A SERVER TO PROVIDE THE STATE OF THE ATMOSPHERE TO DISTRIBUTED INTERACTIVE SIMULATIONS

Eric Schmidt, Maureen Cianciolo, W. Garth Smith and Robert Reynolds

TASC Inc., 55 Walkers Brook Drive, Reading, MA 01867
Internet:winds@tasc.com Phone: (617) 942-2000

ABSTRACT

The rationale for designing a server for ingesting, integrating, storing, and distributing the state of the atmosphere is presented. The server supports distributed interactive simulations and can be used for extensive training exercises. An overview of the server design is provided, with a review of each of its five component modules: the database, integrator, viewer, distributor and simulator. The server supports client applications in several modes, including predistribution of an archived database and active distribution for live exercises in a real-time atmospheric environment. Each component of the weather server is discussed in terms of the factors that influenced the design rationale.

The set of variables representing the state of the atmosphere was selected by a three step process: first, atmospheric features were identified and characteristics of the atmosphere that correspond to the features were specified. Second, the effects of the atmospheric environment were related to features observed, such as thunderstorms, blizzards, etc. Finally, characteristic atmospheric variables were mapped into the effects that they produced. These effects impact decisions made at the command and field levels. A typical scenario is presented to show that a variety of atmospheric conditions and effects must be considered to support informed decision processes and accurate personnel training in live, virtual, and constructed exercises.

Biographical Sketch:
Dr. Schmidt has worked for over 12 years in atmospheric modeling and simulation, with theoretical modeling and field observation experience. He received a B.S. in Physics from Texas Tech (1980), M.S. in Physics from Univ. of Texas at Dallas (1983), and a PhD in Atmospheric Science from Georgia Tech (1991). He is a member of the American Geophysical Union, the American Meteorological Society, and has over 30 articles published in journals, proceedings, and symposia.

A SERVER TO PROVIDE THE STATE OF THE ATMOSPHERE TO DISTRIBUTED INTERACTIVE SIMULATIONS

INTRODUCTION

The near-Earth environment is important for simulations utilizing modern day weapons systems, sensors, and vehicles. The environment affects performance of these systems and the personnel who must operate them from command to field level. Tactical decisions that incorporate the naturally occurring features and effects of the environment will be more likely to have an edge in future encounters. This paper presents a methodology by which information about the atmospheric environment is ingested, assimilated to produce a cohesive, consistent description, then archived or distributed for use by a variety of simulators. The information is fully Distributed Interactive Simulation (DIS) compliant to support large-scale distributed simulation exercises.

Weather has a significant impact on the operation of sensors due to modification of properties of the atmosphere by features such as thunderstorms. Similarly, operational missions are impacted by the naturally occurring atmospheric environment. Trafficability, communications, and visibility all have an impact on battlefield tactics. It is necessary that weather effects be included in modeling and simulation exercises to ensure realistic mission planning and rehearsal, sensor test and evaluation, and personnel training. DIS systems are in development to support multi-fidelity applications for a wide variety of users. The addition of a realistic, multi-fidelity representation of the *weather* environment to DIS systems is the objective of the Weather in DIS (WINDS) project, with emphasis on naturally-occurring atmospheric phenomena.

The WINDS project is developing the software tools necessary to provide tactically significant weather to disparate entities participating in DIS exercises. This task is performed by implementing a software architecture for a *weather server* built around five components: the weather integrator, the weather database, the weather distributor, the weather viewer, and a weather simulator (see Figure 1). These components ingest, integrate, archive, distribute, and then simulate and provide a preview of the local atmospheric conditions and effects within the simulation domain.

SERVER ARCHITECTURE OVERVIEW

Server Design Considerations

The primary design considerations of the WINDS program include support for a generic data fusion system and data transmission paradigm. The context is the description of coherent atmospheric phenomena for the distributed simulation community. The WINDS software infrastucture is a series of object oriented C++ classes. The software is designed to be open to the general community of developers, using state-of-the-art object oriented programming to support software modularity and reuse. The goal is to provide an architecture that creates simultaneous representations of complex atmospheric phenomenon for all distributed simulation participants. A general client/server architecture is modeled such that multiple weather server clients, acting as recursive servers, can be added to the network to provide ever higher levels of environmental data fidelity. These nested servers will support integration of feature and effect models to increase data resolution and derive additional weather variables. An applications programmers interface (API) in C and C++ enables abstraction of participating simulation applications from the details of the weather database. Simulation applications need not make direct use of the WINDS infrastructure in that they can adhere to a documented and public protocol description for environmental protocol data unit (PDU) data.

Figure 2 illustrates the recursive nature of the server design. The server software will typically reside at several regional site(s). Replicas of the server software will also reside at local and sublocal sites, where the higher order servers will act as the primary data source(s). This cascade of server software ensures that simulation applications interested in regional-, local-, and sublocal-scale information will be satisfied. The lower-level servers may have different uses, depending on the needs of the local user community. Figure 4 illustrates the use of the server as a testbed tool for development of new decision support analysis methods. The primary data for the regional server is raw weather data from one of a variety of data providers, such as the Master Environmental Library (MEL), which is discussed later.

CONTENTS OF WEATHER DATABASE

The construction of a database of weather variables is the primary function of the server. The software that populates the database, retrieves data, and operates on the data makes up the remainder of the server components. However, the software is of limited use unless informed decisions are made regarding the contents of the database. A brief description of the process used to define the database contents follows.

First, we built a list of variables and their characteristic attributes. The list extends beyond a mere description of physical structure. It also includes variables that describe characteristic effects of the environment on multispectral and acoustic sensors, personnel, equipment, etc. Table 1 provides a limited list of variables and features unique to the atmospheric environment.

Once a list of atmospheric variables was constructed, the next step to define the contents of the weather database was the review of the input data requirements for a comprehensive set of feature and effect models. Table 2 provides a list of some of the environmental models considered. Comparison of the data sources required to initialize and run each of the feature and effect models with the list of potential database contents provided an initial evaluation. The DIS community, the Air Force Phillips Laboratory (PL), the Army Research Laboratory (ARL), and the Naval Research Laboratory (NRL) were consulted to obtain a viable list of models to consider.

The third step used in the definition of the weather database contents considers the needs of the user community (simulation and operational). Environmental effects of interest to semi-automated force (SAF) systems, electro-optical sensors, and field forces, such as those listed in Table 3, were all considered. Determination of the needs of personnel trainers, simulation applications, and equipment was done as members of the simulation community (DIS, ARL, etc.) and potential Synthetic Theater of War (STOW) 97 participants were consulted. The variables needed to provide environmental effects in existing and future simulations were compared to the list of potential database variables. The list was augmented as needed. The properties of the atmosphere, such as those in Table 1, effect sensors, equipment and personnel (see Table 3). The next section provides more discussion of the mapping of properties and features to effects. Iterations between the data requirements of feature and effect models and the requirements of end users provide an evaluation mechanism for the weather database contents.

The primary data source for WINDS is the Master Environmental Library (MEL), developed by the Navy through the coordination of major regional centers associated with NRL at Monterey, Stennis, and Washington, D.C. MEL is to become a major shared resource and data repository for all DoD agencies. NRL is tasked to determine requirements, builda catalog, and design a distribution mechanism for atmospheric and oceanographic data. Two test case scenarios, located in the southwest United States during the summer and winter, have been provided for use by the WINDS program. Two more datasets are planned which will test the automated subscription service. Although WINDS is currently taking advantage of MEL data, the design of the weather server is independent of the data source or specific versions of most feature and effect models for robustness and to ensure flexibility for future growth.

The design of the weather database to support persistent classes [Ref. 1] is a critical function for supporting feature and effect models and alternative environmental data types. The database is extended to support future data types by simply defining a new class within the C++ paradigm. A well documented wrapper approach instills the persistent nature in the classes for inclusion into the weather database paradigm. C++ programmers can add database elements, i.e., C++ class definitions that map to physical objects, via an open approach with minimal spin-up time. The entire persistence scheme and examples are provided as part of the WINDS infrastructure and are non-proprietary. The notion of intelligent pointers (smart pointers) augments the persistence of the weather objects to support efficient and incremental loading of a weather database into active memory. The weather database models meteorological geometries into object-oriented classes with increasing levels of dimensional sophistication. A weather database consists of objects that encapsulate weather fields, layers, profiles, and observations.

THE WEATHER INTEGRATOR

The weather integrator is the interface to the weather database and the myriad weather data formats provided in the meteorological community. The architecture provides data fusion by providing user defined receivers (object classes) to support each data format. The receiving objects run as separate processes in real-time. An object instance diagram for the weather integrator is detailed in Figure 3 and discussed in a later section.

Integrator Design Issues

Table 2 lists atmospheric feature and effect models that could be supported by the WINDS weather server. These models require a subset of the environmental properties listed in Table 1. Some effects of the environment on exercises are provided in Table 3. A hierarchy of properties, features, and effects is established by mapping environmental properties first to features, then to the effects they produce, and finally to decisions. Mapping the relationships between properties and effects establishes the relative importance of individual properties if one considers a wide variety of environmental states.

Feature and effect model recommendations were solicited from domain experts available through the DIS community. The level of detail required to support different participants defines the scope of the requirements. Integration efforts hinge on careful analysis and application of the requirements to produce a coherent snapshot of the state of the environment that meets the needs of all exercise participants. The 4-D assimilation of *in situ* data and model products is crucial for the support of live exercises and to ensure that the best available product is produced. Validation and verification efforts are performed at all stages of the project to ensure that the representation of the environmental state is realistic and complete.

The data and model requirements for WINDS are driven by the need to support a range of users, from command level decision support to support for field operations and personnel training. The requirements include decision support and realistic effects of the environment as experienced by participants in the field. A later section of this paper addresses a potential scenario and highlights a few relationships between decisions, effects and features. The use of scenarios is not intended to hardwire a specific set of environmental characteristics. Instead, scenarios are used to stress the requirements and determine the relative importance of atmospheric variables. Exercises such as the STOW 97 are defined well in advance, so the use of scenarios to test the environmental product provides a means to mitigate risks to the STOW 97 program. These risks include data availability, requirements definition, and software/algorithm development, among others.

Integrator Structure

The weather integrator is the brain of the weather server. The integrator must include functions which allow the server to receive, decode, register, transform and store data fields. The integrator must also be able to initiate feature and effect models to enhance the resolution of data or to derive data products requested by the users. An object-instance diagram of the integrator is provided in Figure 3. The receiver objects ingest the input data, decode and register the values, and store the information as layers in the database. The assimilator is notified when the database is populated with a new layer of values. The assimilator object is a rule-based module that uses an event manager to route messages and initiate processes. Smart C++ scripts are wrapped around the feature and effect models. These scripts use a simpler version of the list of rules that the assimilator uses. Processes are spawned when a specific set of conditions are met, i.e., the necessary data is stored and pointers passed, the time tag is current, and the "go" message is received from the assimilator. Model output is stored in the database and the assimilator notified once the process is complete.

The use of smart scripts to run feature and effect models through batch processes with an initiating control message generated by the assimilator is a highly extensible design. New feature and effect models or transform processes can be added by encapsulating the new algorithm with a smart script and modifying the set of rules in the assimilator. This provides a "plug and play" capability and allows the user to define methods by which the software will treat specific datatypes.

Pre-exercise Setup

Collection and collation of the model and data requirements are done during pre-exercise setup. Figure 1 indicates that pre-exercise set up definitions initialize the server. The exercise participants will be polled to determine their data needs and specify whether the exercise will be virtual, constructed, or live. The environmental data types requested by the exercise participants determine the data subscription requirements for the environmental server. User data requests normally span a range of fidelities which must be considered to ensure that a consistent description is provided. Some interpretation of the user's actual needs is required, since most users mistake "highest resolution" data as the "best" data. For example, a high flier does not need atmospheric data at the 1-5 m resolution required by dismounted infantry. However, the 1 km resolution data provided to the flier must be consistent with the 1-5 m data provided to the dismounted infantry.

An important action that the pre-exercise setup phase performs is to notify the weather integrator which feature and effect models are selected to run at the server. The rules used by the assimilator to control the operational flow of the integrator are simple check lists. Modification of these lists redefines how the integrator will assimilate (or blend) the model output, live data, imagery, and *in situ* data to produce a consistent dataset.

Fidelity Issues and Atmospheric Models

Often the best datasets available from measurements or feature and effect models may not be sufficient to support simulations because the dataset is less complete than other, lower fidelity datasets. In such cases, a trade-off between the use of interpolation methods and model development will be made to best satisfy the needs of the users. Expert advice will be solicited from the DIS community for further model or data requirement definition in ambiguous situations. One result that is expected from the definition of the model and data requirements needed to support WINDS is code improvement. Research grade models, such as the U.S. Navy's *Coupled Ocean-Atmosphere Mesoscale Prediction System* (COAMPS), require further development and validation before they can be used to support exercises on an operational basis. The need for correlated environmental data emphasizes the need for model development, particularly as regards "near-operational" model codes. WINDS provides an assimilation and validation mechanism that is independent of the data streams.

A corollary to this is the fact that operational model and data requirements are not the same as research requirements. Operational support of exercises is the primary goal of WINDS, though the server supports both types of models. Judicious interpretation of the requested model output and data types will be performed on a case by case basis. Requirements interpretation is a continuous process, so careful definition of common feature and effect models and user requirements will provide the basic rules to an informed decision process. Careful interpretation of the requirements and knowledge of the strengths and weaknesses of the models will help to ensure consistency and correlation within the simulation.

Data Assimilation

Optimal interpolation schemes are often used to assimilate observations and model forecasts. The problem of blending observations and models has a 4-D nature since the assimilation of spatial datasets over long periods of time requires care. Interpolation and assimilation techniques are treated as feature models in the server architecture. Consequently, they are activated by the integrator as needed. We solicit community and user input to determine the "best" interpolation or assimilation method for a particular scenario, based on the user needs and the data and model requirements. Live scenarios prove to be the most difficult to correctly assimilate. It is highly likely that WINDS will help define recipes for the assimilation of different datatypes and model output through use of test cases and participation in exercises.

Validation and Verification

It is necessary to test the data fields that WINDS produces for distribution. It is not sufficient, however, to wait to test these fields at the end of the development period. WINDS provides a series of demonstrations to show the incremental growth of the software. The validity of the results and the consistency with which the state of the atmospheric environment is described must be tested for each server subsystem, such as the database-integrator and database-viewer subsystems. One tool developed under the WINDS program to aid verification and validation is the weather viewer, discussed in the next section.

A first order validation and verification approach requires the definition of WINDS test cases needed to satisfy the accuracy and fidelity requirements. Collection of a consistent set of *in situ* data from a variety of platforms in the environment is necessary. Coordinating the data collection with numerical model predictions to forecast the full state of the environment in a coherent fashion is also necessary. These required tasks are complex, but provide the correlated data needed for verification and validation of the server products. Close cooperation between WINDS and MEL has provided the necessary expertise, data, decoder software, and understanding to adequately address these issues.

Some technical issues that are important to the project include: software migration from a single platform to multiple machines, network loading for live play exercises in a fully integrated environment, the need to incorporate load balancing into the distributed integrator design, definition of the criteria by which the "best" data and models are selected. Distribution of large volumetric, spectral and image datasets, the scope of the feature extraction problem and multi-fidelity support, and 4-D model and data assimilation in support of live play exercises must also be addressed.

Providing information on the location of atmospheric phenomena of interest, or features, is an important part of decision support for a disparate set of participants in a DIS exercise. Extracting information on features from input data fields (2-D or 3-D) via automated analysis is a difficult problem. The WINDS baseline project will provide feature extraction methods for a limited set of atmospheric features, such as thunderstorms.

WEATHER VIEWER

The weather viewer provides an analysis tool to visualize the weather fields stored in the weather database. This visual aid allows participating users to preview the data available at the server and simulation databases, thus validating the transmission mechanism in the process. Therefore, the weather viewer is useful as a simple database browser, a transmission validation mechanism, and as a tool to view the stored data values. A complete discussion of the weather viewer is beyond the scope of this paper, but the following brief description provides a flavor of the module.

The weather viewer is currently based on version 1.3 of the ModSAF software. The weather viewer is a software tool based on the ModSAF plan view display (PVD). The weather associated with a scenario cannot be modified through the tool. Instead, the software accesses the weather database through an API. The user can select weather elements to be displayed by selecting a scenario, the weather element, the time, and the level (height of the layer). This tool allows the user to overlay contours and streamlines on the PVD to preview, or browse, the information available in the database for distribution.

Other features of the weather tool include a textual summary of the scenario information. Typical surface weather information - temperature, wind speed and direction, precipitation, etc. - is also provided to the user by selecting a surface location on the PVD. A feature recently added is the vertical profile of a selected weather variable above a surface location selected from the PVD. The final feature of the weather tool is a vertical cross-section display. The user activates the cross-section widget, selects two endpoints on the PVD, and the display is dynamically updated to show a vertical cross-section of the weather element previously selected.

DISTRIBUTION ISSUES

The ultimate product is an integrated description of the state of the environment. The design supports predistribution of the environmental state and distribution by various means, including Protocol Data Units (PDUs), tape, or data file transmission via file transfer protocol (FTP). Preliminary estimates of the number of PDUs required to describe the state of the atmosphere indicate that the network loading is likely to be limited. The distribution of a few thousand PDUs every 20-30 minutes should suffice to support large-scale live exercises (e.g. 500x500km simulation domains), the most difficult case. Efforts are underway to minimize the network traffic through the use of subscription agents, multicast addresses, and mechanisms for matching data fidelity with sphere of influence. Predistribution of data to users for use in training simulations will primarily be done through the use of tapes or FTP-based delivery. PDU mechanisms are always an option, but are only required when exercises use live observations. Virtual and constructed simulations are much easier to support than exercises requiring live data assimilation, since predistribution methods are sufficient and useful. The use of predistribution mechanisms will require knowledge of the weather database API, while PDU traffic can be read and directly interpreted.

Frequent updates to describe transient features and their effects will be of limited spatial and temporal extent, so that the network traffic is of limited duration and scope. Implementation of feature and effect models at the simulation nodes plays a significant role in the reduction of network traffic. A final benefit from careful requirement definition is the definition of potential new PDU types. The current prototype environmental PDU is designed primarily to distribute information about small-scale atmospheric features. It is recognized to be a member of the environmental PDU family, which will most likely expand to include PDUs for the distribution of volumetric, image, and spectral datasets.

The weather server will distribute data which satisfies the needs of all participating simulators. The server subscribes to the necessary data sources to obtain the observational data, model forecasts, and analysis products it needs to meet the user requests. FTP-based systems are under development by MEL to support both electronic mail and world wide web (WWW) subscription mechanisms. The initial WINDS data subscriptions are via manual FTP. A WWW interface will be implemented at MEL by the spring of 1996, and automated subscription to the library through the catalog will be supported. This automated system is crucial during live exercises.

WEATHER SIMULATOR

The weather server supports at least three types of client applications. The following paragraphs provide a brief description of each of the client types. The WINDS simulator is designed to operate as a local version of the weather server, where feature and effect model processes are run. The resulting specialized weather information is then distributed to all dependent local nodes that have subscribed. The weather database, viewer, distributor, and APIs can all be identical to those that operate at the primary server. The dependent simulation nodes subscribe to the local weather server for the data needed to supply their local nodes. This limits the computation load on the primary server and any single simulation application.

The second client for the weather server is a simulation that receives information, manipulates it in some fashion, then passes it to an application. This is the simplest version of the weather simulator as it merely acts as a transformation mechanism. It can host feature and/or effect models to perform CPU intensive calculations and distribute the results to its subscribers.

The third type of client weather simulator is likely to be used most often. This local simulation node receives data from the weather server to build a local instance of the weather database. This database is generally a subset of the primary weather database, but could contain information received from a local weather simulation node. This simulator also supports feature and effect models and transformation of the data as needed by the application. The simulator can provide the same software capabilities of the weather server - database, viewer, and APIs - if the software is tuned to perform in that fashion. Figure 2 illustrates the recursive nature of the server software, driving simulation applications at different levels of dependency. These client applications are limited and only intended to indicate the types of simulation applications that the WINDS architecture could support.

EXERCISE SCENARIO

Exercise scenarios are required to stress the design of sensors, decision aid support systems, simulations, and provide training. WINDS will provide information to a variety of potential users, so we consider a wide variety of situations where disparate data types might be required. Many potential scenarios will need to be supported and the environmental data requirements can vary dramatically. Many of the scenarios considered are DoD oriented; however, commercially-oriented scenarios, such as those of interest to the air traffic management community, have been considered in our weather database requirements.

It is necessary to know who the potential users are for an exercise scenario to be useful. Table 4 provides a list of potential exercise participants, limited to field operators. WINDS has the task of supporting exercises and providing scenarios with consistent, correlated environmental properties, features, and effects needed by the participants. This task becomes more difficult when environmental components are coupled, as environmental features have more extensive impact. The effects of a natural or anthropogenic feature propagate through more than one environmental component and influence decisions by a larger number of operators and commanders. Battlefield decisions must be supported by integrated representations of the environmental features and effects in order for informed tactical decisions to be made.

Environmental Impacts for Typical Scenario

Shallow water and amphibious warfighting are two typical scenarios that will prove to be important for STOW 97 and other exercises that involve land, air, space, and ocean based forces. We will briefly highlight important aspects of a generic shallow water situation. Training exercises are supported by providing a variety of atmospheric features and effects for use in live or replay mode by the simulation application. Scenarios involving more that one component of the near-Earth environment (atmosphere, ocean, land, and space) are the most difficult and important to incorporate, since they require a higher degree of realism are correlation.

Assume that the exercise occurs in a coastal region and that air, land, sea, and space forces are involved. Further assume that the environmental background is a quiescent mid-latitude summer day, but that a thunderstorm is moving into the exercise area. Prior to the arrival of the thunderstorm, assume that normal subsurface, surface, air, and space activities occur for each of the forces. As the thunderstorm moves through the exercise area:

- trafficability is reduced due to precipitation as the wave heights increase, the land surface becomes saturated, and icing and wind gusts affect the ability to fly.
- the temporal and spatial resolution of environmental state information becomes more critical as the mesoscale phenomenon propagates through the environment.
- intervisibility is reduced due to obscuration by the clouds associated with the storm and the precipitation falling from the cloud.
- background acoustic and electromagnetic (EM) noise increases,
- communications are affected by EM and acoustic noise, loss of visibility, whistlers, ducting, and modifications in the ionosphere.
- navigation is affected by changes in the Earth's magnetic field due to substorms or solar events, magnetic anomalies in the ocean floor, and geographic location.
- navigation is also affected by turbulence, currents, microbursts, space debris/meteoroids, and strong EM fields.
- survivability is influenced by the intensity of the feature and the effects of the feature as it propagates.
- operations of sensors, equipment, weapons, and personnel are affected by the storm as it passes through the exercise area. Lightning is just one example where fatalities can occur due to the naturally occurring atmosphere.
- performance of sensors, equipment, weapons, and personnel are degraded.
- command decisions must account for the presence of the storm and the effects that it will have on friendly and opposing forces.
- understanding the effects that the naturally occurring environment has on all forces in the exercise arena will allow environmentally informed commanders to gain a tactical edge on their opponents.

The highlights provided in outline fashion above address not only the effects of an environmental feature on participants and simulations, but also the impact on commander and field crew level decisions. It is easy to see the mapping of properties to features, effects, and decisions. The evidence is clear that the features related to one component of the environment will have effects that propagate into the other components. A fully integrated approach, or environmental server such as that described in this paper, becomes a necessary tool to develop a better understanding of the impact of the environment on operations.

Decision support and personnel training are the ultimate applications for systems such as WINDS. Figure 4 shows an example of how the weather server could support a decision aid process. The data needed by the decision aid will consist of environmental and exercise information. The weather simulator module provides an interface for the decision aid and acts as a testbed for development of analysis, evaluation, and simulation techniques. The recursive nature of the server software provides the capability of migrating fully developed methods to the integrator module to allow a "production" mode. The simulator can then be used to develop other methods. The extensible nature of the server provides only the data needed by the user, so performance optimization can be attained after an initial development and testing phase.

CONCLUSIONS

The Advanced Research Projects Agency (ARPA), the Defense Modeling and Simulation Office (DMSO), and the Topographic Engineering Center (TEC) lead an integrated effort to address issues associated with dynamic interactions between simulation entities and the environment. Under the sponsorship of ARPA and DMSO, TASC is pursuing an effort to describe and distribute the state of the environment (our focus is the *weather* environment) in a complete, self-consistent fashion for integration into programs such as the Synthetic Theater of War (STOW). This paper presents an overview of the WINDS effort. The Server integrates, manages, and distributes weather data to support enhanced physical realism in DIS exercises. These exercises will provide important opportunities to train personnel through the use of combined field operations and manned simulations, both of which are supported by the WINDS server.

REFERENCES

[1] Shilling, John J., "How to Roll your Own Persistent Objects in C++", Journal of Object Oriented Programming, pp. 25-32, July 1994.

[2] Smith, W. G., Schmidt, E., Cianciolo, M., Reynolds, R., Doren, P., Stanzione, T. and G. Mealy, "A Weather Server to Support Distributed Interactive Simulations," Proc. 12th DIS Workshop, Orlando, FA, (paper 95-12-066), pp 369-372, April 1995.

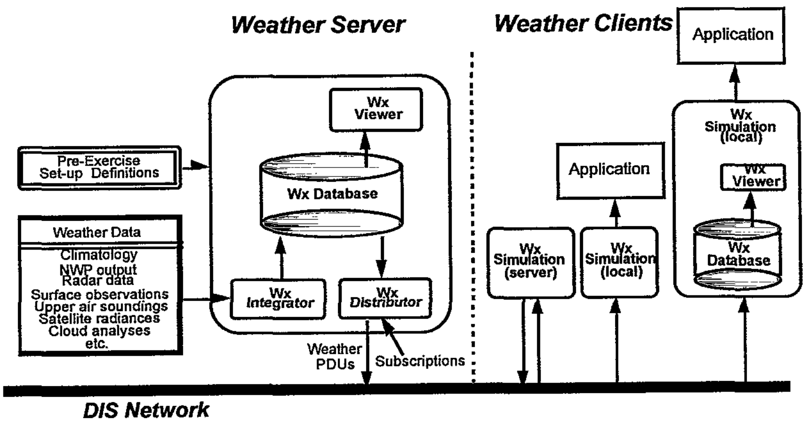

Figure 1. Overview of Weather in DIS (WINDS) server architecture in client-server form.

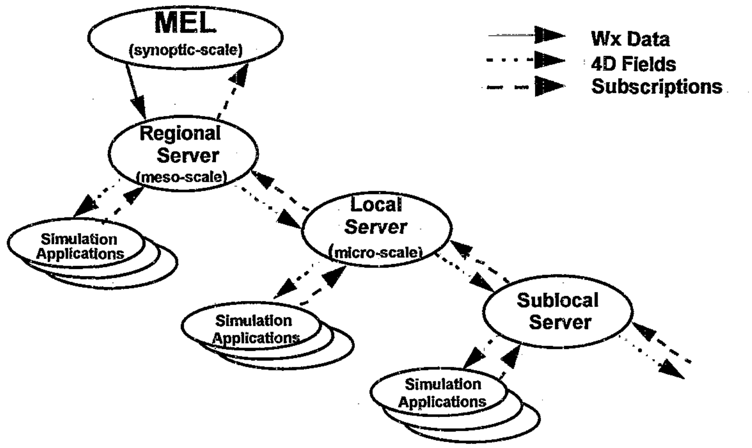

Figure 2. Recursive nature of software as hierarchical distribution of server nodes.

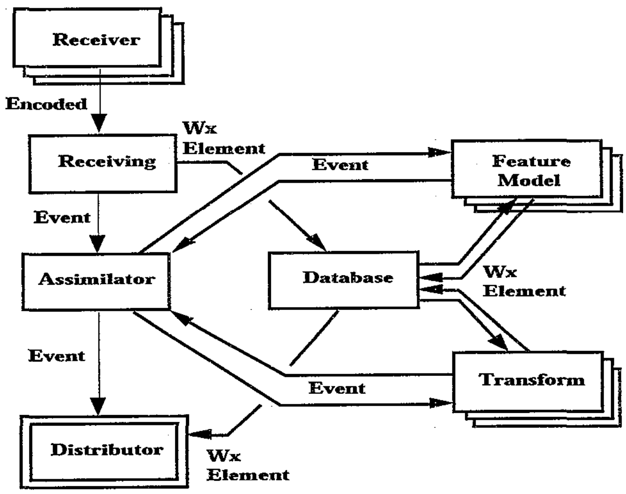

Figure 3. Object-instance diagram of weather integrator.

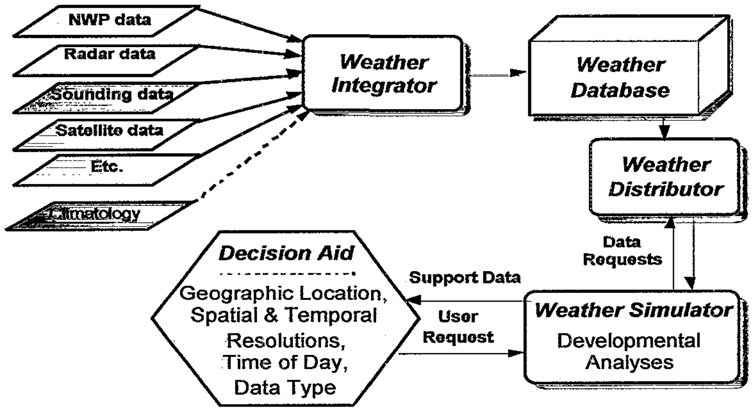

Figure 4. Schematic of WINDS server as it supports decision aids. The weather simulator acts as both testbed and application support mechanism.

Table 1. Properties and Features related to the Atmosphere

pressure	fog	thunderstorms	hurricanes	blizzards
temperature	precipitation	tornadoes	floods	mudslides
humidity	composition	lightning	droughts	plumes
winds	pollution	clouds	wind chill	sprites
turbulence	diurnal cycles	vorticity	heat stress	monsoons
visibility	heat fluxes	divergence	UV index	icing
aerosols	air density	dust storms	microbursts	extinction
acoustic properties	transmission properties	refractive index	frontal systems	tropical storms

Table 2. Sample Feature Models

COAMPS	TSTORM	ATMOS	RAMS
BFM	COMBIC	CSSM	UW Model
HRW	STATBIC	SCGSIM	MM5
EOSAEL Library	BEAMS2D	ATMOS	LOWTRAN

Table 3. Environmental Effects

Trafficability	Survivability
Intervisibility	Operability
Communications	Performance
Navigation	

Table 4. Simulation Participants

Surface ships	Paratroops
Submersible vehicles	Aircraft
Amphibious vehicles	Unmanned air vehicles
Unmanned underwater vehicles	Pilots
Swimmers	Infantry
Ground vehicles	Helicopters