

The Assisted Experimental Designer: A Decision Support System to Optimize Modeling and Simulation Experiment Design

Sylvain Bruni, Kenyon Riddle, Andres Ortiz, Danielle Dumond

Aptima, Inc.

Woburn, MA

[sbruni] [kriddle] [aortiz] [ddumond] @aptima.com

Jay Saffold

Research Network, Inc.

Kennesaw, GA

jsaffold@resrchnet.com

ABSTRACT

Many professionals are involved in the creation of Live, Virtual, Constructive, and Game (LVC&G) simulation environments and scenarios, from simulator programmers to subject matter experts. Currently, there is no systematic method for designing simulation events from these multiple perspectives. To meet this need, Aptima developed LVC&G-AED (Assisted Experimental Designer), a decision-support system that guides individuals through a ten-step simulation-design process, from defining the research question or training goal, to choosing variables of interest and developing relevant measures. A knowledge database, populated by data from past simulations and their results, serves as the basis for an underlying model that recommends simulation configurations that address all relevant goals, while maximizing quality and minimizing cost. This prototype was designed to be usable by any individual, regardless of background, enabling them to create effective simulation designs quickly and efficiently. By utilizing all existing data and available tools, LVC&G-AED saves manpower, time, and financial resources.

ABOUT THE AUTHORS

Sylvain Bruni is a Human Systems Engineer and leads the Training and Feedback Design capability lead at Aptima, Inc., where he provides expertise in human-centered design processes: requirements definition, design development and implementation, and system evaluation. His research focuses on human-automation collaborative systems, multimodal user interfaces, human-in-the-loop experimentation, and the statistical analysis of human-centered system data. Prior to joining Aptima, Mr. Bruni conducted research at the Massachusetts Institute of Technology (MIT), focusing on designing and testing collaborative planning systems, specifically in military environments. Mr. Bruni holds a S.M. in Aeronautics and Astronautics from MIT and a Diplôme d'Ingénieur from the Ecole Supérieure d'Electricité (Supélec, France). He is a member of the Human Factors and Ergonomics Society, the IEEE Systems, Man, and Cybernetics Society, the Association for Computing Machinery, and the DoD Human Factors Engineering Technical Advisory Group.

Kenyon Riddle is a Human Factors Scientist and Manager of Orlando Operations at Aptima, Inc. Mr. Riddle provides expertise in the design, evaluation, and implementation of human-centered automation and decision-support tools, as well as in human-system evaluation, experiment design, and statistical analysis. Prior to joining Aptima, Mr. Riddle was a Research Associate at Honeywell Aerospace in Golden Valley, MN, where he worked on a wide range of projects supporting the design and analysis of human-centered systems in the aviation domain. Mr. Riddle's graduate work at the University of Illinois research compared the effectiveness and human performance implications of perceptually-based and command-based automation. Mr. Riddle holds a M.S. in Human Factors from the University of Illinois at Urbana-Champaign and a B.S. in Psychology from the University of Central Florida. He is a member of the Human Factors and Ergonomics Society, the American Psychological Association, and the Aircraft Owners and Pilots Association.

Andres Ortiz is an Aerospace Engineer at Aptima Inc., with a broad background in modeling and analysis of multi-agent autonomous systems. Particular areas of expertise include continuous and discrete event system modeling, real-time task scheduling and assignment, optimization, vehicle trajectory planning, control of dynamical systems and machine learning. Dr. Ortiz's research involves the development of augmentation and support systems that facilitate and enhance interaction and coordination between autonomous/robotic systems and their human operators.

His Ph.D. research studied the development of a mission planner to reduce operator workload in reconnaissance missions where multiple Unmanned Aerial Vehicles (UAVs) are supervised by a single human operator. To this end, he developed an optimal task scheduling algorithm that enable the operator to individually process information from multiple sources in real time. This time domain solution was implemented via velocity and trajectory modifications to the UAVs' flight plan uploaded in real time. Previous work also focused on aircraft safety and the development of diagnostic health management systems deployable onboard small UAV platforms. Dr. Ortiz received his Ph.D. and M.S. degrees in Aerospace Engineering from the University of Illinois at Urbana-Champaign, and a B.Eng in Electronic Engineering from the Pontificia Universidad Javeriana in Cali, Colombia.

Danielle Dumond is an Analytics, Modeling & Simulation Research Engineer at Aptima, Inc. Her background lies in machine learning, decision making, probabilistic modeling and robotics. Dr. Dumond's interests include using mathematical models to predict and control how different agents within a system will behave and interact. She worked at the Thayer School of Engineering at Dartmouth to develop a system in which an autonomous robot could identify a terrain from proprioceptive sensors onboard the robot. This system could be used to respond to changes in terrain in real time to help avoid immobility. Dr. Dumond received a Ph.D. in Engineering Science with a specialization in Control Engineering from Thayer School of Engineering at Dartmouth College, and a B.S. in Physics from the College of William and Mary. She is a member of the Society of Women Engineers (SWE).

James Saffold is the President, CEO, and Chief Scientist for the Research Network Inc. (RNI). He has over thirty years of experience as an engineer in both the military and industry. He holds a BSEE degree from Auburn University (1983). Mr. Saffold has performed research in RF Tags, UWB radar, Virtual Reality, Digital databases, Soldier Tracking Systems, Millimeter wavelength (MMW) radar, multimode (MMW and optical) sensor fusion, fire-control radar, electronic warfare, survivability, signal processing, and strategic defense architecture (to name a subset). Mr. Saffold's current interests include embedded AAR systems, game-based training, combat identification, virtual scenario simulation, sensor fusion, signal processing, electromagnetic propagation, and phenomenology. Mr. Saffold lectures annually at the Georgia Institute of Technology on topics related to remote sensing, propagation, clutter, smart munitions, and signal processing.

The Assisted Experimental Designer: A Decision Support System to Optimize Modeling and Simulation Experiment Design

Sylvain Bruni, Kenyon Riddle, Andres Ortiz, Danielle Dumond

Aptima, Inc.

Woburn, MA

[sbruni] [kriddle] [aortiz] [ddumond] @aptima.com

Jay Saffold

Research Network, Inc.

Kennesaw, GA

jsaffold@resrchnet.com

THE PROBLEM

The U.S. Army's Simulation and Training Technology Center (STTC) is currently developing the Executable Architecture Systems Engineering (EASE), a unifying platform that connects all stakeholders involved in the conception and implementation of simulation environments -- such as modeling and simulation users (e.g., experimenters or trainers), systems engineers, developers, and other subject matter experts -- and all equipment and apparatus, including models, simulators, scenarios, hardware, software and data repository (Marshall, 2011). The EASE platform is aimed at ensuring interoperability and connectivity between the users and their tools, in a manner that simplifies access and implementation of experimental or training configuration (Figure 1).

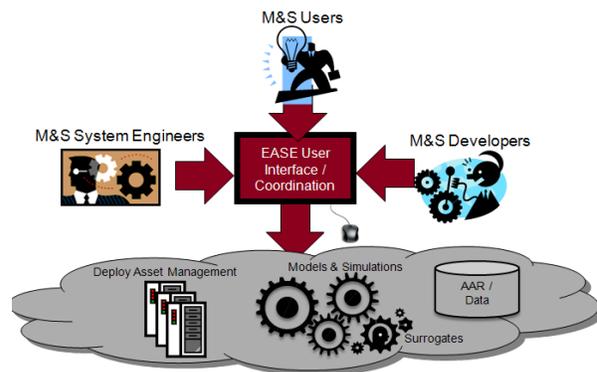


Figure 1. STTC's EASE concept of operations.

From the introduction of new technology to dynamic enemy tactics, the ability to test and train quickly and cost effectively new technologies, operating procedures, and organizational structures is becoming increasingly important. Live, Virtual, Constructive, and Game (LVC&G) simulations allow for efficient and repeatable experimentation. However, many individuals are involved with the creation of an LVC&G simulator and scenarios. An Army analyst understands the phenomena to be studied. An operations research or systems analyst understands research methods, experimentation, and measurement. A simulation engineer knows the intricacies of the LVC&G scenarios and general capabilities specific to the LVC&G platform. In order to create an appropriate scenario, all three individuals must coordinate. The difficulty of coordinating across people and professions can slow development of tests, produce tests that are less decisive and efficient than desired, and add additional financial and manpower costs (McDonnell et al., 2011).

APPROACH

The development of a highly automated solution can streamline the selection and setup of simulation environments, by guiding any one of the three above individuals through the experimentation process, while still leveraging much of the knowledge of any of the given individuals by using a machine intelligent solution. The experiment designs developed with the proposed technology would be well-structured and complete, so design teams can coordinate more productively over these products. To enable this vision, our team developed the LVC&G Assisted Experimental Designer (AED), a machine intelligent and scientifically informed guided research aid to support a variety of networked LVC&G simulations.

As illustration in Figure 2, the foundation of the AED decision-support system is a computer-based automated interview process that enables users to input information relevant to their experimental objectives. This interview is guided by a ten-step research questionnaire, structured as a systematic way of designing an experiment, from the identification of research goals and statistical design to the definition of independent and dependent variables. The input provided by the experimenter is then analyzed by an intelligent algorithm, in the form of a Partially-

Observable Markov Decision-Process (POMDP) model, which explores the domain space of feasible experimental configurations that would satisfy the user's objectives. The POMDP model dynamically queries a knowledge database (KDB) containing information relevant to existing modeling and simulation (M&S) or live assets, platforms or scenarios, as well as records of previously-conducted experiments. With this data, the model infers feasible experimental configuration that optimize the design for cost (i.e., minimizing the cost of running the experiment) and for quality (i.e., maximizing the quality of the expected experimental results). A rank-ordered list of experimental configurations is returned to the experimenter through the AED interface, allowing the user to select the configuration they want to run. Ultimately, AED will allow to push automatically an experimental design file, directly to the M&S or live platforms connected with EASE.

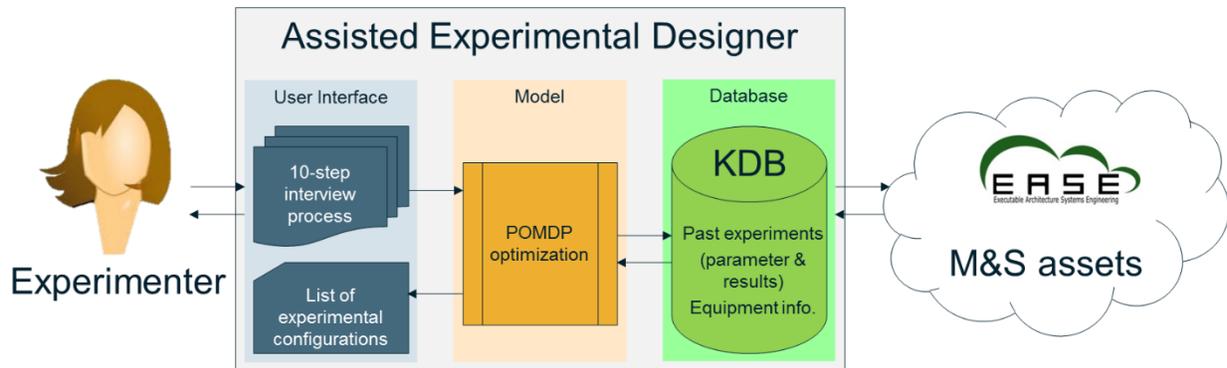


Figure 2. System architecture for the Assisted Experimental Designer.

COMPONENTS

The User Interface

The AED user interface (UI) is structured around a ten-step experimental design process that queries the user for key information related to the experiment they wish to conduct (Figure 3). The first seven steps, listed in a quick-access menu on the left side of the UI, and available as a form in the main area of the UI, include:

1. **Objectives:** the list of experimental objectives, written in plain English – e.g., *to study the impact of using unmanned robots in room clearing tasks*;
2. **Research questions:** the list of specific research questions the experimenter wish to address with the experiment – e.g., *does situation awareness increase when using a blue force tracker handheld display?*
3. **Constraints:** the list of hard and soft constraints that apply to the experiment – e.g., *budget of \$50,000 max*;
4. **Environment:** a specification of the environment in which the experiment is expected to be conducted – e.g., *classroom training or Multi-UAS ISR operations*;
5. **Independent variables:** the list of experimental variables whose impact on system behavior is tested through the experiment, these variables are controlled by the experimenter – e.g., *number of soldiers on a team or level of autonomy of robots*;
6. **Dependent variables:** the list of experimental variables that quantify the system behavior, these variables are measured through the experimenter – e.g., *situation awareness, workload, system performance*;
7. **Statistical design:** a specification of the anticipated statistical design of the experimenter, to be employed to analyze dependent variables against independent variables – e.g., *fully-crossed, repeated design*;

Once the first seven steps have been filled by the experimenter, the model generates experimental configuration based on user inputs and queries to the knowledge database. The experimental configurations generated by the model are pushed back to the user, in Step 8:

8. **Configuration:** a rank-ordered list of experimental configurations generated by the AED model, for the experimenter to select from and modify as they want – e.g., *“OneSAF with Scenario 4 and the Vision Toolkit”*;

9. **Protocol:** an automatically-generated experimental protocol document that lists information pertaining to conducting the experiment with the selected configuration;
10. **IRB:** an automatically-generated Internal Review Board (IRB) document with information pertaining to the submission of an application for IRB approval when research involves human participants.

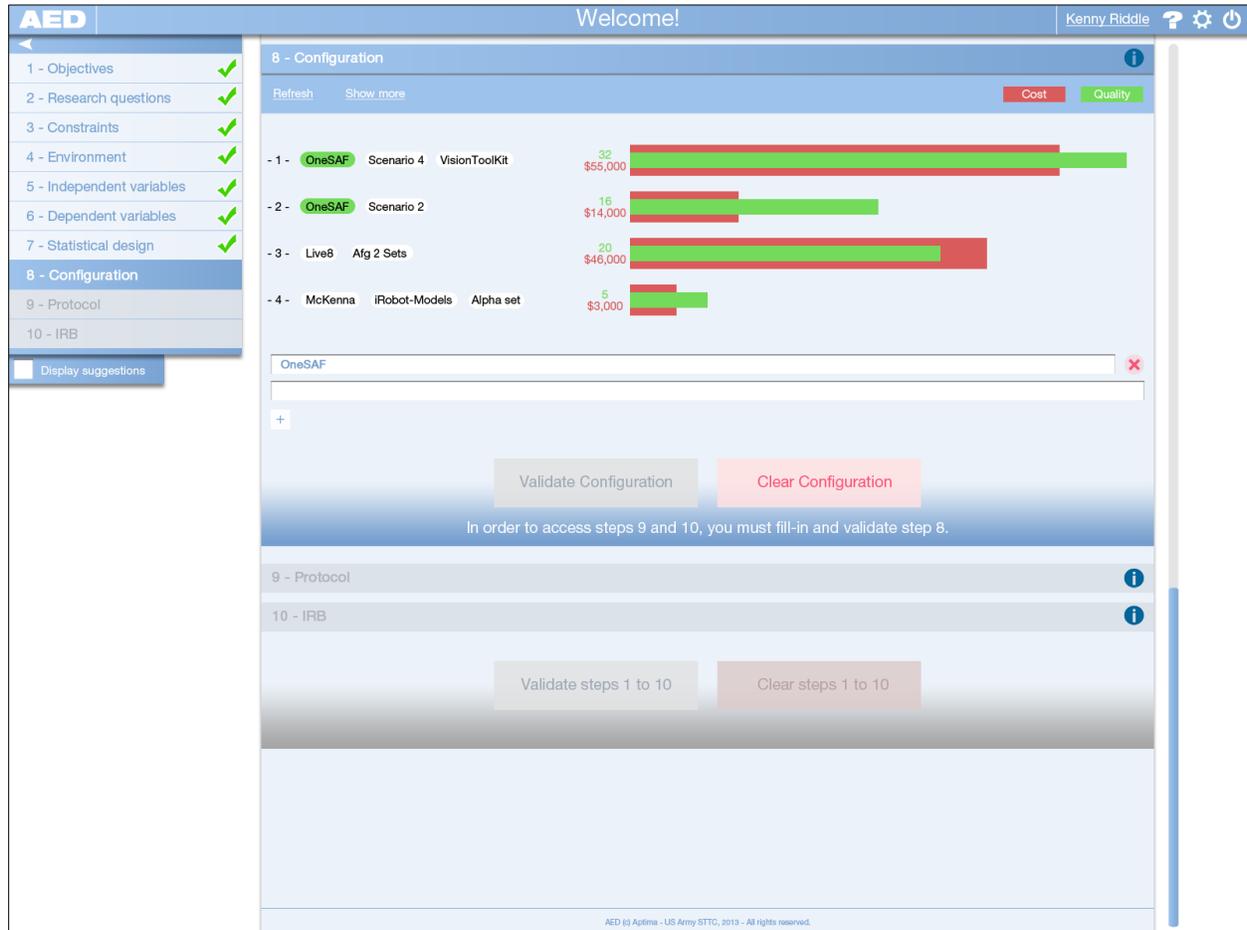


Figure 3. Screenshot of the prototype user interface of AED.

The Model

The AED's decision support system is built by modeling the experimental process as a Partially Observable Markov Decision Process (POMDP). POMDP models have been widely used to represent and optimize sequential decision-making problems under uncertainty and is thus well suited for this particular application (Smallwood & Sondik, 1973, Putterman, 1994). Formally, a POMDP model is defined by a tuple $\langle S, A, P, \Omega, O, R \rangle$. The set S of all possible states is referred to as the state space. The set A defines a set of actions while P defines the state transition model. P is a probability table composed of elements $P(s'|s, a)$, which define the probability that the model will transition from state s at time t , into state s' at time $t + 1$ given that action a was taken. The set Ω represents a set of possible observations and the observation function O provides the probability that a particular observation will occur given a state transition (i.e., $O(o|s, a, s')$). Lastly, the reward function R is used to drive the optimization problem, that is, the goal of the POMDP solver is to create a policy to maximize this reward.

The POMDP models the experimental process by tracking experiment state. For this particular application, the state is an abstract representation of the progress towards meeting the experimental goals. In particular, the state is a representation of the quality of the experimental results that the experimenter has obtained throughout the

experimental process. The state however, is not directly observable (i.e., measurable) but can be estimated through a set of observation which are related to it. As such, the model combines a set of measurable parameters (i.e., observations) such as the quality of the instrument (e.g., reliability and fidelity of a particular apparatus to measure a particular dependent variable) and the quality of the results (e.g., mean stabilization, variance and bound checks for each dependent variable across a number of trials) to infer the experimental state. The set A , is defined by the courses of action that the experimenter can take after each one of the experimental runs has been completed. These actions are represented in the model as a set of experimental configurations that the experimenter can run. Ultimately, the goal of the POMDP model is to recommend to the experimenter the next experiment configuration that should be run to improve upon its current results. In POMDP terms, this translates into recommending an action or set of actions, given the most current observations such that the reward function is maximized. This is done by mapping observations to actions using a policy, developed to maximize the reward.

The AED model works in two stages (Figure 4). In the setup stage, a POMDP solver is used to generate a POMDP policy to be used during the execution stage. This policy is generated using a predefined set of parameters that characterize the observation, transition and reward models and a set of historical data from relevant experiments. The relevant historical data is extracted from the KDB by filtering past experimental configurations to select those which are similar to the initial configuration that was selected by the experimenter.

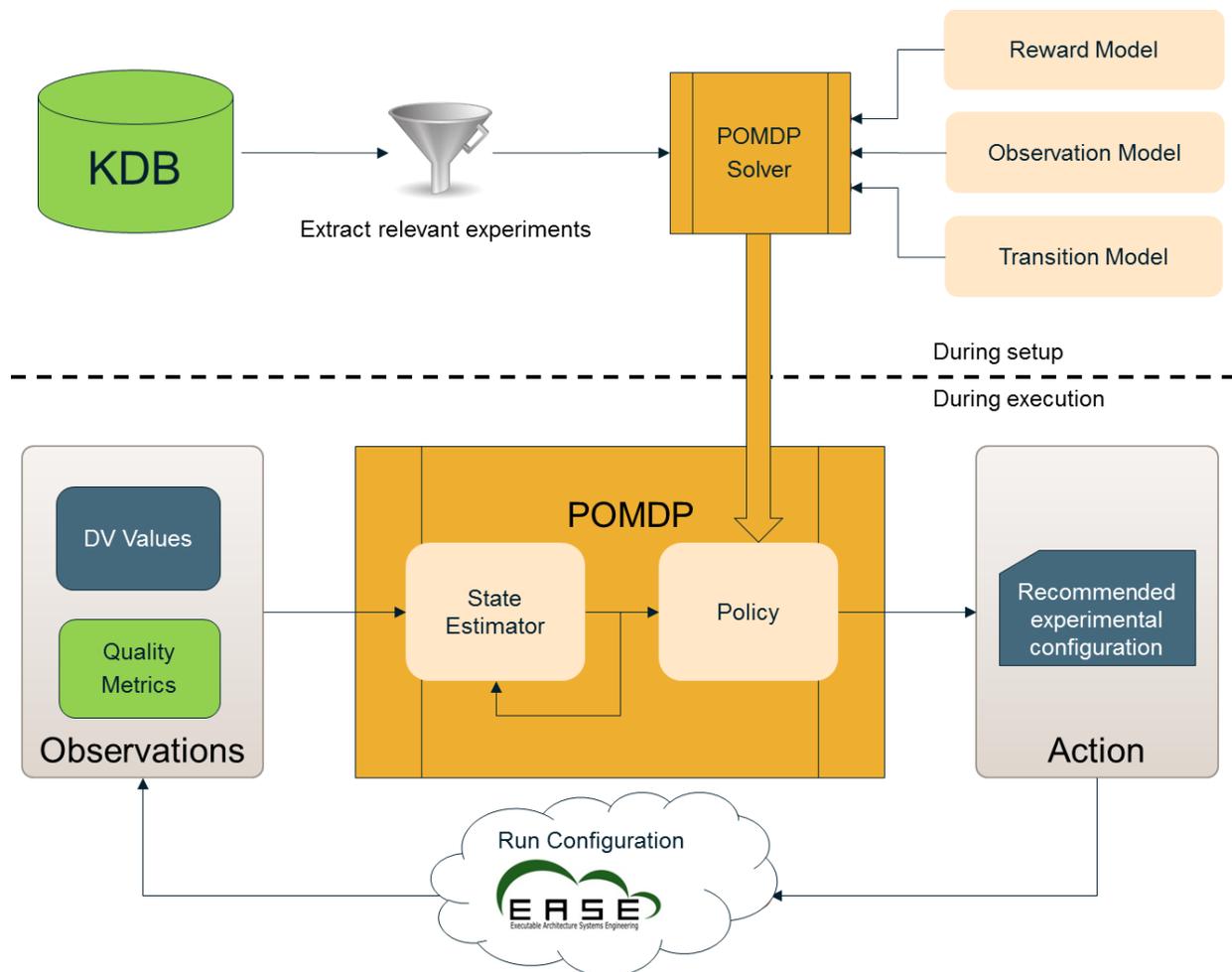


Figure 4. Detailed view of the POMDP model in AED.

Once the POMDP policy is generated, the model enters the execution stage. In this stage, new recommendations are suggested to the experimenter every time the experimenter can provide a complete set of results from an experimental run. These results include the measured values for all of the dependent variables of interest for a

specific set of trials. Given these values and a set of predetermined quality metrics, observations are generated for the POMDP model to update its state. The new estimated state (i.e., the belief state) is then used by the policy to generate a new recommendation. This iterative process may continue until the experimenter is satisfied with the outcome of the results.

The Knowledge Database

The purpose of the knowledge database (KDB) is to act as an information repository available to the model for query when building and evaluating experimental configurations. The KDB can fulfil four queries from the model:

1. **Request for experimental information:** the KDB houses known information about previously conducted experiments. This information is captured in the KDB in a systematic structure using an XML schema (TestExperiment.xml).
2. **Request for apparatus information:** the KDB houses known information about apparatus available to the user for experimentation. This information is captured in the KDB in a systematic structure using an XML schema (TestApparatus.xml).
3. **Request for cost estimate:** the KDB can compute the estimated experimental cost of an experimental configuration designed by the model.
4. **Request for quality estimate:** the KDB can compute the estimated apparatus and experimental quality of an experimental configuration designed by the model.

Request for experimental information

Past known experiments are recorded in the KDB using the schema illustrated in Figure 5. For each experiment, the KDB lists the following critical information: experiment title, point of contact, apparatus employed, location of experimentation, experimental conditions (e.g., weather for Live experiments), recorded research questions, recorded research objectives, independent variables, dependent variables, experimental results (i.e., dependent variable values under combinations of independent variables), measures of quality of the experiment run, post-test analyses conducted, anonymized experimental participant information, experimental procedures followed, validity information, constraints information (such as classification levels, facilities used etc.), lessons learned, cost details, etc.

Request for apparatus information

Similarly, information about M&S and live assets available to the experimenter, for example through the EASE network, is coded using an XML schema (Figure 6). For each platform, scenario, apparatus or item, the KDB features information about: point of contact, access information, missions modeled or available in the asset, available documentation, hosting facility, post-test capabilities, known costs, instrumented variables, variables fit (i.e., the quality of results obtained for each variable when employing this asset), interoperability with other LVC&G assets, etc.

```

TestExperiment.xml - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<Experiment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance">
  <Name>TestExperiment</Name>
  <Created>2014-02-05T15:05:44.8557824-05:00</Created>
  <GUID>e39795a-e3e9-4746-b414-76b5a373116</GUID>
  <Title Title="My First Official Experiment" Confidence="VeryHigh" />
  <Documentation>
    <Document Title="QuickStart Guide revision 12.pdf" Author="Maj Mike Smith" Page="12" />
    <Document Title="My Experiment Analysis and Results rev 12.pdf" Author="Jim By" />
  </Documentation>
  <Description>Comparison of My Experiment Metrics over a correlated database with
  <DateRange StartDate="Jan 2012" EndDate="April 2012" DaysInDesign="30" DaysInExe
  </DateRange>
  <Apparatus>
    <Apparatus Name="GDIS-Unity" GUID="6e9c2a2a-09c1-4e0c-9b4a-b5e9886e4d21" Type="
    <Apparatus Name="Fort Hood MOUT Site" GUID="5c3edd05-eea7-4db2-875c-59ea8fc84c
    </Apparatus>
  </Apparatus>
  <ExperimentLocations>
    <Location>Virtual Portion - Customer Facilities in Orlando</Location>
    <Location>Live Portion - Georgia</Location>
  </ExperimentLocations>
  <ExperimentConditions>
    <Condition weather="Clear" Indoors="true" Confidence="VeryHigh" />
    <Condition weather="Clear" Indoors="false" Confidence="VeryHigh" />
    <Condition weather="Cloudy" Indoors="false" Confidence="VeryHigh" />
    <Condition weather="LightRain" Indoors="false" Confidence="VeryHigh" />
  </ExperimentConditions>
  <ResearchQuestions>
    <ResearchQuestion>Research question 1</ResearchQuestion>
    <ResearchQuestion>Research question 2</ResearchQuestion>
  </ResearchQuestions>
  <Objectives>
    <Objective>To compare and contrast Live and Virtual performance metrics and ec
    <Objective>objective 2</Objective>
  </Objectives>
  <VariablesManipulated>
    <Apparatus Name="GDIS-Unity" GUID="b453e243-337c-44a5-b657-4b4ee7e3c68b">
      <InstrumentedMetrics Type="Burdened" DirectMeasure="true" Accuracy="1" Updat
      <InstrumentedMetrics Type="Unburdened" DirectMeasure="true" Accuracy="1" Upd
    </Apparatus>
    <Apparatus Name="Fort Hood MOUT Site" GUID="33962da2-a018-4ce4-8d02-2ff8c7912f
      <InstrumentedMetrics Type="Burdened" DirectMeasure="true" Accuracy="1" Updat
      <InstrumentedMetrics Type="Unburdened" DirectMeasure="true" Accuracy="1" Upd
    </Apparatus>
  </VariablesManipulated>
  <VariablesMeasured>
    <Apparatus Name="GDIS-Unity" GUID="b453e243-337c-44a5-b657-4b4ee7e3c68b">
      <InstrumentedMetrics Type="PoseType" DirectMeasure="true" Accuracy="1" Updat
      <InstrumentedMetrics Type="Velocity" DirectMeasure="true" Accuracy="1" Updat
      <InstrumentedMetrics Type="LocomotionMode" DirectMeasure="true" Accuracy="1" Updat
      <InstrumentedMetrics Type="Stance" DirectMeasure="true" Accuracy="1" Updat
    </Apparatus>
  </VariablesMeasured>

```

Figure 5. Example XML record for a past experiment.

```

TestApparatus.xml - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="utf-8"?>
<Apparatus xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance">
  <Name>TestApparatus</Name>
  <Created>2014-01-30T10:44:51.8675353-05:00</Created>
  <GUID>482b0141-609c-48b8-a21f-78d6db14f6de</GUID>
  <General>
    <Name>TestApparatus</Name>
  </General>
  <Access>
    <Restrictions>GovernmentOnly</Restrictions>
    <Classification>Unclassified</Classification>
    <TOD>NormalWorkHours</TOD>
  </Access>
  <Mission>
    <MissionType>SearchandRescue</MissionType>
    <TeamType>SmallTeams</TeamType>
    <Locations>Ground</Locations>
  </Mission>
  <Category>Virtual</Category>
  <Documentation>
    <Document Title="TestApparatus Users Manual ver 1.docx" Author="Mike Smith" />
    <Document Title="TestApparatus License ver 1.docx" Author="Tim Oriely (USMC)" />
  </Documentation>
  <RemoteOperation>
    <ControlType>Control</ControlType>
  </RemoteOperation>

```

Figure 6. Example XML record for a test apparatus.

Request for cost estimate

The KDB houses predictive algorithms that compute cost information for experimental configurations generated by the POMDP model, based on the cost data available in the experimental and apparatus information of the KDB. The computed cost of an experiment is based on estimates of costs associated with apparatus acquisition and use, and the embedded costs associated with each of the major phases of an experiment. These include planning, design, conduct, and data analysis. Cost data are compiled in each of these phases based on a range of cost categories which are partitioned into labor, the apparatus used, licensing, facilities, required documentation, potential travel, estimated experiment time (on site and off site), and historical data from past experiments of similar nature. Once the phase costs are estimated, the total estimated cost is produced by summing these individual phase costs using a quality point average (QPA) technique based on confidence factors provided at the time of data collection. When data is collected through a survey of M&S stakeholders, a standard example use case is provided which allows responders to “cost” the use of their respective apparatus and also provide a baseline by which the model can normalize future cost estimates for experimental configurations that would include the apparatus.

Request for quality estimate

Similarly, the KDB is capable of computing a quality metric that estimates the “fit” of an experimental configuration against experimental objectives: in other words, the KDB can return to the model a prediction of how well a specific agency of M&S or live assets will accommodate the objectives specified by the experimenter during the interview process. This is done by producing a combined estimate of: (1) the ability for an LVC&G apparatus to handle a specific set of independent or dependent variables, (2) the confidence and dependability of the outputs of the apparatus, and (3) the intrinsic quality of its components (e.g., the level of granularity of a vehicle model).

In addition to “fit” quality estimates, the KDB also provides the ability to estimate experiment progress and the quality associated with interim and final results. Experimental progress is estimated based primarily on statistical confidence intervals and metrics associated with reaching desired thresholds of confidence in the results’ mean and variance. Result quality is estimated based on interim (or final) results using a set of metrics that are derived from Analysis of Variance (ANOVA) principles, good design of experiment steps, and variable independence measures. The quality in each of these metrics is estimated, normalized into a consistent quality scale, and then combined using a quality point average (QPA) technique to produce the final result.

Quality Point Average

The quality point average method for quality computation is based on a “Quality Points” system similar to that used by academia to assess grade point average for students. Essentially, each quality component is given a “rating” and a “credit” or weight. The total numbers of credits for all the quality components are summed. The products of all the Quality Rating (QR) and Credit (CR) combinations (QR*CR) are also summed. This sum of products is also called “Quality Points” in the literature (Northeastern University, 2014). The overall quality rating is then computed as the ratio of the rating and credit combinations to the total credits (Equation 1). Confidence factors from the structured data collection from apparatus stakeholders are also factored into the QPA algorithm through the credits system.

$$QPA_{Total} = \frac{\sum(QR * CR)}{\sum CR} \quad (1)$$

PROTOTYPE DEVELOPMENT

The three components of AED, the user interface, the model, and the database, were developed following an agile software engineering process. The team followed an iterative schedule of three-week sprints: at the beginning of every sprint, a set of UI, model, and KDB requirements was targeted for delivery, and worked on until completion. The design of the user interface followed the principles of ecological interface design (Vicente, 2002) and decision-oriented design (Metersky, 1993). Quality assurance (QA) testing was conducted in the form of unit testing, to ensure full compliance of the delivered software with the design requirements and expected use of AED by experimenters.

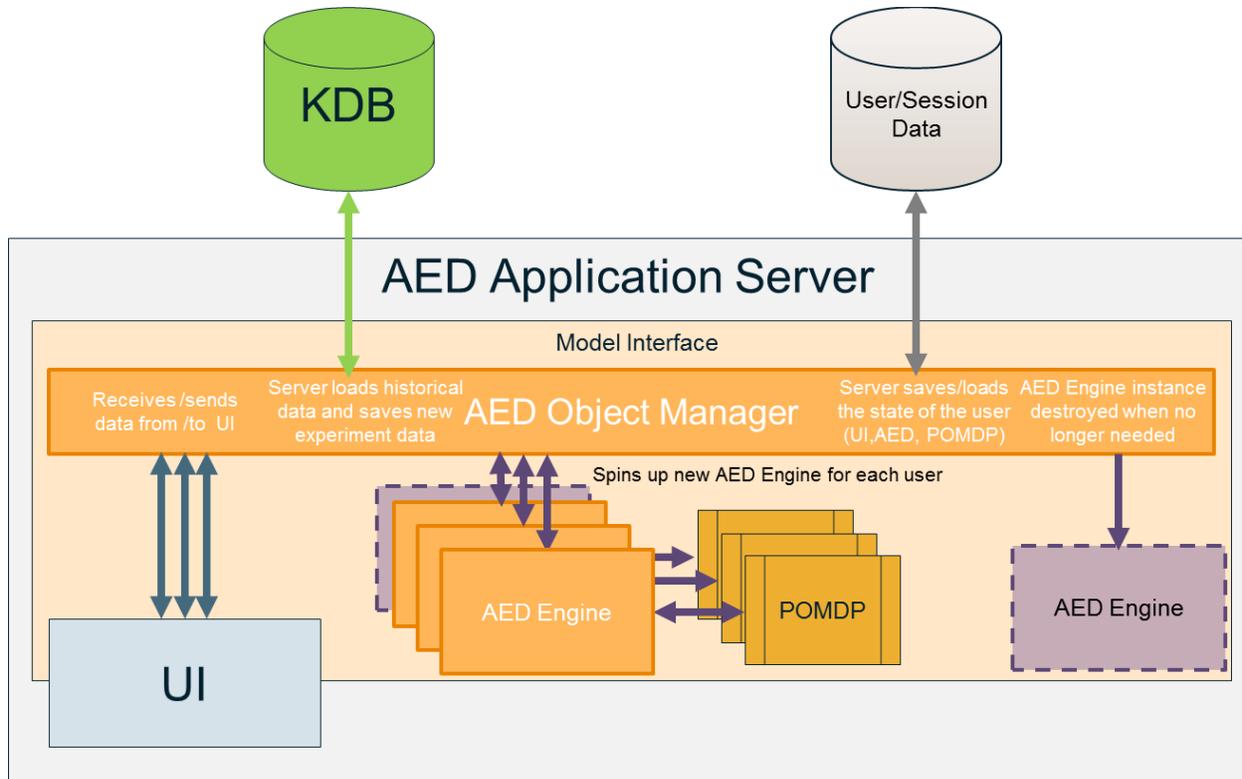


Figure 7. Implementation architecture for the AED model

Building And Testing The Model

The AED model was built within a custom developed C# application. The model consist of a POMDP library wrapped inside the decision support module referred to as the AED Engine (Figure 7). We envision the AED to be a service which can be used by several users simultaneously while running distinct sets of experiments with a variety of custom settings. In order to enable such deployment an application server was developed which facilitates the interaction between the multiple modules within AED.

Within this architecture, the model operates as follows: An instance of the AED Engine module is created for each user that is carrying out an experimental process. Each AED Engine instance has its associated POMDP module and is managed by the AED Object Manager (AOM). The AOM loads pre-existing experiments from KDB and makes them available in memory to all instances of the AED Engine. Similarly, after every experiment is run, the information from that experiment is stored in the KDB through the AOM. Lastly, the AOM facilitates data passing between the AED Engine and the UI in order to provide the model with the necessary configuration parameters from the user and the experimental process and return the generated recommendations.

The functionality of the various modules has been initially tested to ensure that all of the pieces deliver the specified functionality and communicate appropriately. Features like loading and saving KDB data, creating experiment configurations from user defined parameters, generating POMDP policies from a set of experimental configurations and generating POMDP recommendations from observations have all been successfully tested by populating the KDB with scripted apparatus and experiment data. In terms of the validity of the recommendation engine itself (i.e., the POMDP model) it is a challenging task to benchmark its performance given the nature of the decision making problem. That is, the potentially large decision space which can lead to a series of recommendation sequences that may be equally sensible. Despite the limited availability of data and the challenge of generating testing scenarios, a few scripted test cases have been developed in order to help determine the validity of the recommendations. It is anticipated that the fidelity of the testing and evaluation process will increase as the KDB gets populated with real

historical data. In addition, expertise from experimental designers can be leveraged to determine qualitatively the efficacy and sensibility of the recommendations in a systematic manner.

Populating The Knowledge Database

Methods

Populating the KDB primarily comes from two approaches: (1) structured interviews with stakeholders of both apparatus and past experiments, and (2) directly from current experiments using the AED tool. The structured interviews are based on a provided survey questionnaire which follows the database schema but formats the questions and answers into human-friendly forms by providing example answers in both text and tabular forms. An example of the structured interview related to the characteristics of an apparatus looks like:

Example query: Can the apparatus partition areas (cordon) for experiments or training exercises automatically and if so how is this done and are the partitions modifiable at runtime?

Example response: Yes, by setting up route systems in scenarios and/or implementation of entity triggers which can be changed at runtime by the real-time editor (Quality: medium, Confidence: high)

The information is then recorded as indicated in Table 1.

Table 1. Example record of apparatus characteristics in the KDB.

Cordoning	Real-Time Modifiable	Method	Quality	Confidence
<i>Yes</i>	<i>Yes</i>	<i>Real-Time Editor</i>	<i>Medium</i>	<i>High</i>

In the structured interview, the responder's confidence in the answer is requested in order for the model to make an estimate of the associated quality of the characteristic, and the utility of the response in choosing this apparatus for an experimental configuration.

Confidence Factors

In each survey question a confidence factor is required based on the responder's confidence in the level of accuracy of their answer for the apparatus. The confidence factor is mapped to a "confidence percentile" (Table 2) and is used to propagate uncertainty bounds through the optimization system. For example, if the responder is about 90% sure the answer is accurate over most conditions, the factor is high. If the responder is unsure the answer is accurate, the confidence is low. Respondents may be prompted to qualify their confidence answer based the amount of historical use of the apparatus in a variety of conditions.

Table 2. Confidence Factor Map.

Category	Percentile
verylow	0 – 20%
low	21 – 40%
medium	41 – 60%
high	61 – 80%
veryhigh	81 – 100%

The goal of the KDB is to establish quality information on available apparatus and on past (and current) experimental data to provide a valuable resource to the model in the AED tool and to all users of the system.

CONCLUSION AND FUTURE WORK

The Assisted Experimental Designer developed under this effort constitutes a promising first step into increasing accessibility to numerous, distributed modeling and simulation platforms, as a means to conduct more reliable and efficient M&S or live experiments. At its core, AED leverages powerful algorithms, such as POMDPs, to devise an experimental configuration that satisfies as best as possible those objectives specified by experimenters. Through an intuitive user interface, experimenters can navigate the space of possible configurations generated by AED, to select the experimental setup they wish to implement.

This technology has the potential to augment considerably the capabilities of an experimenter using LVC&G assets. Through networks of modeling and simulation assets, such as the Army's EASE, an experimenter will be allowed to build experiments that make use of those previously inaccessible, remote assets, with AED helping to figure out what piece of equipment is the best suited to fulfil the desired experimental objectives. Once the full vision of EASE is implemented, experimenters in one location will be able to drive optimized experiments (defined by AED) in a distributed fashion, using M&S or live apparatus in various remote locations.

Future work to develop AED will include partnering with the United States Military Academy at West Point, NY, to embed a version of AED in their EASE framework. Working with cadets will allow for an early evaluation of the integrated AED prototype, in a realistic environment.

ACKNOWLEDGEMENTS

This research was conducted under SBIR funding from the Army's Simulation and Training Technology Center (STTC). The authors thank Henry Marshall and Christopher Gaughan of STTC for their support and guidance in this research. The views and conclusions presented in this paper are those of the authors and do not represent an official opinion, expressed or implied, of STTC, the U.S. Army, the Department of Defense or the United States Government.

REFERENCES

- Northeastern University (2014). *How to Calculate Your GPA/QPA and Earned Hours*. Retrieved March 14th, 2014 from <http://www.northeastern.edu/registrar/gradecalc.html>.
- Marshall, H. (2011). Executable Architecture Systems Engineering (EASE), A New Vision for M&S. Proceedings of the INFORMS 2011 Annual Conference, Charlotte, NC.
- McDonnell, J.S, Gallant, S., Gaughan, C., & McGlynn L. (2012). Executable Architecture Systems Engineering (EASE). Proceedings of the Systems Engineering Conference (SEDC 2012), Washington, DC.
- Metersky, M.L (1993). A decision-oriented approach to system design and development. *IEEE Transactions on Systems, Man and Cybernetics*, 23:4.
- Putterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.
- Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:5, 1071-1088.
- Vicente, K. J. (2002). Ecological Interface Design: Progress and challenges. *Human Factors*, 44, 62-78.