

## **Dynamo: A Tool for Modeling Integrated Air Defense Systems**

David L Hall<sup>1</sup>  
Pamela Sexton<sup>2</sup>  
Mike Warren<sup>1</sup>  
James Zmyslo<sup>1</sup>

<sup>1</sup>The Pennsylvania State University  
Applied Research Laboratory  
P. O. Box 30  
State College, Pa 16801-0030

<sup>2</sup>Joint Warfare Analysis Center  
18385 Frontage Road  
Dahlgren, VA 22448-5500

### **Abstract**

A continuing challenge for multisensor data fusion systems is the problem of estimating the accuracy or performance of the system. This is especially difficult for dynamic adaptable systems such as modern integrated air defense systems (IADS). These systems utilize multiple sensors (e.g., radar, identification-friend-foe (IFF) systems) to observe airborne targets to develop accurate tracks and to identify the observed targets. Modern IADS have the capability to dynamically reconfigure themselves to account for fault conditions such as the failure of individual sensors or communications links. As a result it is difficult to compute the accuracy of the IADS in situations in which there is dynamic reconfiguration. This paper describes a software system implemented to model IADS. The system, called Dynamo, accurately models the communications links and routing for integrated air defense systems. Dynamo accounts for sensor performance, effects of terrain, and dynamic reconfiguration of the IADS to fault conditions. Near term efforts will focus on modeling the target tracking and identification processes. Planned future capabilities include the ability to account communications delays, data formats, and associated data processing.

# REPORT DOCUMENTATION PAGE

Form Approved OMB No.  
0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 01-01-2001	2. REPORT TYPE Conference Proceedings	3. DATES COVERED (FROM - TO) xx-xx-2000 to xx-xx-2000
---	--	--

4. TITLE AND SUBTITLE Dynamo: A Tool for Modeling Integrated Air Defense Systems Unclassified	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S) Hall, David L. ; Sexton, Pamela ; Warren, Mike ; Zmyslo, James ;	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME AND ADDRESS The Pennsylvania State University Applied Research Laboratory P.O. Box 30 State College, PA16801-0030	8. PERFORMING ORGANIZATION REPORT NUMBER
---	--

9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS Director, CECOM RDEC Night Vision and Electronic Sensors Directorate Security Team 10221 Burbeck Road Ft. Belvoir, VA22060-5806	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT  
APUBLIC RELEASE

13. SUPPLEMENTARY NOTES  
See Also ADM201258, 2000 MSS Proceedings on CD-ROM, January 2001.

14. ABSTRACT  
A continuing challenge for multisensor data fusion systems is the problem of estimating the accuracy or performance of the system. This is especially difficult for dynamic adaptable systems such as modern integrated air defense systems (IADS). These systems utilize multiple sensors (e.g., radar, identification-friend-foe (IFF) systems) to observe airborne targets to develop accurate tracks and to identify the observed targets. Modern IADS have the capability to dynamically reconfigure themselves to account for fault conditions such as the failure of individual sensors or communications links. As a result it is difficult to compute the accuracy of the IADS in situations in which there is dynamic reconfiguration. This paper describes a software system implemented to model IADS. The system, called Dynamo, accurately models the communications links and routing for integrated air defense systems. Dynamo accounts for sensor performance, effects of terrain, and dynamic reconfiguration of the IADS to fault conditions. Near term efforts will focus on modeling the target tracking and identification processes. Planned future capabilities include the ability to account communications delays, data formats, and associated data processing.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:	17. LIMITATION OF ABSTRACT Public Release	18. NUMBER OF PAGES 13	19. NAME OF RESPONSIBLE PERSON Fenster, Lynn lfenster@dtic.mil
---------------------------------	--	---------------------------	--

a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified	19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007
---------------------------	-----------------------------	------------------------------	--

## 1.0 Introduction

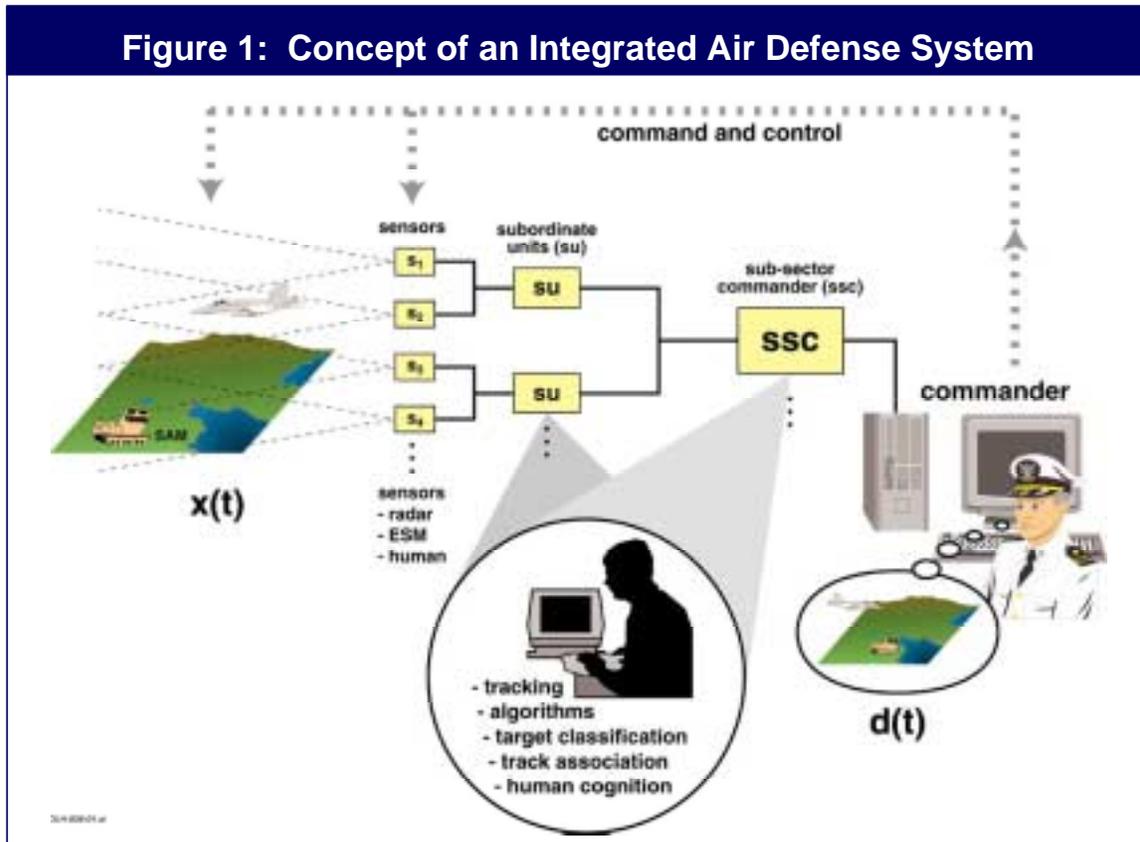
A continuing challenge for multisensor data fusion systems is the accurate prediction of system performance. Typically, fusion systems involve multiple, non-commensurate sensors. In particular, integrated air defense systems (IADS) involve the use of radar and identification-friend-foe-neutral (IFF) sensors to improve the ability to detect, locate, track, characterize and identify objects such as aircraft and unmanned airborne vehicles (UAVs). Several authors including, Hall (1992), Waltz and Llinas (1990) and Hall and Llinas (1997) provide information on data fusion systems. Hall and Linn (1991) performed a survey of data fusion systems.

The concept of an integrated air defense system is illustrated in Figure 1. A battlefield situation (involving aircraft, weapon systems, emitters, etc) is shown on the left-hand side of the figure. Multiple sensors at multiple locations observe the situation and pass information to multiple nodes in the IADS. The sensors include radar, electronic support measures (ESM), human observers, and other sensors. Data from these sensors are passed to subordinate units in the IADS. At these units, initial processing is performed to begin target tracking and identification. Such processing may be performed by a combination of automated algorithms as well as by human "visual" correlation and identification. Typically, track level information is sent to higher level nodes in the IADS (e.g., to sub-sector commanders) who in turn perform track-level fusion and identification. Finally, information is sent to a command post for final processing and ultimately command level decisions for targeting, etc. The command and control ( $C^2$ ) network provides feedback to the sensors and intermediate nodes in the IADS. The concept shown in Figure 1 is merely schematic. IADS may contain tens to hundreds of nodes. Each node may have varying degrees of processing capability and autonomy. The nodes are linked via communications networks, again with varying levels of bandwidth and capability for rerouting (i.e., in an actual IADS, the nodes could have multiple connections to other nodes, unlike the configuration shown in Figure 1).

In Figure 1, the IADS system acts as a highly non-linear filter and *information compression engine* that transforms data about the real situation (denoted by the vector,  $\mathbf{x}(t)$ ), to a representation of the situation (denoted by the vector,  $\mathbf{d}(t)$ ). Here,  $\mathbf{d}(t)$  is a generalized vector that represents all of the target tracks, assigned identities, and raw data fused by the IADS. In general, we are interested in how well the representation,  $\mathbf{d}(t)$ , matches the real-world situation,  $\mathbf{x}(t)$ . For air combat situations it is also important to understand the time delays between  $\mathbf{x}(t)$  and  $\mathbf{d}(t)$ . If the observed aircraft are highly maneuverable, then significant delays could induce a temporal hysteresis effect that would significantly reduce the value of the representation,  $\mathbf{d}(t)$  (viz., we knew where the targets were, but not where they are now). In fact, if there are significant time delays in the IADS process, these delays could result in miss-correlations that would further erode the accuracy of the representation,  $\mathbf{d}(t)$ .

There are several motivations for trying to predict how well a data fusion system will perform. First, such predictions support tradeoff studies related to the selection of sensors, choice of data fusion algorithms, and the optimization of system resources such as computing and communications bandwidth. Second, these predictions support the analysis of how a system would perform in realistic environments. Third, these predictions provide a basis for cost-benefit tradeoffs. Finally, these predictions support the analysis of the vulnerability of a data fusion system (e. g., to the failure of communication links, failure of a sensor node, etc.). Waltz and Llinas (1990) provide a discussion of the evaluation of data fusion systems and discuss the concept of measures of performance (MOP) and measures of effectiveness (MOE). They develop a hierarchy of MOP and MOE to link sensor performance (e.g., probability of detection,

probability of false alarm) with the performance of data fusion algorithms, and ultimately to the accomplishment of a mission.



Extensive research has been published on the topic of performance prediction. Wolf et al (1991) discusses the use of Monte Carlo simulation for analysis of multiple-hypothesis tracking algorithms, and the development of MOP. Jaffee et al (1991) describes the evaluation of correlation and tracking algorithms for electronic intelligence (ELINT). Drury (1991) describes the evaluation of the NATO Iceland Air Defense System. Schweiter and Stromquist (1990) analyzed the sensitivity of tracker/correlation algorithms for multi-target ocean surveillance. Broida et al (1989) presents analytical techniques to estimate the performance of multiple sensor tracking and fire support systems. Belkin (1988) develops predictive models for track-to-track fusion systems and he also presents an analytic model for the effect of false alarms in surveillance tracking systems (Belkin (1987)). Hall (1992) describes the covariance error analysis method for predicting the performance of a data fusion system. Finally, Blackman (1987) presents extensive probabilistic models for target tracking using radar systems.

All of these methods, however, are based on the assumption of a rather static data fusion/observation system. Increasingly, modern data fusion systems are utilizing intelligent sensor management systems to adapt to specific observational conditions (see Denton, et al (1993) and S. Musick (1996)). Modern Integrated Air Defense Systems for example have the ability to reconfigure themselves as a function of mission, the threat environment, and reaction to failure of sensors and communications links. In essence, these systems act as a collaboration of semi-autonomous agents that make local decisions to react to external and

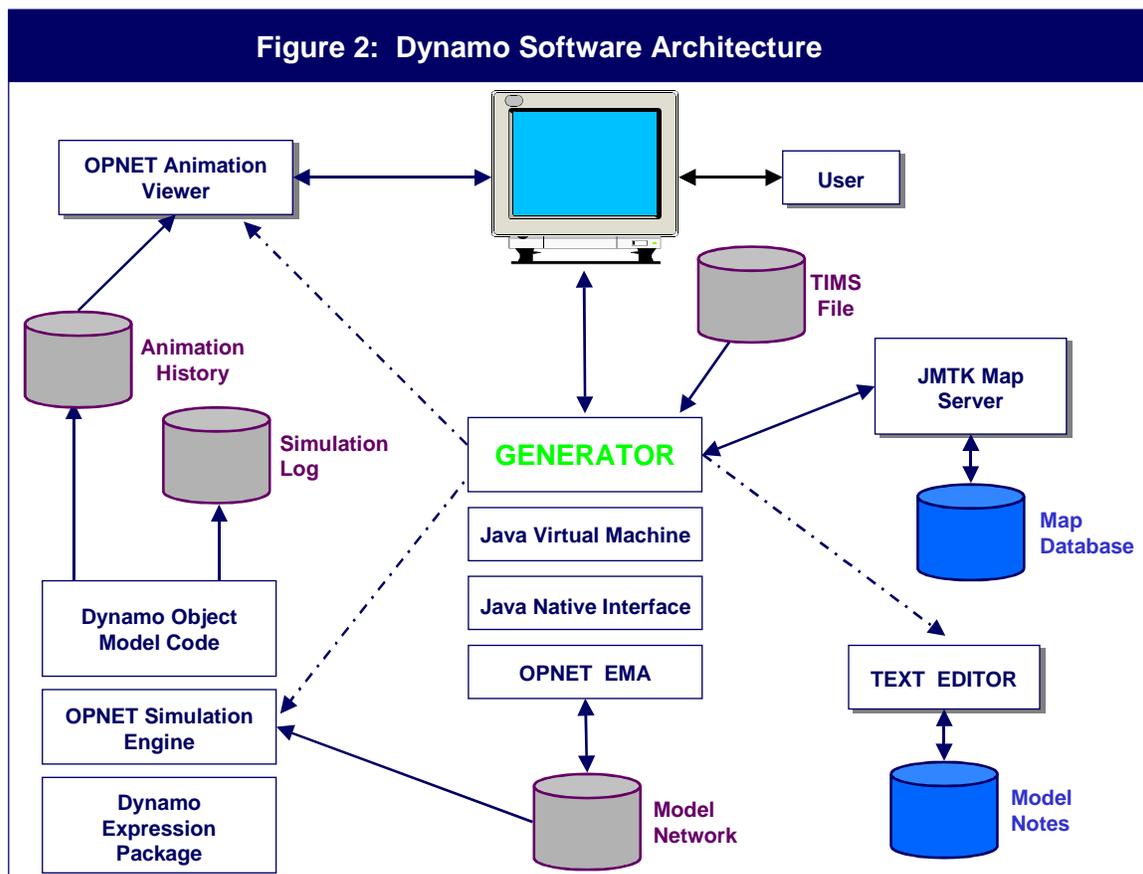
internal conditions. The performance of such systems is very dynamic. In order to accurately model such systems it is necessary to account for effects of sensor performance, the communications infrastructure, fusion processing algorithms, and how the system reacts to fault or failure conditions.

## 2.0 Dynamo: A Performance Analysis Tool

This paper describes a software system called *Dynamo* designed to model an integrated air defense system. The system accounts for the full communications infrastructure, performance of sensor nodes, performance of the fusion algorithms, and the dynamic reconfiguration of the system to account for failure conditions.

### Software Architecture

The basic architecture for Dynamo is shown in Figure 2. Dynamo was implemented using C/C++ and Java (<http://www.javaapplets.com>). The large-scale commercial package OPNET (<http://www.opnet.com>) is used to perform the network communication modeling. In addition, a department of defense (DoD) geographical information system called JMTK (Joint Mapping Toolkit) is used for map displays and overlays. The combination of C/C++ and Java allows Dynamo to be executed on a wide variety of computers and operating system environments. These include Windows NT environments and UNIX-based environments. A key element of the implementation was the development of Java interfaces to the OPNET and JMTK tools.



Key elements of Dynamo include the following.

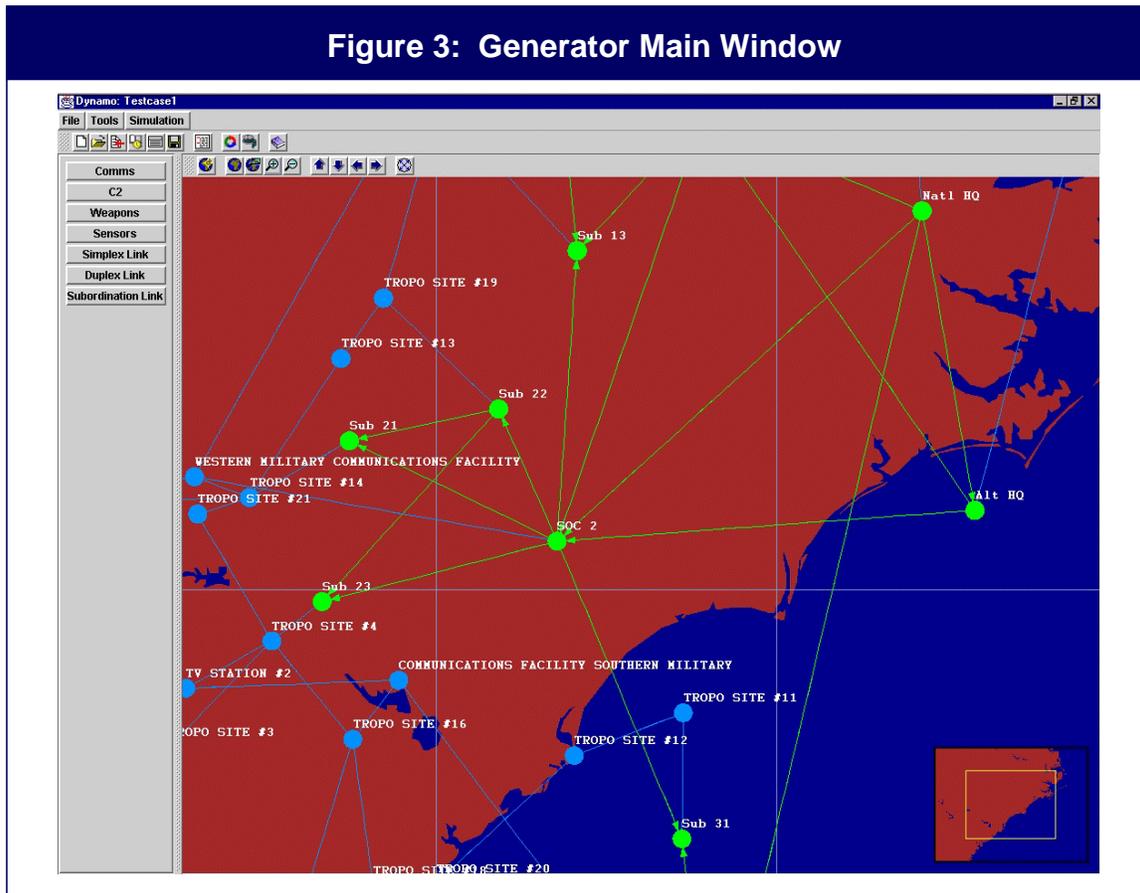
- **OPNET Simulation Engine:** The heart of Dynamo is the OPNET Simulation Engine. This simulation engine uses the contents of a network model to populate the IADS nodes and links used in the simulation. Dynamo network models incorporate custom C language code to provide accurate models of the IADS communications network. The models include dynamic packet routing, event scheduling, and reconfiguration processing. Reconfigurations are modeled as reactions to fault conditions (e.g., failure of a communications link) using a formal logic, which allows specification of the doctrine, associated with reconfiguration actions.
- **Generator:** This Java-based user interface manages the Dynamo OPNET network model file. This component is responsible for creating and modifying Dynamo network models, running OPNET simulations on those models, and running the OPNET animation viewer to view the results of the simulation. Generator was implemented in Java because it is more portable than native code. Java contains an extensive set of interfaces for future expansion including database, Remote Method Invocation (RMI), and CORBA interfaces.
- **Java Virtual Machine (JVM):** JVM is a Java class file interpreter.
- **Java Native Interface:** This acts as a Java bridge between the JVM interpreter and native code libraries.
- **OPNET External Model Access (EMA):** The OPNET native code library is used to manage the OPNET models outside the regular OPNET Graphical User Interface.
- **JMTK Map Server:** JMTK is a DoD networked client/server-mapping package. JMTK provides access to external map databases (e.g., terrain data, geo-political information). It also provides a wide variety of geographical information functions such as map drawings, zoom, and related functions.
- **OPNET Animation Viewer:** An animation viewer is used to read the animation history files. These are rendered into a graphical network representation for use throughout the course of a simulation.
- **OPNET Network Models:** At the time of a simulation, all network objects are contained in an OPNET network model. In general this model contains the geographical position of nodes, simplex links, and duplex link definitions. Information about each node's attributes, hierarchical relationships, and characteristics is captured in the network models.

### *Model Selection and Specification*

The basic function of Dynamo is to allow the creation of an IADS model and the evaluation of the IADS performance against hypothesized scenarios. The Dynamo user begins by using the generator function to create the IADS model. This is done using a graphics user interface (GUI) shown in Figure 3. The user specifies the IADS nodes and interconnectivity using a *drag and drop, point and click* type of operation. The generator GUI supports functions such as:

- New model generation
- Edit of existing models
- Importing external database information
- Creation of temporal scenarios
- Creation of IADS nodes and networks
- Running a simulation

The user can create an IADS node, specify the node characteristics, and link the created node to one or more existing nodes. The specification of an IADS node involves specifying the node's geographic location (including altitude), establishing the node's condition, and describing the node functions. These functions can include data creation (e.g., a sensor node), data transmission and routing, and data processing. The latter includes track processing, and the ability to perform target assessment.



Having specified an IADS node, the user can link the node to one or more other nodes. Specifying a node source and destination for each link performs this. A very general communications model can be specified. Network node elements include the following.

- **Source information:** A node specified as a source generates packet data at regular intervals. The user will ultimately be able to specify the format, characteristics, and even the contents of the packet data.
- **Sink information:** A node specified as an information sink effectively ingests or destroys packet information (an example would be a processing node). The sink destroys the packets that have been successfully delivered to their destination or cannot be forwarded because a communications link or route has experienced a failure condition.
- **Point-to-point transmitters and receivers:** These elements connect network nodes to other network nodes. Each network node has sixteen point-to-point transmitter/receiver pairs. Each simplex link uses either a transmitter or a receiver. Each duplex link uses one of each. Simplex and duplex links may be connected to the same node.
- **Addresser (Addr):** This element receives packets that are generated by the source. If any other network node (through a subordination link) has requested data, the addr processor addresses a copy of the packet and forwards it to the router for transmission. Otherwise it forwards the packet to the sink for destruction.
- **Router:** The router element accepts incoming packets, calculates the next hop in the packet's route, and forwards the packet to the appropriate transmitter.

### *Scenario Specification*

Having created an IADS network model, the user can evaluate the performance of the IADS system against hypothetical scenarios. These scenarios are created using a natural language interface that supports a narrative type of description of actions and conditions experienced by the IADS system. The scenario description specifies a sequence of operational conditions and political conditions experienced by the IADS. In addition, the user can specify the reactions of the IADS to these political and operational conditions.

Political conditions include doctrinally or politically determined circumstances that change the command structure of the IADS, the level at which decisions are made, manning, communications priorities, and the operation of alternative sites. There are two classes of political events, (1) external events and (2) internal events. These are summarized below.

- **External political conditions:** These conditions are called *defense conditions* or DEFCONS in U. S. DoD terminology. External political changes exist for all nodes in the system, regardless of their operation. A DEFCON change results in changes throughout the IADS network.
- **Internal political conditions:** Internal political conditions are called *readiness states*. A readiness state may be applied to a specific site or group of sites, usually by a local commander. These affect local operations rather than theater or global operations. Readiness states may also be set as a result of a specific DEFCON.

Operational conditions are the familiar conditions that result from: (1) the maintenance or physical status of the nodes or inter-nodal links, and (2) the hierarchical relationships

determined by the DEFCONs, readiness states, and physical states. These are summarized below.

- **Physical states:** The physical states of the node (and nodal links) are modeled as simple Boolean on/off condition states. These functions include the ability to transmit information, and the ability to process, assess, or assign tracks. In some cases the capability may physically exist, but some event prevents that event from occurring. For example, a node may have the physical capability to assess and assign tracks; however, a lack of communications may prevent it from exercising those capabilities.
- **Hierarchical relationships:** Hierarchical relationships are organizational in nature. Hierarchical changes involve re-subordination, generally in response to DEFCON changes or changes in readiness states. However, physical states in adjacent nodes and links can also cause changes in the organizational hierarchy.

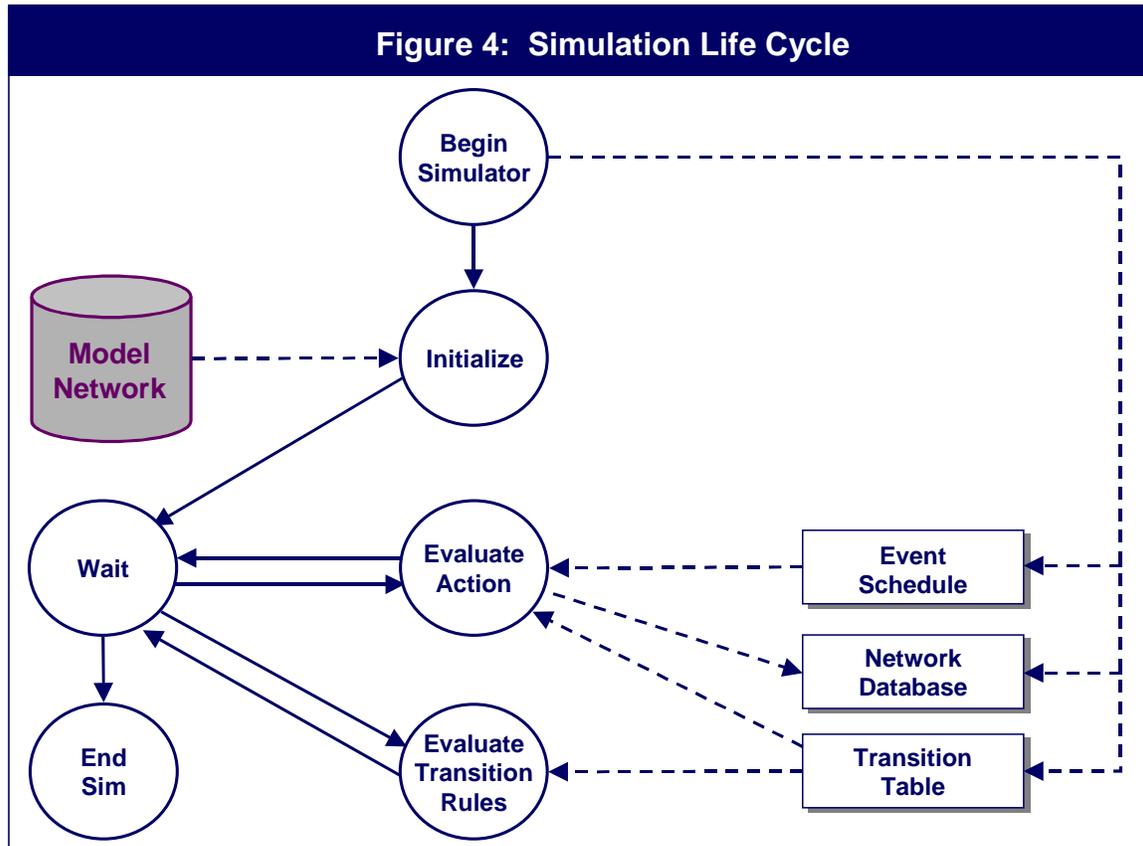
The Dynamo model provides the capability to model all of these conditions and responses. Subordination and routing relationships are defined in models via a routing table. Routing and re-subordination tables change dynamically with changes in the network. The response of the IADS and IADS nodes to these conditions can be defined in a very general way using the rule-based knowledge representation scheme incorporated into Dynamo.

### *Scenario Execution and Performance Evaluation*

Having specified a model for the IADS, and hypothesized a scenario of internal and external events, the user is prepared to run simulations to evaluate the IADS performance. The Dynamo simulation life cycle, represented in Figure 4, is event driven based on a pre-specified simulation script. All processing is initiated from a wait state in response to a simulation event (specified in the script).

The basic steps in a simulation life cycle are the simulation start, instantiation of a scheduled event (specified by the simulation script), modeling the network changes in response to the event, and the simulation end. These are summarized below.

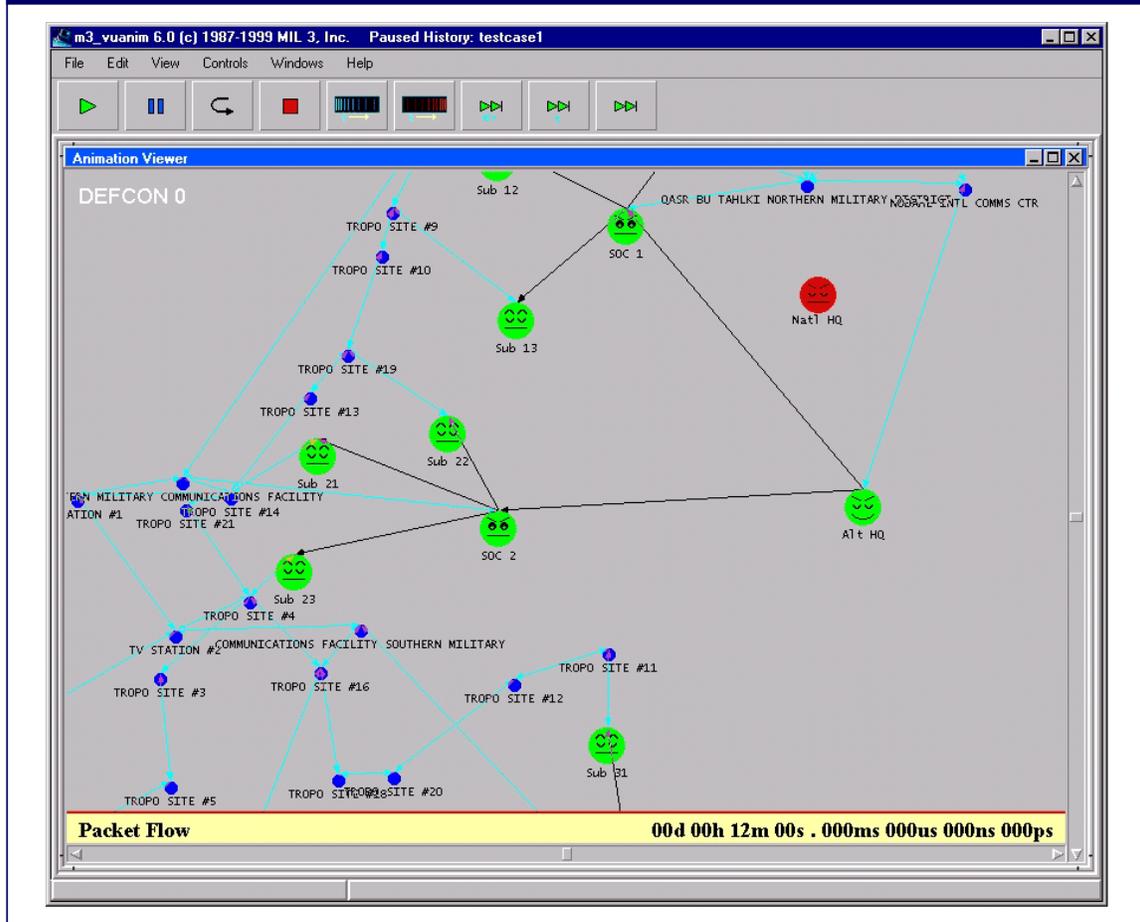
- **Simulation start:** On simulation startup, the network model file is read and the simulation event schedule, network database, and transition tables are initialized.
- **Scheduled event:** When the pre-specified amount of simulation time has elapsed, actions in the simulation event schedule are fired and evaluated. The results of the action may modify the contents of object attributes in the network database.
- **Network change:** When the attributes of network objects change, the Dynamo simulation initiates a re-configuration cycle. During this cycle, each rule in the current transition table is evaluated. If the rule is satisfied, each action in the transition rule is evaluated. The results of evaluating those actions may further change network object attributes.



- **Simulation end:** When all pending simulation events and network re-configuration cycles have been completed, the simulation terminates and information is collected for post-simulation analysis.

During the simulation, the Dynamo display shows the on-going communications in the network (via active icons to show network flow) and any changes in the configuration. An example of an instantaneous display is shown in Figure 5. Note the use of Chernoff faces to illustrate the capabilities of the nodes. The Dynamo user has the capability for extensive control of the simulation process including varying the simulation time rate, stopping the simulation after specified events and re-configurations, etc. The animation or playback time is a function of the input of events (e.g., animation, requests, and changes). The Dynamo viewer allows a user to control the inter-event time delays, which affect the apparent playback speed.

**Figure 5: Example of an Instantaneous Dynamo Display**

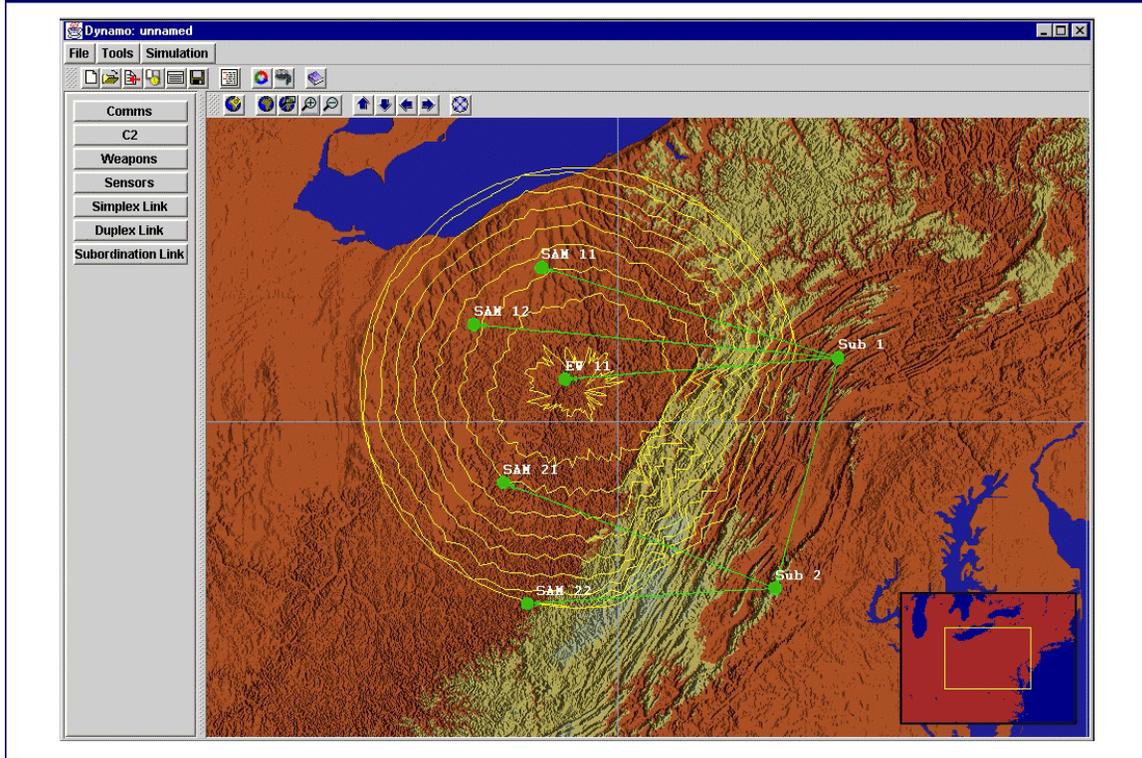


### 3.0 Sensor and Fusion Process Modeling

We have previously described the dynamic model used for the IADS communications network. However, it is still necessary to model the sensor performance and the performance of the data fusion processing. Both of these types of models can be incorporated into the Dynamo communications network model by treating the sensor performance and the fusion algorithms as transfer functions associated with the IADS nodes.

The current focus for Dynamo is to model the performance of radar and the effects of terrain on signal propagation. Extensive research has been performed to develop analytic models of radar performance. Farina and Pardini (1980) provide a survey of techniques. Dynamo uses a variation of the radar equation and Sterling's expressions for modeling the probability of detection for a specified target at a specified sensor-to-target range. In addition, the effects of terrain on the signal propagation are modeled. Figure 6 shows an example of the detection contours for radar in terrain as a function of target altitude. Additional sensor performance models are currently being implemented.

**Figure 6: Example of Terrain Effects on Radar Detection**



Models for the track-to-track data fusion are currently being implemented in a separate computer program named BORG (Bounded Operational Re-Grouping). The models are based on the covariance error analysis approach described by Hall (1992). A key issue in the uncertainty modeling will involve the representation of the effects of delays in the communications network. These delays translate into increases in the uncertainty of the state vector (as it is propagated forward in time by the equation of motion) and potential biases. Effects related to miss-correlation induced by the combination of time delays and target maneuvering are still under investigation. Other effects include issues such as the correlation between tracks generated by multiple nodes in the IADS for the same target. This modeling effort represents work still in progress.

#### 4.0 Summary

This paper has described a dynamic simulation capability to accurately predict the performance of integrated air defense systems. The Dynamo model allows a user to specify an IADS and to execute simulations using a scripted set of political and environmental events. The model provides a means to predict the performance of the IADS communications network, the sensors, and the data fusion processing. Dynamo accounts for potential failure conditions in the IADS communications system and nodes, and models the dynamics of the IADS reconfigurations in response to these conditions. This tool kit should be useful in determining how well an IADS can detect, characterize, track, and identify tactical entities such as aircraft and UAVs. In addition, the tool kit can be used for analyzing the vulnerabilities of an IADS to failure conditions.

## 5.0 References

- D. L. Hall, *Mathematical Techniques in Multi-Sensor Data Fusion*, Artech House, Inc., Norwood, MA, 1992.
- E. Waltz and J. Llinas, *Multisensor Data Fusion*, Artech House, Inc., Norwood, MA, 1990.
- D. L. Hall and J. Llinas, "An introduction to multisensor data fusion", *Proceedings of the IEEE*, vol. 85, No. 1, January 1997.
- D. L. Hall and R. Linn, "A survey of data fusion systems", *Proceedings of the SPIE Conference on Data Structures and Target Classification*, April 1991, Vol. 1490, Orlando, Florida, pp. 13-36.
- R. Jaffee, S. Thode, J. A. Gauss, and F. A. Perry, "Evaluation of correlation and tracking algorithms for transition from research and development to production", *DFS-91 Proceedings of the 1991 Joint Service Data Fusion Symposium*, 7-11 October 1991, pp. 483-501.
- T. O. Wolff, and D. Fixsen, "Standards, metrics, benchmarks, and Monte-Carlo: evaluating multi-sensor fusion systems", *DFS-91 Proceedings of the 1991 Joint Service Data Fusion Symposium*, 7-11 October 1991, pp. 393-446.
- J. L. Drury, "Evaluating automatic track-to-track correlation (ATTC) algorithm performance in a stressful track data fusion/dissemination environment", *DFS-91, Proceedings of the 1991 Joint Service Data Fusion Symposium*, 7-11 October 1991, pp. 524-536.
- G. A. Schweiter and W. R. Stromquist, "The effect of sensor quality on tracker/correlator performance", *DFS-90 Proceedings of the 1990 Joint Service Data Fusion Symposium*, 15-18 May 1980, pp. 198-222.
- T. R. Broida, L. R. Dean, T. T. Nguyen, R. A. Ribas, "Performance prediction and structured branching for multiple sensor tracking systems", *DFS-89, Proceedings of the 1989 Tri-Service Data Fusion Symposium*, Laurel, MD, pp. 195-202.
- B. Belkin, G. A. Schweiter, and R. A. Wenocur, "Predictive models of false alarms", *DFS-88, Proceedings of the 1988 Tri-Service Data Fusion Symposium*, Vol. 1, 17-19 May, 1988, Johns Hopkins University Applied Physics Laboratory, Laurel, MD, pp. 340-346.
- B. Belkin and R. Wenocur, "An analytic model for the effect of false reports on surveillance tracking", *DFS-87, Proceedings of the 1987 Tri-Service Data Fusion Symposium*, Vol. 1, 9-11 June 1987, Johns Hopkins Applied Physics Laboratory, Laurel, MD, pp. 635-640.
- S. Blackman, *Multi-target Tracking with Radar Applications*, Artech House, Inc., Norwood, MA, 1987.
- R. V. Denton, E. I. Alcaraz, J. P. Knopow, J. Llinas, and K. J. Hintz, "Towards modern sensor management systems", *DFS-93 Proceedings of the Sixth Joint Services Data Fusion Symposium*, Vol. 1, 14-18 June, 1993, pp. 659-678.

S. Music, "Comparison of sensor management strategies for detection and classification," *Proceedings of the 9<sup>th</sup> National Symposium on Sensor Fusion*, Vol. 1, 12-14 March 1996, ERIM, Ann Arbor, MI, pp. 367-394.

<http://www.javaapplets.com/frmain.htm>

<http://www.opnet.com>

A. Farina and S. Pardini, "Survey of radar data-processing techniques in air-traffic-control and surveillance systems", *IEE Proceedings*, Vol. 7, No. 3, June 1980, pp. 189-202.