Automated Intelligent Monitoring Systems

J. Wallace1 and S. Kambouris1
1Infinite Dimensions, Alexandria, Virginia, United States of America

Abstract - Some of the primary challenges in the development of cost-effective, next-generation Game Changing Technology for automated learning systems are the effective, efficient incorporation of representative knowledge models: domain or subject matter experts (SME), instructors/trainers, and trainees. Through interaction of the SME, trainer/trainee models, Automated Intelligent Monitoring System (AIMS) assesses what the student knows, and how well the student is progressing. Previous approaches to solving this problem have yet to overcome the inherent challenges long associated with complex, knowledge-based systems. The AIMS architecture framework overcomes limited and outdated psychological/computer science theories and insufficient computer science artifact using semantic ontology, and development engine architecture framework.

Keywords: Learning Systems, Semantic Ontology, Architecture Framework

1 Introduction

Significant challenges in the development of cost-effective, next-generation Game Changing Technology for automated learning systems are the effective, efficient incorporation of representative knowledge models: domain or subject matter experts (SME), instructors/trainers, and trainees. Through interaction of the SME, trainer/trainee models, Automated Intelligent Monitoring System (AIMS) assesses what the student knows, and how well the student is progressing. The AIMS architecture framework overcomes limited and outdated psychological/computer science theories and insufficient computer science artifacts. It does this through the AIMS architecture framework’s functionally realistic mission and environment approach and involvement from the training SMEs.

1.1 Automated Intelligent Monitoring System (AIMS)

The AIMS architecture framework is two things: an automated, integrated, development engine systems integration environment for high performance networked systems with library support for training and monitoring. The development engine is discussed in detail elsewhere [1]. Secondly, the AIMS’ corresponding high performance real-time executive/government off-the-shelf (GOTS) message bus operates in all environments, even resource-constrained ones such as Smartphone applications or in the Battlespace itself.

AIMS is relevant over wide ranges of education and training situations and is a Game Changing Technology. AIMS includes the capability for both single learners and multi-disciplinary group training as it has the ability to instrument and control a number of systems simultaneously. AIMS allow for the distribution of system processing for efficiency and scales in two dimensions: number of learners and number of systems which are instrumented and controlled. AIMS provides a solution for the problem of implementing intelligence and automation by effectively creating the knowledge models for the Expert, Instructional, and Trainee subsystems.

2 Instrumenting Learning Environment

Instrumentation has traditionally been a fundamental problem in traditional automated learning system architecture frameworks as it requires measuring the trainee’s inputs into the system as well as the reverse function of providing feedback to both the trainee and trainer. Correctly instrumented learning environments increase the overall effectiveness of the AIMS as more trainee assessment information becomes available. An example of instrumenting a flight simulator would involve measuring and recording pilot’s stick and switch movements, button presses, and the way in which the pilot is monitoring the gauges.

Greater system complexity arises when addressing multi-modal group instruction as individuals perform and train on different duties simultaneously, as well as participating at different times and separate tasks. Therefore, proper AIMS instrumentation is critical. Utilization of both group and individual assessment data provides sophisticated, complex information that has been rarely accounted for in previous trainee models.

Group instruction compounds training complexity, but is taken into account by AIMS. In group instruction mode, AIMS chooses content for the trainee group and makes multiple decisions: topics to address, problems to solve, and what feedback to provide and in which order, as well as what trainee information will be recorded. When AIMS employs the trainee model, it represents all the individuals and their specific group’s aspects as well as determines their tasks and correct order sequence for the trainees. Multiple possibilities
exist in determining the most appropriate tasks for a training group.

Another fundamental technical requirement for AIMS is to efficiently utilize its SMEs during training as an integrated trainer-in-the-loop. Real-time SME training integration facilitates an evolution in the inherently regimented and highly controlled nature of computerized training allowing for enhanced flexibility, complexity, variation and increased scope of the training itself. In order for a SME to interact with their trainees in a real-time environment, they need a flexible networking technology utilizing an efficient messaging system. AIMS’s real-time interactions allow the training itself to evolve more rapidly based upon feedback from the trainee or trainer/SME. AIMS takes into account a trainee’s needs with regard to communication, personality conflicts, or learning style issues which it rigorously accounts for as more knowledge is provided a priori to AIMS.

2.1 Semantic Ontology Streamlines AIMS Implementation

This paper addresses the ability to reduce the time required to plan, implement, execute, and analyze training in a unified AIMS architecture framework. AIMS is based on the concept of ontologies and employs Semantic Web technologies. This framework is intended to be used by different constituencies and employs a method, vocabulary, and user interface approach that allows the needs of different SMEs to be addressed. At the same time AIMS is functionally capable of operating off a unified conceptual structure.

In general, the major technological innovations are the ability to affordably instrument training/operational systems creating a learning environment to stimulate trainees, record their responses/capabilities, and facilitate their training. AIMS utilizes the ability to create intelligent systems with hybrid algorithm capabilities and synchronizing a large number of them thereby permitting AIMS to be built in a more modular and understandable way as an entire intelligence system. AIMS facilitates the ability to distribute computation easily over a set of resources. Additionally, the authoring and programming of AIMS is done in a way that simplifies the system specification and programming, respectively facilitating the creation of an automated intelligent monitoring system.

2.2 Adaptive Learning System

AIMS takes into account the trainee’s situational awareness, cognitive capabilities, action accuracy, situational distractions, cognitive distractibility and any resulting reduction in accuracy. AIMS accounts for functionally disparate phenomena such as recognition of patterns, adapting new solutions or strategies and rule following in order to represent “intelligence” more effectively. Based upon input from SMEs, current best training practices can be incorporated. AIMS ascertains the most appropriate feedback for the trainee by determining learning styles and communication preferences, in addition to skill level, and provides the most accurate advice for the trainees and SMEs. In reality, as humans, we rarely follow or interpret rules strictly; and thus, should not expect AIMS to do likewise. AIMS will have a great deal of generality and flexibility while maintaining ease-of-use for integrators, developers and users.

AIMS is an adaptive learning system, taking into account the trainee’s situational awareness, cognitive capabilities, action accuracy, situational distractions, cognitive distractibility and any resulting reduction in accuracy. AIMS accounts for functionally disparate phenomena such as recognition of patterns, adapting new solutions or strategies and rule following in order to represent “intelligence” more effectively. AIMS ascertains the most appropriate feedback for the individual trainee by determining learning styles and communication preferences, in addition to skill level, and provides the most accurate advice for them so they may proceed to the next problem-solving step. The advice given to the trainee is appropriate for their capabilities.

2.3 Technological Advances

In order to implement AIMS, advances in several technology areas must be combined to create an understandable architecture, and functional decomposition, that can scale and adapt to training problems and are discussed elsewhere [1]. AIMS knowledge and intelligence representations focus on pattern recognition and the basic steps of decision-making, as opposed to a larger view to include the reasons for decisions, and the timelines in which decisions must be made. AIMS’s complex, distributed application architecture framework instruments a system of training systems quickly.

2.4 Knowledge Types

Employing knowledge types in AIMS exceeds typical inference engine in rule-based environments [2]. Accommodating increased levels of modeling complexity for automating individual and team training requires a richer set of modeling primitives. A key capability involves easily constructing natural inter-component relationships logically and temporally. In this way, a functionally realistic mission and environment approach overcomes outdated psychological and computer science theories, used to represent intelligence as existing methods utilizing computer science artifacts are insufficient. An empirical connection is needed between real psychological measurement and behavior models. Certain sets of factors are more congruent and adaptive with certain mission/environments. Human factors of individuals, teams, and organizations may be capitalized upon and/or compensated for after they are identified and integrated. Through integrated interaction of the knowledge models,
AIMS accurately provides a more reasonable judgment about what the trainee knows and how well they are progressing both to the trainee, as well as their trainer.

2.5 Ontology-Based Architecture Framework Development Engine for Systems Integration

Ontology is a formal representation of domain knowledge that defines the concepts within a domain and the relationships between those concepts. It can be used to reason about the properties of that domain and is often described as a formal, explicit specification of a shared conceptualization as it provides shared vocabularies [3]. In this case, we need vocabularies able to describe: military operations and associated relevant real world activities, live military assets and their virtual reality and constructive representations, state, initialization, and exchange data and overarching conceptual descriptions of the training event.

Figure 1. Composition Automation Approach

To provide composition automation, as shown in Figure 1, the AIMS Architecture Framework interfaces, models, applications (especially simulations), and data involved in AIMS systems and exercises must be described/captured in a machine readable manner. Ontologies are employed for this purpose. Ontology alignment then permits the building of connections or bridges between existing models and enhances the ability to add new models. Model management and exploitation is made more effective satisfying Warfighter needs by promoting increased realism during training. Providing these ontology based tools enables exercise SMEs and military users to employ composition and architecture bridging tools to rapidly go from testing, training, experimentation, and rehearsal requirements to a training exercise or event for single trainees or groups of them.

The understanding of models, specifically Model Semantics, is almost exclusively a human activity. Models are only partially and ambiguously documented; much has to be interpreted and filled in from experience or context closely working with SMEs. As this experience is difficult to acquire and share, using Model knowledge is restricted to experts. The goal is to provide a method to partially encode expert knowledge about: mathematical and logical assumptions about the limits of the description of physical and non-physical phenomena such as human behavior, interfaces and their semantics, identification of patterns and frameworks, and causality information and execution order, such as one event triggering another.

2.6 Metadata and Composability Services

What this means for the Warfighter: a simple but robust method for categorizing everything from a handgun to the newest air superiority jet aircraft. Published ontologies that make previously created artifacts easy to find and retrieve should help alleviate reimplementation due to the common expedient of “I can’t find it so I’ll just create a new one”.

Utilizing the ontology-based development engine, we address the many significant systems integration challenges and create complex, realistic, and scalable networks utilizing component inter-relationships. The engine allows for the distribution of autonomous controls and constantly monitors them and permits the implementation of complex webs of cause and effect and dynamically alters the component execution structure. When information transfer occurs successfully, automatically and without human intervention, the transfer is symptomatic of integration. Systems integrators attempt to reduce overhead costs by reproducing the information flow more accurately and consistently all within the timeframe of the user’s need.

Systems integration teams must adapt, evolve, and facilitate integration of training systems. Successful integration requires the establishment of an AIMS partnership approach in conjunction with agile product development for all stakeholders: users, SMEs, program managers and program developers. Utilizing a flexible learning process, agile systems integration provides rational, logical conclusions that then lead the path forward for training systems integration. Utilizing rapid modification and implementation of SME and user input creates a more effective AIMS architecture framework.

The purpose of a cohesive and robust development engine is twofold: 1) to build and query knowledge bases (set of descriptions, facts, instructions, commands); and 2) to control the execution of numerous hardware and software components of training systems. The engine integrates hardware and software systems by extending Conceptual Graphs (CG) technology improving the resulting integration.
2.7 Flexibility and Realism

AIMS development engine enables the solution of key representation problems, permitting an easy-to-use, adaptive system. The overhead costs of information flow between software and hardware continue to increase as do system complexity. There are many inherent software/hardware challenges in integrating complex systems. We utilize Conceptual Graphs (CG) to integrate complex hardware and software systems [4]. CGs resolve complex issues arising from the following issues:

- Hardware compatibility
- Communication protocols
- Understanding transfer information
- Well defined system interrelationships
- Mapping for adequate neutral-format standards
- Mutual acceptance of the communication purpose
- Operational meaning in terminology
- Advancement in the core development environment and infrastructure

AIMS leverages many important capabilities, integrates many kinds of phenomenology, and encompasses a variety of formats. The ability to integrate a wide variety of “intelligent” algorithms and software is paramount. Several mathematical and causal formalisms must be accommodated by the representation framework. The development engine’s non-invasive, non-destructive integration and interoperability illustrates how the vital problem of synchronization between functionally distinct computational subsystems is accomplished. Accurate, precise synchronized control algorithms and software execution solves one of the key problems in intelligent systems today. Arbitrarily supporting complex causal networks easily is central to the value-added to the AIMS community from the perspective of development engine utility enabling both flexibility and realism.

The inherent ability to provide distributed and federated interactions and object sharing permits a wide variety of data sources to be integrated and interoperated. Functions such as automated, time-based validity and refresh checks and updating functions are a trivial capability. The implementation of more interesting functions such as causal or logical condition combinations triggering refresh or updating is naturally supported. The AIMS architecture framework is based on the abstractions shown in the Figure 2.

The Evaluation function relies on AIMS’ ability to establish a wide variety of knowledge, information, and data feeds. This can be done in two ways by an active push or passive pull. Most systems support one or the other, but not both. Furthermore, the mechanism for doing so must be low-footprint and portable. This permits a wide variety of software systems and devices to be employed.

The Records Database stores Performance information, which is derived from both the Evaluation process and the System Coach. The Records Database holds information about the past, such as Learner history records, the present, such as current Assessments for suspending and resuming sessions and the future, such as Learner or training objectives. The Records Database represents an advance in intelligent repositories in that a wide variety of complex knowledge, information, and data storage and retrieval functions can be easily implemented.

![AIMS Architecture Diagram](image)

Figure 2. AIMS Architecture

The System Coach requests and receives Performance information, such as assessments and certifications back to the Records Database. Typically, historical information is retrieved, but current information such as "bookmarks" for resuming sessions and future information such as template of future training objectives may be retrieved. The System Coach exploits the full capability of AIMS by a complex causal network representation capability.

The Knowledge Library capability permits a variety of knowledge representation formats to be employed that describe actions and responses of the system. Much like the Records Database, the ability to integrate and interoperate a wide variety of databases is crucial, as is the complex database functionality previously described.

Content Indexes are the result of searches of the Knowledge Library, as directed by Query Indexes. The Content Indexes are also known as metadata. Metadata is best known in web content for facilitating searches. However, web content metadata is inadequate for learning content because learning content requires more search criteria (e.g., pre-requisites, co-requisites, learning style) than what is provided for web content (e.g., title, subject, author, keywords). Content Indexes are similar to "call numbers" in a card catalog system. Web URLs are examples of Locator Indexes. For some learning technology systems, Locator Indexes can be implicit because the Learning
Content can be retrieved along with the Content Index returned by the Query Index search. Learning Content is the coded representation of the learning experience, identified by the Locator Index, retrieved by the Knowledge Library, and transformed by the Delivery system into the training systems/environment and learning experience, where the Learner interacts with the real operational systems or training systems/environment.

The Delivery system, much like the Evaluation system, relies upon AIDE’s powerful publication, subscription, and object sharing mechanisms. The low-footprint and portability features are important here, since that enables a wider variety of systems involved in Learner’s real tasks to be employed. This improves training since the systems used at the actual tasks are used in training.

The Learner (process) represents any learner (one, several, or teams operating in different roles). The Learner receives Stimuli, and their Behavior is observed. From a technology perspective, a variety of assessments about the Learner (individual or group) are available to the System Coach and Evaluation processes to more effectively and efficiently guide each training session.

The Learner Assessments are sent to the System Coach. AIMS considers the various types of information which provides more knowledge a priori to the system itself. Figure 3 represents a multi-modeling approach to more easily bring in additional very complex and interrelated Learner Assessment models.

![Learner Assessment data approach utilized in Figure 2.](image)

AIMS has the ability to integrate naturally and efficiently for the new knowledge representation techniques and engines. Problems in communication, personality conflicts, or learning style issues can be rigorously accounted for in the training program. Learner Assessment information utilized may be obtained and include any number of the following:

- Concentration and awareness
- Accuracy of attention: environment, cognitive analysis, problem solving, set of tasks, mission
- Distraction: external, internal, mistakes and ability to recover
- Information and tasking: quantity and range of diversity
- Risk-taking, sensation seeking, thinking outside the box
- Following rules, orders, tradition, deliberation, caution
- Aspects of control, self-confidence, self-criticism
- Dominating and competitiveness: physical and intellectual
- Speed of decision making in given environments
- Listening, verbalizing, group/person selection preference
- Advocating team performance
- Communicating ideas, support, criticism
- Performing in certain types of high pressure environment

The following discussion is intended to illustrate how the AIMS architecture is implemented with the intelligent systems AIDE framework previously discussed.

The Learner construct represents one or several trainees. The Stimuli incident to the Learner requires the ability to interoperate with the learner’s training or operational system. This requires the ability to quickly and easily integrate everything from buttons, switches, joysticks, variety of control knobs or interfaces to elaborate virtual reality or operational displays.

The Behavior flow shown in Figure 2 is a transmission of Learner behavior from the Learner to the Evaluation process. At the Evaluation process, the Behavior is framed in the appropriate context by matching the Learning Content to the range of Behavioral responses which is how information is collected and organized, e.g., key clicks, mouse clicks, voice response, choices, or inputs to an operational system which are sent to the System Coach. The Learner’s observed Behavior is an observation of the Learner’s activity while they are performing a task such as flying an airship for example. Some Behaviors may be physical location behaviors and have several levels of inputs concerning the Learner: location, six degrees of freedom position, velocity with respect to frame of reference, orientation (pitch/roll/yaw), acceleration, and the angular velocity with respect to those quantities. The Learner Behavior physical inputs are sent to the System Coach: 10 degrees of pitch, 20 degrees of roll at that moment in time, etc.

The System Coach sends the Learner Behavior to the Evaluator. Behavior information is obtained with a similar interoperation of user interface mechanisms or training specific observations such as a system of cameras to determine the scan pattern of an operator through their tasks and interfaces and feeding this streams into an Evaluation process which then would be implemented as some number of System Objects [1]. Ultimately all the data streams implemented using System Events and System Messages and the programming
constructs used to implement them are simplified for the clarity and ease of programming.

Learner Assessments sent to System Coach are different than the Behavior Evaluation path. Assessments are things that directly coming out of the training or operation system. These may be the Learner’s problems with understanding when the timing of when to press a training system button. The Assessment part of the architecture represents information that comes from a direct observation of the learning environment and is not explicitly a Learner Behavior. For example, a Learner Assessment might be a warning light is on in a flight simulator and the fact that the light is on as a result of a trainee’s actions and is a normal, internal part of system’s operation.

AIMS observes the trainee’s performance. This is a Learner Assessment and it is directly sent to the System Coach. Learner Assessment is information about what or how the trainee is doing, but it is not an Evaluation. System Coach gives feedback Stimuli to improve which is sent through the Delivery process. One piece of feedback may be “please release the safety”.

The Learner Assessments transmitted from the Learner construct to the System Coach are implemented as System Objects and serve to provide specific inferences about the Learner such as an running Evaluation of the Learner’s skill level at a particular task or other diagnostic that is built into the training or operational system that has some useful data which is not explicitly behavior based observation.

The System Coach is similar to the Evaluation implementation and it receives one or more Learner Assessments. The Learner Assessments are sent as System Events and System Messages. Multiple independent evaluations of behavior and multiple simulation learner assessments occur and feedback can be given and noticeable trainee improvements made over time. The delivery construct is also implemented with multiple system objects and execution engines and serves to mediate between the learning environment and the System Coach.

The Knowledge Library may then be viewed as not just rules or rule specifications, but elaborate combinations of libraries of function and arbitrary combinations of knowledge representation constructions. In both the System Coach and Evaluation, one or more knowledge execution engines can be instantiated to operate on the data being received.

The ability to have more than one knowledge execution engine operating simultaneously would simplify the construction of authoring the knowledge bases from a semantic perspective. This permits more realistic and practical evaluation information, resulting in more realistic and accurate automated inputs to the Learner. AIMS has knowledge of the Learner’s activity, training systems, relationships between them, the ability to index training systems, and AIMS architecture framework system that could realize roles and responsibilities for both AIMS and the trainee.

The AIMS that captures Learner Assessment behavior must accommodate a variety of inputs, and the delivery of the Behavior then triggers the Evaluation process. The Evaluation Assessment process produces information such as where the Learner is at any given time which is then sent to System Coach. The Evaluation process uses the Learning Content to provide context to the Learner’s Behavior to determine the appropriate Evaluation. The Evaluation process also creates and sends Performance information to the Records Database. This active triggering of Evaluation is an aspect of the AIMS complex, distributed application framework. The Evaluation process must then utilize advanced Knowledge and Intelligence Representation in the form of the adaptive SME and student model algorithms, such as genetic or fuzzy logic algorithms, that can then be used, as well as rules and pattern recognition. The AIMS distributed application framework also allows the ability to integrate many varied knowledge and intelligence techniques.

The AIMS complex, distributed application framework is also used to build the System Coach, as it must interact with other applications, and employ knowledge and display “intelligence” in its actions by also utilizing the advanced Knowledge and Intelligence Representation capabilities. The System Coach may store “bookmarks” as Performance information in the Records Database saving the Learner’s session for resumption at some future time. The System Coach also receives the current Assessment from the Evaluation process, and Performance information from the Records Database, to support the decision-making process.

Based on the current Assessment and historical Performance information, the System Coach sends Query Indexes to the Knowledge Library to search for appropriate learning materials. The Knowledge Library returns Content Indexes such as a list of Locator Indexes that match the search query. The appropriate Locator Indexes such as a lesson plan are sent to the Delivery process. The System Coach sends Query Indexes to the Knowledge Library to search for content that is appropriate for the Learner. The Query Indexes specify search criteria based on, in part, Learning Style, Assessment, and Performance information.

The Knowledge Library stores knowledge and other learning materials as resources for the learning experience. The Knowledge Library may be searched by Query Indexes. The matching information is returned as Content Indexes, i.e., a set of content tags that are, conceptually, “card catalog” entries known as metadata. The locator Indexes (conceptually, “call numbers” on the bindings of the “books in the digital library”, e.g., URLs) are extracted from the content indexes. The locator Indexes are used by the Delivery process to retrieve Learning Content. It is unspecified who initiates the
transfer of Learning Content and it can be the Learner, the System Coach, or the Delivery process itself. For example, a Query Index might return a set of Content Indexes and an ontology, which is a conceptual model of the subject represented as generic Learning Content.

The Delivery process receives locator Indexes from the System Coach and retrieves Learning Content from the Knowledge Library. The Delivery process transforms the Learning Content into Stimuli for the Learner. The Stimuli is actually sent to either simulations connected to the operational systems or training systems/environment and in some cases sent directly to the operational systems or training systems/environment. When the Delivery process sends Stimuli to the Learner, the Evaluation process is expecting some Behavioral response to the Stimuli. The Evaluation process is unable to interpret the Behavior without context, so the Delivery process sends the Learning Content to the Evaluation process to understand the context of the Learner's response.

3 Conclusions

In general, the key technological innovations are the ability to affordably instrument training/operational system for learning environments to stimulate learners and record their responses. The second key innovation is the ability to create intelligent systems with hybrid algorithm capabilities. For instance, the evolution of the value of a neural network could be the result of a query of a first order predicate calculus inference engine. The converse can also be implemented where a value in a rule or fact inference based engine’s knowledge base could be calculated by evaluating a neural network.

Additionally, the ability to create hybrid intelligent algorithms and synchronize a large number of them permits the AIMS to be built in a more modular and understandable way as an entire intelligence system. In order to enable the next generation of instructional systems, several technology challenges had to be overcome. The establishment and elaboration of coherent, comprehensive system architecture, relevant over wide ranges of education and training situations, is a major step forward providing:

- Single learner or group training on customized courses
- Ability to instrument and control a number of systems simultaneously
- Allowance of the distribution of system processing for efficiency
- Scales in both dimensions – the number of learners, and the number of systems instrumented and controlled
- Internal and external processes
- Internal and external events
- Efficient simple two-way interactions
- Effectively creating the knowledge models for the Expert, Instructional, and Student Trainer subsystems
- Integrating the most appropriate decision algorithm required to naturally implement functionality – not just rule bases
- Utilizing a novel system development framework that permits the most appropriate type of interaction paradigm required for any component of the system

AIMS takes into account the trainee’s situational awareness, cognitive capabilities, action accuracy, situational distractions, cognitive distractibility and any resulting reduction in accuracy. AIMS accounts for functionally disparate phenomena such as recognition of patterns, adapting new solutions or strategies and rule following in order to represent “intelligence” more effectively. AIMS ascertains the most appropriate feedback for the trainee by determining learning styles and communication preferences in addition to skill and provides the most accurate advice for them.

AIMS’ next generation training interactive engine and architecture framework will revolutionize training and greatly enhance the intrinsic value training provides to our communities. Scalable, robust solutions are provided for complex, real-world training problems and scenarios. In general, unsolved problems become tractable, and integration and interoperability costs no longer dominate project cost profiles and automated, intelligent training is achieved. Integrating SMEs into the development process facilitates better training systems as they are there during the design, engineering, testing, and evaluation process ensuring that their training needs are met at every step of the AIMS development cycle. SMEs understand what the training issues for both the system to be trained and their trainees. SME knowledge, input and feedback are incorporated so the AIMS is customized to meet their specific training needs.

4 References


