DEVELOPMENT OF CATT COMPATIBLE INFRA-STRUCTURE TO SUPPORT AVIATION EXERCISES

Mark Falash, Roger Branson, Eytan Pollak, Ph.D
Lockheed Martin Information Systems
Orlando, Florida

ABSTRACT

The fielding of the Close Combat Tactical Trainer (CCTT) System has established an initial infra-structure and set of tools defining the virtual battlefield environment for the Combined Arms Tactical Training (CATT) family of training systems, primarily focused on ground vehicles. This paper discusses the work undertaken by Lockheed Martin Information Systems Company to add the capability to the current CATT infra-structure to fully support aviation exercises. In addition to providing a selectable infra-structure fidelity and CCTT compatible simulation environment, it emphasizes inter-operability with other training devices. This scaleable, packaged support environment, along with a simulator, constitutes a complete training system.

Modifications have been made to the architecture and applied using a reconfigurable simulator. The existing CCTT infra-structure and code formed the baseline for this work. The system will be used to develop aviation behaviors and functionality to support aviation-oriented planning and debriefing techniques. The system will also be used to investigate high level architecture (HLA) compliance and compatibility issues, upgrade of CCTT to HLA compatibility, and functionality requirements of exercise management, instructor/operator station, computer generated forces and their control, and after action review. The system incorporates web technologies and other strategies to provide independent platform capabilities and to enhance system intuitive control and expanded access.

AUTHOR'S BIOGRAPHIES

Mark Falash is the lead software engineer responsible for the design and development of the initial phase of the Lockheed Martin Information Systems Company CATT Core infra-structure. He previously served as the software architect for CCTT. His responsibilities included software architecture changes, performance modeling and analysis and rehosting CCTT software to symmetric multi-processors (SMPs). He has since worked with and supports UKCATT and WARSIM on a periodic basis. He received his M.S. in Computer Science from California State University, Chico.

Roger Branson is a senior staff project engineer specialist at the Lockheed Martin Information Systems Company in Orlando, FL. His responsibilities include oversight of the CATT Core infra-structure and enhancement for aviation functionality and exercise support, and requirements development for the United Kingdom CATT Battlegroup Headquarters Main cells and role player workstations. He received his M.S. in Aeronautical Sciences from Embry-Riddle Aeronautical University, Daytona Beach, FL.

Dr. Eytan Pollak is currently the IR&D Director at the Lockheed Martin Information Systems Company in Orlando, FL. His responsibilities include the LM Reconfigurable Simulator, Embedded Training Systems, War Gaming, CATT CORE Architecture, Synthetic Environment, and other DIS/HLA research programs. He serves as an adjunct professor of Electrical and Computer Engineering at the University of Central Florida. He received his Ph.D. from Purdue University, West Lafayette, IN.

INTRODUCTION

The Close Combat Tactical Trainer (CCTT) is the first of a series of simulation systems wherein various simulated elements imitating actual combat vehicles, weapons systems, and command and control elements are networked for real-time, fully interactive, collective task training on computer generated terrain. These high fidelity simulation systems are collectively known as the Combined Arms Tactical Trainer (CATT) simulation systems.

While CCTT addresses the collective training requirements for ground based forces with limited aviation support, CCTT's fully compliant DIS implementation provides the baseline infra-structure and core tools for building future systems in the CATT family and extending its training capabilities. This paper looks at using the CCTT infra-structure and tool set as a baseline for CATT Core and extending the functionality to include support for aviation assets and requirements of emerging aviation programs.

CATT CORE ENVIRONMENT

CATT is an acquisition strategy for developing and fielding the US Army's objective CATT virtual simulation program. CATT consists of many components that can be grouped into two categories: 1) manned simulators, and 2) the simulation environment, CATT Core.

Family of Systems

CATT is a set of collective combined arms trainers that will effectively and efficiently provide commanders the opportunity to train in a realistic, force-on-force, virtual battlefield environment representing up to battalion task force level. This set of high fidelity simulation systems is a combination of manned vehicle specific simulators, manned generic simulators, workstations, semi-automated forces, and Distributed Interactive Simulation (DIS) technology. The objective of the CATT is to provide proficient and sustained training of selected individual, crew, collective, staff, and combined arms tasks.

The five systems making up the CATT family are:

1. The Close Combat Tactical Trainer (CCTT)
2. The Aviation Combined Arms Tactical Trainer (AVCATT)
3. The Air Defense Combined Arms Tactical Trainer (ADCATT)
4. The Fire Support Combined Arms Tactical Trainer (FSCATT), and
5. The Engineer Combined Arms Tactical Trainer (ENCATT)

Requirements and fidelity for the trainers are driven by the individual programs.

Commonality of Tools and Services

The CATT Core environment is composed of tools and services common to all combined arms virtual training system requirements. PM CATT, STRICOM, US Army, identifies twelve common elements desired for the CATT Core environment and CATT virtual training systems. Starting with the current CCTT environment as the baseline, work has begun to extend the capabilities of the CATT Core environment focusing on the requirements anticipated to support aviation exercises. In the following paragraphs the CCTT hardware and software architectures are described. While the CCTT architecture was developed for ground vehicle exercises, we can see that the architecture with small modifications can be extended to support aviation simulator applications.

BACKGROUND OF CCTT

CCTT is based on and fully compliant with the Distributed Interactive Simulation (DIS) standard version 2.04R. The combined effects of the manned simulators, semi-automated forces and computer workstation create a complex, highly realistic, doctrinally correct virtual environment to meet defined training objectives. Manned simulators using realistic vehicle crewstations require personnel to perform their respective tasks correctly for the collective mission to be successful. The Semi-Automated Forces provide computer generated BLUFOR and OPFOR vehicles and behaviors, which are highly realistic and used to stimulate the virtual synthetic environment that is perceived by the manned simulators. The computer workstations further augment the synthetic environment with tactical Operation Centers (OC), artillery, mortars and aviation units. The After Action Review (AAR) system supports real-time monitoring of the exercise in a stealth mode, while also collecting data for analysis, playback, and debriefings.

OVERVIEW OF CCTT ARCHITECTURE

Primarily the DIS standard, as well as the intent to be a fully distributed system, drove the CCTT architecture.

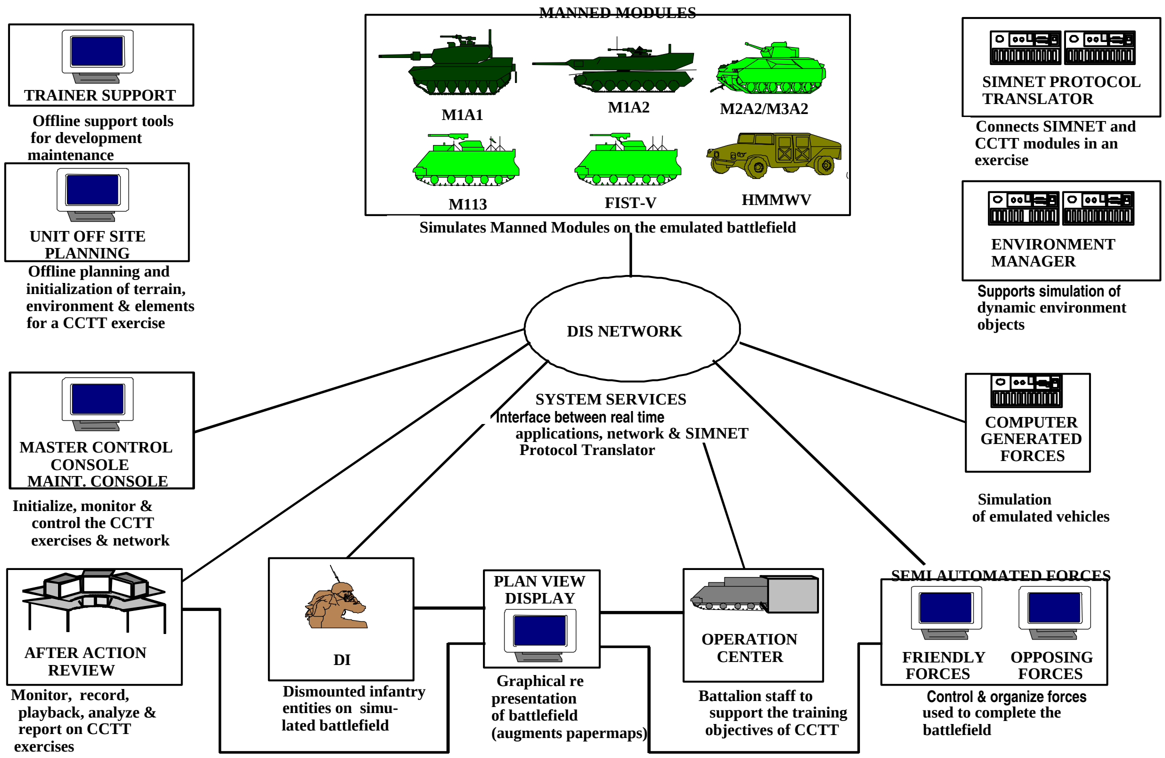

Figure 1 - CCTT Communication Infra-Structure

The following principles represent the overarching design constraints that applied to CCTT development:

- No central computer exists for event scheduling.
- Each simulation application is autonomous and is responsible for maintaining the state of one or more simulation entities.
- Standard protocol is utilized for communicating 'ground truth.'
- Receiving nodes are responsible for determining what is perceived.
- Simulation applications only communicate changes to the state of the entities that are being simulated.
- Dead reckoning algorithms (DRA) are used to reduce communications bandwidth requirements and processing.

Central to CCTT's architecture is the DIS network, which provides the communication infra-structure (Figure 1). CCTT DIS network has been implemented using FDDI networking media and network adapters that support 64 multi-cast group addresses. All the host processors are based upon the PowerPC chip executing the AIX operating system. The only exception is the Unit-Off-Site-Planning Station (UOSP) that has been implemented with a Pentium based processor executing Windows NT. The CCTT tactical areas and manned simulators are shown in Figure 1.

The After Action Review (AAR) provides the operator a set of operator capabilities that can be used to examine and analyze the simulated battlefield. The AAR operator can visually view the battlefield through 3D displays that are under control of the operator. The viewpoint from which the battlefield is displayed can be freely moved around the simulated battlefield. A Planned View Display (PVD) provides the operator a 2D map-like view of the simulated battlefield. AAR's data analysis and recording capabilities support the operator's analysis of the exercise with statistical reports aimed at providing insight into the exercise. The AAR workstation provides four radios that are available to the operator for listening and speaking on any of the radios defined for the exercise. A voice instruction channel is used to speak to any simulator or group of simulators during the exercise

The Computer Generated Forces (CGF) provides for the simulation of computer generated forces with limited operator control of units for both the Semi-Automated Forces (SAF) and Operational Center (OC) workstations. The emulated tactics are documented in Combat Instruction Sets (CIS) and approved by the government prior to implementation. The CCTT design allows for the SAF and OC users and CGF simulation to be distributed among multiple, physical processors. The CGF platforms supplement the CCTT manned modules with the BLUFOR and OPFOR platforms necessary to provide a realistic training environment.

The Dismounted Infantry (DI) station is a man-in-the-loop simulator that provides dismounted leaders the opportunity to practice their tactical decision making skills in the synthetic battlefield by leading, directing and controlling computer generated dismounted forces. In addition, the manned simulators are provided an enhanced training environment by providing dismounted combat activity within the synthetic battlefield.

The Manned Simulators (SI) provide man-in-the-loop training via high fidelity manned modules for M1A1, M1A2, M2A2, M2A3, FIST-V and HMMWV vehicles. Software control of the manned simulators module and hardware components provide a realistic training environment of US Army soldiers. The manned simulators realistically model the vehicle dynamics and behavior of their real-world counterparts. The vehicle dynamics are a result of models based on real-world behavior applied against simulated terrain and other synthetic battlefield conditions.

The Master Control Console (MCC) and Maintenance Console (MC) are similar hardware configurations. Each provides capabilities to initialize the CCTT workstations and modules and execute built-in tests for up to five simultaneous exercises. MCC/MC provides capabilities to start, stop, resume and end exercises using the DIS SIMAN PDUs. Weather and other special effects can be inserted into specific exercises by an MCC/MC operator. MCC/MC network monitoring, control and management of the CCTT physical network is provided by the Commercial-Off-The-Shelf (COTS) software product, Netview 6000 System Management. A Planned View Display (PVD) provides the operator a 2D map-like view of the simulated battlefield.

The Operation Center Workstations (OC) provide the simulated functions of the battalion staff to support the CCTT training objectives. OC provides the capabilities required for trainees in manned simulators to request and receive services, such as refueling and repair, that would be required in the real world. Communications is provided by simulated radios that closely resemble the real-world counterparts with models to simulate the effects of the synthetic environment (i.e. propagation loss, interference). The OC workstations provide the user interface for OC units and platforms that are simulated by the CGF element of CCTT.

- Combat Engineering Support (CES) provides engineering support for such things as breaching and clearing minefields, construction and destruction of obstacles, bridges and fighting positions.
- Combat Trains Command Posts (CTCP) provides fuel and ammunition resupply.
- Tactical Operations Center (TOC) consists of the S2 and S3 work areas.
- Unit Maintenance Collective Point (UMCP) provides for vehicle repair and recovery.
- Fire Direction Center (FDC) provides mortar fire support.
- Fire Support Element (FSE) provides for development and communications of fire support plans, target priorities, and coordination of fire support.
- Field Artillery Battalion Tactical Operation Center (FABTOC) provides artillery support.
- Tactical Air Control Party (TACP) provides close air support.
- Higher Headquarters Support (HHQ) work area emulates battalion access to staff members at higher army echelons.

The Semi-Automated Forces Workstation (SAF) provides the user interface to create, control and destroy vehicles emulated by CGF. The SAF workstation gives the operator the ability to implement pre-exercise activities which include creating overlays and initialization of SAF units.

The Unit Off Site Planning station enables training personnel to perform planning activities prior to arrival at a CCTT training site. The UOSP software executes on an IBM compatible PC running Windows NT with a minimum of one 3.5 inch diskette. The UOSP workstation provides capabilities to select the terrain area, environmental conditions, and initialization parameters for TACP, FSE and CSS, and select BLUFOR and OPFOR units. The initial conditions which are specified using UOSP are transferred to the MCC via 3.5 inch diskette for use in an exercise.

CCTT Hardware Components. The major CCTT configuration items are composed of several sub-components, which when combined with CCTT development software, produce a working manned module or workstation. Each of the CCTT configurations use the same sub-components and software elements to satisfy common requirements. Figure 2 depicts the sub-components used to build any of the specific CCTT configurations.

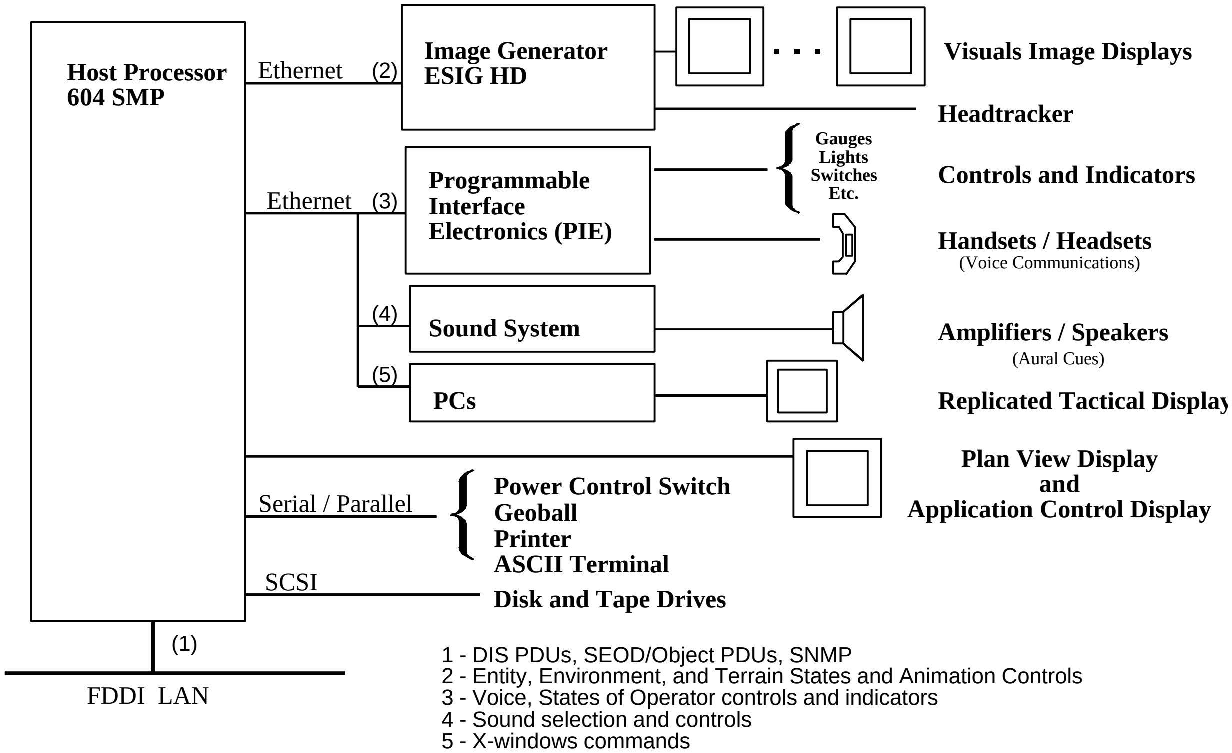

Figure 2 - CCTT Build Sub-Components

1. FDDI LAN
2. Ethernet
3. Host Processor (2-Way 604 Symmetric Multiprocessor (SMP))
4. Image Generator (IG) & Visual Image Displays
5. Programmable Interface Electronics (PIE)
6. Voice Communications (Radio)
7. Plan View Display (PVD)
8. Trainer Unique (TUD) or Application Control Display
9. Replicated Tactical Displays
10. Printer
11. Logger Tape Drive
12. Geoball

CCTT Software Architecture. The CCTT Architecture can be partitioned into two logical groups that work together to accomplish its mission: 1) the DIS specific capabilities that provide capabilities and services common to all CCTT applications (such as Coordinate Conversions and DRA of remote entities) and 2) the Application specific capabilities that provide the function specific to a specific configuration item (such as M1A1, CGF and AAR).

The CCTT software architecture and design took into consideration the implications associated with a system based on the DIS protocol and the requirements for a high fidelity and realistic, distributed simulation training system. As a result, the design constraints were applied both to development of the CCTT application software and to the CCTT software architecture.

Figure 3 illustrates the major software architectural components for CCTT and the high level interfaces between them. The Network, Entity Database, IG and Log Processes provide the bulk of the DIS related processing. The Planning View Display, Communication and PIE processes are common CCTT software elements which are utilized by CCTT applications which require these capabilities. The application process is unique for each configuration item and may be implemented as one or more AIX processes. Although each application process or set of processes represent capabilities that are unique to a particular configuration, there is extensive reuse within the application designs. A prime example of this is the environment functions which have been implemented once and used by all CCTT applications. The environment functions used by an application are bound into its executable and executes under its thread of control.

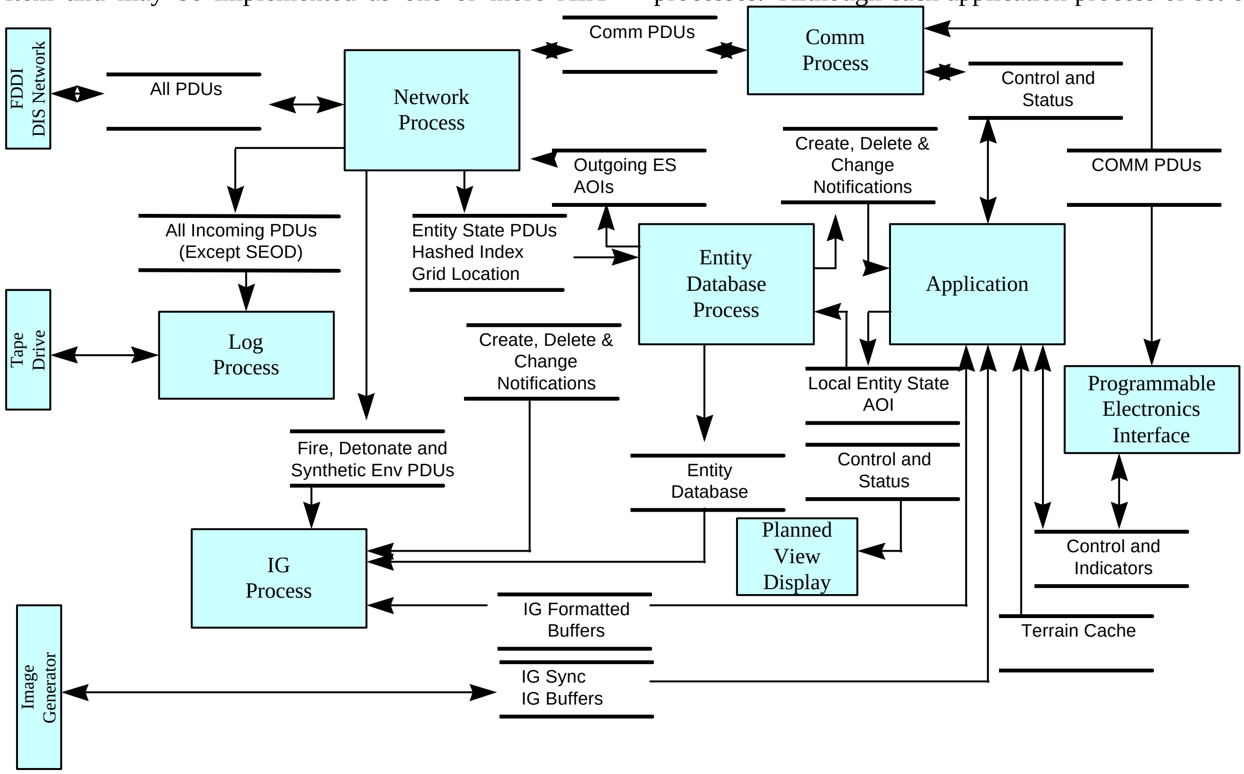

Figure 3 - CCTT Software Architecture Components

The Network Process manages the exchange of information between CCTT configurations. The exchange of information is accomplished by sending and receiving DIS PDUs. The Network Process is the single and only component of CCTT that interfaces directly with the DIS Network. Services are provided to the applications via an Application Programming Interface (API) which allows the applications to send, receive, allocate space for, format, concatenate and filter DIS PDUs. The Network Process stores raw DIS entity state PDUs in the entity database using an API provided by the Entity Database function. This API has been provided to and used exclusively by the Network Process. The API determines if the entity state PDU is for a newly discovered entity and where to store the incoming PDU. In addition, the API queues an indicator to the Entity Processor flagging that a raw entity state PDU for the specified entity has been stored in the entity database and requires conversion.

The Entity Database Process manages and provides access to the entity database which has been implemented as a shared segment. Any entity state PDUs that have been received since the entity processor was last scheduled are converted to an internal representation. The conversion process performs coordinate conversions, determines if smoothing is required, calculates the smoothing points, and reformats portions of the PDU to optimize performance. The entity process applies the dead reckoning algorithm (DRA) to remote entities that are located within the application specified Area Of Interest (AOI). Dead reckoning is also applied to ownship locations to be used when performing threshold checks. System specified thresholds are used to specify when an entity state PDU will be sent to other DIS applications attached to the DIS Network. If the ownship’s current state compared to its dead reckoned state exceed any threshold, an entity state PDU is sent by the entity processor on behalf of the ownship. The entity database provides a common API to all applications which allows them to interface with the entity database. Applications use the APIs to specify AOI, and to update ownship state including articulated parts and query state information associated with remote entities.

The IG Process provides the capabilities required to visually depict the activities of the remote entities, fire and detonations on the virtual battlefield. This is accomplished using a dedicated host processor-to-image generator (IG) connection, an IG, a 3D visual database and display monitors. Ownship specific visual effects are controlled by the ownship application utilizing visual APIs provided by visual software capabilities common across CCTT. The IG Process utilizes the Entity Database capabilities which direct the entity process to notify applications (in this instance, the IG Process) of changes in order to minimize the overhead of communicating changes between the two processes. The IG Process maintains a prioritized list of entities to be displayed. The list is updated whenever the entity process informs the IG Process that it has created, deleted or updated entity state based on incoming PDUs. The IG Process queries the entity database to retrieve the necessary information required to generate the IG buffers which drive the IG when sent over the dedicated Ethernet.

The Log Process exists only within the AAR configuration. When the Log Process exists, the Network Process will route all incoming and outgoing DIS PDUs to be recorded on tape. When directed by AAR to replay an exercise, the Log Process will read PDUs from tape and hand them off to the Network Process to be processed as incoming PDUs.

The Plan View Display (PVD) process generates, displays and manipulates a 2D electronic map of the simulated battlefield. PVD capabilities allow applications to interface with the user by displaying Unit Icons, Overlay Symbols, Menu Items, Unit Status and damage. The PVD supplied application interface (API) allows applications to direct PVD actions and retrieve user actions.

EVOLVEMENT OF THE CATT CORE ARCHITECTURE

CCTT was confronted with many challenges throughout development with regards to technical performance. Significant effort went into reducing data movement, redundant calculations and optimizing algorithms to get CCTT within reach of its performance requirements. Bringing manned aviation simulators into the structure introduces additional requirements and challenges:

- Faster update rates
- Additional PDUs
- Lower number of entities, AOI
- Non-linear dead reckoning
- Sensors, databases and correlation
- Larger databases/lower density
- Aviation unique OCs
- Extensible vehicle architecture for high fidelity dynamics
- Aviation unique mission planning and debriefing techniques and reports.

The following represent a small number of CATT CORE concepts where overall system performance improvements are being investigated and realized using the aviation reconfigurable simulator as a testbed.

Multi-cast. IP Multi-cast is a COTS software implementation which supports the abstraction of multi-cast groups. A set of ‘n’ multi-cast groups are supported for which receiving processors may join and sending applications may send to. Receiving processors join and leave multi-cast groups dynamically under application control. CCTT’s smart network adapters use Content Addressable Memory (CAM) to remember up to 64 joined multi-cast group addresses. If the number of joined multi-cast groups is less than the number of addresses that are supported by the network adapter, the hardware adapter will filter out and discard the network packets with multi-cast groups addresses which differ from the joined groups by the receiving processor. Given these constraints, CCTT has defined five logical partitions of PDU traffic:

1. Exercise Traffic – PDUs whose only distinguishing attribute is exercise. Each exercise is assigned a multi-group address. When multiple exercises are in progress, the receiving host will not suffer the overhead of filtering out traffic from other exercises.
2. CGF Traffic – The CGF protocol (SEOD) PDUs. CGF generates PDUs whose purpose is to communicate information between CGF, SAF and OC workstations.
3. Location Sensitive – Entity Information and Warfare PDUs. The terrain is partitioned into ‘n’ square cells, where ‘n’ is the number of cells determined by the size of the terrain and a specified cell size. The location specified in the PDU is mapped to a multi-cast group which is used by the Network Process when sending the PDU. Location Sensitive PDUs utilize the large number of multi-cast groups. Care must be taken to properly size the cells such that receiving

processors are not consistently joined to more multi-cast groups than the hardware adapters can support. If this condition occurs, the adapter passes all the traffic (similar to broadcast) and the device driver will perform the filtering.

4. Radio Traffic – Signal PDUs. The current CCTT implementation supports mapping of radio nets to multi-cast groups, however this option is not exercised due to the number of multi-cast groups used for location sensitive groups. As technology progresses and network adapters with more than 64 groups become available, CCTT will be able to take advantage of these adapters when they are available.
5. Special Case – Where the intended receiver need not be assigned to a specific exercise_id such as Create Entity.

Area-Of-Interest (AOI). The CCTT databases used to represent the simulated battlefield cover a large geographic region, up to 100km by 150km. In addition, CCTT is required to support exercises which are comprised of more than 800 entities with no constraints as to how these entities are distributed across the terrain. The simulated battlefield size and number of entities, along with the DIS protocol which can be characterized by frequently transmitted network packets, combine to place significant restrictions on CCTT's ability to scale as well as meet its performance requirements. The DIS protocols did not represent a bandwidth problem for CCTT (<15 percent utilization), however they do create a CPU loading problem given the number of interrupts generated by incoming PDUs. In order to contain and bound CPU loading as the size of the exercise grew, CCTT implemented a concept referred to as Area-Of-Interest (AOI). The Entity Database exported the capability to create AOIs to applications through an Entity Database API. By specifying AOIs, the application has specified to the Entity Process the boundary that will contain the remote entities that are of interest. The Entity Process will not convert, dead reckon, etc., any Entity State PDUs that are outside the AOI. In addition, the Entity Process uses the AOI to direct the Network Process to multi-cast groups to join and remove. The AOI represents a simple concept that allows CCTT to control processor loading directly related to incoming PDUs on the DIS Network.

Applications can specify AOI, and reduce the processing load by allowing CCTT to filter entities that are not of interest to this simulator. Once an AOI is defined, the entity processor will manage the AOI for the application by tracking relative to the ownship location. This removes the burden of the application recalculating the AOI as the local entity moves around the synthetic environment.

PDU Concatenation. When PDUs are transmitted, each host processor attached to the network will suffer an interrupt for each PDU received. PDU concatenation is the batching of one or more PDUs onto a single network and then transmitted on the DIS Network. Entity State PDUs represent the bulk of the PDUs generated during an exercise. Therefore, CGF represents the greatest opportunity to exploit PDU concatenation within the CCTT configurations since it hosts the largest number of entities per processor and, in most cases, over 70% of the entities generated.

Dead Reckoning. CCTT's initial implementation of the dead reckoning algorithms mirrored those as specified by DIS. The formulas were implemented verbatim and the resulting overhead was prohibitive given the frequency at which they were executed. Since the dead reckoning algorithms are linear, CCTT modified its implementation. Entity Processor now calculates delta x, y and z values when an incoming entity state PDU is converted to an internal representation. Dead reckoning is reduced to a simple addition for each dead reckoning that occurs. Non-linear dead reckoning algorithms using accelerations is needed to reduce the updates for a fast moving aviation entity.

Entity Database Change Notifications. Many of the CCTT applications are only interested in remote entities that are within the specified geographical region (AOI) and meet application criteria (filtering). An example of this is the IG Process which generates IG buffers for a limited number of remote entities and effects, especially those resulting from Fire and Detonate PDUs. The buffer is required due to the IG limitation on the number of models that can be processed each frame. Currently, the image generators used by CCTT can handle approximately 200 remote entity updates per frame, which is well below the possible 800 or more entities that could be within visual range. Given these conditions, the IG Process maintains a list of prioritized entities to be displayed. The Entity Database Process notifies the IG Process whenever an entity enters the AOI, exits the AOI, is deleted or created. The IG Process is able to process the prioritized list of entities based on the changed notifications, rather than searching the entity database and performing comparisons to determine what has changed. The change notification overhead for entity database proved to be minimal since its processing threads already made decisions required to determine when create, delete and update notifications are generated.

APPROACHES TO INTER-OPERABILITY

Inter-operability is a concept whereby logically different (such as CCTT and FSCATT) distributed and possibly heterogeneous systems exchange data in a manner which results in a training exercise which meets its training objective. To address inter-operability several fundamental concepts must be addressed. First and foremost is the mechanism used to communicate exercise, entity information, environment conditions and exercise management data. The DIS standard is currently the mechanism employed by CCTT. Second is the natural and tactical environments that are generally referred to as the synthetic environment. The tactical environment includes entity platforms such as tanks, soldiers, aircraft, missiles and ships. The natural environment includes terrain conditions, atmosphere and terrain features such as buildings, roads, bridges, etc. The third element is correlation. Each simulation maintains of its own view and local representation of the synthetic environment.

A frequent approach to inter-operability is development of paper standards and interface control documents. Paper standards alone do not assure inter-operability. The underlying reason is that each program will interpret using the standard differently. The standard is not complete (data format specified rather than content) and paper standards can seldom perform a complete analysis of all conditions which can exist and the appropriate response.

Software reuse in combination with paper standards is sometimes employed as well. Reuse of specific algorithms rather than architecture, architectural services and databases is more often the case. Reuse typically falls shorts since system behavior is influenced by architecture and its associated data, rather than by reuse of specific algorithms.

POSITIONING FOR HLA

Inter-operability is achieved using similar architectures, architectural services and databases. Applying the CCTT derived architecture to the reconfigurable simulator has made the device inter-operable with CCTT. The philosophy of a common framework to facilitate inter-operability between all types of models and simulators is the basis of the latest, most visible emerging technology within the modeling and simulation community, the High Level Architecture (HLA). The reconfigurable simulator will be used to investigate the issues of moving from a DIS legacy system to a HLA compliant federate. Re-engineering the legacy system requires software modifications. Compliance becomes a function of how well the resulting federate handles the functions defined in the HLA interface specification. Lessons learned here will be used to migrate the CATT CORE architecture to HLA, and then applied to other DIS legacy systems.

INITIAL APPLICATIONS

LMIS is making these modifications to the baseline CCTT environment and implementing these changes in the LM Reconfigurable Simulator and its supporting environment. This work will be the baseline for CATT Core development and set the ground work for supporting aviation assets in future aviation programs. The LM Reconfigurable Simulator environment is the testbed for this development within the LMIS IRAD department. Analyzing future program solicitations and using the requirements in the US Army Aviation Center “AVCATT-ARMS Operational Requirements Document” and US Army STRICOM paper “Desired After Action Review Capabilities for Aviation ARMS”, the LMIS IRAD staff has identified several key areas for development.

Delta Aviation Requirements

Aviation exercises have additional requirements beyond the current CCTT capabilities. Additional aviation behaviors are being developed and added to the common environment to support aviation missions using manned aviation simulators, and SAF aviation entities and support units. In addition, the use of a standardized aviation mission planner to support aviation missions is being evaluated. Development is required to interface the resulting files of the offline planning sessions with the master control console. This includes preparation of an environment file useable by the mission planner system, an online editor, and implementation of the new behaviors within the CATT Core environment.

Future development will include an instructor/operator station implementation for faults and malfunction control, giving the instructor a choice of faults and complete control of the insertion/repair. Reaction to system failures and degradations is an important part of the training environment for aviators performing critical mission tasks.

MCC/AAR

Combining functional workstations on a single platform reduces the number of computing platforms required to support a system. This investigation addresses the issues of scalability to support simple to complex exercises. Single processors using common processes, such as the PVD, are currently owned by each application on a CCTT workstation. This results in multiple copies of the process being loaded to support multiple applications on the same workstation. Common processes, such as the PVD, are being reworked so that it can be shared by multiple applications, using common databases and selectable overlays for the display. The MCC and AAR are being combined as a single workstation since the MCC becomes a "background" task during the exercise once up and running. A shared PVD process made this possible.

Modified and new debrief report functions are being added to the AAR to support the debrief of aviation missions. These changes include report formats and web access to the summaries.

Aviation Device

The LM Reconfigurable Simulator is the testbed for the new and emerging technologies. The reconfigurable simulator can represent any of several current aviation rotary wing aircraft. The end result of applying these new technologies to the reconfigurable simulator enables the device to inter-operate with the Combined Arms Tactical Trainers (CCTT) to the degree that it will appear to be another CCTT module, become a member of the CATT Core Family. Work in this area is described the paper "A Reconfigurable Software System Architecture." The efforts extended here provide the ground work for future aviation trainer development.

APPLICATION OF WEB TECHNOLOGIES

LMIS is supported by the General Electric Corporate Research Department in applying web technologies to the CATT Core environment. Web "page-like" functionality and user interaction was applied to the after action review capabilities for the LM Reconfigurable Simulator and described in the paper "A Web-Based Approach to After Action Review". This technology is being extended into the CATT Core environment for the control of the PVD display, and AAR debriefing capabilities, especially for presentation of standard, selectable reports. The results of these efforts will be used to drive modification of other CATT Core environment graphical user interfaces.

SUMMARY

CCTT forms the baseline for the CATT Core environment. Modifications are being made to the toolset to support aviation exercises, specifically to extend the toolset and establish services needed to support planning, executing, and debriefing aviation assets. Development of these modifications will be implemented with the LM Reconfigurable Simulator as the testbed. The experience derived here will be used to drive future applications.

REFERENCES

Aviation Combined Arms Tactical Trainer and the Aviation Reconfigurable Manned Simulator Operational Requirements Document (26 September 1996). US Army Aviation Center, Directorate of Training, Doctrine, and Simulation, Fort Rucker, AL.

Braudaway W. K. Ph.D, Harkrider S. M., (1997) Implementation of the High Level Architecture Into DIS-Based Legacy Simulations, 1997 Spring Simulation Interoperability Workshop.

Desired After Action Review Capabilities for Aviation Reconfigurable Manned Simulator System (1997). STRICOM, US Army, Orlando, FL.

Kelliher T., Volpe C., Pollak E., (1996). A Web-Based Approach to After Action Review, Proceedings of the 1996 IMAGE Conference. Scottsdale, AZ: The IMAGE Society.

West F., Pollak E., Falash M., (1997). A Reconfigurable Software System Architecture, Proceedings of the 19th Interservice/Industry Training Systems and Education Conference. Orlando, FL.