Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

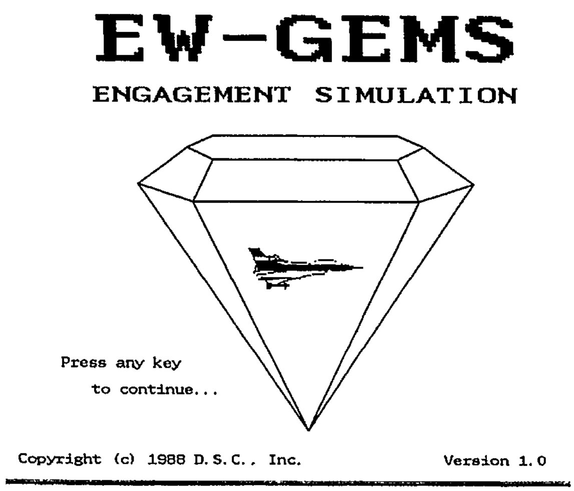
EW-GEMS: System concepts and development process of an electronic warfare simulation

Dr. Raymond E. Gallagher,
Frederick C. Christie, Kenneth R. Kaiser, and John T. Pattee
Defense Systems Concepts, Inc.
1340 Ashton Road, Bay A
Hanover, Maryland 21076

ABSTRACT

The Defense Systems Concepts (DSC), Inc. EW Group is currently developing a discrete time-driven stochastic computer simulation known as EW-GEMS (Electronic Warfare-Generic Enemy Missile Simulation). EW-GEMS will provide for a fully detailed depiction of the in-flight engagement between a high performance tactical aircraft, with on-board state-of-the-art ECM equipment, and a fully operational threat missile. Via EW-GEMS, analysis of the effectiveness of the application of airborne ECM techniques can take place in a real-life engagement without ever leaving the tarmac.

This simulated real-life engagement is the result of a uniquely coordinated effort between EW/Aerodynamic engineers and Software/Mathematics personnel following two parallel paths of software development. An Engineering test bed for model validation is being developed in parallel with EW-GEMS, the full-up formal software simulation. In this manner the models of EW-GEMS will have not only been validated, but verified data will be available prior to the completion of EW-GEMS.



1. EW-GEMS

In the Electronic Warfare (EW) community, there exists the ever present need to accurately, efficiently, and cost effectively test and validate electronic countermeasure (ECM) techniques. These techniques are applied in destroying or degrading the effectiveness of the enemy's electronic aids to warfare, such as ground-based gun fire control radars and surface-to-air and air-to-air missile guidance.

This need for testing and evaluating ECM techniques has not been satisfactorily met in the past. ECM hardware prototyping, by itself, proved to be too costly and time consuming. Computer-aided simulations with actual missile assemblies in the loop were very limited in evaluation capabilities and applications and did not live up to expectations. Moreover, to repeatedly place an electronic countermeasure device on a drone aircraft and fly it against a live missile, for the purpose of analysis, would be prohibitively costly.

EW-GEMS, as a generic EW simulation, is being designed to be an accurate cost effective analysis tool for conducting EW analysis. It will allow its user to represent an engagement of a wide variety of missiles, targets, radars, and electronic countermeasure systems without having to modify or reprogram any of the software. The design currently includes a powerful user interface, a scenario configuration module, a database manager, and built in test procedures, all supported by a foundation of software tools.

EW-GEMS is a time-driven computer simulation of a missile versus target engagement. The target may have two primary lines of defense at its disposal: 1) an evasive maneuver, and/or 2) an ECM device or technique. EW-GEMS will aid EW analysts in assessing the effectiveness of a particular countermeasure in a chosen scenario. An example of a scenario which EW-GEMS can simulate is an airborne F-15 interceptor fighter encountering a threat radar signal with an enemy missile in flight. The F-15 turns on its ECM jamming signal and pulls a 3g maneuver. The missile is driven off target, misses the F-15, and goes ballistic.

In order for EW-GEMS to simulate an engagement, the scenario must be defined by the user. The scenario configuration includes the selection of the target, threat, and missile launch platform, the parameters of which are drawn from corresponding databases. Via the scenario configuration module, the definition of a target maneuver, the setup of the initial geometry of the scenario (position and velocity vectors of the components), and the definition of the desired ECM technique are also entered. During a run of the engagement simulation, all positions and orientations (six degrees of freedom) of the objects are stepped through time and RF signal propagation is modeled. The engagement simulation includes the modeling of an illumination signal, an accurate RCS model which may contain actual glint points on the target aircraft, the effects of a missile radome on signal propagation, and a missile antenna model. The simulated engagement also includes the detailed modeling of the missile's receiver, a missile antenna gimbal model, the modeling of the missile's autopilot, and the complete modeling of the missile's aerodynamics (through boost and sustain travel phases).

Upon completion of the simulated engagement, the user can attain information about the run. First, the user can view an information screen describing an entire simulation run. This information includes the difference vector between the missile and the target at time of closest approach, the time at which the target escaped the main beam of the missile antenna, the final time of the simulation run, and the event which caused the simulation to terminate. Second, the user may choose to use the data frame scroller utility. By using this utility the user can view many simulation run-time parameters on the same screen as they change over time. Finally, the user has the option of viewing the information graphically. SimGraph, a generic graphics software tool, provides the user with the ability to plot any pertinent simulation variable versus any other variable in a two-dimensional graph.

Since EW-GEMS is a stochastic simulation, some of the information produced from one run has only statistical value, which may necessitate the generation of sample sets.

Much effort went into the planning of EW-GEMS in order to allow the user as much capability as possible in not only creating an engagement scenario, but analyzing its outcome. This plan also included an innovative method for developing EW-GEMS.

2. SYSTEM DEVELOPMENT METHOD

An extremely effective method was applied in developing the EW-GEMS system. At the outset a multi-disciplined team was assembled to first determine what capabilities the simulation should have and the user requirements of such a simulation. The team was composed of EW analysts/EEs, software engineers, missile aerodynamics personnel, and mathematicians.

The mathematical models of each engagement scenario element (target, missile, launcher/illuminator, ECM technique) were drafted in terms of geometric and electromagnetic calculations. These originally drafted models provided the basis of that which was to be finally simulated.

At the outset of the development of EW-GEMS, it was agreed that a highly interactive approach to simulation development would be followed. This approach consisted of a group of individuals divided between two parallel paths of development as shown in Figure 1.

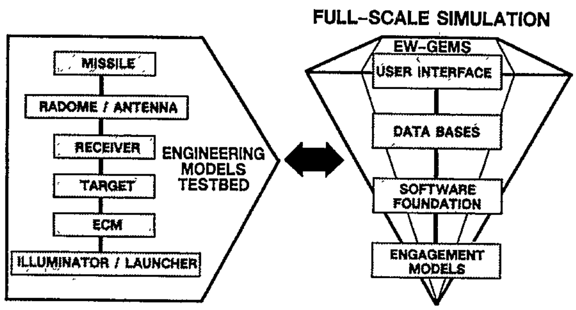


Figure 1: EW-GEMS Development Process

The first path was to take the originally developed models and create an engineering simulation. The second path was to take the originally drafted models and under a formal software development process, prepare them for software implementation in a full scale simulation (EW-GEMS). This method serves many purposes. One is that the engineering simulation actually serves as a test bed for the drafted models. In following the first path of development, the engineering simulation is utilized to determine if the models function properly as separate entities as well as interactively. The engineering simulation provides a means of correcting any models or modifying the original drafts.

The primary benefit of this development process is that those personnel interested in the operability of the models do not have to wait for the completion of the full scale simulation. Once the models have been satisfactorily implemented and debugged in the test bed, time will be saved on the design and implementation of the models in the full scale simulation.

In following the second path, the personnel are concerned with three things. One is formally designing the overall structure of the simulation. This includes the user interface, system utilities, speed of the simulation, and data flow while accommodating all the models. The second is designing the simulation such that the originally drafted models are made to be as mathematically generic as possible. The final concern is the design and development of generic low-level software tools such as a form editor, a menu generator, and a graph generator.

This bi-path development process necessitates a high level of interaction between team members. Routine updates must be made between the two development teams. If a model is changed or updated, the entire development group is aware of the fact. It is emphasized that the idea is not to develop two "independent" simulations, but two "separate" simulations each contributing to a final simulation. This bi-path development process also promotes a team environment by eliminating the subcontractor/contractor relationship which predominately exists between software engineering teams and electrical/physical engineering teams in industry today.

3. EW-GEMS SYSTEM REQUIREMENTS AND DESIGN

The EW-GEMS system requirements and design are user intense. EW-GEMS was designed to be a powerful analysis tool allowing the user full latitude in readily creating or modifying an EW simulated engagement.

The primary user of EW-GEMS is foreseen as being an EW analyst. The level of experience of the user will vary, so EW-GEMS has been designed to accommodate both the EW novice as well as the EW analyst. The EW analyst may, for example, analyze the effects of jamming an air-to-air missile, launched at a target, with some prescribed ECM capability. The analyst effectively creates the scenario by selecting and parameterizing the scenario elements individually.

The creation of the scenario requires that the user be allowed to describe an aircraft, a missile, and an ECM technique in full detail and initialize the geometry (See Figure 2). However the user interface must not only be powerful in providing for this description, but it should be easy to use and not be laborious to the user. The user interface for EW-GEMS was designed using a mix of user input forms, menus, and list selection queries.

The user is allowed to select a tactical aircraft such as an F-14, F-15, etc. as the target. Each target is described in a database in terms of maximum and minimum speed, maximum roll-rate, and RCS. At run-time, the selection of a target automatically triggers the use of associated database information. The user then may enter run-time maneuvers such as a compound maneuver. The user enters the time at which the maneuver is to take place, the associated roll-rate, the g's pulled in a turn, and the direction of the turn (See Figure 3).

The desired threat missile and associated illuminator platform and signal may then be selected by the user. Both surface-to-air and air-to-air missiles are modeled. All missile aerodynamic parameters, characteristic of the selected threat missile, are stored in a database structure and are automatically utilized upon selection of the missile. A threat radar platform carrying the illuminator is selected. The illuminator signal types such as continuous wave, continuous wave-FM, or pulsed signal are stored in the illuminator database. For the illuminator, parameters such as power, gain, frequency, and aperture size are required in the database.

An ECM technique under investigation is then defined.

After selection and parameterization of the scenario elements the scenario geometry may be initialized. The user has the option of initializing in any one of the following three modes: 1) pre missile launch fire control mode, 2) pre missile launch absolute definition mode, and 3) post missile launch condition mode. The first and second initialization modes place the missile at the launcher at time t=0 of the simulation. The third initialization mode places the missile in three space, not necessarily at the

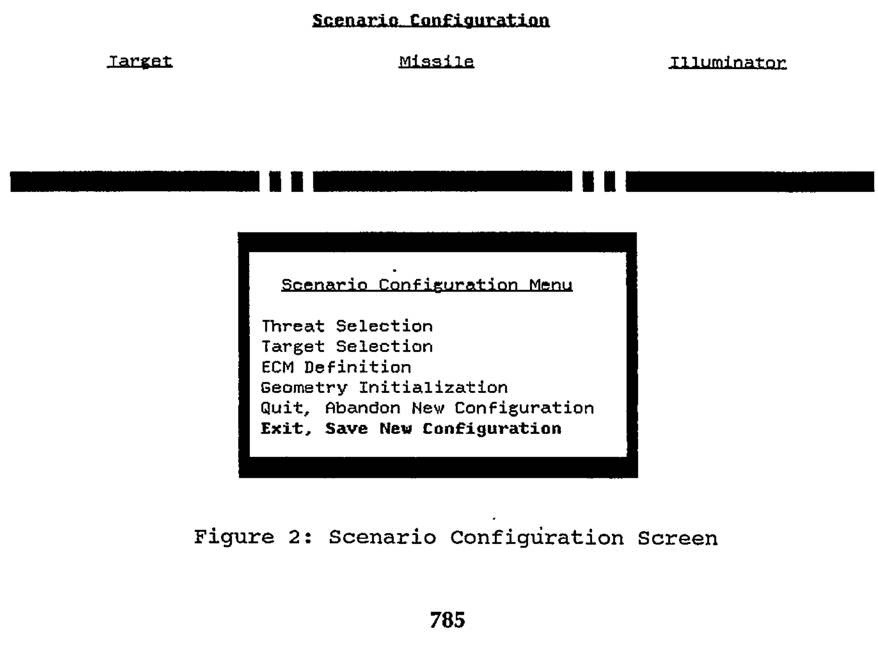


Figure 2: Scenario Configuration Screen

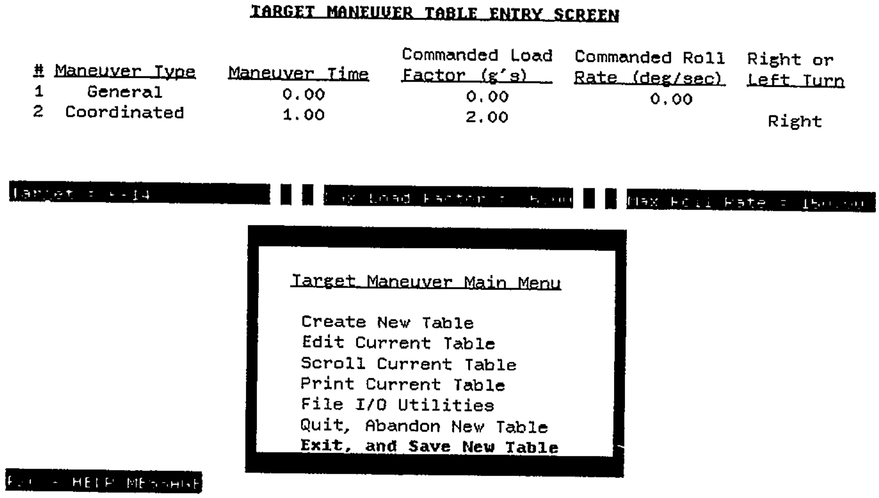


Figure 3: Target Maneuver Entry Screen

launcher (post launch). In all three modes the scenario element position and velocity vectors are initialized.

When the user has entered all required database information, selected the EW engagement scenario elements, defined the target maneuver, and tailored an ECM technique, the engagement may be simulated. The user has two options of running the engagement: 1) open-loop mode, or 2) closed-loop mode. The primary difference between the open-loop mode and the closed-loop mode is that in the open-loop mode the missile is selectively placed at a fixed position and does not flyout. The open-loop mode is designed to mimic an EW test chamber and can thereby be used to validate the simulation results. In the closed-loop mode the missile flies out to engage the target and associated ECM technique. For each mode the user specifies the maximum run-time of the engagement (characteristically in seconds) and the delta time step of the simulation.

A very favorable option made available to the user in the closed-loop run mode is the selection of an end-game mode. The end-game mode allows the user, after an initial run has been made, to select a time at which the scenario is to begin on subsequent runs. For example, suppose the engagement was selected to run for 15 seconds after which the user realized that the last 5 seconds of the 15 second run was of particular interest. The subsequent runs would then dispense with the calculation of the first 10 seconds of missile flyout and just run the engagement scenario for the last 5 seconds of the engagement. No compromise is made by the user. The engagement in subsequent runs is merely initialized with the 10th second geometric, electromagnetic, and aerodynamic data of the first run. The primary advantages in implementing an end-game mode option are conservation of run time and computer storage.

A second type of user of EW-GEMS may be the missile analyst. The EW-GEMS system includes provisions for analysis of individual scenario elements and the functionality of the same. For example, a missile analyst may wish to investigate missile aerodynamics as well as the missile electromagnetic response to an impinging signal. This analysis may be performed without having to run an engagement scenario.

In terms of structure, each mathematical model of EW-GEMS represents individual models depicting geometries, electromagnetic responses, or aerodynamics and as a whole represent the engagement scenario. For example, a generic 6-DOF (six-degree-of-freedom) model exists for the target motion, which includes maneuver capabilities. Details of the desired maneuver are entered by the user at run-time.

A target database structure was constructed containing information describing individual aircraft in terms of maximum and minimum speed, RCS, etc. Likewise, the missile/launcher database contains information describing specific parameters of a missile. Databases serve as a convenient means by which static information may be entered and maintained in a common location.

All inputs of the simulation are of nomenclature and units commonly used in respective fields of flight (target/aircraft description), aerodynamics (missile control), radar, and EW. The system is designed to allow the user to enter parameters which are most likely to be changed run-to-run for the purpose of analysis. Those parameters least likely to be changed run-to-run, will be placed in a database structure. This separation of parameters lends itself to being more user efficient by limiting the barrage of inputs that a user would have to see or enter run-to-run. The scenario configuration inputs and database information together, effectively satisfy the inputs to the models.

The system as a whole was designed to include accurate detailed models of the targets, missiles, and ECM, but not be burdensome to the user in its use. The degree of detail of the models allows the user the versatility and latitude required in a system of this magnitude, and the database structure decreases the user overdose of required run-time inputs. Within these design constraints, the system is being implemented.

4. ENGAGEMENT SCENARIO MODELING DETAIL

EW-GEMS has many physical models which can be subdivided into two major categories; 1) geometric models and 2) electrical models. The geometric models are those which describe the motion of the various vehicles. Electrical models are those which describe the characteristics and interaction of the signals and devices.

Geometric models included in EW-GEMS are: 1) target, 2) missile, 3) missile launcher, and 4) decoy.

Electrical models included in EW-GEMS are: 1) illuminator (transmitter and antenna), 2) missile (radome, antenna, receiver, gimbal, autopilot), 3) target (RCS), and 4) ECM (receiver, antenna, polarizer, transmitter).

The models are detailed to the extent that, for example, any target may be simulated via the target model. The equations of motion for either a small tactical aircraft or a large transport are the same. The two aircraft are differentiated in the target database by virtue of their specific values of maximum and minimum speed, radar cross section, maximum allowable load factor, etc. Once the different aircraft details have been entered in the target database, the user merely selects the target by name in the creation of the scenario engagement. Thus a single generic target model may be used respectively to simulate a multiple number of different aircraft.

The missile, missile launcher, decoy, and ECM technique models are detailed in the same manner, each capable of representing a varied number of scenario elements in their respective classes.

5. BENEFITS OF DETAILED EXACT MODELING

Two important concerns of a simulation developer and the developers of EW-GEMS are the speed of execution and the exactness of the results. These two concerns are characteristically inversely related. If the simulation contains more detailed modeling the results are normally more exact, but the execution time will increase. On the other hand, the execution speed of a simulation determines the throughput and the interactiveness of the system. Therefore faster execution speed means a more efficient simulation and a more productive user.

However, in trying to achieve this execution speed some simulations fall prey to inexact modeling. Some simulation designers either overlook or ignore important events or interactions. The designers usually view these events as insignificant, while the user sees them as the difference between validity and invalidity. This inexactness problem is often caused by over simplification on the part of the system designer.

One major advantage of including detailed exact modeling, as is being done in EW-GEMS, is that the results can be validated by real world measurements. Once the user community acknowledges that the simulation actually does mimic the real scenario, the simulation will be accepted as correct. Only after correctness is established through validation, will the simulation be viewed as an effective tool. In EW-GEMS, only the functionality of each component is modeled exactly. The internal operations of the component may not be of interest because they do not affect the outcome. By modeling the functionality of a component, a black box approach, execution time is decreased while the integrity of the results are left unscathed.

These two concerns of exactness and speed are both critical to the success of the simulation. A tradeoff study must be made on a case-by-case basis in order to determine the detail at which to model.

6. EW-GEMS SOFTWARE DESIGN CONCEPTS

Once the system requirements are well defined and understood and the software design phase of a simulation is about to commence, one should identify certain software design concepts and guidelines that are to be followed. Six such concepts have been identified in the software design of EW-GEMS.

First, EW-GEMS is to be modularly designed so that program parts can be freely interchanged and also so that errors can be isolated, and corrected easily. Second, the program is to be flexible enough to adapt to future modifications and it must not limit the creativity of the EW analyst. If the EW analyst wishes to simulate a new state-of-the-art missile, EW-GEMS should be able to accommodate the user. Third, the design of EW-GEMS is to be robust enough to handle every foreseeable error condition. Fourth, the software designer should maintain simplicity and uniformity in the design of the user interface and also in the design of the entire software product. This will ultimately make the simulation easier to use and the software will be easier to maintain. Fifth, a structured design will be incorporated into EW-GEMS. A structured design ensures that there is only one path into and out of the program or any structured subprogram. Using a structured design will increase the readability of the software and therefore decrease the cost of software maintenance in the long run. Finally, the use of global variables will be restricted to special unique cases and will not be encouraged. By restricting the use of global variables, the probability that any given module will produce side effects decreases significantly.

The concepts listed above are nothing more than good software engineering practice. However, these concepts should be emphasized, because simulations tend to grow into large software systems escalating in complexity and need to be frequently modified and upgraded.

7. ADHERING TO S/W DESIGN CONCEPTS

The overall design structure of EW-GEMS is divided into two major areas: 1) a top-level simulation architecture, and 2) a foundation of generic software tools and low-level routines (Figure 4). Within these two major areas, the EW-GEMS software design concepts are strictly followed.

The top-level simulation architecture is concerned with the computer/user interface and the managing of the entire EW-GEMS process. It will allow the user to configure a scenario, maintain the databases, execute a simulation run, evaluate a scenario component, and use a simulation utility (Figure 5).

The simulation execution subsystem, which executes the engagement scenario, was designed extremely modularly. Because of such a modular design, simulation models, which comprise the execution subsystem, have like interfaces. Each model acts as a transfer function whose only output is the current data frame. These models do not use any global variables and therefore can only affect a limited accessible portion of the data frames.

Data frames contain all pertinent dynamic simulation run-time-data. A data frame is a snapshot of the simulation at a

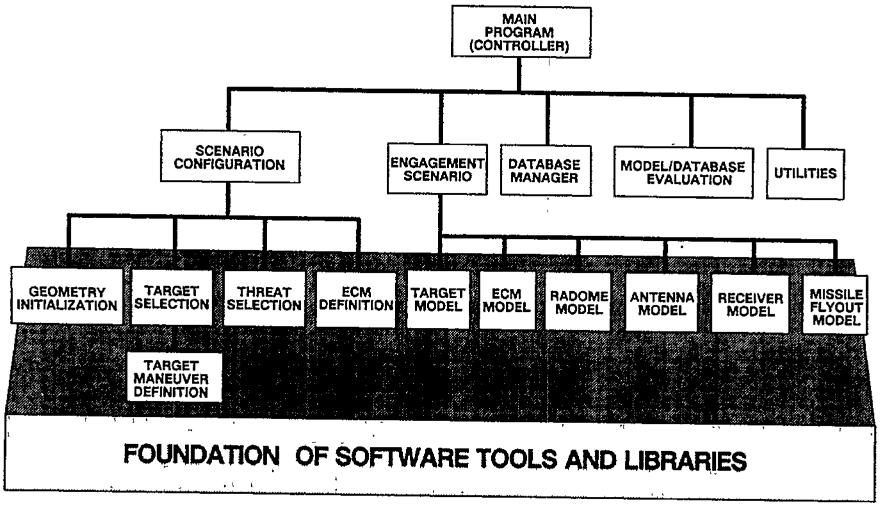


Figure 4: EW-GEMS Software Design Structure

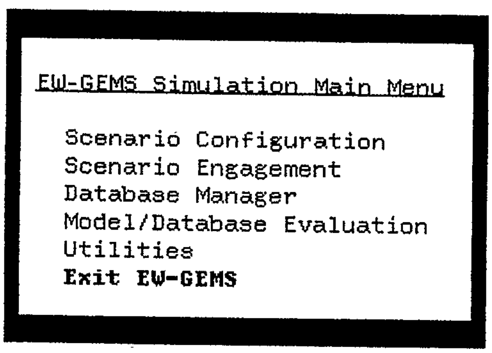


Figure 5: EW-GEMS Main Menu

given time during the simulation (Figure 6). There is a one-to-one correspondence between data frames and simulation time steps. The array of data frames accumulated during a simulation run is used later to generate graphical and tabular output.

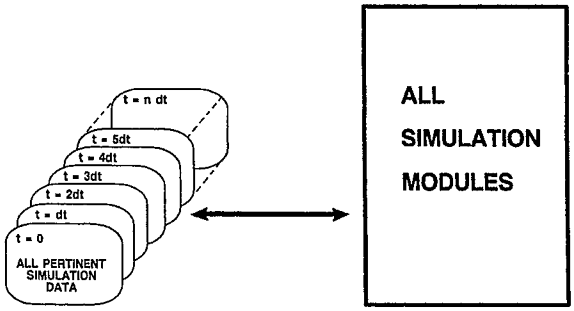


Figure 6: Simulation Data Frames

Since all simulation models have like interfaces, they can be interchanged effortlessly. For example, a slotted waveguide antenna model can be interchanged with a Cassegrain antenna model without producing any negative side effects.

In its final form, the top-level simulation architecture is a modular, flexible structure which will house EW-GEMS. It is designed to be easy to maintain and easy to use.

The second major area of the simulation design is the foundation of software libraries and tools. The foundation corresponds to a bottom-up software design approach.

The design of the EW-GEMS software foundation is concerned with the specific computer hardware and operating system, memory management, and generic software tools. It is not concerned with the actual implementation of EW-GEMS. The foundation performs tasks such as screen handling, keyboard interfacing, operating system interfacing, string handling, and pointer functions. It also contains software tools which are generic modules which perform a specific task to aid in the development of the simulation.

Within the EW-GEMS software foundation exist modules labeled as generic software tools. These tools include a form editor, a menu generator, and a graph generator. Each of these modules perform a task which is required frequently by EW-GEMS. However, these are generic tasks which are not specific to the simulation, and these tools can be reused at any time in the future on any other software program that requires that particular capability. These software tools are not only generic and reusable; they are also extremely powerful. Since the tools only perform one specific function, the software design engineer can focus well on the problem and create a robust solution.

The form editor is a good example of a powerful software tool utilized frequently by EW-GEMS. The form editor is a software tool which creates and implements entry forms. Moreover, it checks the syntactic and semantic validity of each entry in the form, allows the user to scroll freely through the form using the arrow keys, can enable and disable form entries at run-time, can implement a "hot" key facility which is primarily used as a help key, and provides the capability of handling many different data types within one form including enumerated entries.

An enumerated entry is similar to selecting an item from a list. For example, a form is created for personnel information. An entry on the form asks for employment status. If the entry is an enumerated type, it can be defined such that the entry clerk simply uses the arrow keys to scroll through the choices: Active, Terminated, Retired, etc. The entry clerk can only select items from the list, so there is no confusion and no error checking.

Each form created is defined by a form definition record which is loaded and passed to the form editor when called. It is the form definition record which makes a particular form unique. The form editor only creates entry forms to the specification of the definition record. Since all forms are created and managed by the form editor, every form in EW-GEMS has all of the capability described above. The menu generator and the graph generator similarly are definition record driven generic software tools.

Because of the vast amount of data required and generated by a simulation, the design of the major data structures is extremely important. A bad data structure design can complicate and spoil a good overall design. In EW-GEMS, there are three major data structures which contain approximately 90% of the data used in the simulation. The three data structures store the input parameters, the databases, and the data frames (see Figure 7). The input parameter data structure contains all user entries concerning the configuration of an engagement scenario. Some examples of the input parameters are the target selection and maneuver, the threat selection, the geometry initialization, and the ECM definition. The database data structure contains the target, launcher, missile, receiver, radome, antenna, and modulation waveform databases.

Before the development phase of EW-GEMS began, it was decided that this simulation would be a major software system which would have a long usable life. It would need to adapt to unknown future modifications in order to remain valid and useful in this high-tech, dynamic field of Electronic Warfare. It was due to these needs and expectations that a formal software development effort was undertaken. Because of the adherence to good software design

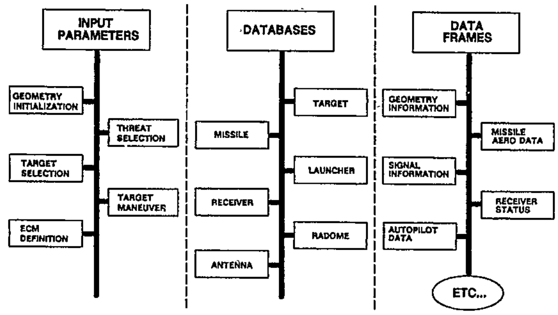


Figure 7: EW-GEMS Data Structures

concepts, the overall design of EW-GEMS is a flexible chassis which should enable the simulation to live a carefree software life-cycle.

8. CONCLUSION

The driving factor behind the system design and software development of EW-GEMS has been and is the development of a usable EW analytical tool. Through good system design the simulated models are proving to be accurate. Through good software development the simulation is functioning efficiently. Finally, by keeping the needs of the user first and foremost, EW-GEMS will be the analysts' tool so highly desired in the EW community.

ACKNOWLEDGEMENT

The authors are very grateful to Mr. Norman Toto and the Avionics Laboratory of the United States Air Force Wright Aeronautical Laboratories for the partial sponsorship of this work.

AUTHORS' BIOGRAPHIES

RAYMOND E. GALLAGHER received a BS degree in Mathematics from Illinois Institute of Technology, and MS and PhD degrees in Electrical Engineering from California Coast University. He also received the degrees of MA and MBA in business from Claremont Graduate School. His background includes experience in analysis and simulation of aerodynamics, fire control radar, and IR and EO seeker systems. He had previously provided program management and technical leadership in these areas at Rockwell International, Hughes Aircraft Co. and Gould, Inc. Dr. Gallagher is presently the Manager of EW Technology at DSC, Inc.

FREDERICK C. CHRISTIE received a BS in Computer Science and a BS in Mathematics from Villanova University. He received his MS degree in Computer Science from Johns Hopkins University in Baltimore. Mr. Christie was previously employed by Gould, Inc. where he worked primarily on software simulations. Mr. Christie is presently a Senior Software Engineer at DSC, Inc.

KENNETH R. KAISER received a BS degree in Electrical Engineering from North Dakota State University. He was previously employed by the U.S. Patent and Trademark Office where he worked in the radar systems technology and subsequently at Gould, Inc. where his responsibilities included the analysis of EW systems and techniques. Presently Mr. Kaiser is a Senior Engineer/EW Analyst with DSC, Inc.

JOHN T. PATTEE received a BS degree in Electrical Engineering from Virginia Polytechnic Institute and State University. He was previously employed by the Boeing Military Airplane Company with primary responsibilities in radar system and software design evaluation and subsequently at Gould, Inc. where his responsibilities included the analysis of EW systems and techniques. Mr. Pattee is currently a Senior Engineer/EW Analyst with DSC, Inc.

Defense Systems Concepts (DSC), Inc., 1340 Ashton Road, Bay A, Hanover, Maryland 21076