Adapting to Urban Warfare

Andy Ceranowicz
Alion Science and Technology
Alexandria, VA
aceranowicz@alionscience.com

Mark Torpey
Lockheed Martin
Burlington, MA
mark.torpey@lmco.com

ABSTRACT

Urban operations are currently of great concern to the defense community. J9, the Experimentation Directorate of USJFCOM, and the Joint Advanced Warfighting Project are currently conducting an experiment to investigate concepts for applying future technologies to joint urban operations. The first phase of the experiment focuses on employing future sensors to remotely monitor and understand enemy operations in a foreign city. Characteristics of the urban environment include high building density, a large civilian population, and a cultural environment. These characteristics pose significant challenges for simulation designers. This paper describes the modifications required to adapt the simulations supporting the experiment, JSAF and SLAMEM, to the urban environment. A landscape with a large number of buildings had to be automatically generated and represented in a space efficient manner. Large concentrations of vehicles and pedestrians had to be modeled moving realistically through the city. This behavior had to be automatically generated since it would be impossible to individually control 100K entities. Embedding cultural features within the database in the form of building functions and other building attributes allows the civilian entities to automatically plan their movements based on generic daily schedules. Sensors had to be modified to detect building properties. The density of both entities and structures made both movement and intervisibility calculations significantly more expensive requiring optimization combined with the application of large amounts of hardware. Computation and control was distributed between three CONUS sites and the High Performance Computing Center at Maui. Limiting and balancing simulation traffic was a major effort. Source squelching was enabled by a distributed data collection system developed to collect data locally on each simulation node while still allowing analysts to perform real time queries during the experiment.

ABOUT THE AUTHORS

Andy Ceranowicz is the technical lead for federation and Joint Semi-Automated Forces (JSAF) development at J9. He led the development of the Millennium Challenge 02 federation as well as the development of JSAF and its predecessors, ModSAF and SIMNET SAF. Andy is a Senior Science Advisor at Alion and holds a Ph.D. in Electrical Engineering from The Ohio State University.

Mark Torpey is a J9 federation developer and is the lead developer and integrator of JSAF. He is a Staff Software Engineer at Lockheed Martin Simulation Training and Support - Advanced Simulation Center (LMSTS-ASC) in Burlington MA. Mark holds an M.S. in Computer Science from the University of Massachusetts.

Adapting to Urban Warfare

Andy Ceranowicz
Alion Science and Technology
Alexandria, VA
aceranowicz@alionscience.com

Mark Torpey
Lockheed Martin
Burlington, MA
mark.torpey@lmco.com

BACKGROUND

Urban Resolve is a three-phase experiment designed to explore new approaches to urban combat for the 2018 time frame. The U.S. Joint Forces Command (USJFCOM) J9 Directorate and the Joint Advanced Warfighting Program (JAWP) at the Institute for Defense Analysis are conducting the experiment. The hypothesis is that future sensors will provide a significantly higher degree of situational awareness than we currently have, allowing us to conduct urban operations with more effect and economy. The first phase of Urban Resolve focuses on exploring the level of situational awareness that can be achieved using the sensor capabilities that we believe will become possible in the next decade. A team of subjects uses these simulated sensor capabilities to monitor enemy activity in the urban Area of Interest (AOI) shown in Figure 1; building up knowledge of the enemy's positions, assets, and activities. Situational awareness is considered to have three levels: perception, understanding, and prediction. The experiment will evaluate how well the test subjects are able to perform at each level. Their capabilities will be explored by varying the sophistication of enemy countermeasures over a set of four trials. This is a discovery experiment conducted at three levels of abstraction. First, concepts are developed in workshops and wargames, then they are refined via constructive simulation, in this case the SLAMEM simulation running closed loop, and then specific scenarios are tried out using a human in the loop (HITL) simulation, composed of the federation of JSAF and SLAMEM. Each level explores the key areas required to implement new urban warfare concepts in greater detail.

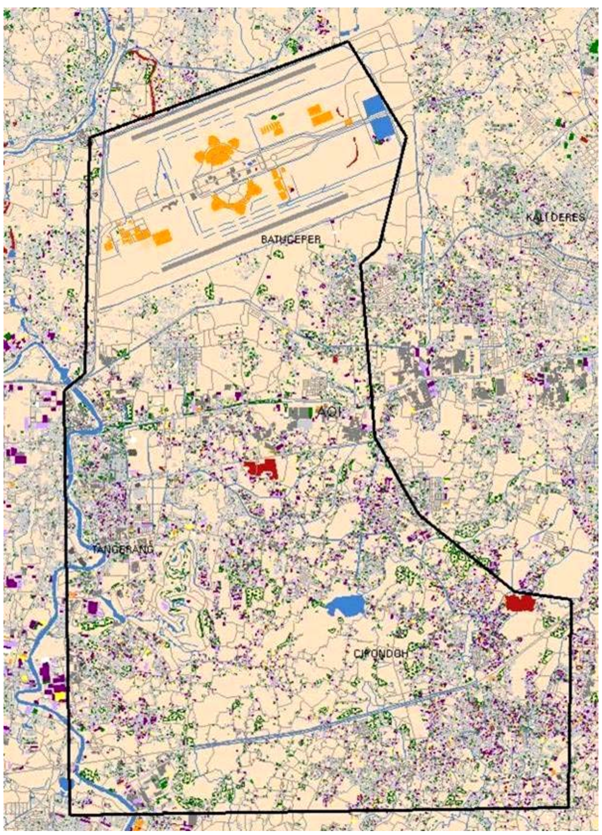

Figure 1. Jakarta AOI

The HITL experiment is a Distributed Continuous Experimentation Environment (DCEE) event. The DCEE (Ceranowicz 2003) was setup by J9 last year to enable an increased pace of experimentation by providing dedicated space, hardware, and network facilities. Urban Resolve takes full advantage of these DCEE facilities. Urban Resolve was also enabled by the Joint Experimentation on Scalable Parallel Processors (JESPP) project (Lucas 2003), which developed ways to use scalable parallel processors (SPPs) to perform large scale entity level simulations. Urban Resolve takes advantage of the High Performance Modernization Program's Huinalu Linux Cluster on Maui. The Urban Resolve experiment builds on the experience of previous experiments at J9, especially the J9901 Experiment (Ceranowicz 1999) where JSAF and SLAMEM were used in a similar configuration to study the potential of future sensors and standoff weapons in a non-urban environment.

THE EXPERIMENT

Phase one of the experiment is situated in Jakarta, due to the availability of urban data for the area. The experiment's subjects man a Joint Intelligence and Fusion Cell (JIFC) that is responsible for monitoring activity in a small section of Jakarta covering roughly one hundred square kilometers including the airport and about twelve kilometers south of it. The JIFC has access to an array of high and medium altitude unmanned aerial vehicles (UAV), low altitude organic air vehicles (OAV) and unattended ground sensors (UGS) plus human intelligence (HUMINT) reports. Of these, the OAVs and the placement of the UGS are under the direct control of the JIFC. The other sensors are controlled by the White cell and the players have to request sensor coverage from those assets. These platforms carry a wide variety of sensors that can be brought to bear on the city.

The target is an insurgent Red force that has taken control of Java from its elected government and is moving into the city to hide its forces and equipment in with the urban population and sensitive sites. Its infiltration into the city is peaceful as a significant minority of the population supports it. As the forces come into the city, they establish hidden fortified positions and anti aircraft sites. They also hide medium range ballistic missiles and weapons of mass destruction. Life in the city proceeds normally during the infiltration, as the Red forces need normal traffic to mask their movements and operations. The Red force is controlled by a team of approximately ten Red operators situated at the Topographic Engineering Center (TEC) at Ft. Belvoir VA. Their operations are controlled by a Red cell, which is in turn controlled by the White cell. Although Red develops and proposes countermeasures to make Blue's job more difficult, the White cell decides which tactics Red can use. They are not considered experimental subjects.

A single operator located at the SPAWAR System Center in San Diego CA controls civilian traffic. While the activities of the civilians are automated and guided by preplanned templates, at some points the operator is required to initiate the formation of crowds and the establishment of exclusion zones. Engineers monitoring the MAUI cluster, data logging and other simulation functions are also located at SPAWAR.

The Blue, White, and Analysis cells are located at the J9 DCEE facility in Suffolk VA and technical control is located in the adjacent J9 Simulation and Analysis Center. Coordination between all sites has been significantly simplified by the use of the Marconi 'ViPr' video conferencing system, which communicates over the same network as the simulation traffic.

Preparation for the experiment started in Sept 2003, with execution starting in June 2004. Four two-week trials will be held from June to October. Preparation for the HITL experiment included a requirements conference, seven one-week developer testing events, two preliminary trials, and a three-week player training session. As is typical in this type of experiment, requirements evolved continuously throughout the development period as JAWP personnel assembled the sensor characteristics and experimented with the C4I displays. The preliminary trials allowed White cell personnel to man the Blue cell and learn how to perform the tasks the players would be asked to carry out. These trials produced many improvements in the system design.

MODELING

The key features of the urban environment for the first phase of the experiment were the dense building environment that occluded sensors and made it difficult to track enemy movements and the civilian population, which Red used to mask its movements. Incorporating these features into the federation caused significant computational loading requiring both optimization of the simulation software and distribution of the computation over many computers including scalable parallel processors.

Urban Terrain

The data for the Jakarta buildings, roads, and rivers was obtained from a commercial source. It was combined with additional coastline, feature, and elevation data. The building data included footprints for approximately one million buildings. However, it did not cover all of Jakarta. This geospecific data was used to create additional geotypical data to fill in the remainder of the city and produce a final database of over 1.8 million buildings. Figure 2 shows the building coverage of the original source data while Figures 3-5 show the resulting terrain database, both from the perspective of an operator at a JSAF plan view display (PVD) and from the perspective of a 3D stealth viewer. The outline of the AOI is shown in the upper left of Figure 3. The target databases were the JSAF Compact Terrain Database (CTDB), the Stealth 3D Visual Database, the dynamic terrain database, and the SLAMEM terrain database. The

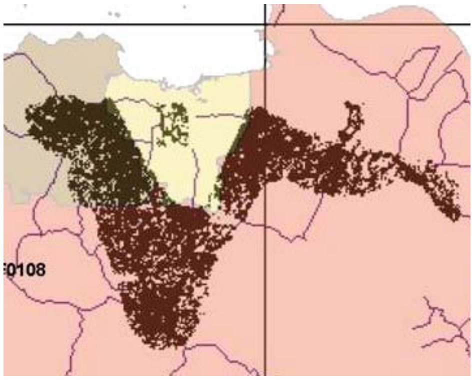

Figure 2. Original Building Data

first three are all constructed via a common process that ensures that they are fully correlated. ArcView Shape files of the linear and areal features were imported into SLAMEM to provide correlated building footprints, roads, and rivers. All the intervisibility calculations are performed in JSAF mitigating issues of terrain correlation between the two simulations. In addition to the building

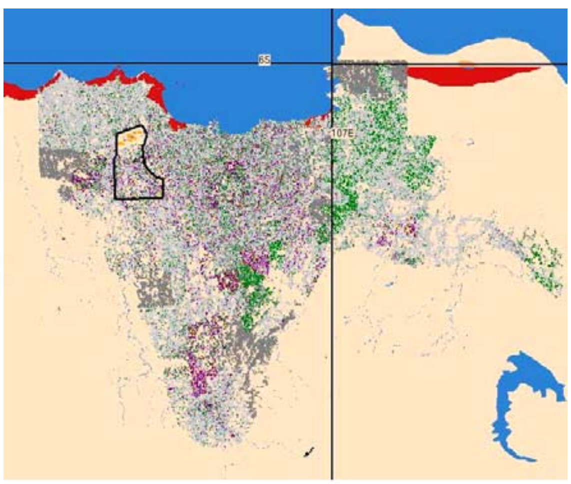

Figure 3. JSAF PVD view of Jakarta buildings

footprints, the source data contained some building names and attributes, but only for major buildings. To fill in the attributes for the remaining buildings, stochastic techniques were developed based on norms for different land use areas. More systemized versions of these techniques are described in Adelson (2004). Although the focus of the experiment is only a small AOI, the database was built prior to this experiment's design as part of our research on using SPPs for Joint Experimentation. It includes all of Jakarta and embeds it in a low-resolution worldwide terrain database that includes the entire world between 75 degrees North and South latitude. To represent this terrain we had to modify the CTDB format; the experiment uses CTDB version 8.7. While there were changes to many areas of CTDB, changes to reduce the size of multiple elevation structures (MES) were especially important. There are two different building representations used in the database. First, there are buildings without interiors, whose walls are extruded from the footprint data. These are extremely space efficient. The second are multiple elevation structures that include interiors,

Figure 4. Aerial Stealth View of Jakarta Database

floors, doors, and other features. The MES format used a great deal of space and the experiment's AOI contains around 47,000 MES buildings while a second area contains 18,000. The MES format was significantly optimized in CTDB 8.7. The terrain database requires approximately half a gigabyte of disk space for Jakarta and one gigabyte for the remainder of the world. JSAF memory maps the terrain into its address space allowing Linux to do the actual paging of the terrain data into memory. Since 1.5 gigabytes is a large fraction of a 32-bit address

Figure 5. Street Level View of Jakarta Database

space, a meta-paging scheme in JSAF dynamically maps and unmaps terrain cells from the address space.

Civilian Population

The civilian population provides the environment for hiding Red movements. In previous experiments we modeled civilian traffic with JSAF entities called "clutter". These low resolution entities drove around the road network along randomly generated routes and occasionally parked. However, for this experiment, we needed traffic whose density would vary throughout the day and would move around the city in a purposeful manner so that it would be hard to distinguish from the Red traffic trying to blend in with it. We created a new class of clutter called commuter clutter for this purpose (Speicher 2004). These clutter entities use timetables to control their movement. Each clutter template (scenario) is provided with a timetable of activities. These activities include going to work, going out to eat, going home and so on. For each activity listed, a time range is specified. Clutter entities randomly pick a time within the range to perform the corresponding activity.

Although modern terrain databases provide highly detailed models of buildings, interiors, trees, vegetation, and weather effects, they are behaviorally barren. In the "The Sciences of the Artificial", Simon (1969) observes that the complexity of an ant's path arises from its environment. So it is with most everyday activities. Our previous buildings provided cover, concealment, and obstacles for combat operations, but did not serve any civilian purpose. If an activity calls for a clutter entity to go and eat, there was nowhere to eat at. To deal with this problem, we attributed all of the buildings in the simulation with function codes, (see Figure 6) including homes, restaurants, houses of worship, offices, stores, etcetera. Although this scheme is simple, it provides the cultural environment necessary to drive commuter clutter. When each clutter entity is created, it selects a nearby building with an appropriate building function code as its home; it selects another building with an appropriate function code as its workplace. When the timetable calls for the entity to go to work, it plans out a route to go to its chosen workplace. If some of the streets are blocked with exclusion zones, the planner finds a detour. When the entity's timetable says it is time to go to a restaurant, it randomly selects a nearby restaurant and drives there. Upon arriving, the vehicle parks and the occupant gets out of the car and

Figure 6. Buildings Colored by BFC Classes

walks into the building. This simple timetable approach allows us to realistically model the ebb and flow of traffic as the day progresses. The experiment uses approximately 100,000 clutter entities in or near the AOI. This includes roughly 25,000 pedestrians and 75,000 vehicles.

Red Forces

The Red Forces are represented primarily by JSAF task frame controlled entities (Ceranowicz 2002). There are several thousand Red entities in the Urban Resolve scenarios. A key requirement was to have the Red forces blend in with the civilian clutter. To do this, the movement algorithms of the Red forces were modified to use the same logic and route planning as clutter when driving down roads. When driving off road, they revert to the traditional JSAF movement algorithms; thus Red road movement is indistinguishable from that of the civilian vehicles. Additionally, the Red forces needed to modify buildings to create fortified positions. External barriers were modeled as entities and placed around fortified areas. To avoid making these prepared positions too obvious, jersey barriers and other construction obstacles were also placed around the AOI in unfortified areas. To represent internal fortifications, we used building properties. An editor was developed that works with the dynamic terrain federate to modify building properties on each federate's copy of the terrain database. This editor allows the Red cell to mark buildings as modified, fortified, or fortified and occupied. Using appropriate sensors, the Blue players can detect these building properties thanks to a modified version of the distributed sensor proxy. This federate responds to sensor footprints by returning information about buildings instead of entities.

Sensors

Sensor modeling is performed by SLAMEM using confusion tables to model the classification and identification process. A confusion table is a matrix (see Table 1) whose rows correspond to the ground truth types of the entities being sensed and columns correspond to the outcomes of the process. A cell of the matrix gives the probability that a sensor analyst will classify or identify the target as the type at the top of the column.

Table 1. A Simple Confusion Table

	SA-15	Truck	Fuel Truck
SA-15	.7	.05	.25
Truck	.1	.6	.3
Fuel Truck	.2	.3	.5

The operators control SLAMEM platforms via an editor in JSAF. SLAMEM (see Figure 7) receives commands from the editor and models the positioning of the platforms and the orientation of their sensors. It translates the field of view of the sensors into footprints and sends those footprints to JSAF. JSAF entities respond by calculating whether they are within the footprint and whether they have a clear line of sight to the sensor. If so, they respond with a detection report to SLAMEM. These reports are slightly misnamed, as they indicate not that the entities have been detected, but that they can be detected. SLAMEM rolls the dice to determine which of the entities returning detections are actually detected. Entities that are detected are then fed through the confusion tables to determine their perceived classification. There are confusion tables for different types of sensors, different sensor ranges, and different times of day. Once a detection has been classified, the SLAMEM fusion center associates the detection with an existing track or decides to create a new track. The tracks are held by the JSAF track database, which sends track updates out to the Blue workstations for monitoring. The tracks are also used by SLAMEM to perform automatic retasking of sensors to maintain existing tracks on interesting targets.

Blue C4I

Each of the Blue players is given a suite of equipment including a JSAF workstation that serves as their primary C2 system. On one screen there is a 2D tactical map of the area, including a display of the positions of their sensor assets, the fields of view of their sensors, the resulting tracks and their histories. A second screen provides editors for examining tracks, commanding assets, and studying the layout of the AOI. The Blue players translate the sensor data presented on these screens into a picture of the situation by creating situational awareness objects

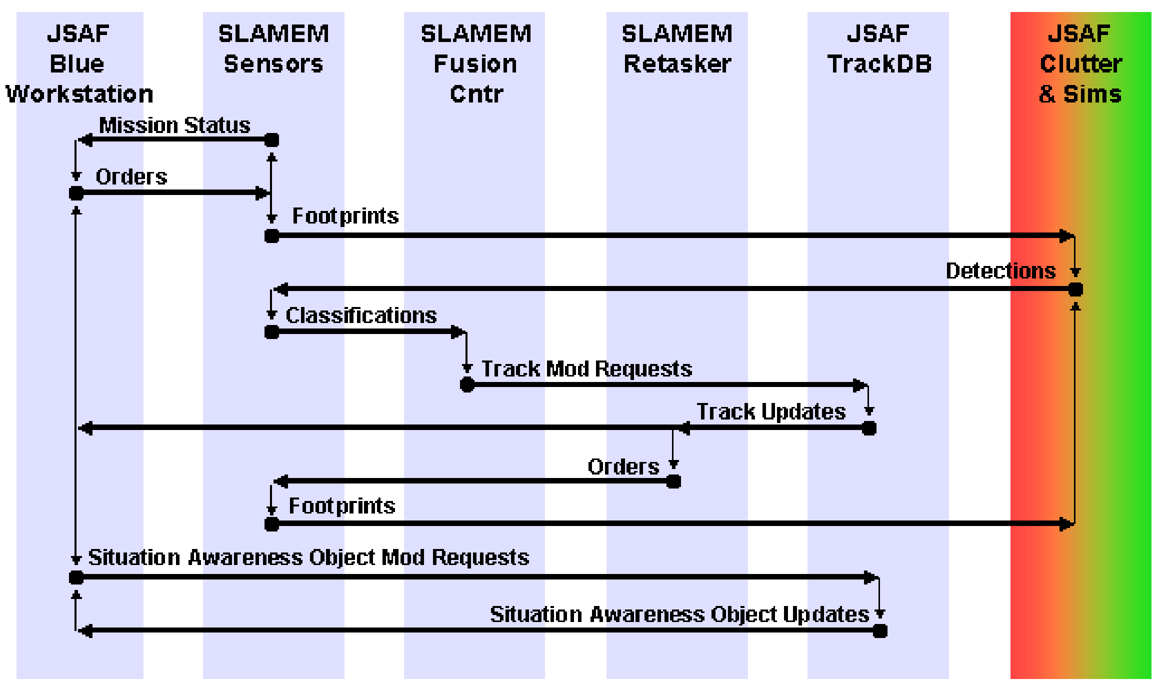

Figure 7. Sensor Control, Interactions, and Data Flow

that indicate what they believe the enemy is doing. A third screen provides access to the InfoWorkspace (IWS), ForceView, and a survey application. IWS is a collaboration tool for interplayer communications. ForceView is an experimental C4I system that displays a 3D view of the AOI. The survey application is used to collect subjective analysis data from the players.

COMPUTATIONAL ARCHITECTURE

Due to the large numbers and high densities of buildings and civilian entities, significantly more computer power was required to run this experiment than previous ones. We attacked the problem through both the optimization and distribution of the computations. Some of the optimization highlights include modifying our intervisibility calculations to avoid premature sorting, adding a maximum structure height in the terrain patch headers to avoid unnecessary checking of building obstructions, ignoring building interiors for external intervisibility calculations, and replacing the JSAF obstacle avoidance algorithm with a grid based route planner derived from a DISAF route planner. These and other changes made a significant difference in the number of entities we could simulate. Prior to the changes, the 933 MHz nodes at Maui were being overwhelmed by the flood of intervisibility calculations required to process SLAMEM sensor footprints and could only simulate several hundred entities without overloading. After these optimizations, they could simulate several thousand.

Distributing SLAMEM

We first used the distributed sensor protocol to distribute some of SLAMEM's calculations in 1999. It allows JSAF processors to do the intervisibility calculations while SLAMEM performs the detection and identification portions of target acquisition (McGarry 1999). However, due to the higher density of entities in this experiment, a single SLAMEM processor was unable to keep up with the load. The sensor entities were distributed to multiple SLAMEM federates, which modeled their movement, sensors, and detection processing. The resulting classifications are sent to another SLAMEM federate (see Figure 7), the fusion center, which produces tracks and track updates. A separate JSAF track database holds the track state and publishes it to the remainder of the federation. Yet another SLAMEM federate, the retasker, controls the automated retasking of sensors to maintain the current tracks. A total of eleven computers were used to run SLAMEM.

Distributing Clutter

Initially, we required over a hundred SPP nodes to simulate clutter. After optimization, the number of nodes required was reduced significantly, but we kept most of the nodes as a safety factor. The key scalability concept for clutter has been to limit its subscriptions to essential information such as detonations and sensor footprints. In particular, clutter did not subscribe to entity updates, so clutter entities have been ignorant of traffic from other simulations and simply drove through each other. For this experiment, we sought to create more realistic clutter movement without the loss of efficiency that listening to other entities would entail. Because each clutter simulator has thousands of entities that can each decide to travel to any part of the AOI, using geographic filtering to limit subscriptions is not effective. The approach developed to control clutter movement is based on deterministic algorithms replicated on every simulator modeling entity road movement (Speicher 2004). Each intersection has a traffic controller for all movement into that intersection. Each vehicle pulling onto a road connected to that intersection sends a message to the traffic controller to register with it. The controller then sends commands to the entity to control its movement as it approaches the intersection and goes through it. The controller takes care of multiple traffic lanes and pedestrian traffic. Because this control algorithm is replicated on each node, each entity receives its commands from the local copy of the intersection controller. However, each copy of the controller listens to all the registration messages for that intersection from the entire federation. The magnitude of this control traffic is approximately 10% of the total entity updates. An alternative approach would be to migrate entities so that all entities controlled by a particular intersection would be on the same computer. However, implementing migration would have been significantly more complicated and could potentially cause computational hot spots if many clutter entities and sensors converge in the same location.

Source Squelching

Because of intersection control, entities are now constantly starting and stopping as they make their way through the city. In addition, the Jakarta road network is much denser and full of turns and intersections than previous databases. This makes entities break their error thresholds much more frequently producing significantly higher update rates and bandwidth requirements than in previous experiments and exercises. In addition to entity updates, our bandwidth budget must also support the intersection control logic interactions, the constant sweeping of sensor footprints, the returning detections, as well as track updates. With widely distributed sites, the challenge was how to balance the network traffic. Early network measurements showed that 5,000 clutter entities generated approximately 7.5 Mbps (Megabits/sec) of network traffic. Extrapolating to 100,000 clutter entities the network bandwidth required would have been 150 Mbps just for clutter updates alone. Since our wide area network links were only capable of supporting around 50 Mbps, we clearly could not simply put all the traffic out on the network.

Runtime infrastructures (RTIs) have long had the capability to perform source squelching. That is, to turn off the publication of simulation data that no one has subscribed to. However, it has been very hard to take advantage of this capability in practice. Most simulations are not significantly distributed, so they do not benefit from DDM and simply use class based subscriptions; listening to all the state for each class of federation object. The worst offenders are loggers that typically try to record everything and white cell displays used by simulation controllers who want to see everything happening in the simulation. These practices defeat source squelching. To make it work for Urban Resolve, we had to implement technical and cultural changes. First, we replaced our centralized logger approach with a distributed design. A distributed logger process runs on each node that has a federate producing simulation state updates. The logger process records any publications the federate produces directly to the local disk drive of the computer the federate is running on. It is implemented using an RTI intercept that allows the logger to catch any publications the application sends to the RTI irrespective of whether the RTI actually sends them out or not. Thus the logger is not a federate and does not generate any subscriptions. One of the advantages of a centralized logger is that the data is collected in a single database that can be queried while the experiment is running. To retain this capability, the distributed logger supports simple SQL queries while it is recording. A tree of aggregators takes a query and delivers it to all the loggers in the federation and assembles all the responses back into a single query response. Naturally, the volume of such queries needs to be limited or the advantages of source squelching will be lost. Each evening, after the conclusion of the experiment day, data from the loggers is downloaded to a mass data storage facility at J9 and is then inserted into a relational database.

The other major problem for source squelching is subscribers that can expand their view and subscribe to too much data resulting in a network overload. If a map display or PVD is subscribed to clutter entities, the operator can zoom out causing subscriptions for all the clutter entities to be turned on; forcing 150 Mbps of traffic to attempt to go through 50 Mbps of network bandwidth. To prevent this, we limited the scale at which operators could view individual ground entities to 1:5000 or a view covering several square kilometers. This was sufficient to maintain network stability while still allowing operators to perform their tasks.

RTI Transports

Just as we had to optimize our simulation algorithms, we also worked to optimize the transmission of simulation data. This was made easier by the RTI we used, RTI-s, which implements all the transport types internally rather than making each application implement them independently. This arrangement allows more flexibility in tuning the transports for a particular experiment. A variety of transports were used along with rules that would switch between transport types depending on the level of traffic generated. For example, an object using a minimum rate transport with a heartbeat of thirty seconds would automatically change to using a state consistent transport if its state was not changing for a while. Similarly the object could change to a faster minimum rate transport if required. Substantial effort went into tuning the transports for different objects and interactions.

Data Distribution Management

In our previous distributed simulation architectures, we implemented data distribution management (DDM) using multicast addressing. Simulation state was divided into interest regions that the RTI associated to multicast addresses. Updates for each interest region were published to the corresponding multicast address. Simulations requiring data from particular interest regions subscribed to the appropriate multicast addresses. The entire simulation state was published to the network but individual simulations could listen to as much or as little of it as they needed. For Urban Resolve, we switched to a DDM implementation based on tagging simulation state updates with an interest vector and using a network of interest management processors (IMPs) to route the information between federates (Helfinstine 2003). Each federate is connected to a single IMP and the IMPs are connected into a tree. A federate's interests are expressed in a sparse bit vector. This interest vector is propagated around the tree so that each federate's output link knows what the listeners on the other side of the link have subscribed to and only those data items matching the subscriptions are sent forward. This implements source squelching both for simulation federates and IMP federates. This ensures that if Maui does not have any federates subscribing to tracks, no track data is sent to Maui. We located our federates so that track updates were confined to the J9 LAN while all Red control information was confined to TEC to minimize the traffic going over the WAN.

One advantage of this method of implementing interest management is the ability to have far more interest regions than are available using multicast addressing on an IP switch. High end CISCO switches like the 6509 have a theoretical maximum of 7000 multicast groups and the rate at which subscriptions can be changed is globally limited. We are using 175,000 interest regions for Urban Resolve, and a federate can change as many subscriptions as it wants at any time.

Another advantage of this approach is that in practice running multicast over WANs tends to be difficult. Since there are often multiple remote routers involved in the network path, it is difficult to get all of them configured properly and maintain that configuration over the course of the experiment. Additionally, since multicast routing is dynamically recomputed, it is not unusual to occasionally drop multicast routes leaving some simulation nodes without state updates. The best reliability we have achieved in multicast based events has been through the use of the MRouteD multicast tunneling program. We used it to connect together sites in the US, Canada, Germany, and Australia for the Multi-National III Experiment. That only worked reliably when the remote MRouteDs were connected in a star configuration to a central MRouteD in the US.

The IMPs allow you to set the transport protocol on each link to TCP, UDP, or rate limited UDP; point to point protocols that are easy to route around the world. The rate limited UDP transport allows you to prioritize which interest regions are dropped first. You can also enable compression on the links to reduce bandwidth. It took us about three test events to balance the traffic on the network. The biggest problem we ran into was overloading the output Ethernet connection of an IMP. Most of our connections were 100 Mbps full duplex and we would get bursts of traffic that exceeded this if an IMP had too many links. So we limited the number of links to around five unless the IMP had a Gigabit connection. We also reduced the bursting problem by using TCP whenever the latency on a link was low. All links on LANs were TCP. For long distance links with high latencies such as those to Maui and California, we used UDP because the TCP bandwidth was too low. It was important to monitor network latency between sites because WAN networks can change at any time. At one point, our connection between J9 and TEC was rerouted causing a very high latency between the two sites. This reduced the TCP bandwidth severely and caused packet loss between the sites.

Another reason to use IMPs instead of multicast is that not all SPPs have IP connections between their nodes. Instead they may use another interconnect technology such as the message passing interface (MPI). An MPI transport layer was added to RTI-s to allow it to run on non-IP based SPPs in the future. We developed two different types of IMPs: tree connected and mesh connected. For this experiment we used the tree connected IMPs, because they where ready first and although the mesh structured IMPs can theoretically provide more bandwidth, in practice our underlying network architecture was tree structured, precluding the benefits of a mesh topology.

Queries

A side benefit of having large number of interest regions was that the number of RTI objects in any particular region is relatively small. This made it possible to use interest region based queries frequently. Conventional class based queries are severely disruptive because the number of objects in a class can be extremely large, so that a large network spike is created when they are used. Object queries are not particularly useful because you need to have discovered the object prior to querying. Interest region based queries are much more practical because they can be used to rapidly discover objects when subscribing to a new interest region. Thus there is no need for the objects to continue publishing if they are not changing. It solves the late joiner problem as long as the late joiner does not subscribe to too much at once. It also reduces the latency between joining a new interest region and finding out about the objects in it.

Federation Configuration

Figure 8 shows the Urban Resolve federation at the end of the player training period. The federation was constantly changing as the experiment developed and computation was redistributed. The top level IMP is located at J9. It exceeds the five connection rule since it has a Gigabit link. The IMPs were primarily limited by their local connections to the switch so the availability of more Gigabit connections would have significantly reduced the number of IMPs required. Clutter was simulated on the MAUI cluster and a small 16 node cluster at J9. All Red entities are simulated at TEC to confine the control traffic to that site. The TEC machines are organized by operator with one IMP connecting the operator's JSAF PVD with his three JSAF simulators. We made several attempts to run the Red simulator federates on MAUI. First, we simply ran the simulators on MAUI and the PVDs at TEC. The Persistent Object (PO) control protocol proved to be too verbose and fragile to support high latency links. PO is currently being rewritten to make it 90% lighter and more fault tolerant. Then we placed both the PVD and the simulators at MAUI and used VNC servers to allow operators at TEC to control them. Unfortunately, for

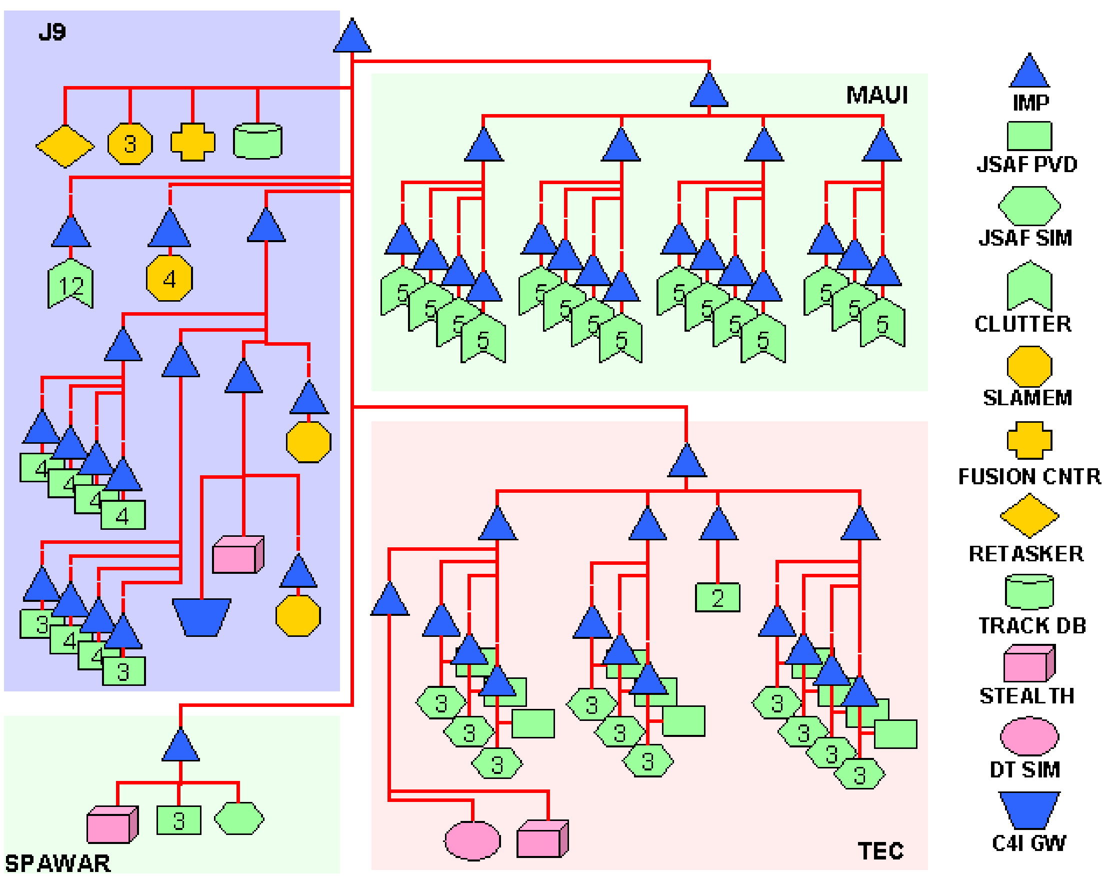

Figure 8. Urban Resolve Federation Snapshot

rapidly changing displays VNC breaks down and reverts to a slow linear scan update. All the SLAMEM assets were located at J9 together with the track database and the Blue player workstations. Figure 8 only shows the simulation federation. It does not show the aggregator hierarchy for the loggers, the data analysis hardware, the non-JSAF C4I equipment, or the networking hardware.

CONCLUSIONS

The dense environment created by urban areas challenges the capabilities of our current combat simulations. It requires significantly more computational power and new and more efficient algorithms. The incorporation of civilians into the simulation underscores how barren our simulations are and the need to incorporate the cultural environment. While this federation was able to satisfy a few of these requirements much more work is needed to represent urban combat accurately.

Significant progress was made in federation scalability during this experiment. Distributed logging showed that it was possible to collect all the data without creating a central bottleneck for the federation. This enabled source squelching. The practical application of very large numbers of interest management regions was demonstrated. This enabled the use of queries to reduce the network requirements of the simulation. A large highly detailed urban area of was represented embedded in a worldwide terrain and a large population of civilian entities was demonstrated responding to a simple cultural environment. To a large extent, this experiment was made possible by the use of an experimental RTI, RTI-s. Our ability to customize the RTI to support new requirements was essential to making this federation work.

ACKNOWLEDGEMENTS

The authors thank J9 and the JAWP for supporting this project, and the many people who contributed to its success. Bill Helfinstine of Lockheed Martin developed the IMPs and the RTI-s enhancements. Dan Speicher of Lockheed Martin developed the clutter and the intersection control algorithms. Ke-Thia Yao, Dennis Allard, Brian Barrett, and Gene Wagenbreth of ISI developed the distributed loggers. Phil Colon of Toyon distributed SLAMEM. Dave Bakeman developed CTBD Format 8. We thank the J9 Experiment Engineering Team for their dedication and the many innovations they produced to make this experiment work.

REFERENCES

Adelson, S., Salemann, L., Farsai, S., Miller, D., Miller, T. and Nakanishi, M. (2004). Complex Terrain Databases for Urban Operations. *Proceedings of the Spring 2004 Simulation Interoperability Workshop,* Paper 04S-SIW-007.

Ceranowicz, A., Torpey, M., Helfinstine, B., Bakeman, D., McCarthy, J., Messerschmidt, L., McGarry, S. and Moore, S. (1999). J9901: Federation Development for Joint Experimentation. *Proceedings of the Fall 1999 Simulation Interoperability Workshop,* Paper 99F-SIW-120.

Ceranowicz, A., Nielson, P. and Koss, F. (2000). Behavioral Representation in JSAF. *Proceedings of the 9th Conference on Computer Generated Forces and Behavioral Representation,* Paper 9TH-CGF-058.

Ceranowicz, A., Dehncke, R. and Cerri, A. (2003). Moving toward a Distributed Continuous Experimentation Environment. *Proceedings of the 2003 Interservice/Industry Training Simulation and Education Conference.*

Helfinstine, B., Torpey, M. and Wagenbreth, G. (2003). Experimental Interest Management Architecture for DCEE. *Proceedings of the 2003 Interservice/Industry Training Simulation and Education Conference.*

McGarry, S. and Torpey, M., (1999). Back to Basics: Balancing Computation and Bandwidth. *Proceedings of the Fall 1999 Simulation Interoperability Workshop,* Paper 99F-SIW-188.

Simon, H., (1969). *Sciences of the Artificial*, Cambridge: MIT Press.

Speicher, D., (2004). Simulating Urban Traffic in Support of the Joint Urban Operations Experiment. *Proceedings of the 2004 Interservice/Industry Training, Simulation, and Education Conference.*