Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A SCENARIO GENERATION TOOL FOR DDF SIMULATION TESTBEDS

G Srimathveeravalli
N Subramanian
T Kesavadas

Virtual Reality Laboratory
809 Furnas Hall
State University of New York
Buffalo, NY 14260, U.S.A

ABSTRACT

An interactive tool has been developed for visualizing and creating scaled battlefield based scenarios for use in a simulation testbed to develop and test distributed data fusion and ad-hoc networking algorithms. This paper discuses the design requirements and implementation issues for developing such a tool. Two main design goals were to enable design of complex scenarios in an intuitive and easy fashion, and provide a complete set of decision support utilities. The tool, called *SceneGen*, supports 3D visualization for creating scenarios and overcoming challenges including geospatial (GIS and Terrain) data management, entity information management and waypoint/path specification. The scenario generator includes a number of sensor and target models and provides database support to manage different fusion and network algorithms. This tool was implemented and tested successfully with several sample scenarios.

1 INTRODUCTION

The concept of Network Centric Warfare (NCW) is the centerpiece of the US Department of Defense's Joint Vision 2020 approach to modern-day and future warfare. This concept involves a multiplicity of platforms and sensors observing many targets in a battlespace, these platforms interconnected by a communications infrastructure in order that a common operating picture (COP) can be developed for shared understanding and decision-making by friendly commanders. The key building block of NCW and a step to producing a COP is distributed data fusion (DDF).

Achieving this vision involves development of new distributed fusion and networking algorithms, evaluating network designs, testing effect of fusion algorithms, developing methods for information transfer and analysis of post-battle results. The first step in this development cycle is acquisition or creation of data for input, testing and validation of these algorithms. Utilizing real life battle scenarios for data is unsuitable as data acquired is not ideal and would require refinement to suit development and testing of distributed fusion concepts. Hence at a design and development stage there is a need for a reliable data source that would provide realistic data, where users would have control over system variables and sensitivities that would allow the algorithm developers to create specific test scenarios.

Our group at the University at Buffalo is involved in the development of a DDF simulation testbed. This paper discuses the visualization and simulation support for this DDF testbed. Presently the important modules of this testbed include (see Figure 1), This paper reports on the design experience and the implementation issues faced during the development of the *scenario generation* tool that facilitates generation of test scenarios for use in this simulation testbed.

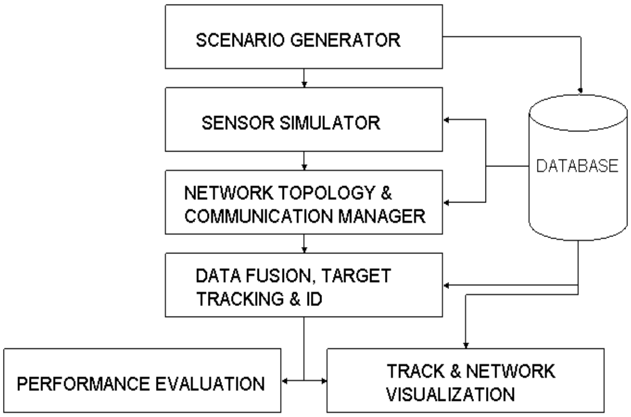

SCENARIO GENERATOR
SENSOR SIMULATOR
NETWORK TOPOLOGY & COMMUNICATION MANAGER
DATA FUSION, TARGET TRACKING & ID
PERFORMANCE EVALUATION
TRACK & NETWORK VISUALIZATION
DATABASE

Figure 1: The Scope of Scenario Generation in the DDF Testbed Developed at University at Buffalo

The features of the testbed include,

- Scenario generation tool
- Sensor modules & Suite of target motion models
- Tracking & Distributed fusion algorithms
- Dynamic network topology managers
- Human centric information visualization set up

2 MOTIVATION AND PROBLEM DESCRIPTION

A DDF simulation testbed needs multiple scenarios covering specific problems to evaluate different algorithms and architectures in battlefield simulations that emulate reality. However, creation of these realistic scenarios is a laborious process and involves various problems. Creation of one such scenario for use in a simulation testbed would involve placement of several sensors and targets on a digital terrain model, specification of the attributes of these sensors and targets, specification of trajectories (waypoints) of these entities and if necessary an initial communication network topology. As the fidelity of each of these sensor and target models increases, the effort involved in specification of custom behavior for these sensors and targets increases several fold. Scenario scaling, which is another factor, encompasses the number of targets and sensors (few to an entire battlespace with many hundreds of them), terrain size (from a small county to an entire state or country) and time over which the simulation needs to run (few hours to few days). Another issue is the need for a tactical 3D view of the scenario. The problem of target tracking is highly geospatial; factors such as line of sight affect the performance of the sensor. Hence choice of sensor placement in a 2D setting is difficult as the user cannot evaluate the tactical significance of the position. Creation of scenarios randomly does not assure that the data generated will be adequate to test the algorithms sufficiently.

These factors led to the development of a dedicated scenario generation module with the aim of reducing the drudgery of scenario creation while creating best possible data for the simulation testbed. The following design requirements were identified for a scenario generator to support a DDF simulation testbed:

2.1 Definition of Scenario/Entity Behavior

The scenario generator should allow selection of entity or platform models from an existing source and should support the creation and definition of new types of models. In addition to this there should be means for defining the behavior, i.e. sensing and motion behavior for an entity. Prior to the creation of the scenario the user should be able to define simulation variables like simulation time of the scenario, the time step interval, simulation type and algorithms to be used. When completed, the scenario created should provide all possible data that may be required to run the simulation successfully.

2.2 Terrain Database Creation

The scenario generator should support creation of a terrain database for use in the simulation. As mentioned earlier, the algorithms are affected by geographical factors, especially the terrain. Hence simple 2D GIS data for scenario creation will prove to be insufficient. There should be a provision for the user to load up a file providing GIS and 3D terrain information for desired locations. The depiction of the geography as 3D with overlaid with 2 dimensional geographic information will aid in tactical and strategic decisions made during creation of a scenario.

2.3 Interactive Scenario Creation

The creation of a scenario should be easy, interactive and intuitive. The scenario creation process should involve least possible steps from start to end, while giving the user easy access to all associated data through a user interface. The user should have the option of entering the scenario in an automated or manual process. In the automated case, the scenarios will be generated by simply selecting certain parameters, while in the latter case the user would create the scenario one sensor or target at a time through a user interface.

2.4 Scenario Size and Re-Usability

There is a great monetary and time cost involved in the development of scenarios; hence there should be a provision for the reuse of scenarios. This means that the scenario generated should be saved and made available for later reuse and edits. The size of the scenario should not be a limiting factor during the creation of scenarios. In most situations due to the amount of data generated during process of scenario generation means that database support must be provided for effective storage and maintenance of the scenario data. The database support should allow; (i) storage of entity behavior and model information and (ii) storage of data generated during the simulation itself.

2.5 Visualization and Representation of Tactical Information

The scenario generator should provide the user with a capability to navigate and interact with the scenario easily. The scenario should cater to both two and three dimensional information representation. The scenario generator should provide the user with information at a rate that is easy for the user to absorb. The scenario generator should provide the flow of information in an easy to understand fashion. The scenario generator should be able to deliver tactical information to the user.

3 RELATED WORK

There has been extensive work in the general area of visualization support for simulations and more specifically in the area of battlefield visualization. The number of groups conducting dedicated research in the field of DDF is few and the people working on providing support for this type of visualization, fewer. A survey of existing work with regard to generation of data/scenarios for DDF algorithms revealed that the COTS software available did not match all the requirements listed in the previous section. FLAMES™ (FLAMES 2004) is a commercial military scenario generator. This scenario generator supports incorporation of sensor and target modules through the usage of scripts. It incorporates a terrain database generator and also supports consideration of terrain masking characteristics in the scenario. (Nicholson, Lloyd, Julier and Uhlmann 2002) and (Svensson and Horling 2003) have used this software for generating data (and also simulation) for their applications. From their experience and from the capabilities that FLAMES provides the importance of geospatial capabilities and the 3D visualization is reinforced.

FbSim™ is an information fusion simulation testbed that offers limited scenario creation support. This is one of the earliest simulation support tools developed and used by (Horling, Mojtahed, Svensson and Spearing 2002) for their applications. This scenario generator contains an integrated terrain data manager and path calculation algorithms. However this scenario generator lacks support for 3D terrain data. It was also found that this scenario generator was difficult to adapt to the growing requirements of users (3D visualization) and lacked the support for creating terrain databases. The introduction of new simulation entities was also found to be very cumbersome.

Research groups in United States military and the Air Force have also addressed the need for a scenario generator to support the development of their DDF testbeds. Development work for this need is reported by (Blais, Brutzman, Horner and Nicklaus 2001) and (Dawidowicz, Tran, Wong, Mayk and Klose 1997). These groups have addressed some of the requirements that are enlisted in the previous section. However both the groups have reaffirmed the need for terrain support and 3D visualization in their work.

An application that comes closest to requirements identified has been developed by Toyon (SLAMEM 2004) and MRS (Carlotto and Kadar, 2003). While Toyon has developed a complete simulator for ground/air tracking, there are certain areas in which it is lacking. First among these is the lack of 3D support. The next being lack of terrain support. Inspite of these short comings SLAMEM is one of the better software available to handle these problems. The best software that has been reviewed is MRS developed by (Carlotto and Kadar, 2003). The software that has been developed by this group satisfies most of the listed requirements.

Other commercial products are available from Boeing (BattleScape 2004) and TNO-FEL (HAREM 2004) for the purpose of battlefield simulation and scenario creation. However these applications seem to be very special purpose in nature and little information is available about the architecture, utilities, modules etc.

4 SOFTWARE ARCHITECTURE AND FUNCTIONALITY

Kim et al. (Kim, Singh and Llinas 2002) had previously developed a proof of concept simulator for distributed target tracking. A simple scenario generation module using Matlab was developed to support this simulator. However, this scenario generator was found to be insufficient for generation of realistic scenarios as it lacked terrain support, and could not satisfy the previously listed requirements for a scenario generator.

A scenario generation tool has been developed, "*SceneGen*" , for creation of scenarios for use in the DDF simulation testbed, keeping in mind the requirements listed in section 2. The following sections describe some of the design issues faced and approaches taken in the development of *SceneGen*.

A modular architecture has been developed for *SceneGen*. The primary modules in this software are the terrain engine, entity behavior description, visualization module, user interface (UI) components and finally the database support (Figure 2). This architecture allows for easy extension and customization of the scenario generator. For instance given a simulation testbed that utilizes a different database schema, the database module could be re-written to support that schema. This would enable use of *SceneGen* for creation of scenarios for a wide range of fusion testbeds. This architecture also enables plug & play capabilities. *ScenGen* can be plugged and used to create data for any type of networking, tracking or fusion algorithm.

The application (*SceneGen*) has been developed using C++ in a Linux environment. Graphics and Visualization capabilities are provided through the use of the OpenInventor API (Coin 2004). OpenInventor was chosen because it is a scenegraph based 3D graphics API that gave a lot of provisions to support a large 3D visualization with animation involving a wide scaling of entities. The user interface has been developed using Trolltech's Qt GUI (Qt 2004) development toolkit. MySQL was used for data warehousing and the KDevelop IDE was used as the development environment. The following sections provide a detailed description of the aforementioned modules:

Figure 2 shows the logical division of modules in *SceneGen*. The data that is generated by user interaction with the graphics window (mouse picking) and the user interface is gathered through the Qt widgets and is routed to the classes that describe the entity. Copies are sent to the graphics window for any updates and the database for warehousing. The copy of data received by the graphics module is used to render information onto the screen. The copy of information stored in the database is the actual scenario data that will be used to design and test DDF systems. Alteration of the entity models and the database schema can be done to provide data for any kind of scenario.

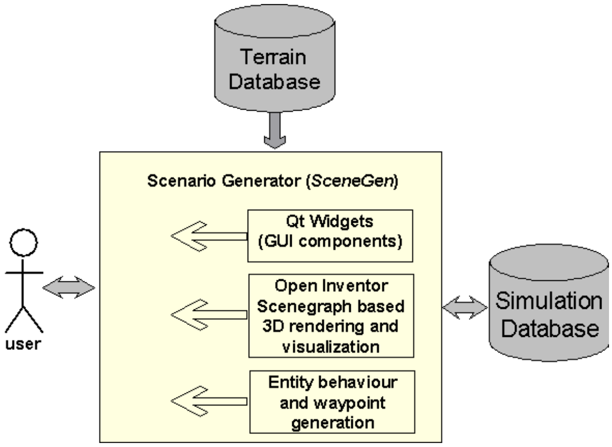
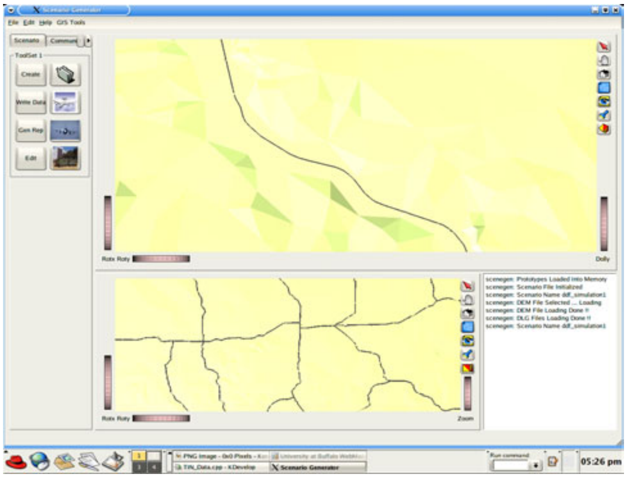

Terrain Database
Scenario Generator (SceneGen)
Qt Widgets (GUI components)
Open Inventor Scenegraph based 3D rendering and visualization
Entity behaviour and waypoint generation
Simulation Database
user

Figure 2: Software Architecture Used for *SceneGen*

4.1 Terrain Database

The first step in creation of a scenario is the preparation of the terrain data for the location of interest. Digital Elevation Model (DEM) files are used for getting terrain data of the desired location. DEM files are easily obtainable from USGS repositories and can be downloaded freely using the internet. However in its raw form, DEM files do not contain the polyhedral information describing the terrain skin. As a result, they cannot be directly used for 3D visualization. The DEM files are triangulated to compute the Triangular Irregular Network (TIN) representation that gives the terrain skin. DLG files from USGS are then parsed and information regarding roads and cultural entities are extracted. The GIS data is then mapped onto the corresponding location on the terrain. Then maps of the region providing other important data such as rivers and foliage are textured onto this terrain to provide adequate geographic information to support users during creation of scenarios. Another provision provided in this tool allows users to select and place foliage and buildings in areas otherwise devoid of them, this will help in creation of highly customized scenarios. Figure 3 shows a location where the terrain has been recreated using a DEM and has been overlaid with road information from DLG files for that location.

In this application a static triangle mesh without LOD is used. The reason for this is that with LOD, tasks like ground vehicle animation and mapping of GIS data become difficult. Additionally the terrain has been extensively used as a point of reference. All entities in the scenario are created and placed initially by picking on the terrain. All paths associated with the entities are also generated using the terrain. Using dynamic or static LOD meshes would mean some of the points associated with the data may become invalid when generating the new mesh. Hence for purposes of ease and to avoid corrupting the generated data a static triangular mesh has been used for this application. Due to availability of high performance computers and good graphics cards, a high resolution static mesh with textured GIS information has been used. Work is currently on to add functionality to support usage of Digital Terrain Elevation Data (DTED) 1, 2 and shape files.



```
scenegen: Prototypes Loaded into Memory
scenegen: Scenario File Initialized
scenegen: Scenario Name ddf_simulation1
scenegen: DEM File Selected ... Loading
scenegen: DEM File Loading Done !!
scenegen: DLG Files Loading Done !!
scenegen: Scenario Name ddf_simulation1
```

Figure 3: Terrain and GIS Overlay in *SceneGen*

4.2 Targets, Platforms and Sensor Modeling

SceneGen supports the creation and modeling of a number of platforms with specific sensor and communication models. Here the term 'platform' has been used to define vehicle; whether on land or air. The prototype of a platform carries generic information about any class of entities within a scenario. The user can select the prototype information from a database and use that prototype information to model the entity in the scenario. Any given platform can support a number of communication and sensor units on it. The communication and sensor units have their own prototypes associated with them. But a class of entities with same prototype information may also carry a separate set of individual information. For example (Figure 4), in a battle scenario there maybe many fighter planes of type F-14. They will all have common prototype information (same wingspan, fuel rate etc …), but additionally individual planes in the battle scenario can carry different individual information like type of ammo carried, mission type or types of sensors on board. *SceneGen* allows association of both types of information during modeling of an entity. The prototype information of various kinds of aircraft, sensors and communication units are provided in the database.

The other key issue related to creation and modeling of entities is their placement and definition of their motional behavior. While it is desirable to have entities following complex paths in 3D mimicking their real world counterparts, it is difficult to model or pick points and create a path to approximate the same. Another advantage of using 3D is that entities can be placed at points on the geography based on its strategic and tactical merit. For creation of paths the user picks a series of points that would be part of the desired path. As picking points in empty 3D

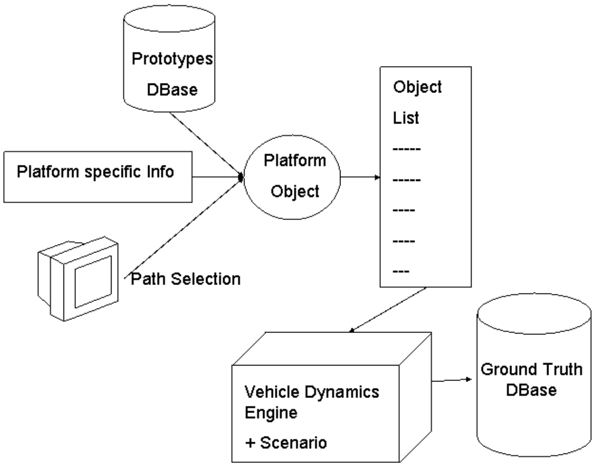


Figure 4: Scenario Generation in Progress

space (for aircrafts) is not possible, the user uses the terrain as a reference and picks points on the terrain. The picked points are then displayed to the user, who then can alter the Z elevation of the points as desired. These points are then fed into a vehicle dynamics engine that generates the entire path based on the vehicle properties.

4.3 Database Support and Reusability

The scenario generator uses MySQL for data warehousing. The database is used for storage of data generated during the scenario and for the storage of entity model and prototype information library. The use of the database allows for easy storage and retrieval of scenarios of any size. The database facilitates loading and editing of previously created scenarios. The database schema can be changed to store data in different formats for different information fusion testbed needs. The biggest advantage is once a database has been created and validated for a particular scenario it can then be reused many times over for different algorithms.

4.4 Information Visualization

Scenarios created for DDF can easily get cluttered with a large number of entities and their associated data. Hence it is necessary to represent the entities and data in fashion that transfers maximum information to a user at a glance, but at the same time can also lend itself for in depth analysis. For this purpose 3D icons and symbols have been used to represent information. There are two main sources of information which are of interest to the user,

1. Vehicle specific information
2. Terrain information
3. Vehicle dynamics information

In *SceneGen* traditional conventions for representation of targets (red color) and friendlies (blue) has been followed, but at the same time have adopted further concepts for iconic representation of vehicle specific information. The following metrics are being used for vehicle specific information transfer; transparency, color, shape and size.

For example, consider representation of sensor range (Figure 5), the sensor range is represented by a simple spherical shape in *SceneGen*, the importance of the sensor can be found by the color of the sphere representing its range (red for most important and blue for least important) and the sensing strength of the sensor is represented by the transparency (least transparent is most powerful and the most transparent is the weakest sensor). Logically the size of the sphere will represent the coverage of the sensor. These metrics while being simple to implement, can be used to depict a wide variety of information at different levels. As these icons are generated and updated at runtime, a large range of values can be displayed without the overhead of having to store 3D models or bitmaps for each level of information.

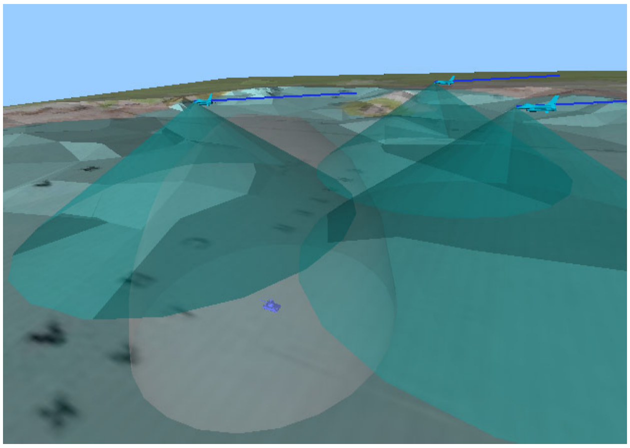


Figure 5: 3D Iconic Representation of Scenario Data

Entity and target trajectory is another important data that needs to be conveyed effectively to the user. This information is represented as three dimensional curves and lines on a temporal basis. The trajectories can be animated to show the users the change in a scenario over the course of the simulation. This could help in altering entity trajectories or sensor positions to obtain specific datasets for testing. High resolution VRML models of various aircraft and ground sensors are used to represent the entities. Individual VRML models have been used for the objects as there is simulation of discrete events like tracking performance for a sensor tracking a single aircraft or a fixed number of aircrafts.

4.5 User Interface and Navigation

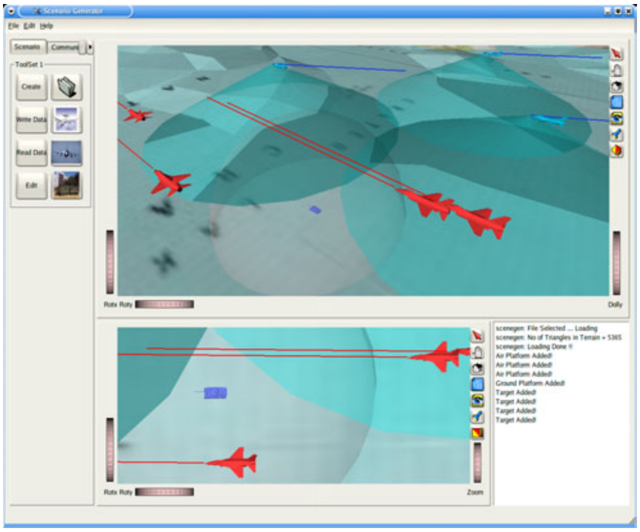
SceneGen provides the user with interactive navigation support. The user will have the capability to view the scenario from any desired viewpoint and then manipulate the environment to examine regions of interest. *SceneGen* also provides the user with capability to zoom in and out of regions of interest. The navigation and interaction is mainly using the mouse except in cases where the user has to explicitly set or specify some value. For instance the placement of vehicles and vehicle behavior is set using the mouse, which is the most intuitive way to do it. When explicit values for vehicle parameters are to be set, dialog boxes are used to take input from the user through the keyboard. Large part of the navigation was implemented using the tools that OpenInventor provided. Interactive navigation is important as it would facilitate in depth analysis of the scenario created. The user will be able to go to the position of a sensor and physically evaluate its line of sight.

While 3D visualization helps decision making, studying vehicle motion, sensor coverage behavior, it was found that for tasks such as selection for placement of platforms and tactical decisions in general, a 2D view was preferable. When doing a comparison of sensor range overlap or network connection visualization, 3D visualization does not offer the best picture. Using a 2D top view is better for these tasks. Hence *SceneGen* has been provided with two windows for viewing, one window provides a 3D representation of tactical information and the other window provides a 2D top view of the scenario. Cohesive tactical pictures with 3D icons are displayed in the 3D window, while information that is easily comprehended in 2D is displayed in the 2D window. Using the combination of the two windows makes process of scenario examination and validation more effective.

5 RESULTS

The application was implemented on a Pentium 4 desktop with 512 Mb RAM, a 64 Mb NVIDIA GeForce graphics card with Red Hat Linux 9 operating system. *SceneGen* was then loaded up with a DEM of representing a location of size 108 by 108 kilometers in North America. The terrain loaded using the DEM was textured with a political map of that region to provide the user support for scenario creation. A scenario was then created with 10 ground based radar units and 15 air based units carrying various sensor specified by the users. A total of 10 targets, all of them aircraft, were created as two groups flying in formation with crossing trajectories. Four ground based sensor units and six air based sensor units were designated with fusion capabilities. The scenario was the run to obtain the ground truth and the data generated was stored in a database. This scenario was created to test out the algorithms developed by group developing the data fusion testbed.

Figures 6 (a, b and c) shows screenshots of *SceneGen* after the creation of the sample scenario. The larger window provides three dimensional viewing and navigation, the smaller window provides a dedicated top view. There is a tool bar to the left that is used for data collection and user interaction. A command prompt displays the previous command history.



scenegen: File Selected ... Loading
scenegen: No of Triangles in Terrain = 5365
scenegen: Loading Done !!
Air Platform Added!
Air Platform Added!
Air Platform Added!
Ground Platform Added!
Target Added!
Target Added!
Target Added!
Target Added!

Figure 6a: Screenshot of the *SceneGen* Interface

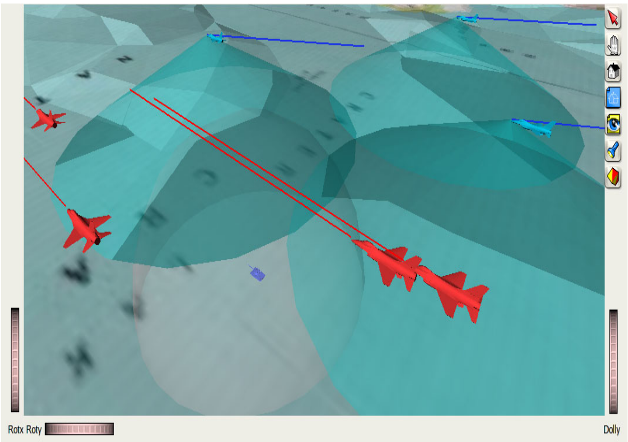
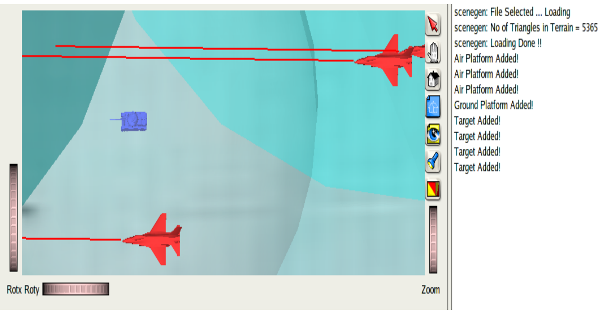


Figure 6b: Main Window Providing 3D Visualization



scenegen: File Selected ... Loading
scenegen: No of Triangles in Terrain = 5365
scenegen: Loading Done !!
Air Platform Added!
Air Platform Added!
Air Platform Added!
Ground Platform Added!
Target Added!
Target Added!
Target Added!
Target Added!

Figure 6c: Top View Window with Command History Box

Apart from the test case discussed above, scenarios involving a number of other geographical locations were also created and tested A wide range of scaling in the number of entities with the range being 2 to 125 individually managed simulation entities were used. *SceneGen* showed adequate performance in all the test cases. Frame rates of about 50 fps and uninterrupted interactivity were achieved. Bulk of computationally intensive activities (like sensor report generation) is performed at the end of the scenario generation; hence the process of *SceneGen* is not overly influenced by post computational processes.

6 CONCLUSIONS

The features and architectural details of *SceneGen*, a support tool for development of DDF systems have been described in this paper. *SceneGen* was developed to satisfy common requirements expected of scenario creation softwares. *SceneGen* has been used extensively as a support tool for the DDF system development done by the group at SUNY at Buffalo. The framework of *SceneGen* was found to be flexible and extensible. With changes in the database schema and the methods used to commit the data to the database, *SceneGen* can be used as a tool for generating scenarios for any kind of data fusion application. User evaluations of the visualization scheme has currently been evaluated only by our group, further tests need to be conducted. Currently *SceneGen* has been tested for air tracking scenarios, but the same framework can be easily extended to include ground tracking. In the future, the *SceneGen* will be made more generic in nature to cater to any geo-referenced Fusion applications.

ACKNOWLEDGMENT

This research was performed under the sponsorship of NIMA and managed by Rosettex Technology and Ventures Group under Task Order # RTVGC-02-216-03-C-CR-2-001. The authors gratefully acknowledge their support. We also thank Dr. Tarunraj Singh, Dr. Rakesh Nagi and Chandresh Mehta for their support for this work.

REFERENCES

BattleScape. 2004., Battle Space Visualization Software, Boeing, Available online via `<http:/washops.boeing.com>` [accessed February 2004]

Blais .C, D. Brutzman, D. Horner and S. Nicklaus. 2001. Web-Based 3D Technology for Scenario Authoring and Visualization: The SAVAGE Project. In *Proceedings of Interserivce/Industry Training, Simulation and Education Conference*.

Carlotto .M and I. Kadar, 2003. A Multisource Report level Simulator for Fusion Research. In *Proceedings of Signal Processing, Sensor Fusion and Target Recognition XII, SPIE Vol. #5096*

Coin. 2004., Open Inventor 3D Graphics API, Systems in Motion, Available online via `<www.coin3d.org>` [accessed August 05, 2004]

Dawidowicz .E, L. Tran, R. Wong, I. Mayk and D. Klose. 1997. Scenario Animator and Generator as a Command and Control Decision-Making Tool for Humanitarian Assistance. In *Proceedings of 4th International Command and Control Research and Technology Symposium, June 17 -20, Washington.*

FLAMES. 2004., Battlefield Simulation Software, Ternion, Available online via `<www.ternion.com>` [accessed August 05, 2004]

HAREM. 2004., Air defense model simulator, Available online via `<http://www.tno.nl/instit/fel/div1/harem.html>` [accessed August 05, 2004]

Horling P., V. Mojtahed, P. Svensson and B. Spearing. 2002. Adapting a Commercial Simulation Framework to the Needs of Information Fusion Research . In *Proceedings of Fifth International Conference on Information Fusion,* pages 220 - 227, Annapolis, USA, International Society for Information Fusion, Sunnyvale, CA.

Kim .J, T. Singh and J. Llinas. 2002 Large Scale Simulation of a Distributed Target Tracking System. . In *Proceedings of Fifth International Conference on Information Fusion,* pages 220 - 227, Annapolis, USA, International Society for Information Fusion, Sunnyvale, CA.

Nicholson, C. M. Lloyd, S. J. Julier and J. K. Uhlmann. 2002. Scalable Distributed Data Fusion. In *Proceedings of Fifth International Conference on Information Fusion,* pages 220 - 227, Annapolis, USA, International Society for Information Fusion, Sunnyvale, CA.

Qt. 2004., User Interface Development Toolkit, Trolltech, Available online via `<www.trolltech.com>` [accessed August 05, 2004]

SLAMEM. 2004., Simulation software for surveillance and tracking, Developed by Toyon, Available online via `<www.toyon.com>` [accessed August 05, 2004]

Svensson P. and P. Horling. 2003. Building an Information Fusion Demonstrator. In *Proceedings of Sixth International Conference for Information Fusion,* Cairns, Australia, International Society for Information Fusion, Sunnyvale, CA

AUTHOR BIOGRAPHIES

T.KESAVADAS is a professor of Mechanical and Aerospace Engineering at the State University of New York at Buffalo and the founder director of the Virtual Reality lab. He completed his PhD from the Pennsylvania State University in 1995 and M.S from Indian Institute of Technology, Madras in 1988. Prior to joining SUNY Buffalo, he worked as a scientist at thesVR center at the Iowa State University. Kesavadas's main areas of research are design of VR based engineering systems, computer aided design, human machine interface, and biomedical simulation. He is a member of ASME and IEEE. His email address is `<kesh@eng.buffalo.edu>`

G. SRIMATHVEERAVALLI is a Masters student in the Department of Mechanical and Aerospace Engineering at the University at Buffalo. He has a BE degree from the University of Madras and will soon be joining the PhD program at SUNY Buffalo. His email address is <gks2@eng.buffalo.edu>

N. SUBRAMANIAN did his MS in the Department of Mechanical and Aerospace Engineering at the University at Buffalo. He is currently working with GE Medical in Bangalore, India.