

SABRE

William C. Branley, Jr. and Earnest D. Harris *

U.S. Army Artificial Intelligence Center
ATTN: SAIS-AI
107 ARMY PENTAGON
WASHINGTON, DC 20310
branley@pentagon-ai.army.mil

Abstract

The Single Army Battlefield Requirements Evaluator (SABRE) is a decision aid that combines rule-driven scoring strategies and domain-specific heuristics to solve a constraint satisfaction problem. The main SABRE algorithm performs a function called "sourcing," which refers to the allocation of resources to satisfy a set of requirements that may or may not be feasible in terms of the resources available. In the context of this application, the resources are military organizations, and the requirements refer to the needs of a commander-in-chief who is responsible for conducting a military operation. SABRE is presently used in both an operational and an analytical role. The primary operational user is an Army command that is responsible for choosing specific Army units for overseas deployment during a national emergency. Before SABRE, this function was performed manually, involved nineteen or more people and could take up to nine days, depending on the number of requirements. Using SABRE under ideal conditions, a single person can perform the function in a couple of hours. However, SABRE is designed to be a *partner* in the sourcing process. There are times when the sourcing solution involves highly subjective trade-offs that require human involvement. Therefore, the organization that is using SABRE has instituted procedures that permit human expert involvement when required; otherwise, a single "SABRE operator" can perform most resource allocation tasks.

Introduction

When leaders of the United States opt for military intervention during a crisis, military planners must immediately begin grappling with numerous assumptions about the nature and duration of the military action, and assumptions about the current disposition of military assets. These uncertainties must be managed

*Many members of the U.S. Army Artificial Intelligence Center have contributed to this project from 1987 to the present. (File name: sabre-iaai4)

by analysts and decision makers in a severely time-constrained environment.

The U.S. Army has been addressing part of this problem using an assortment of standard artificial intelligence techniques embedded in a program called the Single Army Battlefield Requirements Evaluator (SABRE), a prototype developed "in-house" by the U.S. Army Artificial Intelligence Center, Washington, D.C. SABRE is currently used in both operational and analytical settings by the Army.

SABRE considers current information about Army assets and combines this with a set of weighted decision variables to produce an optimal solution for determining which assets to employ. In cases where a complete solution cannot be obtained, SABRE explains which decision variables had the greatest effect on the solution and how they might be relaxed to obtain a more desirable outcome.

This paper describes SABRE and is organized as follows: Problem Description provides a detailed description of the decision problem faced by users of SABRE and the algorithm that was designed to solve it; Application Description is a description of the overall system; Implementation and Use describes SABRE's deployment and use; and Technology Transfer briefly covers plans for transferring SABRE into the formal software acquisition process. We close the paper with a Conclusions section.

Problem Description

Decision variables modeled by SABRE are derived from attributes of military units (a unit is an organization, such as a transportation battalion, which may consist of several hundred trucks and truck drivers). The task is to choose units that fulfill all requirements and have the most desirable attributes, as determined by the user of SABRE. The relevant data needed to process the decision variables, which SABRE treats as constraints, is contained in many different databases. SABRE contains the necessary rules about which databases to consult when sourcing a given kind of force subject to stated criteria. Sometimes, processing a single constraint involves checking data elements

Criteria	Meaning	Allowable Values
Equipment Readiness	The present state of an organization's equipment.	C1, C2, C3, C4, C5
Personnel Readiness	The extent to which an organization has the people needed to do its mission.	C1, C2, C3, C4, C5
Training Readiness	An assessment of a unit's ability to do its mission.	C1, C2, C3, C4, C5
Overall Readiness	An aggregate measure that considers equipment, personnel and training readiness.	C1, C2, C3, C4, C5
Component	The component to which the unit belongs.	Active Army, National Guard (NG), Army Reserve (USAR)
Region	The geographic region to which the unit is assigned.	US, Europe, Korea, Panama, Alaska, Hawaii, Japan
Availability	Whether or not a unit is deployed elsewhere.	Yes (the unit is available), No (it is not)
Status	An indicator of whether or not a unit is scheduled to activate or be deactivated.	Various codes represent all of the possibilities.
Capability	The kind of mission that a unit is expected to perform, such as medical, engineer, field artillery.	A 12-character code.
Time Rotation Dates	Projected duration of the operation.	Start date, end date
Sourcing Sourcing Pool	The pool of Army units from which the selection occurs.	A set of units

Table 1: These are the most commonly used decision variables considered by SABRE. There are 17 variables in all, each having from 2 to 8 alternatives, which may be ranked.

in more than one database. The issue of how the data is accessed is addressed later in this paper. The most commonly-used criteria modeled by SABRE are shown in Table 1.

The criteria listed in Table 1 can be weighted according to user preference. (Actually, there is a default cardinal ordering determined by applying rules. The cardinal ordering, whether it's by default or user-determined, maps to a default numerical weighting scheme, which the user can also override if necessary.) The criteria weights are used to by the sourcing algorithm to determine the order in which to relax constraints. Furthermore, each criterion's allowable attribute values can be arranged according to preference, from least desirable to most desirable. This is also done by applying rules that can be overridden by the user. For a particular sourcing task, screening criteria (minimum acceptable attribute values) are set by a plans officer, who is in charge of selecting units for the force under consideration.

However, from what source does the plans officer get the weights and screening criteria? Typically, they are derived from "sourcing guidance" from many possible sources, such as the National Command Authority (NCA), the Joint Chiefs of Staff, commanders at various levels, or from existing policies and procedures. For example, if the guidance is "Reserve forces will not be used unless absolutely necessary," then the Component criteria may be given a higher weight than, say, Region. However, if the guidance is "Do not use Reserve forces under any circumstances," then the Component attribute becomes a screening criteria with a value of "Active Army". To the algorithm, this becomes a hard constraint.

SABRE attempts to find units that, at a minimum, satisfy the screening criteria, while giving preference to those that have more desirable attributes values. The solution is then presented in the form of an explanation that tells the user how various criteria impacted on the sourcing solution. The user can relax constraints by interacting with a graphical user interface and then continuing the computation to obtain an estimate of what the result is likely to be, given the relaxed settings.

A general statement of the problem is as follows:

Given:

- a pool of military units, each uniquely identifiable;
- a proposed plan containing requirements for types and quantities of units;
- a set of weighted constraints, with minimum values indicated;

From the sourcing pool, attempt to satisfy all requirements with units that possess minimum values, or, optimally, have the most desirable attribute values.

Given the preceding problem description, SABRE can be viewed as solving a problem of finding an opti-

mal subset from a set of units. This problem is well-suited for AI techniques. The procedure starts by using as many domain-specific heuristics as possible to reduce the size of the set of units to be considered, and to reduce the number of requirements to source (by finding those that may be unsourceable for an obvious reason). Then, among the the remaining candidates, an exhaustive search is conducted to find optimal units using a numerical scoring procedure. Throughout each step of the algorithm, information is retained to support extensive explanation and feedback. As noted in previous paragraphs, rules are used to process constraints and to form the basis for the scoring procedure.

Even with all of the searching and storing of results, the following performance is achieved: given a sourcing pool of 8025 units, and a list of 1825 requirements to source, the algorithm sourced 1750 of the requirements in 17 seconds during the first sourcing iteration. However, of the 1750 filled requirements, 123 were newly filled, and 1627 were units that were already in the force and were allowed to stay. Verifying that existing units meet the criteria is a valuable heuristic because keeping an existing unit is much faster than finding a new one.

Consider a contrasting sourcing iteration with the same sourcing pool (8025 units) and the same requirements list (1825). This time, however, parameters were set so that the algorithm attempted to source all requirements, regardless of the status of existing units. The performance was 258 seconds to source 1629 requirements, of which all were newly sourced. (The 258 seconds included a 72-second garbage collection.)

The recommended approach to using SABRE is to avoid screening criteria in order to let SABRE find the best units that it can, based largely on its default rule base. On the other hand, the user could choose a more manual approach by setting very high screening criteria, and then relaxing them during successive sourcing iterations, with each iteration encompassing a wider search space. (Ultimately, it would be up to the user to relax the criteria in a manner that is consistent with the sourcing guidance.) This approach takes much longer, as many as nine or ten sourcing iterations to satisfy a few hundred requirements. Some users prefer this approach, however, because they can inspect SABRE's results at each iteration.

Application Description

In general, the impact of AI on the success of this project is in the use of rules to drive a numerical scoring procedure and in using English-like explanation to support user interaction with a complex model, namely, the sourcing algorithm. The application also makes extensive use of rules to help a user to correctly interpret raw input data that describes the force to be sourced. Almost all of the rules in SABRE were gathered from domain experts over a period of two years. The net

result, we feel, is a tool that functionally-oriented end users, who are not operations researchers or computer scientists, can use and understand.

The current version of the system, called SABRE 33, is written in Common Lisp and runs on both Unix and Symbolics platforms. The interface is the Common Lisp Interface Manager. It is currently being converted to a client-server application with a Motif interface.

The system consists of three modules: Force Maps, Warplanner and the Alternatives Analyzer, Comparer, Editor & Sourcer (AACES). The sourcing algorithm, explained in Problem Description, is contained in AACES (pronounced "Aces"). Typically, sourcing is not the only task a user wishes to perform in a session. Warplanner and Force Maps provide capabilities that complement the sourcing task in order to create a more useful environment for the user. The three modules of SABRE are described in the following paragraphs.

Force Maps contains the Common Lisp Object System (CLOS) class hierarchy and methods that are used throughout SABRE. It contains all of SABRE's data loading utilities plus rules for applying various data sets to the sourcing task. Presently, all data is loaded from ASCII files. This is because SABRE has always worked with data stored in many different databases on different mainframe computers. The easiest way to make an integrated database available to analysts using SABRE was to download data to tape. When SABRE is running, all of its data is stored in an object structure that resides in memory. This is saved as part of a SABRE image in order to improve start up time. The current version of SABRE contains code to permit data loading from relational databases, but most user organizations are not yet ready to take advantage of this feature.

Warplanner was created as a companion to AACES to help it interact better with existing systems. For example, users of SABRE often want to use SABRE's output on the Joint Operational Planning and Execution System (JOPES), another planning system used throughout the Department of Defense (DOD). Warplanner helps by extracting data from SABRE objects and generating files of transactions, in ASCII format, that can be input into mainframe based systems, such as JOPES.

AACES, as stated earlier, contains the sourcing algorithm and all of the rules that drive the algorithm. In addition to the capabilities that we've described, AACES contains many query utilities that help support analysis of the content of forces and of the overall force structure of the Army.

Implementation and Use

SABRE is currently used by both operational and analytical communities. Thus far in this paper, we have described SABRE's operational role. That mission is performed at one site: U.S. Army Forces Command (FORSCOM) in Atlanta, GA. However, several other

sites have used and are scheduled to use SABRE in a purely analytical role. These include the Combined Arms Support Command (CASCOM), Fort Lee, VA; Concepts Analysis Agency (CAA), Bethesda, MD; Logistics Evaluation Agency (LEA), Harrisburg, PA; and the Combined Arms Center (CAC), Fort Leavenworth, KS. In addition, several other groups from the Army Staff and the Joint Chiefs of Staff have used SABRE for specific projects.

The use of SABRE in a purely analytical role is distinct in the following way: when a user sources a force, he or she is usually not concerned with which specific units are selected during sourcing; the concern is whether or not enough units of the correct type exist in the Army to make a given force option feasible. For example, an analyst could use a sourcing tool such as SABRE to answer the following questions: Given a proposed force for a peacekeeping operation, how many distinct forces of this type can be sourced from the Active Army?, How many can be sourced if we allow National Guard and U.S. Army Reserve units?, and so on.

We make these distinctions in the use of SABRE because it has impacted on SABRE's development, deployment, use and return-on-investment. Early users of SABRE were all from analytical communities. The Army AI Center supported these users by loaning the necessary equipment and software; the typical "package" consisted of a MacIntosh computer with a Symbolics MacIvory board inside. This got SABRE on the desktops of users; and they described great time-savings when using SABRE for studies and analysis. However, no organization stepped forward to take SABRE out of the lab and into the formal software procurement process. In other words, analytical needs alone were not enough to justify the cost.

In 1993, we began applying SABRE to operational needs at the site in Atlanta. In a six-month period, SABRE was completely reoriented from analytical applications to crisis-action planning applications. This involved adding new rules to reflect this community's view of how sourcing is done. In May 1993, during an initial exercise using SABRE, it was reported that users could perform in nine hours a sourcing task that had previously taken nine days. In August 1993, SABRE was used in a two-week exercise that again revealed an order of magnitude reduction in the time required to source forces. To help with this exercise, the AI Center networked three MacIvory machines in the FORSCOM Operations Center and conducted "SABRE training" for a group of nineteen end-users, many of whom were not familiar with computers. All of them were able to use the tool to source forces.

Following the August 1993 exercise, the end users at FORSCOM appointed a "SABRE operator," who then instituted procedures for making SABRE part of the normal workflow. In many cases, this single operator can source large forces with minimal assis-

tance, thus achieving greater efficiency gains. In January, 1995, the Unix version of SABRE was installed at FORSCOM headquarters. As of this writing, development of a client-server version of the SABRE prototype is in progress.

In spite of these inroads, there are still many things that must be done before SABRE can be considered a "fielded system." Right now, it is officially a prototype. Its internal sourcing algorithm has not been formally validated, although it has been informally validated by domain experts who are familiar with the sourcing task. Also, users have verified that the sourcing algorithm can at least reproduce the manual process that SABRE has replaced. It has taken two years for the users to conclude that an automated solution to the sourcing task is feasible. The primary obstacle has been that users have little confidence in the reliability and completeness of the data contained in many military databases. In spite of this, formal requirements for SABRE are now being established. This will start the technology transfer process described in the next section.

Technology Transfer

One of the many functions of the Army Artificial Intelligence Center is to develop advanced AI prototypes to solve complex problems for the Army. These prototypes are designed and built in-house by military officers and full-time civilian programmers for a specific end-user and problem domain. Though developed for a specific user, the prototype may have applicability to other users and problem domains. This is certainly the case with SABRE. Our goal is to have SABRE fielded to all sites requiring this capability.

Critical to the successful transfer of SABRE to many more end-users is working within the federal government's software procurement procedures. The government software acquisition process will ultimately serve as the vehicle by which SABRE will be installed on the Defense department's new Global Command and Control System (GCCS). GCCS is an ongoing effort to provide current information technology to support military planning needs around the world. It is replacing many existing systems. Once installed as an application on GCCS, SABRE modules will not only be available to planners, analysts and decision makers throughout the Army, but also to the Joint Chiefs of Staff, and other Defense commands and agencies.

Meanwhile, developers at the AI Center are extending SABRE by addressing other aspects of this resource allocation decision problem. For example, we recognize that planners seldom do this task in a vacuum. They have a need to coordinate their actions and decisions with other workers at their site, and with others at remote sites. Supporting these activities with intelligent tools is viewed as a key area for exploration in the future.

Conclusion

SABRE provides intelligent support for an extremely complex decision process that is central to the Army's mission. It has proven itself by showing analysts and decision makers that a great deal of time and effort can be saved by using an intelligent decision support tool to properly manage a set of decision variables where uncertainty and time-constraints are key characteristics. The developers of SABRE have shown that a technical solution to the problem is within reach, using many standard AI techniques.