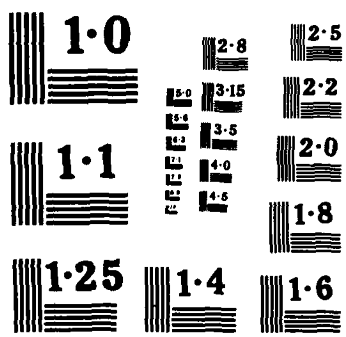
AD-A158 337 AUTOMATED WAR GAMING: AN OVERVIEW OF THE RAND STRATEGY ASSESSMENT CENTER(U) RAND CORP SANTA MONICA CA H J SHUKIAR MAY 85 RAND/P-7085 1/1

UNCLASSIFIED F/G 15/7 NL

END

FILMED

DTIC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

AUTOMATED WAR GAMING:
AN OVERVIEW OF THE RAND STRATEGY ASSESSMENT CENTER

Herbert J. Shukiar

May 1985

P-7085

85 8 26 046

The Rand Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

AUTOMATED WAR GAMING:
AN OVERVIEW OF THE RAND STRATEGY ASSESSMENT CENTER*

Herbert J. Shukiar

The Rand Corporation

Santa Monica, California 90406-2138

*This paper was prepared for an invited address at the C^2 Battle Management session of the *Artificial Intelligence Applications to the Battlefield Seminar* held at Fort Monmouth, New Jersey, May 21-22, 1985. The seminar was sponsored by the Armed Forces Communications and Electronics Association (AFCEA).

PREFACE

The Rand Strategy Assessment Center (RSAC) is a highly ambitious, multiyear research program to develop a system and methodology combining the systematic reproducibility of analytic modeling/simulation with the contextual and behavioral richness of war gaming. The system integrates an adaptive simulation model of precombat and combat operations with three rule-based artificial intelligence agents that represent *Red*, *Blue*, and *nonsuperpower* behavior.

The paper first outlines the RSAC's origins and objectives. It next describes the hierarchical decisionmaking structure within which superpower decisions are made as well as the interaction among the rule-based agents and the simulation model. It then describes in some detail the high-level decisionmaking procedure. With an example the paper next describes the mechanisms used at the lowest decisionmaking level. The paper then turns to the issue of human intervention and closes with a brief description of the next steps in the RSAC system's development and transfer to the government.

SUMMARY

The RSAC grew from a 1978 Defense Science Board concern about the state of the strategy assessment art, conducted at that time on two separate but related paths. War gaming was contextually rich but lacked systematic reproducibility. Analytic modeling/simulation provided systematic reproducibility but lacked contextual richness. The RSAC is developing an *automated* war-gaming system that integrates both paths, capturing the best features of each.

~~To capture war gaming's contextual richness while still providing systematic reproducibility~~, the RSAC replaces human teams with rule-based expert systems of *Red, Blue*, and *nonsuperpower* behavior. Red and Blue decisionmaking is structured hierarchically, in a manner that reflects the Red and Blue command structures as well as their command and control characteristics.

The *National Command Level (NCL)* is the highest superpower decisionmaking level. When it *awakens* it *examines* the complete world situation to determine context and develop escalation guidance. It then determines new objectives and strategy, and it finally selects and evaluates (through the mechanism of look-aheads) an *analytic war plan* that indicates what the NCL's subordinate command levels must do. During its deliberations the NCL can *learn* about its adversary's nature, adjusting its behavior in response to the knowledge thus gained.

The *Global Command Level (GCL)* is responsible for coordinating the activities of the war plans associated with each active region of the world. It also handles all communications with other countries.

The *Supertheater Command Level (SCL)* is also a coordinating plan, responsible for those world regions having more than one active theater. For example, Europe contains three theaters: Northern, Central and Southern Europe, and the European SCL serves the SACEur function.

The *Area Command Level (ACL)* is the lowest superpower decisionmaking level. It handles precombat preparations and combat activities in a specific area, e.g., Central Europe. It breaks the military campaign into precombat, combat, and termination phases,

issuing orders to the simulation model appropriate to each phase. When conditions occur that are inconsistent with the ACL plan's premises, the ACL requests guidance from its superior.

The *Operational Command Level (OCL)* is responsible for the daily detailed combat management adjustments needed to ensure a reasonable execution of the war. It is also a superpower decisionmaking level, but the human rule-writer is generally not interested in the details of combat management.

The *Force Agent*, the RSAC system's analytic modeling element, simulates precombat and combat operations. It is not a rule-based agent but rather a goal-directed simulation model. It accepts orders from the rule-based agents and tries to move military assets from their current states to the states indicated in the rule-based agent orders.

While the RSAC system is an automated war-gaming system, humans can intervene at all decisionmaking levels. Through this intervention the humans can participate in the exercise, i.e., the RSAC system can support exercises with human teams.

While much development work remains to turn the RSAC prototype into a mature production system, the prototype system can support substantive research today. Technical and methodological challenges continue to abound, and the maturation process will be a long one, but work is already under way to utilize the system. Work is also under way to transfer partial system capability to the government.

I. INTRODUCTION

The Rand Strategy Assessment Center (RSAC) has successfully completed an intensive two-year development effort to design and implement a prototype automated war-gaming system.[1] Although enormous improvements will occur over the next few years, even the prototype system, which combines distinctive features of war gaming and analytic modeling, is sufficient to support a number of long-term strategic analysis research projects, and parts of the system have already been used to study NATO defense issues.

This paper describes the RSAC and the system therein developed. We first describe the RSAC's origin and objectives, tracing the RSAC's roots to a 1978 Defense Science Board review of the state of the nuclear strategy analysis art. The Defense Science Board noted some important deficiencies and recommended that the Secretary of Defense take steps to reduce those deficiencies. Next, we present a software system overview, but from a decisionmaking rather than a software system perspective, i.e., we examine the RSAC system's decisionmaking architecture. Indeed, the system's decision elements are artificial intelligence rule-based agents designed to reflect U.S. and Soviet decisionmaking hierarchies.[2] These rule-based agents are integrated with a highly aggregated, flexible, and sophisticated continuous simulation model that simulates the precombat and combat operations of opposing forces. Next, we examine each decisionmaking level in some detail, describing the types of decisions made and the mechanisms for implementing those decisions. Finally, we look at the near-term future and the challenges facing the RSAC today.

---

[1]For RSAC objectives and a general approach, see Paul K. Davis and James A. Winnefeld, *The Rand Strategy Assessment Center: An Overview and Interim Conclusions about Utility and Development Options*, The Rand Corporation, R-2945-DNA, March 1983.

[2]For a description of the RSAC's hierarchical decisionmaking approach, see Paul K. Davis, *Rand's Experience in Applying Artificial Intelligence Techniques to Strategic-Level Military-Political War Gaming*, The Rand Corporation, P-6977, April 1984.

II. ORIGINS AND OBJECTIVES

RSAC ORIGINS

In 1978 (and indeed today) this country's strategic analysis capability rested on two independent but modestly overlapping tracks. The first track utilizes *war gaming* with human teams to examine the interaction of U.S. and Soviet strategy and tactics. These human teams can represent strategic and/or tactical decisionmaking entities, and the resulting war games provide contextual richness and realism. This richness and realism stem both from the ability to treat political and behavioral issues and from the ease with which unexpected events can be placed within the context of a larger conflict. A team's decisions often reflect the team's concern about political consequences and the opposing team's responses, with the threat of geographical and nuclear escalation being only two examples of such responses.

Important insights spring forth from the conduct of such war games. But while we recognize the relevance of such insights, we cannot usually extend these insights to general truths. Different teams respond to the same scenario differently. Further, there is only limited capability to ask such *what if* questions as, *"What if, early during the game, you had chosen this course of action instead of the one you picked?" "What if, at the game's outset, the initiating scenario was like this instead of the one we gave you?"* A human team could certainly respond to such queries, but the fact that they have already *played* the game provides an unwanted bias that colors the response. While we enjoy contextual richness and realism, we lack systematic reproducibility.

The second strategic analysis track utilizes *analytic modeling/simulation*. Using sometimes very sophisticated simulation models, analysts can examine how systematic changes to independent variables affect other dependent variables. Analysts can *explore the response surface* in systematic ways, developing intuition about the relationships among the various explanatory variables.

While these models often have very sophisticated analytic foundations, they lack the behavioral elements inherent in war gaming. In the behavioral sense, the analytic models are unrealistic and quite sterile. It is easy to ask *what if* questions based on model parameters, but it is impossible to ask *what if* questions based on behavioral assumptions. As a result, analysis based on such models frequently reduces to *bean count* comparisons or exchange calculations, lacking the contextual richness and realism found in war gaming.[1]

The Defense Science Board in 1978 noted the lack of an integrated methodology combining the best features of war gaming and analytic modeling and recommended that government take the necessary steps to develop such a capability. This ultimately led to the creation of a Strategy Assessment Center within The Rand Corporation and the development of the strategy assessment system described in this paper.

RSAC OBJECTIVES

The RSAC has four related and general objectives.[2] First, we wish to provide an *integrating framework within which to* analyze worldwide military strategy, from localized, limited conventional conflicts to general, prolonged conflicts and nuclear war. Second, we wish to lay the groundwork for the conduct of *multiscenario analysis*, asking the imprecise behavioral *what if* questions as well as the highly precise analytic *what if* questions. This implies the ability to characterize various U.S. and Soviet decisionmaking styles.

Third, we wish to increase *analysis realism* by treating such often ignored factors as operational constraints, asymmetries in U.S. and Soviet objectives, attitudes, and military style, and the effects of third-country decisions on military operations. And, fourth, we wish to better understand a situation's *strategic dynamics* at various decision

[1]We overstate this point to some extent. It is, of course, possible to design an analytic model to include different adversary responses to military conflict, with the analyst able to select the set of such responses he wishes to examine. What is lacking, however, is the rich set of responses a human team can enjoy, with the team altering its approach based on its war-game-gained experience.

[2]See Davis and Winnefeld, op. cit.

points, looking at the specific interaction of U.S. and Soviet strategy in specific situations as well as the possible second-order effects not readily revealed by traditional analysis techniques.

III. THE RSAC SYSTEM'S DECISIONMAKING ARCHITECTURE

The RSAC system integrates war gaming and analytic modeling by replacing war-gaming's human teams with rule-based agents of Soviet, U.S., and third-country behavior, called the *Red, Blue*, and *Scenario Agents*,[1] respectively. These agents interact with a highly aggregated and flexible simulation model of precombat and combat operations, called the Force Agent. The Force Agent is a sophisticated model in its own right, can be operated in stand-alone mode with a human operator, and is currently supporting a NATO defense study.

Both Red and Blue Agents contain several hierarchical levels, each level being a rule-based agent itself. Each level reflects a decisionmaking level in the agent's command hierarchy, and these hierarchies impose a degree of realism in the Red and Blue Agents' representation of Soviet and U.S. decisionmaking.

DECISIONMAKING HIERARCHY

Figure 1 illustrates the several decisionmaking levels within a Red or Blue Agent as well as the mechanisms whereby each decisionmaking level can observe the unfolding exercise[2] and awaken to respond to specific situations. The figure shows the four major rule-based decisionmaking levels, namely the National Command Level (NCL), the Global Command Level (GCL), the Supertheater Command Level (SCL), and the Area Command Level (ACL). The rules associated with these decisionmaking levels are written in ABEL, a language developed specifically for RSAC rule-writing. ABEL's intent is to permit

---

[1]This paper will not discuss the Scenario Agent and the influence that third-country actions can have on an automated war game's outcome. Interested readers are referred to William L. Schwabe, *Strategic Analysis as Though Nonsuperpowers Matter*, The Rand Corporation, N-1997-DNA, June 1983; and William L. Schwabe and Lewis M. Jamison, *A Rule-Based Policy-Level Model of Nonsuperpower Behavior in Strategic Conflicts*, The Rand Corporation, R-2962-DNA, December 1982.

[2]We think of the RSAC system as a mechanism for conducting automated war games, and we therefore use the term *exercise* instead of simulation.

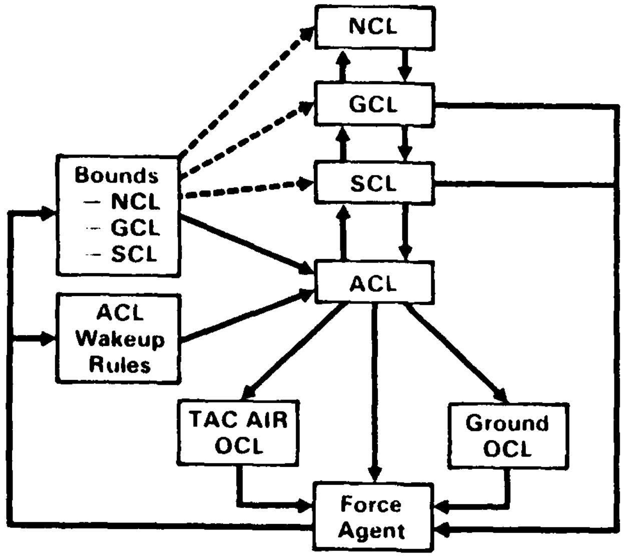


Fig. 1 -- Hierarchical Decisionmaking

substantive experts to review and critically comment on rules, modify those rules, and develop thereby new rule-sets.[3] Using the United States as a descriptive model, the NCL represents the National Command Authority, i.e., the President and the National Security Council. The NCL's function is to examine the current global situation and select a course of action in response to that situation. In selecting its response, the NCL effectively chooses an *analytic war plan* (AWP). The AWP is really a set of operations plans for each active theater plus the coordinating plans that ensure the cooperative execution of the operations plans.[4]

---

[3]The reader may justifiably ask the question, "Why another language?" We felt the need for ABEL based on both substantive and computer efficiency grounds. Our rationale is treated in Norman Z. Shapiro, H. Edward Hall, R. Anderson, and M. LaCasse, *The ABEL Programming Language: History, Rationale and Design*, The Rand Corporation, forthcoming. For a description of the ABEL language itself, see Norman Z. Shapiro, H. Edward Hall, Robert H. Anderson, and M. LaCasse, *The ABEL Programming Language: Reference Manual*, The Rand Corporation, forthcoming.

[4]For a good discussion of the AWP concept as it was originally conceived, see Paul K. Davis and Cindy Williams, *Improving the Military Content of Strategy Analysis Using Automated War Games: A Technical Approach and an Agenda for Research*, The Rand Corporation, N-1894-DNA,

It is possible that the NCL will not like the results of either look-ahead. In this case the NCL will select another vector of plans for testing, if a suitable one is available. In the worse case the NCL will terminate the exercise with a message stating that no plans are suitable.

The mechanics of the look-ahead process take advantage of the fact that the exercise context is entirely contained in the WSDS. When the NCL calls for a look-ahead, a copy of the WSDS is made for later restoration. The exercise then proceeds in a normal manner. At the look-ahead's conclusion, the NCL saves the results and restores the original WSDS, in effect turning the exercise clock back to the point at which the look-ahead was requested.

NCL Learning. At any point in time the NCL has a perception of its opponent that comes in two parts. First, the NCL does not know the actual disposition of its opponent's military assets but rather its perception of those assets' disposition. Second, the NCL also has a perception of the opponent's *nature* or *mindset*. In fact, we speak of a number of *Sams* and *Ivans*, each Sam and Ivan reflecting a different approach to political and military conflict. One Sam might be an *avoid nuclear war at all costs* Sam, making decisions to engage in military conflict from this perspective. Another might take a different approach.

As an exercise unfolds, an NCL can change its view of the opponent based on the exercise results thus far. An Ivan, for example, that originally thought it was going up against a docile Sam can change its perception if the real Sam behaved in a manner inconsistent with that assumption. Ivan would indeed learn from experience. Since look-ahead forecasts are based on the perceived opponent, such learning can markedly influence the results of those forecasts. An NCL rule-writer not wishing to assume a specific perceived opponent might in fact ask for forecasts based on two or more likely opponents and two or more likely plan vectors, selecting the plan vector to implement after appropriately trading off the risks and benefits of alternative plan selections. Of course, it is also possible for an Ivan or Sam to overreact and select a perceived opponent more extreme than is in fact the case.

currently selected plan vector need be different from those already active, and it is possible that no plans will be different. The NCL in fact may just choose to alter bounds or change plan parameter settings. In any event, before the NCL passes this plan vector to the GCL for implementation, it goes through one additional process. It *tests* the plan in what we call *look-ahead* mode.

Look-Aheads. The NCL actually performs two look-ahead tests. It first asks the RSAC system to forecast the next several days of the exercise assuming that no plan change takes place. This look-ahead forecast, and all look-ahead forecasts for that matter, differs from the *real* exercise in that the system performs the forecast using the NCL's perceived opponent, both in terms of opponent mindset and the perceived state of the opponent's military assets. Thus the look-ahead result can differ markedly from the real game. Later in the NCL's deliberations it asks the RSAC system to forecast the effects of the newly selected plan vector. In deciding whether or not to implement the newly selected plan vector, the NCL takes these two forecasts into consideration.[6]

In fact, the results of the *first* look-ahead, where the NCL asks the question, "What happens if I make no changes?" are part of the NCL's high-level situation assessment process. This process is one of *synthesis*, where the NCL distills the numerous detailed world situation variables, e.g., attrition rate and FLOT location, both current and projected, into a small number of higher-level variables that give a more intuitive flavor of the current state of affairs, risks, and future prospects. These synthesized variables are used in other parts of the NCL's deliberations. The NCL's decision to make a change is based on these variable values as well as other high-level variable values the NCL determines in the course of its deliberations.

The second look-ahead, the one that tests the acceptability of the proposed alternative plan vector, has its acceptability determined for the most part by threshold criteria contained in the proposed plans themselves. This look-ahead will be deemed successful if the proposed plans meet their objectives during the look-ahead.

---

[6]Keep in mind that the NCL is simply a rule-set developed and modifiable by substantive experts, and therefore the number of look-aheads and comparisons made is a direct reflection of the substantive experts' desires.

through a multiple-step process in making this determination, and if a plan is deemed inconsistent with the current world situation, the NCL selects another plan to replace the inconsistent one.[5]

The NCL first determines the current context, i.e., determines the state of affairs in each active theater. The NCL next considers what we have called *escalation guidance*, i.e., the NCL determines the degree to which it wishes to escalate the conflict above the current level. This decision is based on the NCL's perception of how it is currently doing as well as on its perception of its opponent. Indeed, the current awakening may have occurred because the NCL is doing too well, i.e., the NCL's previously selected course of action is achieving its objectives far more rapidly than expected. Because the NCL is doing so well, it may wish to markedly increase its objectives by escalating geographically. It also may wish to revise downward its perception of its opponent. Alternatively, the NCL may be awakening due to poor performance, and it may choose to escalate in response to the degrading military situation and revise upward its perception of its opponent.

Once the context and escalation guidance are established, the NCL next determines if the objectives it had previously established need modification. For example, the NCL might on a previous move have chosen a course of action with the expectation that its adversary would respond only minimally. On the current NCL move, the NCL might be confronting an unexpected situation because its adversary responded much more harshly than originally anticipated. This could cause the NCL to alter its objectives.

Finally, the NCL selects a strategy that reflects the current world situation, escalation guidance and revised objectives as well as the possibly revised perceived opponent.

The net result of all these choices is the selection of a plan vector, i.e., an AWP that contains appropriate plans for each active supertheater and theater in the exercise. Not all of the plans in the

---

[5]This paper's discussion of NCL decisionmaking is highly simplified. For more complete discussions, see Paul K. Davis and Peter J. E. Stan, *Concepts and Models of Escalation*, The Rand Corporation, R-3235, May 1984, and Paul K. Davis, Steven C. Bankes, and James P. Kahan, *Prototype Models of National Command Level Decisionmaking for Analytic War Games and Simulation*, The Rand Corporation, forthcoming.

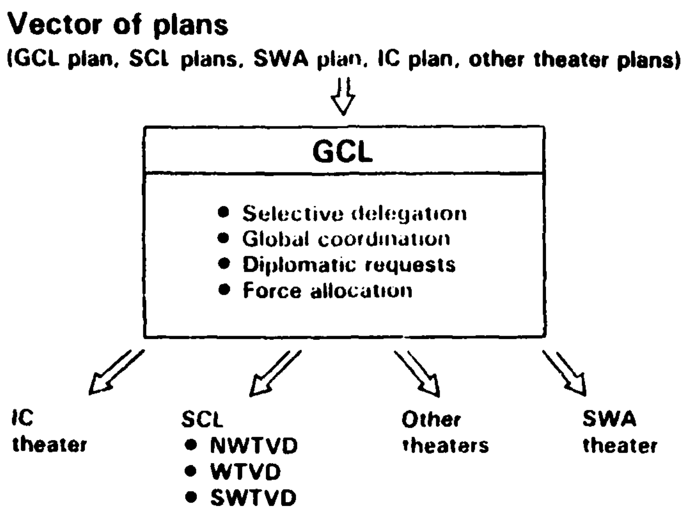


Fig. 6 -- GCL Responsibilities

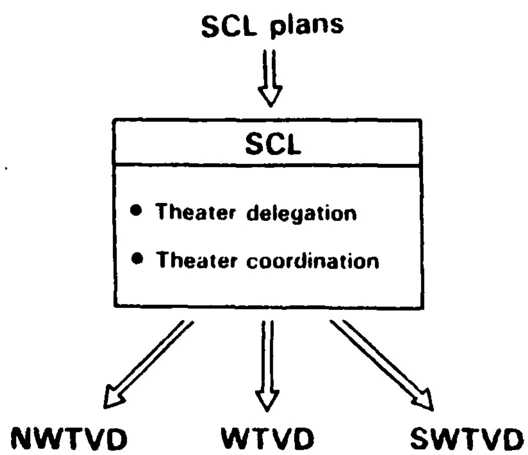


Fig. 7 -- SCL Responsibilities

When the NCL awakens, either at the beginning of an exercise or during the middle, it must determine if the plans currently in effect are still consistent with the current world situation.[4] The NCL goes

[4]Even at an exercise's outset each active theater must have a plan, even if that plan is only a place-holding peacetime plan.

NCL DECISIONMAKING

Figures 5 through 7 illustrate NCL, GCL, and SCL responsibilities, respectively. We shall dwell only on the NCL in this paper. At the present time the GCL and SCL plans are little more than control-passing plans, with little or no military content. As the RSAC matures and we attempt to deal with more complex, global military situations, the GCL and SCL plans will increase substantially in military content and influence.[3]

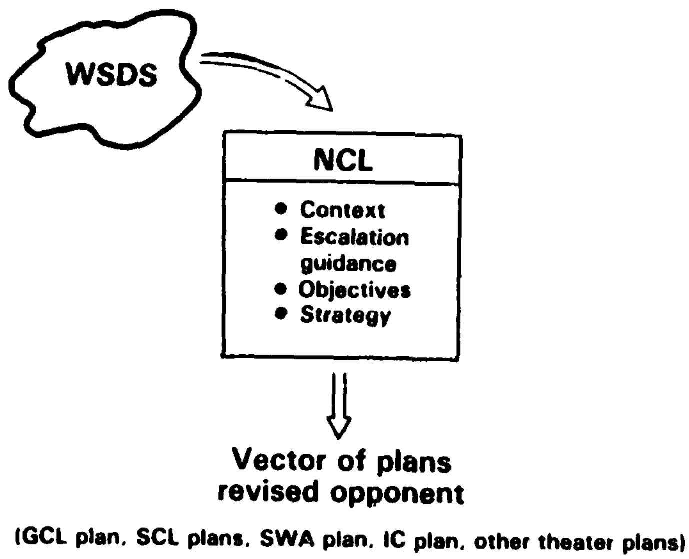


Fig. 5 -- NCL Responsibilities

---

might simply create a perceived world situation that is nothing more than the real world situation of, say, 24 hours ago. More complex agents might incorporate different intelligence gathering philosophies that may prove difficult to capture in rule-sets. Much work lies ahead before we can say that we have created a suitable intelligence agent.

[3]The figures contain several acronyms not previously described in this paper. *SWA* stands for Southwest Asia (the Persian Gulf area). *IC* stands for the intercontinental theater, i.e., the theater that handles homeland-to-homeland strategic nuclear exchanges. *NWTVD, WTVD,* and *SWTVD* are designations for the Soviet European theaters, the Northwest, Western, and Southwest TVDs, respectively. *TVD* is an acronym based on the Russian words for theater of operations. The equivalent theaters in U.S./NATO parlance are Northern, Central, and Southern Europe, respectively.

The WSDS contains all the information about the current state of an exercise, i.e., the exercise's current *context*. When the wakeup rules/bounds examine the world situation at each tick of the exercise clock, they in effect examine the WSDS contents. When a major agent issues an order to the Force Agent, it modifies the WSDS to reflect the new desired state of the affected military assets. When the Force Agent determines how to alter the world situation at each tick of the exercise clock, it compares the actual and desired states of all military assets, those states being contained in the WSDS. When a major agent's subordinate decisionmaking level awakens its superior to ask for guidance, the subordinate effectively posts a message in the WSDS that the superior monitors through a wakeup rule. When the message appears, i.e., when the WSDS is appropriately altered, the superior's wakeup rule fires, and the superior is thereby awakened.

Access to the WSDS is strictly controlled through the Data Dictionary. Each global data item in an AWP must be defined via the Data Dictionary, and this definition includes the agent to which the data item belongs, the agents that can have access to the data item, and the agents that can modify the data item. The Data Dictionary can therefore enforce access restrictions, ensuring that an agent cannot access data to which it does not have access rights, e.g., Red is forbidden from accessing Blue data.

This raises an important point. When one agent wishes to examine the state of affairs of another agent, the Data Dictionary forbids him from doing so directly. But agents must still make decisions based on what they believe the adversary's state to be. This is possible in that the agent can examine a *perceived* opponent rather than the real opponent. And the agent can make decisions based on the perceived opponent. The RSAC system provides the ability to distinguish the real from the perceived opponent, with both the real and perceived opponent data appearing in the WSDS. One can in fact conceive of an *intelligence* agent, whose sole function is to create and adjust perceived opponent data as the exercise unfolds.[2]

---

[2]While we can conceive of an intelligence agent, creating one can prove to be a significant rule-writing challenge. The simplest agent

IV. DECISIONMAKING DETAILS[1]

THE WORLD SITUATION DATASET

In the discussion thus far we have stated that the rule-based agents, their bounds, and wakeup rules all examine the world situation in the course of making their decisions. Figure 4 illustrates how this process takes place. In fact, the figure illustrates how all communication among the agents takes place. None of the agents directly communicates with any other agent. All communication takes place via a *World Situation Dataset (WSDS)* through the vehicle of a *Data Dictionary*.

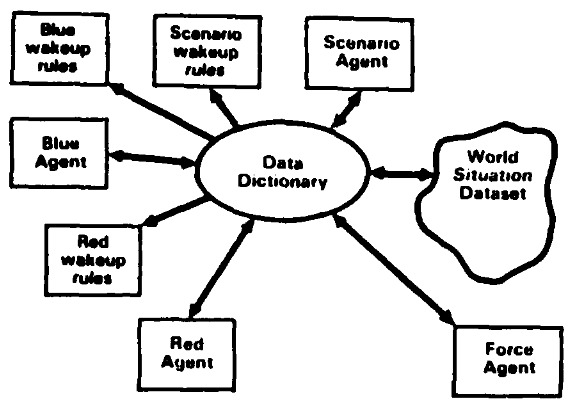


Fig. 4 -- RSAC System Data Paths

---

[1]This section discusses a number of RSAC system capabilities in some detail. While the RSAC system design supports all the capabilities described in this section, some have not as of this writing been implemented. For the most part this implementation involves rule-writing as opposed to system enhancement. The most notable capabilities not yet implemented as of this writing are those dealing with a major agent's learning as well as its perception of the world and the adversary, e.g., Red's Force, Red's Scenario, and Red's Blue.

Third, each rectangle is an independent computer program in its own right, called a *coprocess* in RSAC parlance, and coordinating the execution of these agents is a complicated process in its own right. Indeed, at any one time an exercise may have as many as thirty or more active coprocesses. Figure 3 illustrates how this coordination takes place. A module called the *System Monitor* serves as the *baton passer* among the various agents, determining the agent to awaken by making its own interrogation of the bounds/wakeup rules. In fact, when the Force Agent gives up control, it gives this control to the System Monitor, which then determines which rule-based agent to awaken. Given that two or more agents wish to awaken, System Monitor uses a complex polling mechanism to determine which agent awakens first, which agent second, etc. The highest priority goes to processes the figure labels the *Data Editor*, which is equivalent to saying that human observers can interrupt the exercise at any point and examine selected exercise parameters, changing parameters if necessary. Next in line are the rule-based agents in *Scenario/Blue Agent/Red Agent* order. When no rule-based agent wishes to awaken, the System Monitor then passes control back to the Force Agent, which then resumes the exercise.

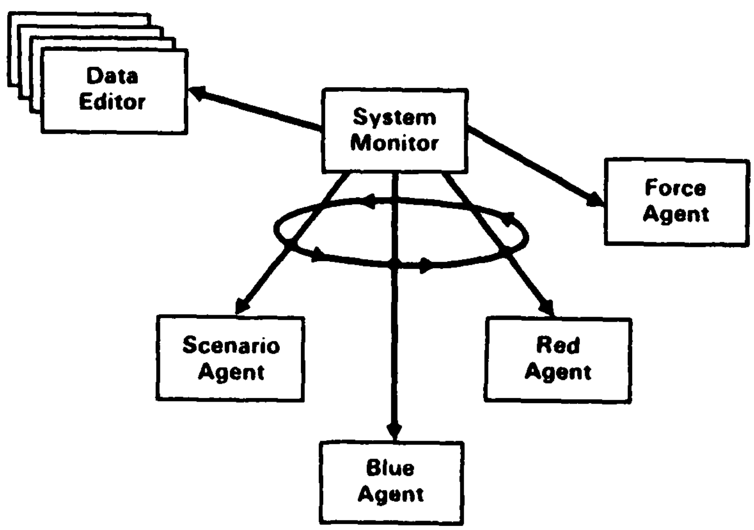


Fig. 3 -- Organization of the Mark III RSAC

subordinate plan the authority to move to the attack phase when ready. In other instances, the superior can require the subordinate to wait until other parallel plans are also ready before it authorizes a move to the next campaign phase.

THE "COMPLETE" SYSTEM

Figure 2 is a simplified version of all the agents contained in a complete RSAC system. On the right and left are the Red and Blue Agents, respectively, and their segments of the figure look much like Fig. 1. In the center is the Scenario Agent and its associated wakeup rules. The important point of this figure is that each of the rectangles in the figure, with the exception of the bounds and wakeup rules, are independent agents in their own right, with communication among those agents precisely defined. Further, it is important to note that only one agent can be awake at any one time, with a rule-based agent automatically putting itself to sleep whenever it calls its superior or passes control to its subordinates.

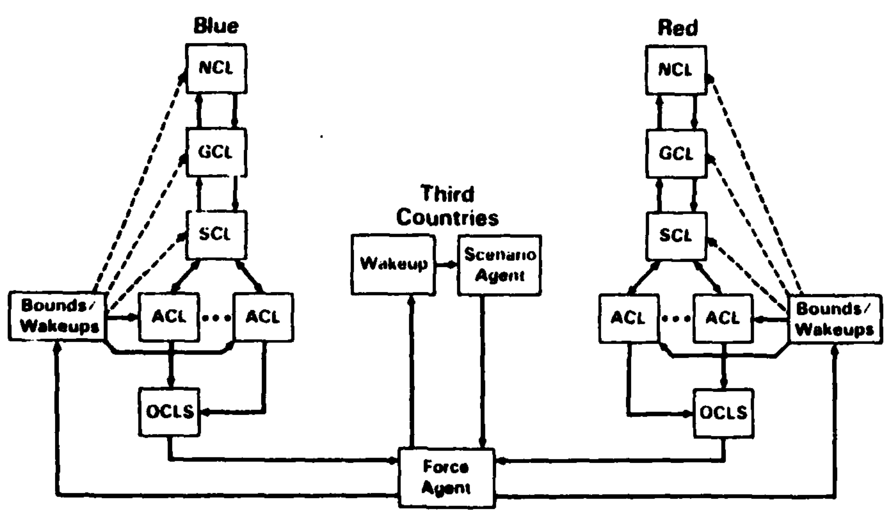


Fig. 2 -- Full System Schematic

COMMUNICATION AMONG DECISIONMAKING LEVELS

As Fig. 1 indicates, each rule-based level can communicate directly only with its immediate superior and its immediate subordinates. By rule-writing convention we do not permit more extensive communication. However, also by convention, if a rule-based level does not recognize a message from its subordinate, it automatically passes the message to its own superior.

Because we want the rule-writers for a given rule-based level not to have to worry about the rules at its subordinate and superior levels, we have defined a precise set of messages that one level can pass to its superior. A subordinate level that wishes to awaken its superior because of a bound violation must provide a *reason* for the awakening as well as a *recommendation*. Both the reason and recommendation must come from a precise set of acceptable reason/recommendation pairs. The superior in turn can respond by relaxing bounds it previously established or altering plan parameter settings to permit execution of additional subordinate plan alternatives. Intraplan communication, in other words, is highly parameterized, and an ACL rule-writer does not have to be intimately familiar with the rules contained in the plans of its superior levels.

PLAN COORDINATION

In an exercise where multiple theaters are operating simultaneously, it is important that the theater plans act in a coordinated fashion. For example, if we are conducting an exercise where Red is planning to attack Northern, Central, and Southern Europe simultaneously, it is desirable to have those attacks synchronized. In other instances, where theaters are relatively independent, such synchronization is unnecessary. Rule-writing conventions and standards have been adopted that permit this coordination.

In order for an ACL plan to move from one campaign phase to another, e.g., from the preparation phase to the combat phase, that ACL plan must have been given authorization to do so by the GCL (and possibly the SCL) plan. In some instances this authorization can be given by the superior in advance, with the superior delegating to the overtaken by events. As an alternative, we could have created wakeup rules for each of the higher-level plans, with those plans awakening directly to deal with bound violations. We chose not to do so for three reasons. First, in the process of debugging ACL plans (keep in mind that ACL plans are written by substantive experts and not necessarily computer scientists or programmers) we want to avoid a situation whereby a higher-level plan awakens without the subordinate plan's knowledge. Second, and substantively more important, we wish to replicate the command and control situation as realistically as possible. In the real world it is usually the theater commander who detects and reports up the chain of command the occurrence of bound violating conditions requiring decisions by the theater commander's superiors.[5] Third, and most important from a rule-writing perspective, while we do not wish to overburden the ACL rule-writer by requiring him to understand NCL, GCL, and SCL plan details, we do want him to be aware of and test the bounds those plans can place on the ACL plans. Indeed, many of those bounds are ACL plan specific. Giving the ACL plan-writer responsibility for bound testing reduces the chances of inconsistent behavior at different decisionmaking levels.

THE EXERCISE CLOCK AND FROZEN TIME

The Force Agent advances the exercise clock, and on each tick of the clock the Force Agent causes the testing of all active wakeup rules and bounds. If any active wakeup rule or bound *fires*, the Force Agent gives up control, and all rule-based agents that wish to awaken are given an opportunity to do so. These awakenings occur in *frozen time*, i.e., the clock does not advance when a rule-based agent is awake. It is possible for a rule-based agent to awaken more than once during this interval of frozen time. And more than one agent can awaken during the same frozen time interval as well.

---

[5]See the section below entitled *An ACL Plan Excerpt* for two examples of how an ACL plan can awaken its superior.

rule-based agent intervention. Wakeup rules and bounds are similar in that they both monitor the unfolding world situation for specific conditions. They differ in that wakeup rules are proactive, prompting the next ACL action (move), whereas bounds are reactive, prompting reconsideration of a plan's validity. In fact, from the ACL plan's perspective, wakeup rules detect the occurrence of expected or planned conditions, i.e., conditions anticipated by the ACL plan that require the plan's intervention. Bounds on the other hand detect the occurrence of conditions inconsistent with ACL plan premises, thus requiring the attention of the higher-level decision models, which are also represented as plans. An example of a wakeup condition might be, during the conduct of a military campaign, the attainment of an intermediate objective, thereby requiring the ACL plan to move on to its next phase and toward a new objective. An example of a bound violation might be, in a conventional warfare ACL plan, the opponent's use of theater nuclear weapons. In this case the ACL plan would notify its superior that nuclear weapons have been used. The higher command-level action could be to change plan parameter settings, relax bounds, or replace the ACL plan with one more suitable to the current state of affairs, as for example the replacement of a conventional warfare ACL plan with one designed to deal with the adversary's use of theater nuclear weapons.

Figure 1 contains two rectangles labeled *Bounds* and *ACL Wakeup Rules*. From the *Bounds* rectangle emanate three dotted lines, leading to the higher decisionmaking levels, as well as a solid line leading to the ACL. The three dotted lines appear in the figure simply to emphasize that bounds belong to the higher-level plans. The solid line appears to emphasize that the ACL plan is responsible for testing whether or not its bounds have been violated. If it detects a violated bound, it is the ACL that reports this to its immediate superior, i.e., awakens its immediate superior. If the superior cannot deal with the situation (whether or not it can will depend on predelegation from its own superior) it will report the situation to its superior, i.e., causing its superior to awaken.

The RSAC has adopted this *ACL tests-for-bound-violation* convention because we believe it is important for the ACL rule-writer to consider and plan for those conditions that would cause an ACL plan to be

The Force Agent is not a rule-based agent but a sophisticated simulation model of precombat and combat operations. It is a goal-directed model in that it accepts *orders* from the rule-based agents. Those orders cause the *desired* state of one or more military assets to change, with the Force Agent then attempting to move the assets from their actual state toward the desired state. Note that the GCL, SCL, and ACL each can issue orders to the Force Agent. However, the bulk of the military orders originate at the ACL level. The GCL and SCL levels issue mostly resource allocation orders.

COMBAT MANAGEMENT

Turning back to Fig. 1, two additional rectangles lie between the ACL and the Force Agent, namely the ***TAC AIR OCL*** and ***Ground OCL***. The term OCL stands for Operational Command Level, and TAC AIR stands for tactical air. It is an OCL's job to manage the daily low-level combat to ensure reasonable combat simulation. The OCLs are highly parameterized, and they must indeed be activated by the higher decisionmaking levels. Hence Red and Blue rule-writers can have a major influence on the nature of combat management decisions. But those rule-writers need not worry about the combat management decision details.

MONITORING THE EXERCISE

The four decisionmaking levels do not simply pick an AWP to deal with a specific world situation and then disappear. They also monitor the AWP's execution, i.e., they monitor the changing world situation as the exercise unfolds. When the NCL is in the process of assessing the world situation and picking an appropriate AWP, it is said to be ***awake***. After it has picked the AWP, it goes to *sleep*. The same is true for all the rule-based agents. Most of the time they are asleep. However, when a situation arises that requires a rule-based agent's attention, the agent is awakened. If these agents are asleep most of the time, how do they monitor the exercise's state?

Wakeup rules and bounds are the mechanisms whereby the rule-based agents monitor the unfolding world situation. These are rules also written in the ABEL language designed to detect situations requiring

Once the NCL makes its choice, it passes the selected AWP to the Global Command Level (GCL) for implementation. The GCL is intended to represent the Joint Chiefs of Staff and State Department, coordinating the execution of plans and handling communications with other countries. The GCL activates the selected plans for each theater, mobilizes and assigns forces to the various theaters, establishes the predelegation bounds and constraints consistent with the NCL's guidance, and then passes control to the various plans for their simultaneous implementation.

Each major agent, i.e., Red and Blue, has only one NCL and one GCL. The NCL acts as the plan selection agent, and the GCL acts as the global plan coordination agent. There can be a number of subordinate plans, however, depending on the number of supertheaters and theaters identified in the exercise. One such supertheater is the European region, where a European Supertheater commander has three subordinate theater commanders reporting to him. Another supertheater might be the Pacific region. When it is necessary to simulate the existence of subordinate commands, each with its own set of rules, then a coordinating supertheater plan is required.

In the current RSAC system implementation two supertheaters exist: one for Red and one for Blue, both representing the European region. The Supertheater Command Level (SCL) in Fig. 1 is intended to depict this. Just as the GCL fills a plan coordinating role, so too does the SCL. Where the GCL's coordinating role is a global one, the SCL fills a regional coordinating role, ensuring the cooperative execution of the several theater plans subordinate to it.

The lowest level of the decisionmaking hierarchy is the Area Command Level (ACL), and the plans at this level have theater responsibilities. No plans are subordinate to them.

June 1982, pp. 5-29. Also, see Paul K. Davis, *Concepts for Improving the Military Content of Automated War Games*, The Rand Corporation, P-6830, November 1982, pp. 8-14. For a discussion of AWP structure and rule-writing conventions and standards, see William Schwabe, *An Introduction to Analytic War Plans*, The Rand Corporation, N-2254-NA, forthcoming.

Flag Wars. The NCL process described above is one of reducing what can be very complex combat and political situations to representations by a few variables. The context, escalation guidance, objectives, and strategy determinations are indeed distillations of those complex situations into precise, well specified and intuitive synthesized variable values. It is therefore possible to run the NCL in a stand-alone mode, i.e., without the other agents, and explore how a specific NCL would respond to changes in these synthesized variables. It is also possible to pit a Red and Blue NCL against each other, with a human in the loop to interpret how a major agent's actions would affect the detailed world state, at least in sufficient detail so that the adversary agent can perform its own world state synthesis. We have labeled this process *flag wars* in that the NCL is responding to flag settings controlled by the NCL rule-writer or human operator.

This method of exercising the NCL has many advantages. First, when developing NCL rules, we can get a feel for how those rules respond to different world situations without having to carry the burden of a complete exercise. The NCL runs very quickly, and it is easy for a rule-writer to test his rules as he is developing them. Second, once a specific NCL rule- set is developed, the rule-writer can readily perform sensitivity analyses on behavioral parameters as well as analytic parameters. One can, for example, examine how alternative Ivans respond to systematically varied world states. One can also examine how a specific Ivan interacts with a specific Sam, the human in the loop determining how specific Ivan and Sam actions affect the world situation.

The flag war technique does not permit the look-ahead projections possible with the full system. But it nevertheless provides some heretofore nonexistent and useful behavioral analysis capability.

ACL PLANS

The NCL, when it awakens, always begins with a fresh look at the world situation, remembering the context of its previous awakening only from the flag settings made at that time. We say that the NCL *starts at the top* each time it awakens. The ACL plans do not. In fact, ACL plans, because they deal most directly with theater-level preparations for combat and combat itself, i.e., with the phases of a specific theater's military campaign, are structured around those phases and conform more to scripts of operations. Thus, while the NCL always awakens at the top, the ACL plans awaken at the next appropriate point within the plan or at a point decided by the GCL or SCL.

ACL Plan Phases and Moves. ACL plans are structured in sequential campaign phases, and movement from one phase to the next requires approval from the ACL's superior. While the specific phases are determined at the rule-writer's discretion, by plan-writing convention they generally include the following:

- *Preparation phase*, including force mobilization, deployment, and alert. This phase generally includes all precombat preparations and posturing.
- *Combat phase*, including initial attack orders and periodic adjustments to combat orders that the rule-writer wishes to handle personally, instead of via the OCLs. The combat phase can be split up into a number of combat phases, e.g., conventional combat, tactical nuclear combat, combat phases with successive intermediate military objectives, etc.
- *Termination phase*, the phase that brings a campaign to a close. The termination phase is triggered by the NCL's authorization to seek combat termination.

Within a phase there can be a number of moves. For example, the preparation phase can include the mobilization and deployment of forces, and both those can be handled as moves within the preparation phase. Placing forces on alert can also be considered a move. Some moves may be executed recursively.

An ACL Plan Excerpt. The following ACL plan excerpt tests for completion of Red deployment in the WTVD, i.e., the western TVD:

```
Perform Execute-WTVD-deployment.
Let deployment-completed be No.

While deployment-completed is No
{
    Let reason-for-move be
    ( the report from Start-move
       using WTVD-deployment-completed     as planned-wakeup,
         and M-day + Preparation-duration as time-limit,
         and WTVD                          as theater ).

    If reason-for-move is bound-broken
    Then perform Notify-higher-authority
       using ( Bound-status of WTVD ) as reason
         and   no-recommendation      as recommendation.

    If reason-for-move is planned-wakeup-triggered
    Then let deployment-completed be Yes.

    If reason-for-move is time-limit-expired
    Then perform Notify-higher-authority
       using preparation-time-limit-exceeded as reason
         and extend-deadline                 as recommendation.
}
```

This is a simplified excerpt from a real ACL plan, the simplification removing much of the boilerplate for local variable definition. The first two statements issue the deployment orders to the Force Agent (the *perform* is simply a call to a lower-level function that contains the specific deployment orders) and set the local variable *deployment-*completed to *No*, thus indicating that deployment has not as yet been completed. Recall that all plan moves take place in frozen time. Even though the deployment orders have been issued, the Force Agent has taken no steps to implement those orders.

The next series of statements is executed as long as the local variable *deployment-completed* remains equal to *No*. Indeed, the *while* statement's bounds are defined by the two braces ({ and }). The first statement within the *while* phrase puts this ACL plan to sleep so that the Force Agent can begin to implement the deployment orders. This statement is actually a call to the function *Start-move* with three parameters: the planned wakeup, the time limit, and the theater. In this case the planned wakeup is defined by the rule (or simply the function)

WTVD-preparation-completed. The time limit is defined to be *M-day* (for mobilization day) plus the duration contained in the variable *Preparation-duration*, and the theater is specified to be the *WTVD*. Both *M-day* and *Preparation-duration* were set previously in the GCL. Further, the function *Start-move* returns a value which is placed in the local variable *reason-for-move*. When the plan reawakens, i.e., when we return from the call to *Start-move*, this local variable will contain the reason for the reawakening. We emphasize that, when *Start-move* is called, the ACL plan is in effect put to sleep.

There are three reasons why the ACL plan might reawaken. First, a plan bound might break. Second, the planned wakeup rule might fire. And third, the plan's time limit might have passed. The next three statements test for the three conditions. If the awakening has occurred because a plan bound has been broken, the plan calls the function *Notify-higher-authority*, i.e., awakens its superior, reporting the bound violation. This act of awakening the superior also puts the ACL plan to sleep until the superior responds. If the ACL plan reawakens from the higher authority notification, then the superior has responded possibly by changing plan parameters or relaxing plan bounds, but the superior has not chosen to replace the plan. If the superior chooses to replace the plan, then this plan will not reawaken from the higher authority notification.

If the *Start-move* awakening has occurred because the planned wakeup rule has fired, then the ACL plan sets the local variable *deployment-completed* to *Yes*, thereby resulting in the plan's leaving the bounds of the *while* phrase. If the *Start-move* awakening has occurred because the time limit has passed, then the ACL plan awakens its superior to ask for a preparation deadline extension. As with any call to the plan's superior, this has the effect of putting the ACL plan to sleep. The superior can respond by extending the deadline, i.e., increasing the value of the variable *Preparation-duration*, relaxing bounds, or it can change the plan.

We have included the above example, which is admittedly a simplified one, to present an example of ABEL rules and to make the point that the ACL plan rule-writer must worry about testing for broken bounds. We further wish to illustrate the rule-writing conventions we have adopted to define moves and notify the plan's superiors. Finally, we wish to emphasize the fact that all plan decisions are made in frozen time, with orders to the Force Agent having the effect of changing the desired state of military assets. The Force Agent controls the exercise clock's advance.

V. HUMAN INTERVENTION

The RSAC has developed the automated war-gaming system in order to remove human teams from the war-gaming loop. By doing so, we have laid the groundwork for asking the behavioral *what if* questions that traditional war gaming cannot readily answer.

However, the fact that we can replace human teams by automated rule-based agents does not mean we wish to preclude human intervention altogether. In fact, the RSAC system has been designed to support several levels of such intervention. This intervention can run the gamut from simple mid-exercise parameter changes to actually selecting and testing complete AWPs for implementation. Indeed, an important role the RSAC system can serve is one where the system supports a complete human team war game.

An observer can, at any point in an exercise, interrupt the system and examine the WSDS. The Data Editor permits this examination, and it also permits the observer to change WSDS data items. The observer can also change any of the rules contained in the NCL and the AWPs, and thereby he can alter the wakeup conditions and bounds associated with a plan that is being implemented.

A human team can also replace the NCL in the sense that the human team can select and test an AWP directly, overriding the NCL's logic for making such choices. When a human team replaces the NCL, the human team's choices are constrained to be grossly consistent with the underlying AWPs the system can support. The human team can use the system's look-ahead mechanism to test the effects of using alternative AWPs. The team can also set plan parameters and bounds and change those as the exercise unfolds. In this way the team can make major changes to the exercise scenario.

The human team, however, should ordinarily refrain from issuing orders directly to the Force Agent. All such orders should come via the AWPs themselves to ensure that the orders are consistent with plan premises. An example where a direct order to Force Agent is inconsistent with plan premises is a direct order calling for the use of tactical nuclear weapons where the underlying plan is a conventional warfare plan.

VI. NEXT STEPS

Two years ago, the main technical risk in the RSAC's research agenda was the question of whether a system of the complexity envisioned could be implemented in the short time frame allowed. Today, given the system's existence, the technical risk has shifted considerably. Today there are two new technical risks. The first revolves around the issue of whether we can actually construct the war plans needed to properly exercise the system. The issue is not just their construction but their construction in a manner whereby substantive experts can review, critique, and modify them with minimal assistance from professional programmers.

The second technical risk is a methodological one. The RSAC system presents a tool unique to the strategic analysis community, and we have only begun to scratch the surface in this tool's utilization. The issue here is not whether a set of useful methodologies can be developed to exploit the tool's potential. Rather, the issue is the length of time it will take us to learn how to fully exploit the tool.

We have confidence that we will successfully meet these dual challenges, but we realize that the methodological development needed for full system exploitation can take years. We have already begun by modulating the emphasis of our research efforts. While the emphasis during the past two years has been almost exclusively on system development, both of the integrated system and of the stand-alone Force Agent, today's emphasis has shifted toward a balance between development and applications. This year several Rand research projects are using the RSAC system, and we are continuing along the methodological development path staked out during the past two years.

In addition, and in parallel to the above, we are positioning ourselves to transfer the RSAC capability back to the government. A number of agencies have expressed interest in acquiring and utilizing the RSAC system, and we expect during the next several months to provide one such agency with a stand-alone Force Agent. We also expect next year to provide the fully integrated system to a number of government agencies.

END

FILMED

10-85

DTIC