is available upon request to the Government, and will be demonstrated at ITSEC.

REFERENCES

Anderson, Michael; Anderson, Susan Leigh (2007-12-15). "Machine Ethics: Creating an Ethical Intelligent Agent". AI Magazine. 28 (4): 15–15. doi:10.1609/aimag.v28i4.2065. ISSN 2371-9621.

Bartneck, C. (2002). Integrating the OCC model of emotions in embodied characters.

Campbell, G. E., & Bolton, A. (2005). HBR Validation: Integrating Lessons Learned from Multiple Academic Disciplines, Applied Communities and the AMBR Project. In K. Gluck & R. Pew (Eds.), Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation. Matawan, NJ: Lawrence-Erlbaum Associates.

Ezhilarasi, R., & Minu, R. I. (2012). Automatic emotion recognition and classification. *Procedia Engineering*, *38*, 21-26.

Garibay, I., Oghaz, T. A., Yousefi, N., Mutlu, E. C., Schiappa, M., Scheinert, S., ... & Zhang, X. (2020). Deep agent: Studying the dynamics of information spread and evolution in social networks. arXiv preprint arXiv:2003.11611.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, 11(1), 10-18.

Hanks, S., Pollack, M. E., & Cohen, P. R. (1993). Benchmarks, test beds, controlled experimentation, and the design of agent architectures. *AI magazine*, *14*(4), 17-17.

Hemalatha, G., & Sumathi, C. P. (2014). A study of techniques for facial detection and expression classification. *International Journal of Computer Science and Engineering Survey*, *5*(2), 27.

Kirkpatrick, D., & Kirkpatrick, J. (2006). *Evaluating training programs: The four levels*. Berrett-Koehler Publishers.

Laird, J. E. (2019). *The Soar cognitive architecture*. MIT press.

Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892.

Ovalle, D. A. (2019, June). Intelligent Agents System for Adaptive Assessment. In Methodologies and Intelligent Systems for Technology Enhanced Learning, 9th International Conference (Vol. 1007, p. 164). Springer.

Rosenbloom, P. S. (2013). The Sigma cognitive architecture and system. *AISB Quarterly*, *136*, 4-13.

Somers, S., Oltramari, A., & Lebiere, C. (2020). Cognitive Twin: A Cognitive Approach to Personalized Assistants. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (1)*.

Toubman, A. (2019, July). Validating air combat behavior models for adaptive training of teams. In *International Conference on Human-Computer Interaction* (pp. 557-571). Springer, Cham.

Ustun, V., Kumar, R., Reilly, A., Sajjadi, S., & Miller, A. (2021). Adaptive Synthetic Characters for Military Training. arXiv preprint arXiv:2101.02185.

Van Dyke Parunak, H., Bisson, R., Brueckner, S., Matthews, R., & Sauter, J. (2006, May). A model of emotions for situated agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems* (pp. 993-995).

van Lent, M. (2019). Artificial Intelligence defined. Tech talk on Artificial Intelligence at the 2019 Interservice/Industry Training Systems and Education Conference, Orlando, Florida. Soar Technology, Inc., Ann Arbor, Michigan.

Vygotsky, L. S. (1993). The collected works of L. S. Vygotsky: Vol. 2. The fundamentals of defectology (abnormal psychology and learning disabilities) (J. E. Knox & C. IB. Stevens, Trans.). New York: Plenum.

Wallace, S., & Laird, J. E. (2003). Behavior Bounding: Toward Effective Comparisons of Agents & Humans. Paper presented at the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico.

Winograd, T. (1972). Understanding natural language. Cognitive psychology, 3(1), 1-191.

Wray, R. E., & Stowers, K. (2017). Interactions between Learner Assessment and Content Requirements: A Verification Approach. Paper presented at the Proceedings of the 8th International Conference on Applied Human Factors and Ergonomics (AHFE 2017) and the Affiliated Conferences, AHFE 2017, Los Angeles.

Wray, R. E., Woods, A., Haley, J., & Folsom-Kovarik, J. T. (2016). Evaluating Instructor Configurability for Adaptive Training In Proceedings of the 7th International Conference on Applied Human Factors and Ergonomics (AHFE 2016) and the Affiliated Conferences. Orlando: Springer.