MILITARY SIMULATION
Techniques & Technology

Roger D. SMITH

"Military Simulation Techniques & Technology" is a 3-day training course designed to teach engineers and project leaders the essential techniques required to design, build, and operate a military simulation system.

The outline of the course is:

- Introduction to Simulation
- History of Simulation
- System Architecture & Design
- Interoperability
- Event Management
- Time Management
- Principles of Simulation
- Physical Modeling
- Environmental Modeling
- Behavioral Modeling
- Multi-Resolution Modeling
- VV&A

The topics in this course have evolved over several years of experimentation with different materials. These appear to be universally applicable to all simulation systems and most simulation projects. Each topic has some relevance to both the engineer and the manager.

Course Principles:

- Educate
  - Teach the most valuable and current simulation information available today
- Communicate

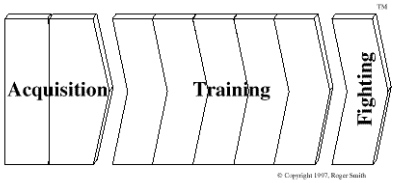
INFORMATION & SECURITY. An International Journal, Vol. 3, 1999, 170-184.

- Present information in a form that can be understood
- Provide materials that are useful in work environment

- Entertain
  - Maintain interest, enthusiasm, and participation
- Inspire
  - Trigger new ideas in the minds of students
  - Serve as the catalyst for innovation

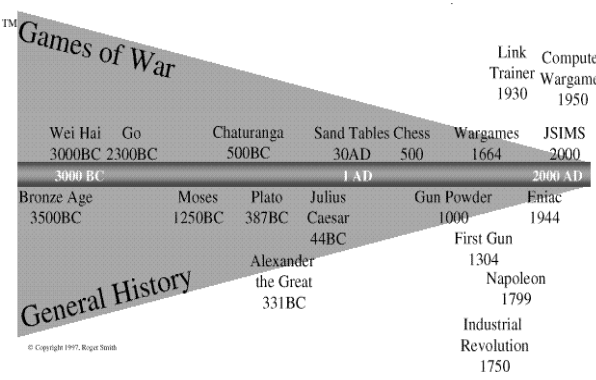
Fighting and Training

"Train the way we Fight" is not correct.
"Fight the way we Train" is more accurate.
Our society and government spends a great deal of time and money in the acquisition and training phases, long before entering into the fighting phase.



The experiences of each soldier during years of training have a direct impact on his or her performance when the fighting actually starts. If we provide inadequate or inaccurate training we are not preparing soldiers for real combat. This could result in the loss of their lives and, potentially, reduced influence of the United States in world affairs.

Games of War: A 5000 Year History

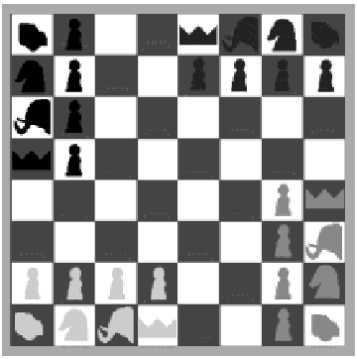


The history of military training goes back 5000 years to the Bronze Age. One may speculate that military training has existed as long as combat, armed conflict, and tribal feuds. The first references we can find to games of war are to Wei Hai around 3000 BC.

This timeline illustrates how long the history of this field really is. War gaming had its start long before the time of Moses.

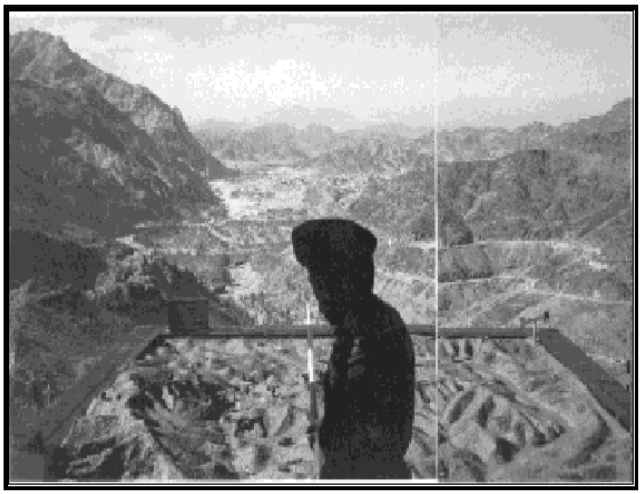
Chaturanga

The game of Chaturanga originates in India around 500 BC and is found in both 2-player and 4-player versions. The name literally means "four limbs", referring to the four branches of the military represented by the pieces. It makes use of pieces that roughly equate to those familiar to us in Chess. The pieces represent the Infantry, Rajah, Elephant, Cavalry, and Chariot (or Boatman). This game is the direct ancestor of Chess.



Originally the piece to move was selected by rolling dice. But Hindu law prohibited gambling and the dice were eliminated to avoid violating this law.

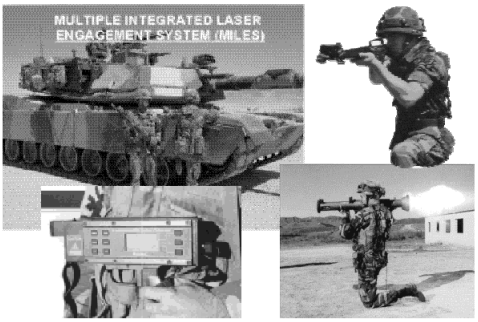
Pakistani Sandtable



This sandtable overlooks the actual battlefield for the Khyber Pass, Pakistan.

From a safe vantage point commanders may experiment in miniature before committing their decisions to actual troops. Once commands are issued, the sandtable can be used to track the execution of these against the planned/expected outcomes.

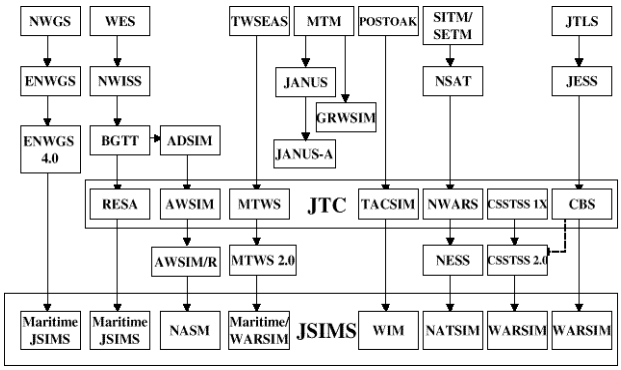
NTC Laser Warfare



The center of NTC force-on-force training is the Multiple Integrated Laser Engagement System (MILES). These are laser devices and sensors attached to each vehicle and soldier (e.g. 8 detectors on a soldier's vest and 4 on his helmet). Laser beams are triggered when any weapon is fired. Microphones detect the sound of an exploding blank round and trigger the firing of a laser. This forces soldiers to load their weapon and limits them to actual magazine capacities.

If this laser beam hits the sensor on another vehicle or human a hit is scored. Killing one of these results in that weapon shutting down so it can no longer fire.

Warfare Model Evolution

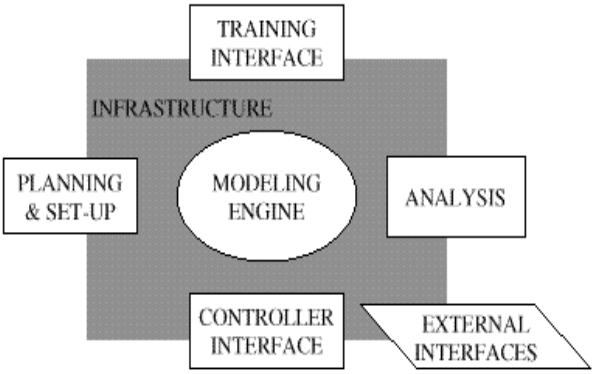


This is the model genealogy beginning in the early 1970's and leading up to the current development of the members of the Joint Simulation System (JSIMS). The Joint Training Confederation (JTC) was an interoperability program that joined together several models that had originally been designed to operate independently. JSIMS is attempting to design the entire family to operate together from the beginning.

Joining models after they are created has proven to provide only a very limited degree of interoperability. Each model has a specific representation of the world that allows it to share/export information in very limited ways.

However, the JTC program proved that interoperability at this level is feasible.

Simulation Architecture

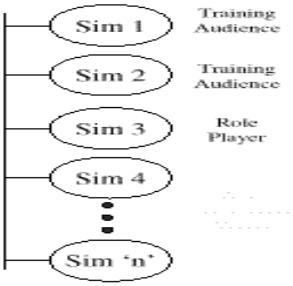


Following the establishment of six major components of a military training simulation, there evolved an infrastructure to tie all of these together. This infrastructure functionality had to be previously embedded in each of the components, usually in different forms with differing capabilities.

1) **Modeling Engine** - The representation of the objects, behaviors, and environments of interest in the training or analysis.
2) **Infrastructure** - Commonality of interfaces with the hardware, operating system, time management, event synchronization, network distribution, etc.
3) **Planning & Set-Up** - The process and tools used to prepare data, software, hardware, facilities, documents, and plans for an exercise.
4) **Controller Interface** - Interface that supports the organization, starting, stopping, and efficient progression of an exercise.
5) **Training Interface** - Interface that provides the training audience's experience of the simulation. This may be special displays and devices, C4I connections, etc.
6) **Analysis** - The process and tools used to organize and study the results of a simulation and the events that have occurred.
7) **External Interfaces** - Interfaces that allow one simulation to operate or exchange information with another.

Distributed Interactive Simulation

- Join virtual-level simulators
  - Tanks, Aircraft, Helicopters, Soldiers, Vehicles
- Provide common digital environment
  - Terrain, Features, Vehicles, Lighting
- Support fair-fight between different types of simulators

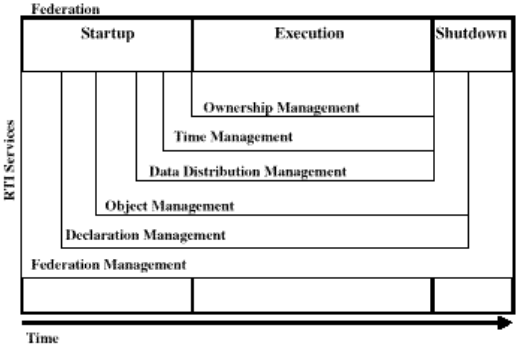


- Detection, Engagement
- Run in real-time
  - No perceptible lag for users

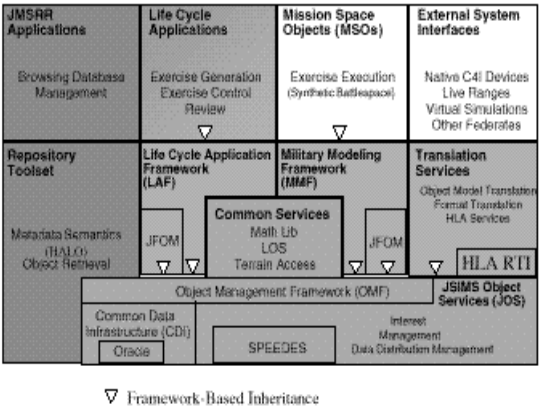
Distributed Interactive Simulation (DIS) is best known for a catalog of message protocols, but it also included extensive descriptions of how to make simulators work together and how to manage a DIS-driven training event.

RTI Service Life-Cycle

The six categories of RTI services are useful during specific phases of an exercise. This diagram emphasizes that most services are needed for both federation startup and execution.



JSIMS Architecture v2.0

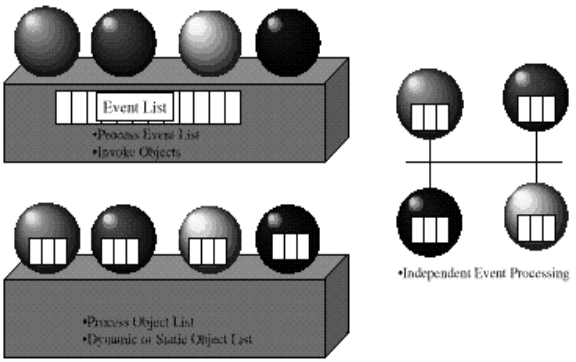


This architecture picture further refines the JSIMS architecture. Oracle has been selected as the Common Data Interchange product. The HLA RTI has been relegated to the role of a Translation Service for communicating with non-JSIMS simulations and other computer systems. Framework-Based Inheritance is specifically identified.

This provides base classes from which high layers inherit

capability. This is unique from the traditional Application Programming Interface (API) in which high layers call functions in lower layers.

Event Management Options



Events may be owned and processed by either the infrastructure or the objects themselves. However, if the objects are synchronized in time the infrastructure must be able to influence the time at which events are processed.

1. A master event list can be held by the infrastructure. The infrastructure has primary thread of control and invokes objects as specified in the event being processed.
2. The event lists can be owned directly by the objects that will execute them. However, the infrastructure must have some control over when the events are processed. This can be accomplished by allowing the infrastructure to process through a list of objects, telling them what time they are allowed to process to. The Object list can be static (everyone all the time), or it can be based on a registration process in which objects schedule themselves for some future processing.
3. Each object can be a separate thread of control and own its own events. These objects may be unconstrained by time, or they may be tracking time through the exchange of time events.
4. Hybrid combinations of all these methods can and are used in real systems.

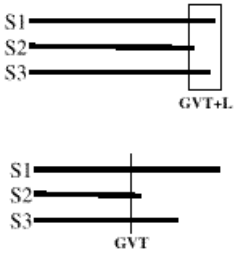
Time Management in DIS

- Paced (real-time or scaled real-time) execution
- Simulation time essentially the same as wallclock time (plus an offset)
- Autonomous simulation nodes, each broadcasts state changes (PDUs) as they occur

- Receiver determines information relevant to it
- Messages typically processed as they arrive (receive order)
- Message latency tolerance bounded by limitations in human perception (typically, up to 100 millisecond delay can be tolerated)
- Receiver may compensate for delay due to message latency.

A member of DIS exercise will receive event messages on the network that have a simulation time stamp slightly in the past of their own system clock. This delay can make the distributed world disjoint. Network delivery speed is an essential factor in ensuring that messages are delivered fast enough that they can be meaningfully applied in the other systems.

Distributed Time Management

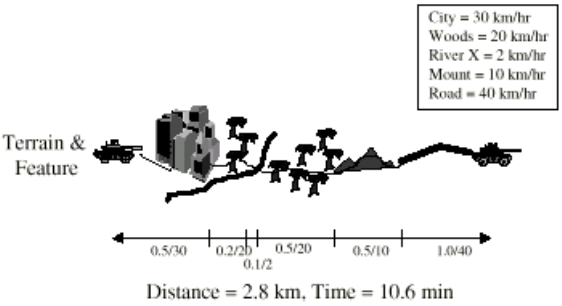


We will explore mechanisms to handle time management in two very different manners. These are Conservative and Optimistic synchronization.

- Conservative
  - Keep all simulations and events synchronized at all times
  - Regulate progress of all by rate of slowest
- Optimistic
  - Allow independent progression, but insist on event synchronization
  - Allow "run forward' and "roll back"

Simple Movement

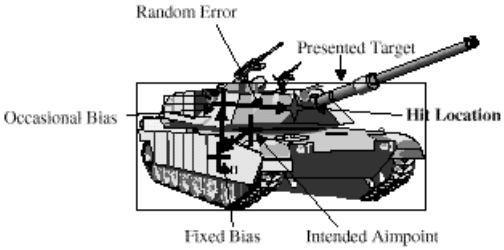
Adding detail to the terrain that the vehicles move on will impose additional time to cover the distance, and will



cause the vehicle to deviate from the straight line route. These deviations require the addition of logic to insure that the objects remain oriented toward their original objective. This is the beginning of the imposition of behavioral modeling in the physical modeling domain.

It is actually very difficult to make clear divisions between physical, environmental, and behavioral models. The current practice is to treat certain factors of each as if they actually belong to the other class. For example, though all decisions for traversing a route are behavioral/cognitive tasks, these are so essential to movement modeling that they are often included in the physical model. At each boundary a decision has to be made about the category for every representation - no matter how arbitrary this decision may be.

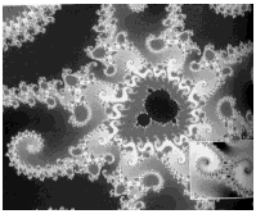
ModSAF Weapon Accuracy



- Fixed Bias - Constant ordnance error
- Occasional Bias - Ordnance error under condition
- Random Error - Round to round dispersion

The accuracy of a weapon in ModSAF plays a part in the PK that is assigned to the engagement. Each target provides some presented area to the shooter. The shooter places an aim point on this vehicle and fires the weapon. However, the engagement is modified by a fixed bias for the type of weapon used, an occasional bias for the variations caused by the positioning of the two vehicles, and a random bias for the round-to-round differences found in any form of munition.

Chaos Theory



- **Traditional Thinking:**

Minor Changes in Input Yields Minor Changes in Outcome

- **Chaos Theory:** Minor Changes in Input Yields Major Changes in Outcome
- Threatens to invalidate predictive power of simulations

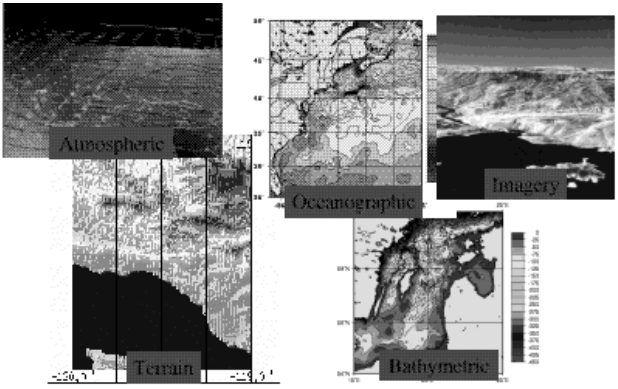
"Very small differences in a system result in very large differences in the behavior of the system."[1]

As modelers we have attempted to insure that input data and model parameters are as accurate as possible. However, we have also assumed that minor errors or variations in this data will only have minor effects on the outcome of the simulation runs. Following the explosion in Chaos Theory in the late 1980's simulation scientists began to question whether our input data could be chaotic variables.

If Chaos Theory is applicable to military simulations this means that minor variations in input data can create huge changes in the outcome of the simulation. At the extreme, this could mean that small changes in the placement of units, slightly different effectiveness variables, and minor changes in the timing at which orders are entered can totally change the outcome of the war.

Since simulations are very complex systems with many interactions between the variables involved, it is possible for two adjacent values to be transformed along totally divergent paths. In practice we believe that most variations dampen out, but the potential does exist for chaotic behavior. This has potentially dire implications on the ability of a model to do analysis and prediction.

Environmental Data

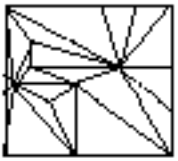


An essential step in environmental modeling is the collection or creation of data to represent the specific type of environment required. This includes information about the terrain, atmosphere, ocean surface, ocean, floor, vegetation, imagery, etc. The list is extensive and the storage requirements are large.

Luckily, this type of data is also essential for modern war fighting. Therefore, the information is available from the same sources that provide it to combat systems. All libraries of data usually go through a transformation, enhancement, or optimization process to prepare them for use in simulations, image generators, and other tools.

TIN

- Triangulated Irregular Networks
- Flexible terrain fitting algorithms
- Reduce sampled data volume, retain higher data accuracy
- Variable resolution based on sampled terrain and features



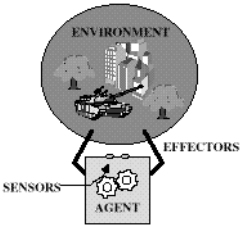
Triangulated Irregular Networks (TIN) are a very efficient and flexible method for storing environmental data. These allow the data modeler to represent the environment at different levels of detail and to transform data between these layers.

When fitting polygons to an underlying data source either of the tessellating shapes described earlier can be used (triangle, rectangle, and hexagon). However, the triangle is unique in its ability to subdivide itself, which the hexagon can not do. Three points are also guaranteed to lie in a definable spatial plane, which is not true for either the rectangle or the hexagon.

Behavioral Modeling

The generation of a decision usually begins with a situation. This enters the model where it is processed and an appropriate set of responses are located.

From these a single decision/action is selected and propagated as the behavior that will be exhibited.



Basic Intelligent Agent

- Encapsulates behavior
- Behavior model is distinct from environment
- Perceives the environment through sensors
- Acts on the environment through effectors

Agents are an encapsulating technique for organizing the behavioral models interactions and effects on the outside world.

"anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors."[2]

"a self-contained software element responsible for executing part of a programmatic process, usually in a distributed environment."[3]

"communicates correctly with peer entities by exchanging messages in an agent communication language."[4]

"makes use of non-procedural process information - knowledge - defined in and accessed from a knowledge base, by means of inference mechanisms."[5]

FSM in Military Training

Vehicle Towing Operation

Start

Enter MOVE State

Move

MOVE until distance closed

Hitching

First switch out of move

Enter HITCH State

End HITCH, enter MOVE

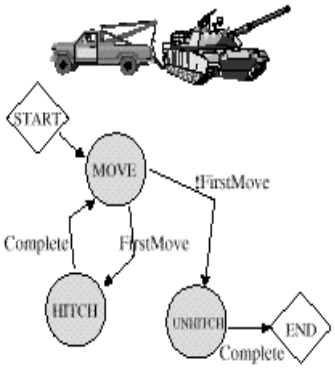
Unhitching

After First switch out of move

Enter UNHITCH State

End

End UNHITCH



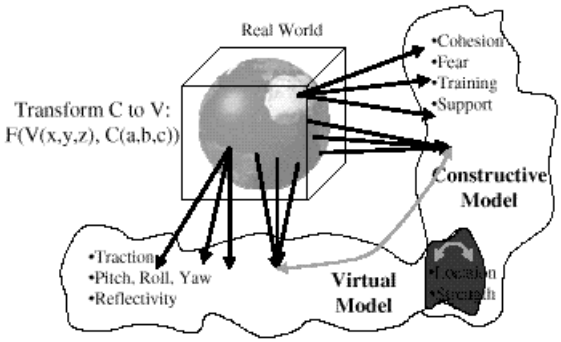
FSM Complete

This is the FSM used within the CCTT system to represent the recovery of a damaged tank by a "tow-tank."

The two-tank first enters a "Move" state which is used to bring the vehicle into the general vicinity of the damaged tank to be rescued. When the tow-tank is within range it switches into the "Hitch" state. This includes all of the backing and positioning to align the tow equipment with the tow point on the damaged tank. This part of the

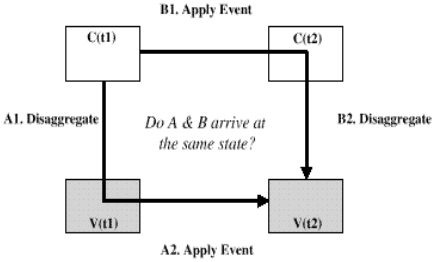
FSM may become very complicated to account to the presence of terrain and other vehicles that interfere with the operation. However, the FSM was selected for the problem specifically because this complexity could be contained and not interfere with the other phases of the operation. Following the "Hitch" state, the tow-tank returns to the "Move" state to drag the damaged tank to a location suitable for repair. Upon arrival at the repair station, the tow-tank enters the "Unhitch" state which models the processes of getting the damaged tank off of the towing equipment and positioned properly in the repair area.

Two World Representations



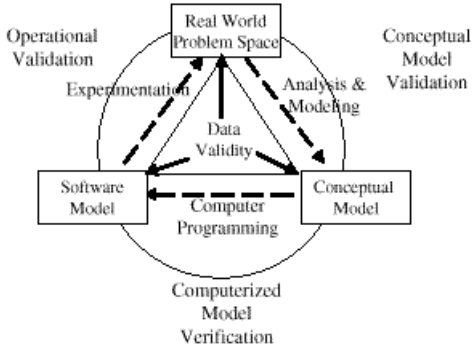
Current MRM prototypes have only touched on the easiest variables to transform from one domain to the other. A small set of variables exist in both models and can be translated more-or-less directly from one form to the other. However, a much larger set of variables are represented very uniquely in each model. When these are transformed to another model you must establish the relationship between them. This relationship may not be obvious or traceable in either model, but may require tracing the variables back to their original source and identifying the relationships between them in that domain (usually in the real world).

Multi-Resolution Model Consistency



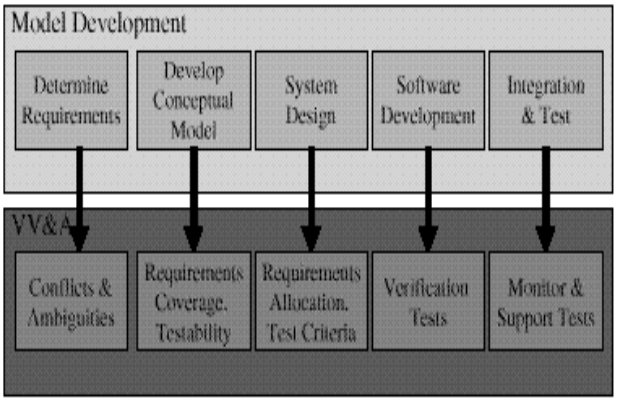
Paul Davis of RAND Corp. pointed out that consistency between the aggregate and entity level view of the world could be tested by running experiments in which one applied the disaggregation operation and then the combat event, then compared this to the result of applying the combat event at the aggregate level followed by the disaggregation operation.

Verification & Validation



The transformation between each of these representations of the system presents a potential for error. Conceptual model validation is conducted to insure that the creation of the conceptual model captured all of the important aspects of the real system. It also insures the proper balance and interactions between objects in the conceptual model. Computerized model verification is performed to insure that the software model is an accurate representation of the conceptual model. The creation of software is a very error prone activity and it is likely that the ideas so carefully crafted in the conceptual model were not accurately captured in software. Operational validation is conducted to compare the software model to the real system. This is the final check to see that the final product does behave in a fashion that is similar and representative of the real world.

Collaborative VV&A



For each step in the development of a simulation system there is a corresponding activity in the VV&A realm. VV&A is not meant as a surprise test that must be passed at the end of a project. It is a continuous guiding light to help lead the developers to a valid solution to the problem presented. Deviations from the appropriate solution should be detected and addressed early by the VV&A practitioners and model developers.

DiSTI Courses

- DiSTI offers courses on:
  - Military Simulation Techniques & Technology
  - Simulation Foundations
  - High Level Architecture
  - Distributed Interactive Simulation
  - Real-time Platform Reference Federation
  - SEDRIS

This and other DiSTI courses are taught regularly throughout the year. They are also available for in-house presentation. Customized versions have been created to focus on specific areas of military simulation. Updated information, schedules and registration forms are available at http://www.simulation.com.

---

1. Henri Poincare, late 1800s.
2. Russell & Norvig, Artificial Intelligence: A Modern Approach.
3. Guifoyle & Warner, Intelligent Agents: The New Revolution in Software.
4. Riecken, Communications of the ACM 37(7), 1994.
5. Guifoyle & Warner, Intelligent Agents: The New Revolution in Software.

ROGER SMITH is the instructor for this course. He has been collecting and refining the content of the course since 1996. The course provides the most current and important technologies for developing simulation systems. The author is an Award Winning simulation developer who is intimately involved in designing, developing, and fielding military simulations. He has contributed to JSIMS, WARSIM, NASM, J-SIGSIM, TACSIM, AWSIM, ALBAM, and F-16 flight simulators. He is a leader in the simulation industry, serving as the Chairman for the ACM Special Interest Group on Simulation, the General Chairman for the Electronic Conferences on Training Simulation, and on the Editorial Boards of *ACM Transactions on Modeling and Computer Simulation and the International Journal of Computer Simulation, Modeling, and Analysis*. He is a Technical Director for STAC Technologies, an Adjunct Professor at Florida Institute of Technology, and a regular lecturer at Georgia Tech and other universities. Address: 424 Research Parkway, Suite 380, Orlando, Florida 32826. Tel.: (407) 206-3390. E-mail: registra@simulation.com.