Improved Decision Making through Simulation Based Planning

Paul A. Fishwick, Gyooseok Kim, Jin Joo Lee
Computer and Information Science and Engineering Department
University of Florida

Abstract

Real-time military planning and decision making involves several different modeling techniques, including rule-based, operator-based and dynamic-model based approaches. While rule-based approaches are generally fast and are more appropriate for simple scenarios, simulation methods and dynamic models, indigenous to the simulation literature, are necessary to plan within environments involving large-scale uncertainty, multiple interacting elements and complex dynamics. Planning techniques must inter-operate to yield the best decisions, and we have found that simulation based planning serves as an architecture for detailed model levels for both real-time and off-line decision making. We introduce *Simulation Based Planning* as a methodology for addressing the complexity involved in Air Force missions while employing an example of *air interdiction*.

1 Introduction

Decision making and planning are critical operations for all military missions. Moreover, planning occurs over several different time scales depending on the amount of time that one has to plan prior to committing to a particular plan. Planning is a hierarchical enterprise since many techniques can be used to determine near-optimal plans. For example, if one has information on costs between events during a mission, and the goal is to minimize cost, then a mathematical programming approach, based on lowest cost path determination, may yield satisfactory results. An even higher level type of planning is possible by using heuristics in the form of operators and rules [Fikes 72]. Rules can use certainty factors or fuzzy sets.

Our long-term goal is to explore this hierarchy of planning approaches, and our first step toward this goal is to provide high level planners with a technique we call *Simulation Based Planning* (SBP). Military missions involve many interacting elements including concurrently active adversarial tasks and uncertain information regarding ground-based anti-aircraft capability. As the complexity of a mission and knowledge-base increases, it is valuable to use computer simulation and more detailed dynamic models to obtain an answer to the broad question "Which is the best approach to take given our mission and all currently available knowledge[1]?" A good method for answering this question is to use simulation since the simulation technique is generally useful for obtaining answers to "what if" scenarios. The quality of the answer depends on how much time is available prior to committing to the plan. If more time is available, more simulation experiments can be executed in real time. If time is of the essence, higher level models will need to be simulated. In any event, our goal is to plan using the most detailed dynamic models available rather than to limit all planning to the use of a singular planning technique such as a decision tree.

[1] Where the knowledge about the terrain, enemy unit motions and postulated enemy plans incorporate a great deal of uncertainty and can change over time.

Our contribution in the area of planning is to develop a method that allows simulation to be used in real-time, where the simulation is embedded within the decision making system. Consider a simulation system that manages force engagements up to the battalion level. CASTFOREM [CASTFOREM 96] provides one such modeling capability [Wargames 96]. CASTFOREM is driven by decision trees and rules to guide what actions occur at any given time. For interactive graphical output on the state of units, JANUS [Janus 96] can be used. Our goal to to allow a program such as CASTFOREM the ability to make decisions based, not on rules or decision tables, but on multiple simulations that are run *within the overall simulation*. Typically, simulation has been used widely in the military for offline decision analysis and "what if" weapons effectiveness assessment. Our proposal suggests that we enable simulation to support "course of action" (COA) analysis, and embed it directly within the force simulation. The approach yields a two-level simulation procedure: simulations for COA analysis guiding the decisions that drive the simulation of units, platoons, companies and battalions. This "simulation within a simulation" approach is novel, but it can be time consuming. For that reason, our work stresses the use of multi-level models so that different aggregation levels can be executed so that the planning can be performed with real-time constraints.

Planning, regardless of the specific domain, involves three components: 1) model type, 2) plan set, 3) plan evaluation. The first step in planning is to determine the modeling language (or type) to use. For rule-based approaches, this language can be "rules" or "predicate logic." For other approaches, there are many alternatives: equation sets, finite state automata, Petri nets, functional block models, queuing models. Often the model type is visual in structure [Fishwick 95]. The next step is to create a set of candidate plans. For rules, this set is often created through backward chaining. For more detailed model types, the set is created by creating an experimental design and performing simulation (i.e., a type of forward chaining). Plan evaluation is the key step where elements in the plan set are simulated to determine the best plan(s). For our purposes, we view all modeling approaches as being definable hierarchically, so that model types can include both rules and detailed queuing models, for instance, defined at different abstraction levels. These hierarchical model types are termed *multimodels* [Fishwick 95].

We will first discuss the application of simulation-based planning in Section 2: air interdiction. Then, we define the method of simulation-based planning, and finally we illustrate our prototype simulation application which serves as an aid to plan interdiction missions. In Section 3, we focus on route planning using a low-level strike mission on a munitions factory. Information on the implementation is included in Section 4, followed by conclusions in Section 5.

2 Air Interdiction

A typical use of the application of force is air interdiction, where the purpose is to destroy, delay, or disrupt existing enemy surface forces while they are far from friendly surface forces [Drew 92]. The interdiction mission includes attacks against supplies and lines of communication. The objective of the interdiction mission is to reduce the enemy threat by diminishing enemy combat effectiveness or by preventing a buildup of combat capabilities.

To achieve this objective, careful and comprehensive planning is required to isolate an area and to stop all support from reaching the theatre of conflict. One must systematically attack the significant elements of the enemy's logistical structures (transportation lines and centers, supply depots and storage facilities, repair and modification centers, staging areas, and industrial installations) and maintain a high degree of destruction until the desired effect is achieved.

There are two levels of the interdiction plan : the interdiction plan at the theatre level and at the tactical air force level [AFM 1-7 54]. A theatre interdiction plan establishes the general scheme of employment, and enumerates available forces by type and number. The plan also outlines logistical support, delineates force responsibility, establishes the general system of targets, prescribes the priority of target systems, and describes the anticipated results. A theatre-level plan is developed through tactical air force planning. This planning involves the day-to-day conduct of operations for the implementation of the assigned portion of the broad theatre interdiction task. It covers specific and detailed actions of the forces to be employed. A large part of the mission is dependent on which particular route or air corridor is used.

The task of the attack aircraft is to strike the target swiftly and accurately with whatever munitions are carried, and then to return safely to base. To carry out this task, we must penetrate the enemy defense. Most difficulties arise here because methods of penetrating enemy defenses can vary according to the strength and sophistication of the hostile detection, reporting, command and control network, and how much intelligence is available on the nature of the defense capability [Spick 87]. Mission planning will also involve maintaining a balance between fuel and munitions resources to determine the load to be carried. Considering these constraints, selecting the best routes can be a complex undertaking.

2.1 Simulation Based Planning (SBP)

SBP refers to the use of computer simulation [Law 91, Fishwick 95] to aid in the decision making process. In much the same way that adversarial trees are employed for determining the best course of action in board games, SBP uses the same basic iterative approach where a model of an action is executed to determine the efficiency of an input or control decision within the given situation. However, board game trees implement a *static* position evaluation function whereas, in SBP, a model (serving as a *dynamic* evaluation structure) is executed to determine the effects of making a "move." With the ability to simulate models at different abstraction levels, SBP executes detailed models of a physical phenomenon when there is sufficient planning time, or when fast computation and parallel methods are instrumented.

The military has been using simulation-based planning for many decades in the form of *constructive model* simulation. A constructive model is one based on equations of attrition and, possibly, square or hexagon-tiled maps using discrete jumps in both space and time. To decide whether to accept a course of action, one can use a constructive model (a "wargame") to evaluate the alternatives. Related work by Czigler et al. [Czigler 94] demonstrates the usefulness of simulation as a decision tool. Our extension in SBP is one where we permit many levels of abstraction for a model, not just the aggregate abstraction level characterized by Lanchester equations and combat result tables. The idea is to allow the planner the flexibility to balance the need between the required level of detail and the amount of time

given to make a decision.

Although the planning system shown in Figure 1 is divided into 5 functional blocks, we will describe the overall framework in terms of three components: the experimental design component, the output analysis component and the simulation component. Experimental design is a method of choosing which configurations (parameter values) to simulate so that the desired information can be acquired with the minimal amount of simulation [Law 91]. Since we treat uncertainty in the planning domain as random variates based on probability distributions, repeated simulations (i.e., replications) using sampled data are necessary in order to perform the proper analysis, which includes confidence intervals about the mean for each result. We apply both heuristic and standard experimental methods to reduce the overall simulation time in two aspects: 1) in reducing the number of replications, and 2) in reducing the overall computation time spent on a single simulation of a plan on a particular route.

Following the experimental design, we simulate models of individual objects. This component is called Trial (block 2) as shown in Figure 1. Figure 2 displays the lower level model of the Trial block where each entity of the planning domain is modeled individually using the appropriate model type. We assume that there are seven types of physical objects: *BlueAC, RedAC, Radar, SAM, Wx, Tgt, Zones* and one abstract object called *Eval*. The class *BlueAC* stands for Blue Aircraft and *RedAC* for Red Aircraft. *Radar* represents a ground radar site. *SAM* represents a ground Surface-to-Air Missile site. *Wx* represents the weather. *Tgt* represents the target that needs to be destroyed. In the current prototype, the target is the red force munitions factory. *Zones* represent the area of defense zone. For the zones, we assume that there are a set of radars strategically located inside the zone such that when an enemy plane *BlueAC* flies inside the zone, it is detected.

During a typical simulation loop, every object updates its local state and perform actions, which in turn may affect other objects in the following time slice. The last object to be called within a simulation loop is *Eval*. *Eval* is responsible for three functions:

1. Maintaining a consistent "current state" of the world that is an aggregated state of all the current states of each object.

2. Deciding the outcome of inter-object events such as an engagement event between a *BlueAC* and a *RedAC*. By allowing either of the objects involved decide the outcome of an event—which would normally be beyond their control— we would violate symmetry and self containment among objects.

3. Evaluating each situation (from the planner's point of view) for every time slice and maintaining a score which represents the goodness of the plan.

We now perform output analysis using the set of output data produced from the replications using the following blocks: Replicator (block 1), Evaluator (block 3) and Analyzer (block 4). Output analysis is concerned with obtaining the appropriate interpretations of the output data. The Replicator controls the random number streams for each replication. Different random number streams are used for each run, so that the results are independent across runs. We also allow for *common random numbers* (CRN) to provide a controlled environment for comparison among alternatives. This is to eliminate any "environmental

differences" that can exist between different simulations. CRN is a standard variance reduction technique in simulation and we use it across different alternative route plans within the same replication so that we may expedite convergence to the true placement of the mean.

In its simplest form, the Evaluator serves as the accumulator of any relevant simulation data that is produced from the Trial Block. If the objective function within the Trial block produces a set of scores for each alternative, a straightforward evaluation approach is to total the scores produced from the replications for each alternatives. Using the accumulated data produced by the Evaluator block, the Analyzer block calculates the mean, variance and the confidence interval for each alternative. The mean of the replication results serves as the basic "data" point for the response surface representing the goodness of a plan. Variance can be a measure of predictability or stability when the variance is small. Confidence intervals are useful because given a sample output distribution and a confidence level x, the interval states that, within $x\%$ confidence, the true mean lies within the stated interval [Lee 96].

2.2 An Air Interdiction Scenario

As one of the applications of SBP, we have chosen a typical air interdiction scenario, and developed its Simulation Based Planner (C++) and graphical user interface (Tk/Tcl) within our Multimodeling Object-Oriented Simulation Environment (MOOSE) initiative. To illustrate the usefulness of the SBP approach, we consider the air interdiction scenario depicted in Figure 3. Figure 3 defines a scenario with dynamically moving objects. The mission of the blue force aircraft is to destroy a red force munitions factory. There are three Radars (*R1, R2, R3*) and two Surface-to-Air Missile (SAM) sites (*S1, S2*), each with different effective detection ranges. Two red force fighters (*A1, A2*) are located in air defense Zone2 and Zone3 respectively, while one red force fighter (*A3*) is located outside of the air defense zones. At first glance, the problem of guiding the blue force around the radar, SAM and air defense zone coverage, and toward the factory seems like a simple problem in computational geometry. The geometry approach is used frequently for route planning problems. A typical rule might be formed as follows "To locate a path, avoid radar and SAM fields, and avoid fighting against enemy fighters." The problem with this simple approach to route planning is that the reasoning becomes difficult when uncertainty and dynamics are present. This complexity manifests itself as an increasingly large rule base which often proves difficult to create, maintain and verify for consistency.

To illustrate the kind of uncertainty and dynamics which are involved, consider the following available information at some point during the mission.

- Uncertain location and range : Radar *R1* and *R2* have been identified as permanent fixtures, but a land based scout report suggests that *R3* may have mobility. Moreover, the ranges (track, missile, arm range) of SAM site *S1* is well known, but *S2* has been reported to have a better guidance system including swift mobility, improving its surveillance capability.

- Uncertain enemy mission : red force fighter *A1* and *A2* are known to be on a Combat Air Patrol (CAP) mission, since they are always detected around Zone2 and Zone3; however, *A3*'s mission type is unknown.

In these examples, the behavior of each object is simplified as much as possible, since our purpose is to demonstrate how to handle uncertainty in SBP, and not to focus solely on the complex behaviors of objects. However, we have a plan to include the sophisticated behaviors of each object incrementally. This can be considered as an advantage of the SBP approach: the ability to increase the level of detail of the simulation object model as desired.

3 Route Planning Examples and Results

Figure 4 shows two possible routes (*Route1, Route2*) under the environment defined in Figure 3. The goal of blue force aircraft is to destroy the red force munitions factory while satisfying 3 constraints: time or fuel level, safety, and destruction of the target. Given the possible routes, the role of SBP is to choose the best route minimizing time and fuel consumption, and maximizing safety and target destruction. In Figure 4, *Route1* is more attractive than *Route2* if we value mission time above all others, but seems less safe since it is vulnerable to an attack by red fighter *A1*. *Route2* might be considered more safe and achieve higher target destruction than *Route1* by avoiding the attack from fighter *A1* and SAM site *S1*. However, it will be detected by radar *R2*, increasing the probability of losing blue force aircraft or damage to blue force aircraft. Moreover, there is a big chance of being detected by radar *R3* even though its location is uncertain. The table at the lower left of Figure 4 shows the result of the SBP. We display the mean score and the confidence interval half width of each mean at a 90% confidence level. As can be expected, *Route2* is more successful since it avoids direct attacks from the highly destructive enemy fighter and the SAM site (mean score of *Route2*: 69, mean score of *Route1*: -54).

If we delete *Route1* and consider another route based on the result of the previous situation, we may have two routes we want to analyze. Figure 5 illustrates these two candidates. *Route3* was chosen to avoid direct attack from *A1*, but for a short time period it will be detected by *R1*. *Route3* also takes the blue force into the track range of *S1*, but not into its arm or missile range. Being detected in the track range of *S1* does not seem very dangerous since only tracking functions may be performed by *S1*. We can expect its success to depend largely on the result of the samplings for uncertainty factors: specifically, the location and guidance capability of SAM *S2* and the mission type of *A3*. If the powerful guided system of SAM is sampled close to this route, or *A3* has a intercept capability, then the chance of success will be very small. Otherwise, the chance of mission success will be very good. These nondeterministic and stochastic characteristics can be resolved by multiple simulation with varying values for the uncertainty factors. The confidence interval of the mean score of *Route3* is wide in comparison to that of *Route2* due to the reason previously discussed; however, the overall mean score is better than that of *Route2* because of the small chance of being detected by *S2* or intercepted by *A3*.

We can now delete *Route2* and insert a route, *Route4*, which is carefully chosen to minimize the amount of time that a blue force aircraft will be within the detection ranges of *R2* and *R3* as in Figure 6. The result of the SBP shows almost the same mean score for *Route3* and *Route4* (*Route3* : 110.36, *Route4* : 103.08) with *Route3* being slightly better[2]. But

[2]The goal is to maximize the mean score for determining the better plan.

we can select *Route4* as the best overall route based on its more narrow confidence interval (*Route4* : 1.3, *Route3* : 6.0).

4 Implementation

In this section, we briefly introduce an example of the multimodel which we developed for the Air Force Route Simulation and two analysis methods in the multiple simulations for dealing with uncertainties, which arise from Simulation Based Planning. Additional implementation issues and their potential solutions can be found in [Lee 96].

The purpose of the multimodel is to create a heterogeneous collection of connected submodels so that one can simulate different parts of the system at different abstraction levels. The choice of a dynamic selection of abstraction level provides flexibility to the simulation based planning activity; real-time constraints can be met by tuning the multimodel. To implement the multimodel, the generic Route Simulation Model in Figure 1 was instantiated to the Air Force Mission Route Simulator, and each object resides inside the Trial block as in Figure 2. Among the seven types of physical objects, we chose one object, *BlueAC*, to explain how we could capture the object's multimodel behavior. The model presented here is not complete since it has not been validated by a Subject Matter Expert (SME). However, the model represents the kind of model that one could obtain from the SME through knowledge acquisition methods. Since building sophisticated and realistic models is not the issue in our current research, simple yet sensible models were built to prove our SBP approach.

The toplevel model of the *BlueAC* object is shown in Figure 7. It is modeled as an FSA with three phases: *Approach Target*, *Return to Base* and *End Mission*. Figure 8 shows the refinement FSA for *Approach Target* phase in Figure 7. Going another level down from the *Traverse Route* phase, Figure 9 illustrates the functional block model for updating the location while traversing the route. Figure 10 illustrates the refinement of *RedAC Alert Mode* in Figure 8.

Assuming that a set of alternate routes and environment data are given through the GUI, dynamic models are simulated and evaluated for each route. The simulation process is replicated and its output results are accumulated and then analyzed by the Analyzer (ref. Figure 1).

For the object, we categorized the uncertainty into several types.

- uncertainty of existence: the object may or may not exist.
- uncertainty of location: an area of uncertainty of the object's location is available but the exact location of the object is uncertain.
- uncertainty of range: the exact detection range or firing range is not known.
- uncertainty of mission: the exact mission type of an object is unknown.
- uncertainty of fire power: the destruction capability of the object is uncertain.

These nondeterministic and stochastic characteristics were resolved by multiple simulations using different samplings of the uncertainty factors. The planning problem becomes

one in optimization for an objective function representing the cost of traversing a route. This cost is currently a function of elapsed time, remaining strength of the unit and the level of success regarding achieving the goal. To reduce the total number of replications in the simulation, we used two different output analysis methods: *iterative* and *non-iterative*. The *iterative* method attempts to quantify significant pairwise differences among the alternatives' means within a given confidence interval. The method is referred to as "*iterative*" because he algorithm *iterates* performing for every iteration, a set number of replications and analyzing data to see if there are any significant differences among each route. Whenever a route is found that is significantly worse than all other routes, this route is then eliminated. The iteration continues until only two routes remain and a difference exists between the two of them.

The *Non-iterative* method is a method that avoids making an unnecessary number of replications to resolve what may be an unimportant difference. When two alternatives are close, we may not care if we erroneously choose one system (the one that may be slightly worse) over the other (the one that is slightly better). Thus, given a correct selection probability P and the indifference amount D, the method calculates how many more replications are necessary to make a selection with the probability of at least P, the expected score of the selected alternative will be no smaller than by a margin of D. In our experiment, we have chosen $P = 0.95$ and $D = 13$. A smaller D will produce more accurate results, but with many more replications.

Recently, we have begun construction of a system, called MOOSE, to enable users to interactively specify multimodels through a modeling window. Output is viewed via a scenario window, similar to those shown in Figures 3-6. MOOSE (Multimodeling Object-Oriented Simulation Environment) represents an implementation for a simulation system that is under construction, and based on an extension to object oriented design (`http://www.cis.ufl.edu/~fishwick/tr/tr96-026.html`). MOOSE is the next generation of SimPack (`http://www.cis.ufl.edu/~fishwick/simpack/simpack.html`, which was initiated in 1990 for providing a general purpose toolkit of C and C++ libraries for discrete-event and continuous simulation.

5 Conclusions

We have discussed the method of simulation-based planning within the confines of an air interdiction example. Our view is not that SBP replaces other forms of planning, but that this new approach can be used in conjunction with existing, higher level planning approaches. This way, given a set of alternatives to consider, SBP is able to extend the planning horizon in three aspects: probabilistic uncertainty is handled through detailed and replicated simulation of models rather than solving them analytically using probability theory; it extends the level of reasoning to a finer level of granularity, producing plans that are closer to the level of execution and discovering subtleties that may be missed by a higher level planner; and finally, it breaks down the complexity of multiagent adversarial planning by employing object-oriented multimodel simulation.

Once the simulation results have been produced, the data can be analyzed and interpreted in several ways to choose the "best" plan. For instance, we can choose the plan which has

not only a good mean score but also the minimum confidence interval width to ensure that it is the safest plan possible. We may also decide to choose a plan that has the most number of highest scores even though the confidence interval width may be large in order to select a plan that has the best potential in spite of risks involved. We can even decide to choose a plan at random (given that the scores are above some threshold) which will produce nondeterministic planning. This is particularly useful for mission planning—opposing forces should not be able to predict one's plan. In addition, similar to how simulation is used for visualization, simulation can be easily used to perform visual playback of how a plan was simulated to explain the planner's decision. This can be very useful for the military since much of the military training is done through *after action review*.

Prior to the advent of fast low-cost personal computers, few researchers would consider simulation of a fairly extensive experimental design to be a possible candidate for real-time mission planning. However, as the speed of low-cost computers increases, the simulation-based planning technique presents itself in a more attractive light. Our longer range goal is to explicitly link several plan model levels together so that, for instance, a rule-based level can be identified from a lower-level simulation. One of the authors (Kim) is studying effective consistency measures which will rectify differences in rules produced empirically (through knowledge acquisition) and rules generated automatically from multiple low-level simulations.

6 Acknowledgments

We would like to acknowledge Karen Alguire, Project Manager, at Rome Laboratory, and sponsorship under United States Air Force grant F30602-95-1-0031.

7 Biographies

Paul A. Fishwick is an Associate Professor in the Department of Computer and Information Science and Engineering at the University of Florida. He received the BS in Mathematics from the Pennsylvania State University, MS in Applied Science from the College of William and Mary, and PhD in Computer and Information Science from the University of Pennsylvania in 1986. He also has six years of industrial/government production and research experience working at Newport News Shipbuilding and Dry Dock Co. (doing CAD/CAM parts definition research) and at NASA Langley Research Center (studying engineering data base models for structural engineering). His research interests are in computer simulation modeling and analysis methods for complex systems. He is a senior member of the IEEE and the Society for Computer Simulation. He is also a member of the IEEE Society for Systems, Man and Cybernetics, ACM and AAAI. Dr. Fishwick founded the `comp.simulation` Internet news group (Simulation Digest) in 1987, which now serves over 15,000 subscribers. He was chairman of the IEEE Computer Society technical committee on simulation (TCSIM) for two years (1988-1990) and he is on the editorial boards of several journals including the *ACM Transactions on Modeling and Computer Simulation*, *IEEE Transactions on Systems, Man and Cybernetics*, *The Transactions of the Society for Computer Simulation*, *Interna-*

tional Journal of Computer Simulation, and the *Journal of Systems Engineering*. Dr. Fishwick's WWW home page is `http://www.cise.ufl.edu/~fishwick` and his E-mail address is `fishwick@cise.ufl.edu`.

Gyooseok Kim received the B.S. degree in Electronics from Korean Air Force Academy in 1984 and the M.S. degree in Computer Science from Korean National Defense College in 1989. He served in the Korean Air Force as a chief programmer of the Korean Air Defense System. He is currently pursuing a doctoral degree in the Computer and Information Science and Engineering department at the University of Florida. His research interests are focused on knowledge acquisition and validation within qualitative and quantitative simulation. Mr. Kim's E-mail address is `kgs@cise.ufl.edu`.

Jin Joo Lee received the B.S. degree in Computer Science from Ewha Womans University, Korea in 1988 and the M.S. degree in Computer Science from Brown University in 1991. After receiving the M.S. degree, she was a research engineer at Human Computers Inc., Korea until 1992. She received a PhD degree in the Computer and Information Science and Engineering department at the University of Florida in 1996. Her research interests are in AI planning, simulation and control. Ms. Lee's WWW home page is `http://www.cise.ufl.edu/~jl1` and her E-mail address is `jl1@cis.ufl.edu`.

References

[AFM 1-7 54] Department of the Air Force. *Air force Manual No. 1-7 : Theatre Air Forces in Counter Air, Interdiction And Close Air Support Operations.* Department of the Air Force, 1954.

[CASTFOREM 96] CASTFOREM, Web Reference: `http://hp01.arc.iquest.com/mosaic/060.html`

[Czigler 94] M. Czigler, S. Downes-Martin and D. Panagos. Fast Futures Contingency Simulation: A "What If" Tool for Exploring Alternative Plans, In Proceedings of the 1994 SCS Simulation MultiConference, San Diego, CA, 1994.

[Drew 92] D. M. Drew. *Air Force Manual 1-1: Basic Aerospace Doctrine of the United States Air Force*, Volume II, Department of the Air Force, 1992.

[Fikes 72] R.E. Fikes, P.E. Hart, and N.J. Nilsson. Learning and Executing Generalized Robot Plans, *Artificial Intelligence*,3,1972.

[Fishwick 95] Paul A. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*, Prentice Hall, 1995.

[Janus 96] JANUS, Web Reference: `http://hp01.arc.iquest.com/war/janus.html`

[Law 91] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis* McGraw-Hill, 1991

[Lee 93] J.J. Lee, W.D. Norris and P.A. Fishwick. An Object-Oriented Multimodeling Design for Integrating Simulation and Planning Tasks, In *Journal of Systems Engineering*, 3, 220-235, 1993.

[Lee 94] J.J. Lee and P.A. Fishwick. Real-Time Simulation-Based Planning for Computer Generated Force Simulation, *Simulation*,299-315, 1994.

[Lee 95] J.J. Lee. and P. A. Fishwick *Simulation-Based Real-Time Decision Making for Route Planning.* In Proceedings of the 1995 Winter Simulation Conference, 1087-1095, 1995

[Lee 96] J.J. Lee. *A Simulation-Based Approach for Decision Making and Route Planning.* PhD Thesis, University of Florida, 1996.

[Russell 95] S.J. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*, Prentice-Hall, 1995.

[Schoppers 87] M. Schoppers. Universal Plans for Reactive Robots in Unpredictable Domains, In *Int. Joint Conference on Artificial Intelligence*,1987.

[Spick 87] M. Spick. *An Illustrated Guide to Modern Attack Aircraft.* An Arco Military Book, Prentice Hall Press, New York, 1987

[Salisbury 93] M. Salisbury and H. Tallis. Automated Planning and Replanning for Battlefield Simulation, In Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation,1993,243-254,Orlando,FL.

[Wargames 96] Wargame Catalog, Web Reference: `http://hp01.arc.iquest.com/war/war.html`

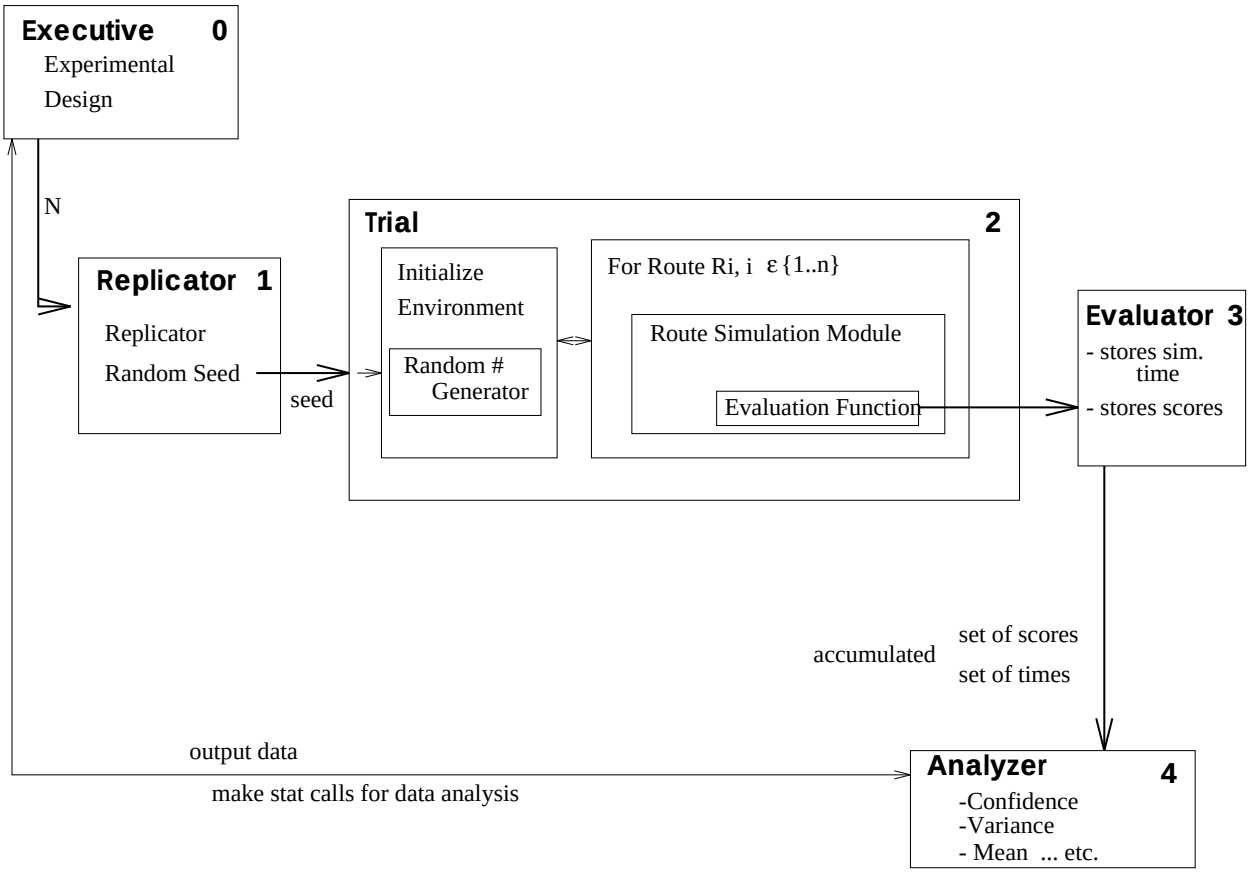

Figure 1: Generic Top Level Architecture of a Route Planner

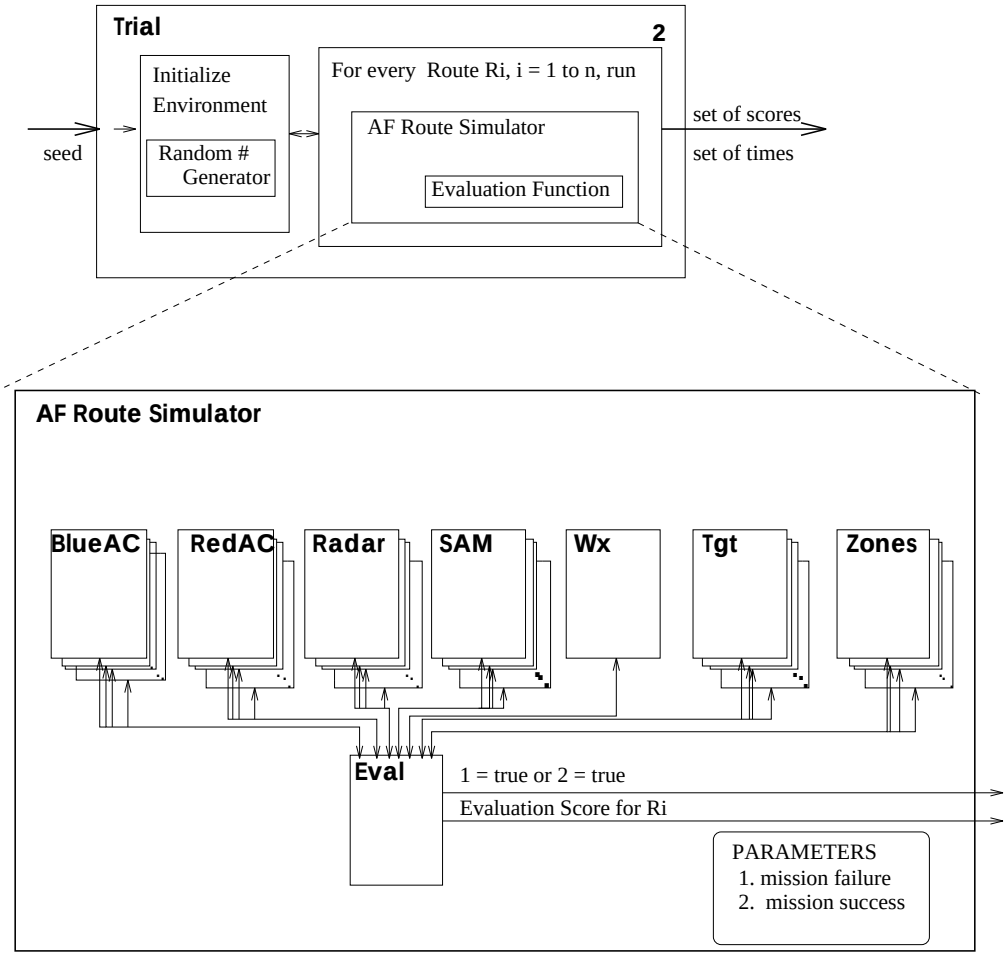

Figure 2: General Simulator Module

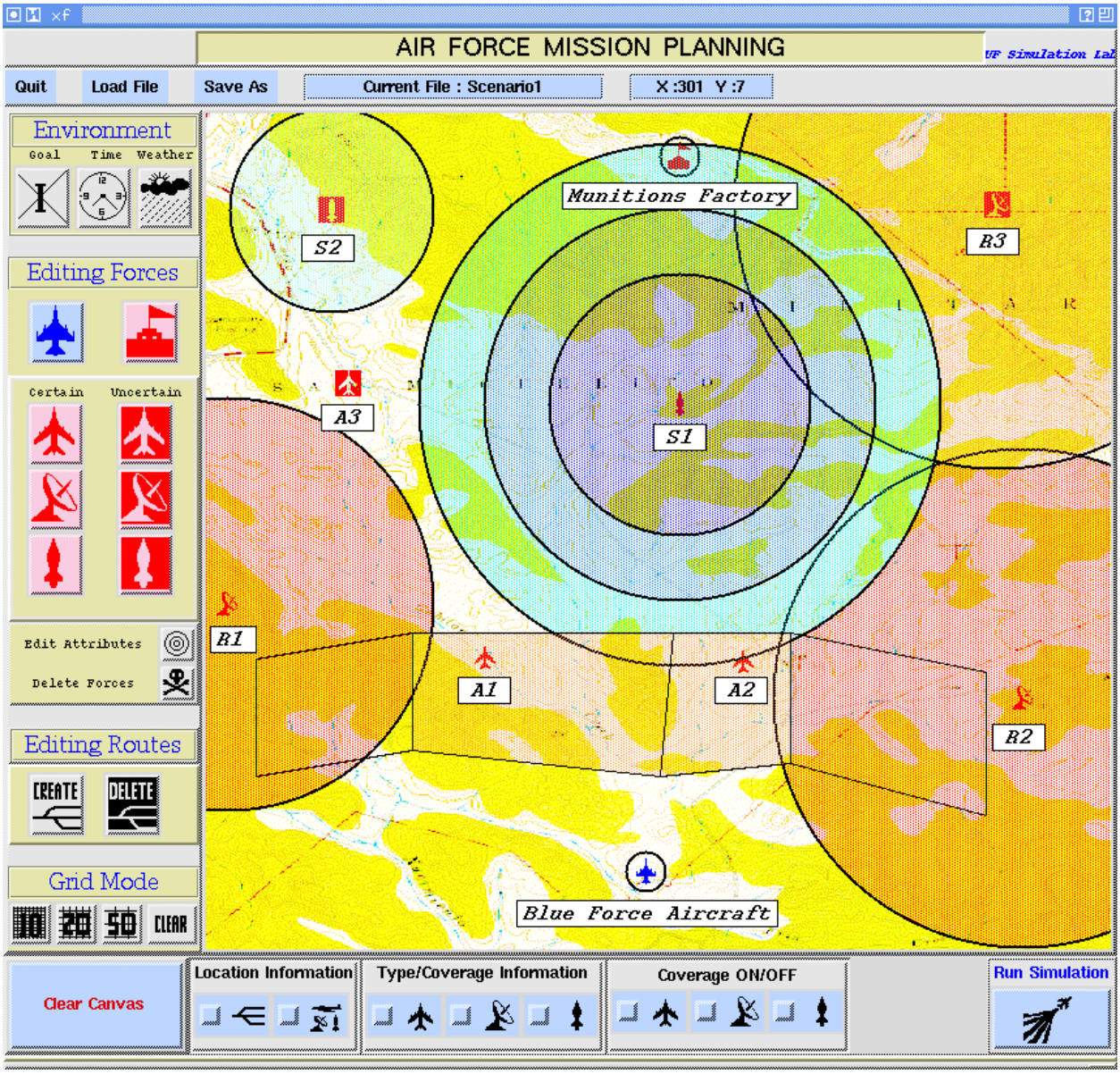

Figure 3: A Typical Air Interdiction Scenario

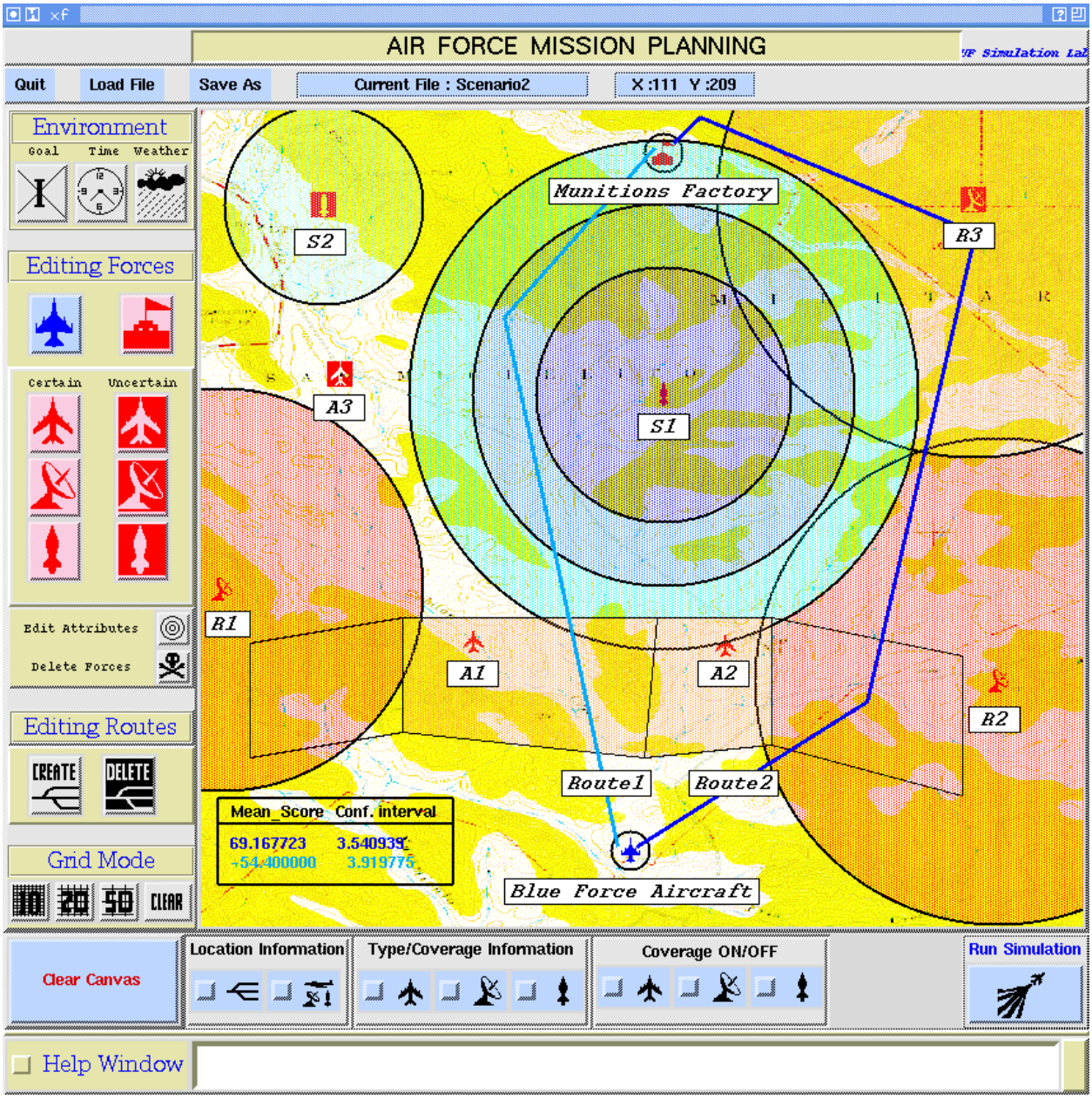

Figure 4: Two Possible Routes in the Figure 3

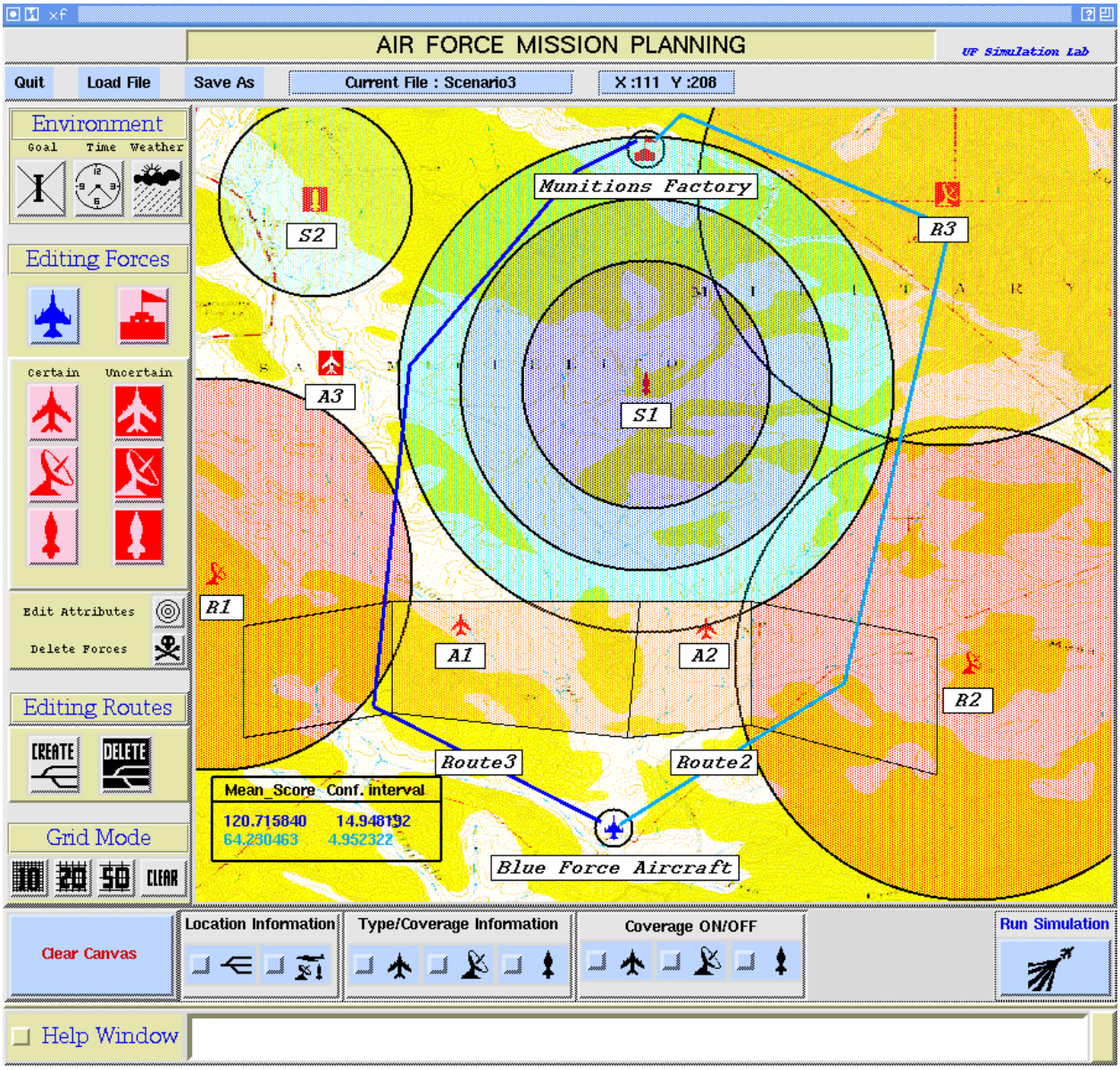

Figure 5: Deleting Route1 in the Figure 4, and Inserting Route3

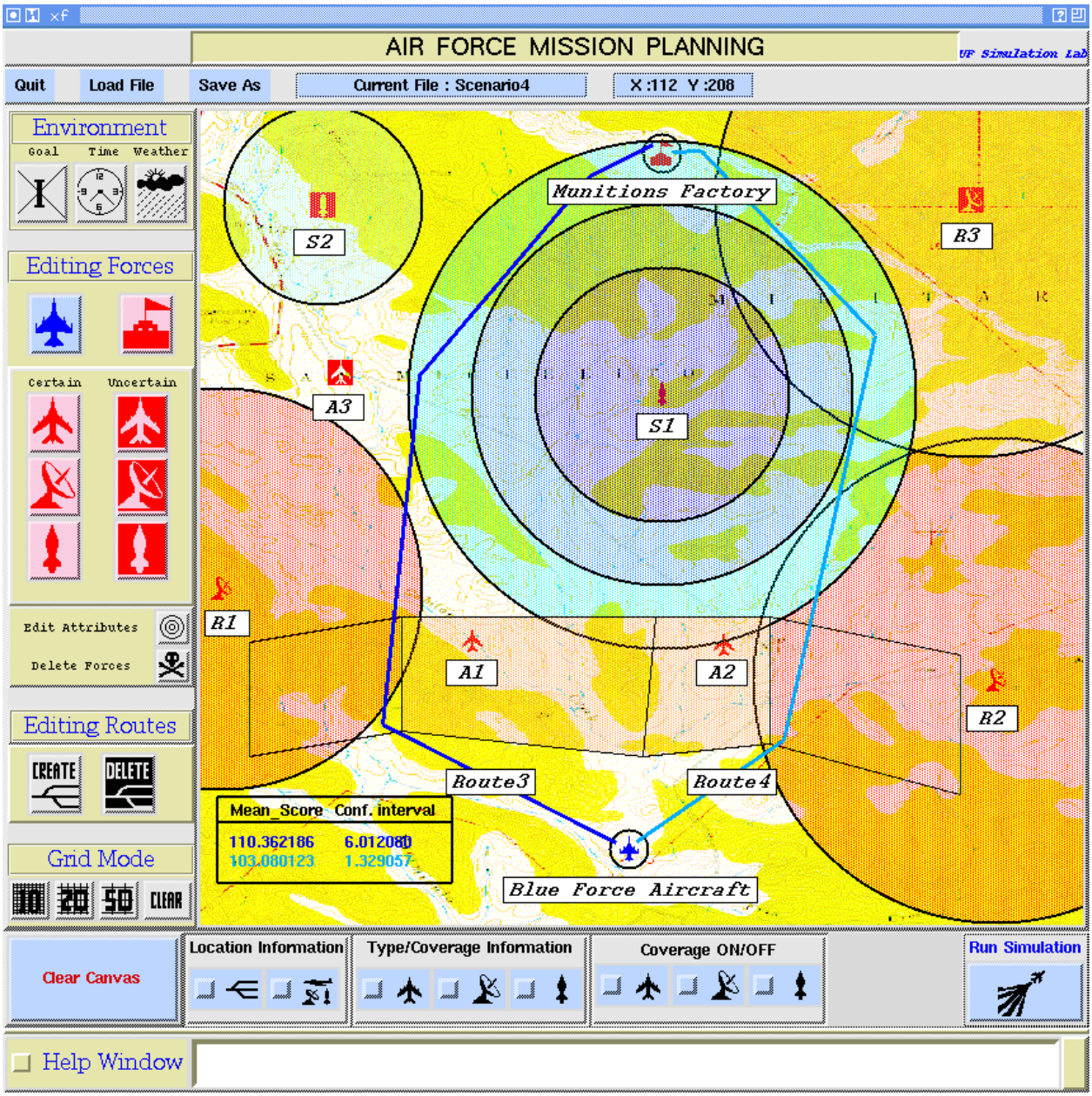

Figure 6: Deleting Route2 in the Figure 5, and Inserting Route4

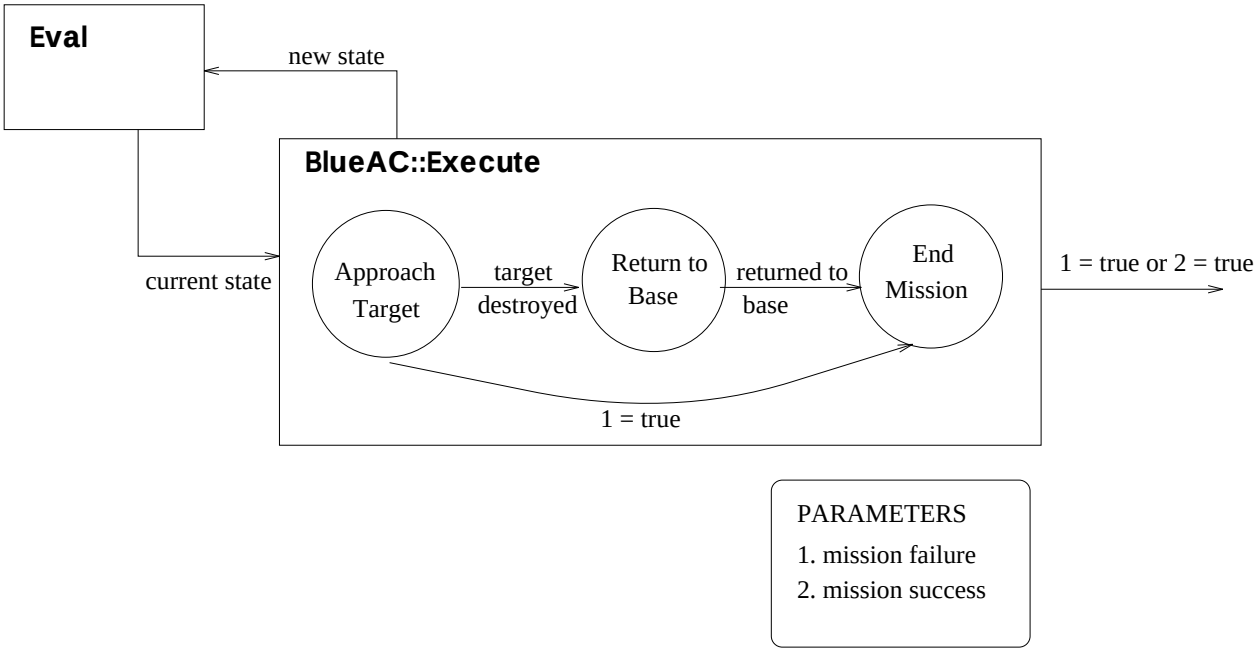

Figure 7: Blue Aircraft($BlueAC$) object model

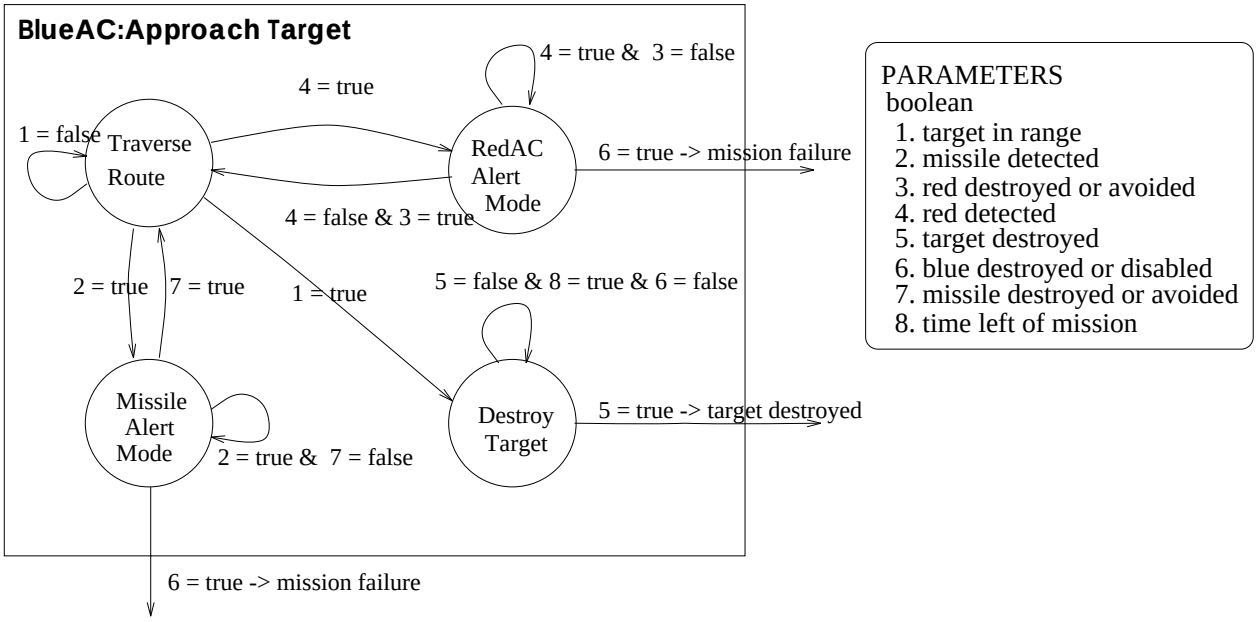

Figure 8: Approach Target for $BlueAC$

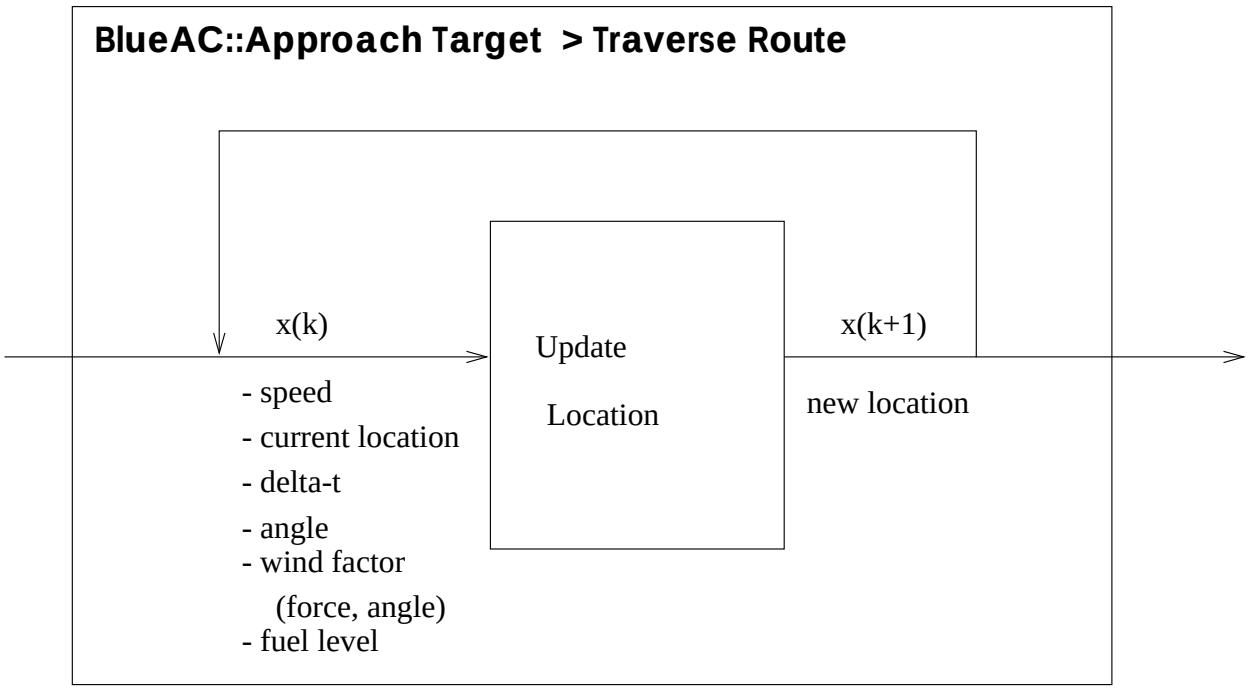

Figure 9: Traverse Route Function for *BlueAC::Approach Target*

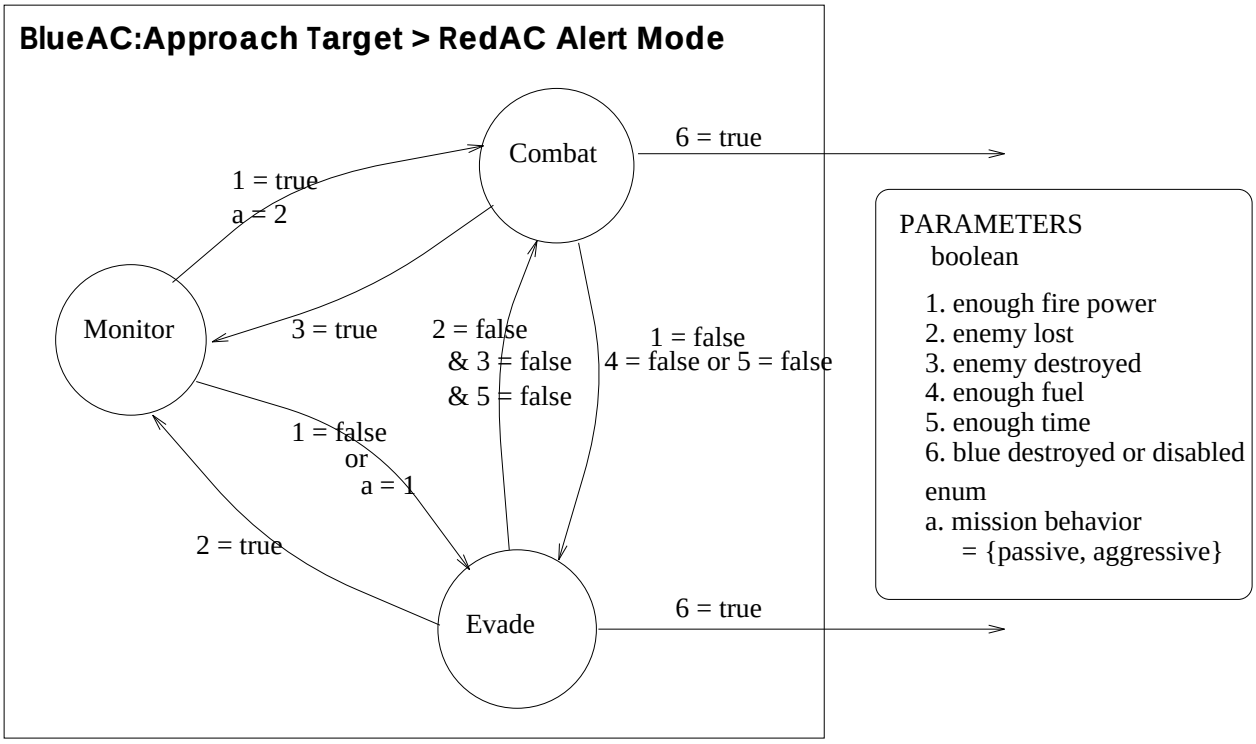

Figure 10: Red Aircraft($RedAC$) Alert Mode for $BlueAC$ object