Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

REDUCING TRAINING COSTS THROUGH INTEGRATION OF SIMULATIONS, C4I SYSTEMS, AND EXPERT SYSTEMS

Mark Whelan
John Loftus
David Perme
Richard Baldwin

Gestalt, LLC
1040 First Ave, Suite 302
King of Prussia, PA 19406, U.S.A.

ABSTRACT

The Department of Defense doctrine of "train as you would fight" exemplifies the military's focus on training as a key component in the preparation of today's warfighter. The positive result of implementing this doctrine has been clearly demonstrated. The costs of implementing training programs, however, have been high. Simulations have historically supported this training by augmenting and supplementing real world scenarios and data. The incorporation of these tools is often problematic as the simulations may not have been designed for integration with operational Command, Control, Communication, Computer, and Intelligence (C4I) systems that are used in training events. These integrations are too often manual. The integration of simulations and C4I systems through the use of software tools and intelligent agent technologies results in significant cost reductions and allows for increased flexibility and effectiveness of training.

1 INTRODUCTION

The Department of Defense Services (Air Force, Army, Navy, Marine Corps) has clearly stated a desire to "train as you would fight." Today's warfighter is challenged to develop skills and expertise in a variety of areas. In order to develop these skills, the Services have instituted broad and robust training events, exercises, and experiments. Typically, these training activities have involved large expenditures, driven in no small part by the desire to make the training experience closely approximate the real world scenarios and situations facing the warfighter. The training expenditures include the development of explicit training capabilities in the form of training versions of operational systems, tools, models, and simulations. Additionally, the Services have established training centers dedicated to the preparation of military and non-military personnel. In addition to these "development" expenses, the operations costs of staging an exercise and the maintenance costs associated with keeping these capabilities "fighter ready" contribute significantly to the total budget needed to "train as you would fight."

It should be no surprise that the Services are increasingly interested in methods and approaches that can reduce these expenses in light of ever tightening budget constraints. The value of simulations in these training exercises is high. Simulators have long been utilized to enable training scenarios that would not otherwise be developed and enacted given the high cost and logistics of real world training.

Often, however, the use of simulations can also drive cost because of the effort required to integrate the simulation output to operational systems. If the simulations have not been adequately designed to integrate with operational systems, the integration of simulations requires the use of manual procedures to effect an interface. These manual integrations themselves often necessitate additional training for the personnel selected to support the exercise so that they can perform the manual tasks in an acceptable manner because these individuals are generally unfamiliar with the operation. The results can vary depending upon the skill sets of those receiving the just-in-time training to support the exercise.

By converting these interfaces into software-driven integration capabilities and toolsets, the costs of training can be reduced while the effectiveness and realism of the training exercise can be increased. In particular, the integration of C4I systems with simulation tools and models provides drastically improved training quality and cost reduction. The insertion of technologies such as expert systems, when applied appropriately, provide much greater cost reduction by automating time consuming and highly repetitive tasks.

2 SIMULATOR ROLES IN TRAINING

Training is performed in a variety of modes. Live training involves high costs and logistics problems. It is best suited for individual training on equipment or small unit scenarios emphasizing unit level tactics and employment techniques, for example, a Wing level deployment and fast attack scenario. Live training can range up to events that encompass multi-national and multi-based units (such Red Flag held at Nellis AFB). These events are extremely costly, and can be less realistic as the scope of the exercise grows due to logistical constraints and safety issues. Thus many live events are scripted thus affecting the realsism of the resulting outcome. To address these issues (cost, realism, scripted outcomes), simulation training is employed.

The next mode of training is the virtual mode. In this mode participants are modeled in simulations as individuals, the equipment is modeled to individual pieces as opposed to subsystems, and interactions are represented between individual entities, for example an air-to-ground missile launched from an aircraft at an armored vehicle. The actual launch, the representation of the missile, and the resulting hit or miss on the armored vehicle is communicated and modeled. This level is most suited for small unit and small scenario (or parts of a larger scenario) training.

Lastly, training can be conducted using constructive simulation. In this mode units are modeled as a whole or single entity. The individual components of the units themselves are not modeled, but are represented within the modeled entity. Models at this level normally represent an Army Battalion or higher, a flight of aircraft and not individual aircraft, or a squadron or higher. These types of simulations are largely stochastic, Monte Carlo type simulations and are best suited for large scenarios.

3 INTEGRATION WITH C4I SYSTEMS

The use of constructive simulations in wargaming events has allowed the scope of these training exercises to be broadly expanded such that single service, joint, and coalition exercises can be successfully conducted. This success, however, has also borne the cost of piecing together the disparate elements of the exercise. Often the elements used to support the exercise have not been designed to work together, or interoperate, in a seamless fashion. C4I and simulation integration has proven to be especially challenging due to three factors :

1. Translation Needs
2. Transformation Needs
3. Fidelity Differences

Translation refers to the differences in the representation of data elements in C4I systems and simulations. In a simplistic form, this is essentially a mapping exercise of a data element in one system to a corresponding data element B in another system. In reality, translation can be much more complex, as data elements present in a C4I system may not be present in a simulation and vice versa. Simulations may also present aggregated versions of multiple C4I data elements. For example, a simulation may represent an air squadron as its finest level of representation, while a C4I system will track individual air assets such as bombers, fighters, helicopters, and tankers as its finest data element level. In some cases, the simulation model may in fact more discretely define certain data elements (this often has fidelity implications as well, see below).

Transformation refers to the differences in the data element formats. When translation, or mapping, can be performed, the format of the data elements themselves may not match. An example of this would be the use of a character string “F15C” in the C4I system and the representation of the equivalent data element in the simulation model as “F15-C”. While this is a simplistic example, more complex transformation issues are present and can be easily conceptualized.

Fidelity differences are perhaps the most problematic of the factors. It is often assumed that an operational system will represent a higher level of fidelity and that simulation tools by their nature are lower fidelity representations of real world elements. In the case of C4I systems, however, there are planning elements that are in fact at a lower degree of fidelity than are typically present in simulation tools. Examples of this are route planning and weapon target pairing. This apparent lack of fidelity in the C4I plan is in fact an acceptable approach that clearly indicates the commander’s intent while not burdening higher echelons with details best left to lower echelons who are prepared to, and desire to, add the appropriate level of detail.

Integration of these two systems (C4I and simulation) must address each of all three factors (translation, transformation, and fidelity). In the case where an established interface is not present, the training event must accomplish the integration by other means. Too often, the use of manual processes to overcome these factors has been employed. In some cases these manual processes are necessary to provide intuition and cognition. In many instances, however, these manual processes are simply repetitive, laborious tasks that are performed in a rote, untimely fashion.

By implementing software tools this integration can be delivered more effectively and with less cost. User interfaces can be established that aid the human interaction elements involved in translation and transformation. This computer-aided capability significantly reduces the manpower required. With the establishment of a common data representation (a common data model), translation can be automated via data insertion and extraction routines. These routines can also accommodate a certain level of transformation as well. Intelligently designed software tools provide the appropriate mix of software directed integration and human assisted processes.

4 THE CONTRIBUTION OF EXPERT SYSTEMS

While the use of software and computer assisted human interaction provides obvious and substantial benefits to the training environment, some elements of the integration can not be automated without the use of additional technology approaches.

Often, complex fidelity issues require the use of a human-in-the-loop to decide the appropriate steps to follow. Standard programming approaches simply do not suffice to insert the required logic to maintain or increase the realism of the training environment. In these cases, subject matter expertise is needed. This expertise replicates the real world processes of fidelity addition, and conversely the aggregation of detail into summary level representations.

Fortunately, technologies do exist that can employ subject matter expertise in software applications. Expert systems technology has long been viewed as the solution to such problems. Too often, expert systems have been misunderstood and misapplied. Expert systems can be designed such that their scope is well defined and a set of rules (subject matter expertise in a codified form) determined so that appropriate reasoning can occur.

In the case of C4I and simulation interoperability, expert systems can be employed to further automate the integration. The key success factor in the use of such systems is to apply them intelligently. In reality, this means expert systems should be defined such that the complexity of the rule base does not outweigh the value gained by their use. Seeking a complex rule base to address all issues results inevitably in failure. By segmenting scope and functionality, reasonable rule sets can be established. Using the example of route planning, rule bases can be defined to automate the ingress and egress routing with a minimum of complexity while maintaining a high degree of realism. In this case, the subject matter expertise required to perform route planning can be reliably determined, represented in a rule set, and implemented in an expert system. Nearly as important, rule sets can be saved for future use or modified to reflect changes in doctrine, practice, or procedure.

5 CASE STUDY – INTEGRATING AIR FORCE SIMULATIONS AND C4I SYSTEMS

During execution of an Air Operations Center (AOC) or Joint Forces Air Component Commander (JFACC) training exercise, the Theater Battle Management Core System (TBMCS), the primary Air Force C4I system, is used to generate Air Tasking Order (ATO) and Airspace Control Order (ACO) messages. The Air Force's Air Warfare Simulation (AWSIM) executes simulated air mission and provides feedback on mission success and status. An interface between TBMCS and AWSIM, the AWSIM TBMCS Interface (ATI), processes the ATO messages and produces AWSIM compatible input called "order stacks." Model controllers further refine the mission by using a tool, the Mission Planner Workstation (MPW), to edit the order stacks. Editing the order stacks involves defining appropriate equipment, ingress and egress routing, resolving data ambiguities, and the addition of standoff shoot locations. The model controllers must complete the editing within timelines established by the AOC staff. The order stacks are submitted to AWSIM which executes the simulated mission. Mission results are sent to the AOC staff, via the ATI interface, directly into TBMCS.

The realism and quality of the model controllers' mission editing is affected by several factors:

- **ATO message length.** The number of missions in the ATO can vary from 150 to over 2000 depending on the objectives of the JFACC and the scale of the exercise.
- **The time available to perform mission editing.** As little as two hours to as many as twelve hours may be available for editing. A typical exercise could involve a 30 member team editing 150 missions over a 12 hour period. This provides for two person-hours per mission. However, the scenario is also likely to involve 2000 missions over just a two hour period with the same 30 person editing team, which only provides for a two person-minute review per mission.
- **The number of model controllers available to support the exercise.** Most simulation centers have a limited number of dedicated model controllers on staff. Military personal or contractors may supplement this staff to support an exercise. Additionally, the availability of experienced personnel can be limited by schedule conflicts, travel budgets, and leave.
- **The model controller's knowledge of air operations**. Dedicated model controller staff normally train supplemental controllers over a two day period prior to the exercise. Personnel with prior experience in day-to-day air operations are more valuable as supplemental model controllers. Experience levels can range greatly from extremely knowledgeable to those with no prior experience or training.
- **The level of mental fatigue of the model controller.** Exercise schedules can run from 12 to 24 hours a day, 7 days a week. Model controllers have often worked ten straight days on twelve-hour shifts. A simulation center may run ten to twelve exercises during the year. Prolonged work schedules have a direct impact on the mental and physical capabilities of the model controllers.

- **The quality of the ATO and ACO messages.** ATO and ACO messages may be syntactically correct but not executable without further research. Different ACOs may use the same fields to convey different information. Route information may be blank, define a specified path, or simply indicate use of a defined airspace. Equipment information may similarly be blank or ambiguously defined, requiring the controller to perform additional research prior to final editing. For personnel unfamiliar with air planning and operations, this type of editing may be daunting.

Many additional examples can be cited to illustrate the myriad of complexities confronting the model controllers. Tools and technologies that alleviate redundant work or simplify mission editing can free the model controllers to concentrate on other areas of mission planning and execution. Expert system technologies can do just that. The expertise of senior model controllers can be captured in the form of defined rules, which can be applied systematically across a broad range of missions. One rule can be logically applied (or “fired”) across all of the missions in an ATO, instead of model controllers manually editing 40 or 50 missions assigned to them. The result of applying these expert system tools is a reduction in the number of hours required to refine mission tasking for simulation use. More importantly, the expert system allows a senior controller the ability to leverage his experience and knowledge across many more missions. Lastly, the novice controller has the ability to reuse the expertise of a trained controller to more accurately affect mission execution.

5.1 AWSIM TBMCS Interface (ATI)

The ATI (and its predecessor system, the AWSIM Contingency Theater Automated Planning System (CTAPS) Interface (ACI)), system has been used by the Joint Synthetic Battlespace (JSB) communities since 1996. The system is based on the initial design of Project Real Warrior (PRW) developed at the Warrior Preparation Center (WPC) in 1995. PRW was one of the initial prototypes to interface operational C4I systems, in this case CTAPS, with the JSB simulators. The PRW prototype evolved into ACI and finally, with the AOC shift from CTAPS to TBMCS, into what is the current ATI system.

The ATI system has five main functions:

- Extract Scenario data from the TBCMS Air Operations Database (AODB).
- Map simulation database entries to the TBMCS database entries.
- Translate the ATO and ACO messages.
- Provide a Mission Planner Workstation (MPW) for elaboration, checking and editing of the ATO missions
- Provide mission feedback to TBMCS.

Figure 1 shows the ATI system architecture which details the data flow between TBMCS, AWSIM and IMCN (see Section 6.2). The ATI system receives ACO and ATO messages from the TBMCS system. These messages are parsed into an internal database representation for easy data quality editing of missions. The ATI operator edits lookup tables with data needed by AWSIM to execute orders. The Graphical Input Aggregate Control (GIAC) Data Server (GDS) interfaces with AWSIM to pull information from the simulator that ATI will need for data quality checks between TBMCS and AWSIM. The model controller also uses GIAC to view the mission.

MPW gives the model controller the ability to view the ATO, view and modify AWSIM order stacks, and visually modify missions using GIAC.

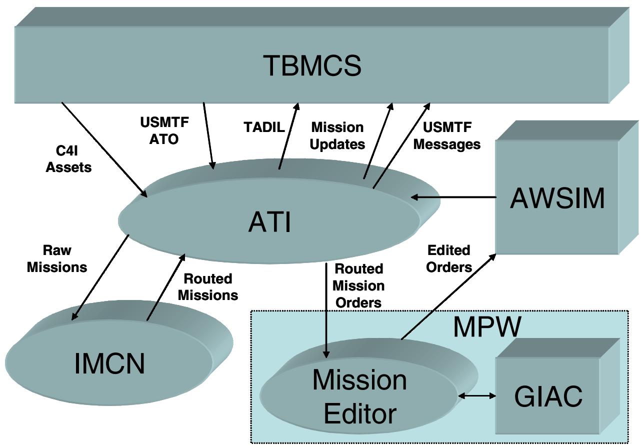

Figure 1: System Architecure

5.1.1 Command Control Data Interchange Format (C2DIF)

The ATI system needed to interoperate with a number of simulators used by the JSB simulation centers. This created the need to define a normalized data layer from which the suite of simulators can build simulator orders. The Command Control Data Interchange Format (C2DIF) is a package of classes that provides the normalized data layer. It captures the planning elements of the ATO and ACO, as well as the data elements that simulators require to generate orders.

Analysis of the ACO and ATO messages, and the simulation suite (AWSIM, RESA, JTLS, NASM, and ENWGS) was performed. The results were merged and refined to create the C2DIF.

In the C2DIF, the air mission is defined as a series of classes that describe the mission and a list of sequential events that the mission will perform. Using the C2DIF, a simulation order writer for the simulator builds the simulator order syntax required to execute the mission according to the ATO planning.

5.2 The Intelligent Mission Controller Node (IMCN)

The IMCN system is an expert system that aids the model controller in refining and editing a mission plan. Developed as a sub-module of the ATI and MPW system, IMCN uses the Java Expert System Shell (JESS) inference engine to process the data received from TBMCS by ATI and fire rules when criteria meeting a specified rule is encountered.

Model controllers are able to add, modify, or delete rules to get the desired behavior in a mission. For example, rules can be easily developed to determine appropriate equipment loads for a mission based upon the type of aircraft and other mission parameters. Model controllers can create rules specifying altitude ranges for an aircraft based upon aircraft and mission type. These rules can then be applied across all missions in the ATO. Final review and modification of the order stacks occurs after IMCN processing is accomplished through MPW.

Controller workload reduction enables time for additional mission planning refinement, or a reduced number of controllers. Over the course of time, as rules are developed, the IMCN system is "trained" to perform better mission editing. As such, model controllers realize continued increases in the amount of time available to review and refine missions. With the aid of the IMCN system, a smaller number of model controllers can work with a larger number of missions. Model controllers are able to work in areas that traditionally have received less support, such as threat cell augmentation. More complicated mission tasking can be processed. As the IMCN knowledge base grows, the ability of the system to process missions rapidly outperforms inexperienced model controllers. Rules can be retained for future use when the same AOC returns for an exercise.

Figure 2 shows the high level architecture of the IMCN system. IMCN is comprised of modules that function as independent threads. The operator utilizes only those modules necessary for the task that is being accomplished. Message interactions between the modules are event driven, as opposed to polling driven, improving run time efficiency. The IMCN GUI provides operator control of each of the threads independently.

5.2.1 Adapter

The adapter interfaces with ATI via the High Level Architecture (HLA) or directly through a JDBC connection. The ACO message is translated into defined air space objects, the ATO message into mission objects, the units into entity objects, and the Surface to Air Missile (SAM) units into emitter objects.

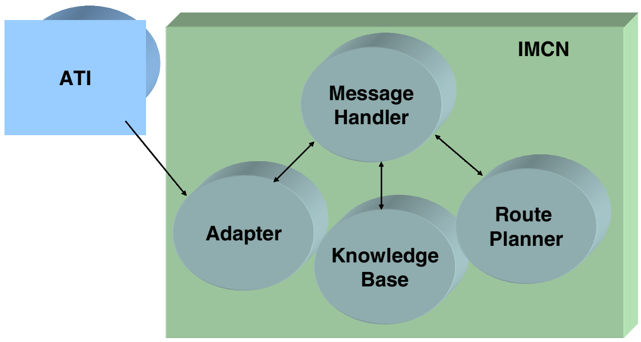

Figure 2: IMCN High Level Architecture

When using HLA as the transport mechanism, Java reflection is used to encode or decode HLA objects and interactions being sent over the Run Time Infrastructure (RTI). This offers the flexibility of creating new C2DIF objects without having to rewrite any portion of the adapter. This interface is configured based on an XML file that is parsed to determine the C2DIF elements required by either system.

The adapter module facilitates the interface of IMCN with external systems. To create a new external interface, the adapter is extended and becomes the middle layer between the external system and the IMCN system. The ATI adapter uses JDBC to pull database records, convert the record to a C2DIF object, and send the object to the message handler. The HLA must be adapted to the Federated Object Model (FOM), the messages converted to C2DIF, and the objects forwarded to the message handler.

5.2.2 Message Handler

The message handler processes message traffic between modules. Each module attaches to the message handler and identifies the message types that they require.

The message handler is designed on a hub and spoke architecture for passing objects. It is the module that IMCN uses to pass C2DIF objects internally. Each module attaches to the message handler and subscribes for the objects for which it is interested. When the message handler receives an object it forwards it to all modules subscribing to that object type. The message handler has performed well for the prototype and initial release of IMCN. It lends itself to be easily replaced with distributed system architecture when the IMCN needs to scale across several systems.

5.2.3 Expert System Knowledge Base

The knowledge base module modifies the mission based upon rules that have been loaded into the knowledge base. Within the knowledge base, rules fire as matches on facts are discovered. A mission may be sent to the route planner for egress and ingress routing; an appropriate equipment load may be automatically defined based upon the mission facts. The model controller interacts with the module via a command line editor. The model controller can add or modify rules. In addition, IMCN has a rule editor that provides a visual interface for rule management by the model controller.

For the expert system components of IMCN, JESS was selected. JESS is an open source, Java-based rule engine and scripting environment. Used with success in the government, commercial, and academic sectors, JESS's platform independence greatly enhances the ability of IMCN to interface with multiple simulators. JESS uses Java reflection to cast arbitrary Java objects as shadow facts within the knowledge base.

5.2.4 Route Planner

Mission routing information is included in the ATO messages produced by TBMCS. This routing information is basic in nature (take off base, destination, recovery base). The messages do not, for instance, include detailed ingress/egress routing. In the real world, this level of planning is handled by the wing. In the exercises, however, this level of detail must be supplied for the simulators, and the model controllers are responsible for creating this detail. Improved routing increases the realism of the exercise. As an example, the lack of realistic routing leads to unacceptably high and artificial "kill rates" caused by simplistic straight line flight paths over threats. While parameters had been tuned to compensate for this effect, removing the artificiality adds to the credibility of the exercise.

The route planner module estimates ingress and egress routes for the mission around or through the known threat areas. The model controller can modify these estimated route paths using the MPW software. The routing algorithm examines the battlespace grid containing threat areas and defined airspaces. Coupling this information with mission and aircraft parameters, the route planner module automatically plans a route to/from the target using the defined airspace.

Figure 3 illustrates the route planning. The process iterates, seeking to minimize the calculated threat level of the flight path.

5.3 Layered Architectural Approach

By interfacing IMCN with ATI, existing interfaces to TBMCS and the simulators are reused. IMCN utilizes the capabilities of the ATI system providing model controllers with a familiar system in which to perform mission editing without the need to learn an entirely new environment.

Figure 4 shows the layered architecture of the ATI/IMCN system. By providing a layered architecture, the system benefits by isolating the impact of change to the

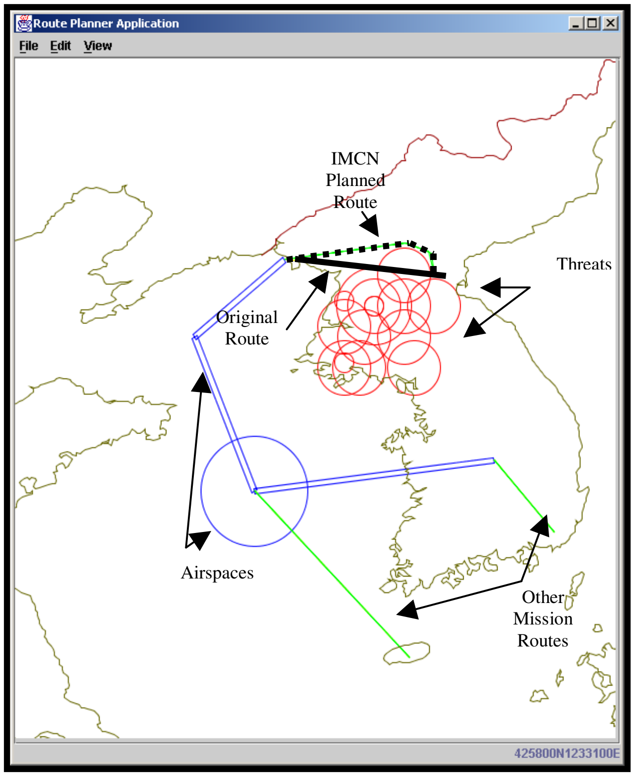

Figure 3: Route Planning

target C4I system or the simulation. Additional C4I systems can be (and have been) accommodated by changes to the C4I layer only. Similarly, additional simulations can be (and have been) integrated by modifying the simulation layer.

5.4 Results

In the exercises and events that utilize C4I and simulations to provide training to military personnel, a large number of individuals are necessary to stage the event. Prior to the integration of the C4I systems with the simulation models as described above, at a large event such as Ulchi Focus Lens (UFL) up to 120 people were required to process the output of the C4I systems to prepare it for use by the simulation models. These personnel were also responsible for preparing the feedback of the simulation models for insertion back into the C4I systems. With the implementation of the ATI system, this effort has been reduced to approximately 60 individuals, representing roughly a halving of training costs. This reduction has largely been due to the complete automation of many manual tasks. In other cases, the ATI system has provided a powerful user interface to assist personnel in translation, transformation, and fidelity adjustment.

The recent introduction of the IMCN system provides even greater cost reduction. By capturing the subject matter expertise inherent in the knowledgeable individuals supporting the exercise, IMCN has demonstrated that an order of magnitude decrease in personnel is achievable.

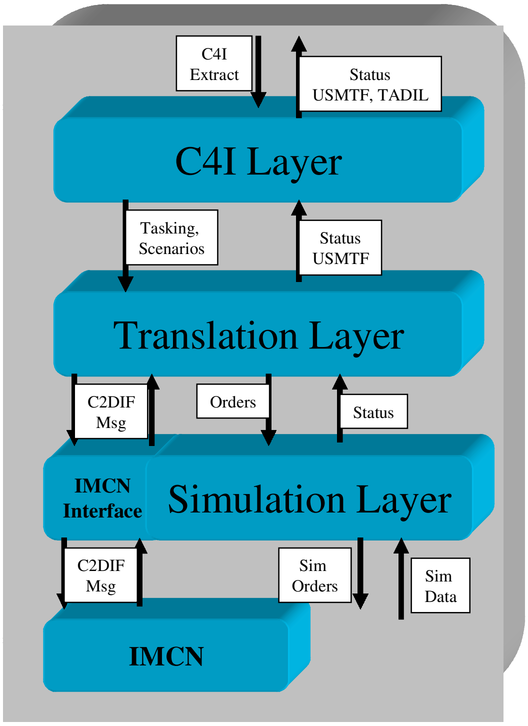

Figure 4: Layered Architecture

When fully implemented, the IMCN rule-based intelligent agent approach will allow for single digit numbers of personnel to support these large scale events.

6 SUMMARY

The integration of simulations with C4I systems provides great opportunity for the military services to reduce the costs associated with executing large scale training events and exercises.

Through the use of integration software and expert systems, true interoperability can be achieved. This results in order of magnitude reductions in the personnel required to support the event. In addition to these reductions, an increase in the realism of the exercise is achieved through the use of the rule-based engine approach.

The development of the ATI/IMCN system reflects the benefits of appropriate technology selection. Modular design, open standards, and a system perspective have been applied to develop the system in a cost effective, time sensitive manner. The system architecture provides for ongoing evolution in the form of increased functionality, expandability, and scalability.

The significant investment in the development of existing C4I systems and simulations can thus be exploited through rich interoperability. This leverages the legacy capabilities, avoids significant development costs associated with creating new systems, and enables dramatic recurring cost reductions at every training event and exercise in which the value of simulation-C4I interoperability is employed.

ACKNOWLEDGMENTS

The authors would like to acknowledge the various members of the technical project team responsible for the development of the ATI and IMCN systems, in particular Larry Crochet, Larry Esmonde, Jim Gustafson, Paul Gallagher, Dan Loftus, Richard Long, John Neyer, Kevin Sheehan, and Ruben Villanueva.

AUTHOR BIOGRAPHIES

MARK WHELAN has 20 years experience in information technology, systems integration, and digital services. He is currently a Managing Director for Gestalt, LLC, a leading provider of collaborative technology for governments and Fortune 500 customers. Prior to Gestalt, Mr. Whelan was the Vice President of Operations for the Mid-Atlantic Region for Breakaway Solutions, a full service provider of digital solutions, served as CIO for a division of a Fortune 100 electric and gas utility, and also served as a Project Executive for this utility where he was responsible for managing enterprise-level development and implementation projects. Mr. Whelan has a B.S. in Computer Science from Penn State University, and an MBA from St. Joseph's University. His email address is <`mwhelan@gestalt-llc.com`>.

JOHN LOFTUS has 19 years experience in the management, business justification, and implementation of advanced software solutions. Mr. Loftus has been a consultant to numerous Fortune 500 and emerging companies as well as a number of investment bankers and venture capitalists. Currently he is a Co-Founder of Gestalt, LLC, a leading provider of collaborative technology for governments and Fortune 500 customers. Mr. Loftus focuses on digital strategy development, technical due diligence, ROI analysis, and e-business application development. Previously, Mr. Loftus was the Sr. Vice President of Professional Services with Breakaway Solutions where he managed all strategy and e-business consulting engagements including the m-Business Practice. Mr. Loftus holds a B.E.E., M.S.E.E., and M.S.C.S from Villanova University and an M.S.O.D. from the University of Pennsylvania. Mr. Loftus serves on the Board of Directors of Avista Advantage, the leading provider of facility intelligence for large multi-site organizations. In addition, Mr. Loftus serves on the Computing Advisory Board of LaSalle University. His email address is <`jloftus@gestalt-llc.com`>.

DAVID PERME has 10 years experience in the executive management and operations of advanced software solution providers. Mr. Perme is an entrepreneur and recognized expert in interoperability. Currently he is a Managing Director and Co-Founder of Gestalt, LLC, a leading provider of collaborative technology for governments and Fortune

500 customers. Prior to Gestalt, Mr. Perme was Director of Delivery for Breakaway Solutions, responsible for managing multiple teams delivering commercial B2B collaboration services. Mr. Perme holds a BS in Aerospace Engineering from Kent State University and MSCS from Boston University. Mr. Perme is a former US Air Force fighter pilot, serving with distinction for over 10 years, and culminated his flying career as a Major Air Command evaluation pilot in USAFE (US Air Forces in Europe). His email address is <dperme@gestalt-llc.com>.

RICHARD BALDWIN is the Regional Director of Gestalt's Florida office. He has over 15 years of experience in the software development arena, including extensive experience in the modeling and simulation world. Mr. Baldwin is an expert in Java-based technologies included object oriented analysis and design and intelligent agents, as well as extensive knowledge of XML and Oracle. Prior to Gestalt, Mr. Baldwin held leadership positions in the private sector with Breakaway Solutions, Sterling Software, and Mystech Associates. Mr. Baldwin started his career in the military and is a 10 year Air Force veteran.

Mr. Baldwin is currently completing his Master of Science degree in Computer Science from Webster University. He holds a B.S. degree in Computer Science from Webster. His email address is <rbaldwin@gestalt-llc.com>.