Building, Remembering, and Revising Force Deployment Plans

Alice M. Mulvehill
The MITRE Corporation
Bedford, MA USA
amm@mitre.org

Abstract

This paper describes how the prototype system ForMAT (Force Management and Analysis Tool) can be used to construct, store, retrieve and revise force deployment plans. The paper also describes some future work that is being done to: (a) speed up retrieval through the use of high performance AI computing techniques, (b) represent and maintain temporal relationships, and (c) provide revision suggestions to the user.

Introduction

ForMAT is a knowledge-based prototype that uses Case-Based Reasoning (CBR) techniques to build, modify, index, store, and retrieve references to the military forces that are part of a force deployment plan that supports an operation plan (OPLAN). Within the military deployment planning domain, this force deployment planning information is often encapsulated in a data structure called the force module (FM) that is part of the OPLAN's Time Phased Force and Deployment Data (TPFDD). In ForMAT, the FMs are the cases in the case base.

Case-based reasoning systems capitalize on the observation that human problem solvers often derive all or part of a solution to a current problem from all or part of a solution to a problem that they "remember" encountering and solving in the past. In a case-based reasoning system, a case is a memory of some situation and some resulting action; therefore the case base is a database of memories. CBR systems are designed to offer the user access to past experiences (cases stored in the case base) for use in solving current problems, and to store the solutions to current problems as they are being made.

Recalled cases are used both to interpret situations and to solve the problems they pose. The common ways we reason with cases are to adapt old solutions to fit new situations, to use the outcomes of old situations to warn of the potential for failure or success in a new situation, to compare and contrast cases to each other and to a new situation in order to interpret a new situation or critique a solution, and to merge pieces of several cases to form new solutions (Kolodner 1993).

Each case can record a specific situation or can represent composite, prototype, or even abstractly defined situations. For example, in ForMAT, a case or force module might describe a "security police unit" that is needed to support a particular mission; or it might describe all of the army units that are needed to support a given mission. Whatever the case represents, each case in the case library is indexed such that it can be recognized as applicable when appropriate. Access to relevant past experiences, and often some notion of how successful the solution was, is provided through case indices. For example, a FM in ForMAT that contains the elements of a security police unit could be indexed by "function = security"; the name of the mission, e.g., "mission-name = typhoon-mary"; and/or descriptions of the geographical location to which the force will be deployed, e.g., "climate = tropical"; "terrain = desert". Any of these individual indices, or some combination of them, can be used to retrieve the FM.

System Parameters

There are currently two versions of ForMAT. ForMAT 1.3.1 is the released version of the system. It has been deployed to several operational sites, used to support numerous exercises, and is currently being used to guide the development of an operational version of the system. ForMAT 1.4 is the research version of the system.

ForMAT 1.3.1 is written in Common Lisp with the CLOS object-oriented system using the Lucid Lisp 4.0.1 compiler, while ForMAT 1.4 uses Allegro 4.2 as the compiler. The graphical interface of both versions is written in Garnet, an object-oriented GUI tool kit built on top of CLX, a Common Lisp interface to X windows.

ForMAT runs on Sun Workstations, with ForMAT 1.3.1 running under both (Sun OS 4.1.x) and Solaris 2.x, and ForMAT 1.4 running only under Solaris 2.x. ForMAT requires 130 megabytes of disk space (primarily to store deployment plan data) and a minimum of 64 megabytes of RAM. Sufficient swapping space is also required.

The system can operate as a stand-alone system on a single platform or can be operated on a properly configured Sun workstation with other tools such as DARTs TPEDIT, LOGGEN, and TARGET loaded (Walker & Cross 1994). While ForMAT uses its own internal database, a flat file version of the TPFDD can be imported to/from systems like TPEDIT that store the TPFDD in an Oracle database.

Domain Background

A force module, or FM, is a structure that describes a military force (and its associated support) independently or as it is required to satisfy a given military planning objective. The United States deploys military forces to support a variety of problem situations (missions); from combat to humanitarian, large and small, anticipated and crisis (Armed Forces Staff College, 1991).

In order to build a deployment plan, representatives from the various services and commands involved in developing the plan must collectively decide how best to allocate the limited transportation resources (aircraft, ships, trucks and trains) so as to achieve the many military objectives of the mission.

In the early phases of deployment plan development, military forces and support units may be specified generically (e.g., fighter aircraft). Movement times (i.e., latest departure, earliest departure, latest arrival, earliest arrival), origins and destinations are also specified generically. As the operational plan for which the deployment plan is being developed matures, so too does the deployment plan; and in the process each specification for a generic requirement is replaced by information associated with actual times and with the actual unit(s) or equipment that will be used. Once the planners have identified what is to move, they must check to see if the plan is logistically and transportationally feasible.

The initial development of the deployment plan (or TPFDD) is often based on how forces were deployed in the past or on a generic description of a particular force and it's support. In a crisis situation, time becomes a critical factor and one way the planner can save time is by reusing all or part of deployment plans that were used to support similar missions in the past. ForMAT was developed to facilitate the retrieval and reuse of all or part of plans from the past that are somehow similar to the current situation.

Development of a deployment plan is iterative, with changes necessitated by changes in the planning objective, or by problems encountered when logistic and/or transportation support is considered. In order to provide support to the user as the plan changes, ForMAT also provides tools that support the monitoring, maintenance and revision of the plan.

Using ForMAT

ForMAT provides an environment in which a force development and planning specialist can build, modify, and monitor deployment plans. ForMAT provides the user with access to a library (case base) of FMs (each belonging to one or more TPFDDs) for use in building new FMs or in building a new TPFDD. In ForMAT the TPFDD can be comprised of one or many FMS, and a user can use all or part of existing TPFDDs (deployment plans) in order to build or revise another plan.

To illustrate, assume that a typhoon hits a country in Southeast Asia. The ForMAT user can query ForMAT to quickly determine if there are any plans in the case base that were used to provide support for a typhoon disaster relief mission in Southeast Asia or anywhere else in the world. The user finds relevant information by executing exact or generalized queries against all of the FMs in the case base. As the user executes queries, ForMAT records all FMs that are retrieved and presents them to the user as a enumerated list. The ordering is based on how many queries each FM satisfies. The report also indicates the name of the plan that the FM belongs to, and typically, the user discovers that a number of related queries will yield references to one or more plans. The user can view the plans in order to determine what forces and other types of support were used to provide support during typhoon disaster relief missions in the past.

If the user decides that the retrieved FMs are useful for creating a new plan to respond to the current problem, then the user can use functions provided through ForMAT to "create" a new TPFDD. The user is then free to mark and copy one or more relevant FMs, or even a hierarchy of FMs from an old plan (as stored in the case base) into the new TPFDD. The user then needs to set or modify the ULNs (Unique Line Numbers) that uniquely identify the deployment requirements of the TPFDD to specify the details of deployment, e.g., origins and destinations for the current mission. This can be done within ForMAT or in a related system called TPEDIT.

When there are no FMs that satisfy a user, the user can build a new FM based upon an already existing FM by copying an individual FM, or a hierarchy of FMs from one TPFDD to another; or the user can copy the force requirements (ULNs) of an existing FM from any plan into the new FM. ForMAT will guide the user with naming and numbering schemes in order to preserve the consistency of the developing TPFDD. Within the new plan, the user can add and/or remove ULNs from the FMs through cut, copy and paste operations. The user can also build a "Query-based FM" to track or organize information within a given TPFDD. A Query-based FM defined by the TPFDD query, e.g., (Service = Army) would collect all of the ULNs within the given TPFDD that satisfied the query "Service = Army" into the defined FM. Once defined, the contents of a Query-based FM are automatically updated to reflect ULN changes in the TPFDD.

Using ForMAT, parent/child relationships can be "explicitly" specified among FMs within a loaded TPFDD. The user can develop a hierarchy of lower level force modules which describe the capability and/or the organization of some "parent" force module. These lower level force modules (i.e., children) can consist of existing force modules that have been copied from the force module (case) library or they can be created by the user at any time. ForMAT automatically maintains consistency when changes are made between force modules based on these relationships. For example, when a ULN that the user wants to add to a child FM is added to the TPFDD, ForMAT automatically includes the ULN in all relevant parent FMs within that hierarchy. This consistency maintenance supports plan integrity and relieves the user of the current labor intensive process in the current operational system of having to rebuild all FMs when a ULN is added.

ForMAT also provides the following reports to support the user in analyzing FMs and in tracking changes:

- Basic FM Report - Lists the FM Short title, each ULN in the FM and the FM features.
- FM Comparison Report - Compares two FMs.
- ULN Feature Ratios Report - Describes how many ULNs make up a feature's value.
- ULN FM Membership Report - Describes which FMs a ULN belongs to within a TPFDD.
- ULNs Not In Any FMs - Lists the ULNs that are in a TPFDD, but not contained in any FM.
- Current TPFDD Status - This is a summary report that describes the changes made to the TPFDD since the last save.

FM Indices

In ForMAT as the user builds a plan, the system and the user build indices to describe the FMs that comprise the plan. The indices are later used to support the retrieval of those FMs. Users can build highly personalized indices or ForMAT will automatically generate indices that are descriptive of the elements that make up the FM. For example, relationships between TPFDD data fields and indices can be specified and automatically maintained through ForMATs feature augmentation rules (FARs). When the user changes the composition of a FM (adds an air-force fighter aircraft), the FM indices that are defined by FARs (e.g., aircraft-type, function and service) will be automatically updated to reflect this change. While most FARs are invariant across theaters and services, (e.g., FARs that parse country codes, service codes, and transport-type codes) some of the FARs, particularly those that parse the unit function codes, are service and/or theater dependent. Additionally, a user may modify a FAR to fit his/her specific needs.

ForMAT currently contains 47 indices, however, 7 of these indices have never been used to index more than one FM. Research efforts are underway to study the usage of indices: which indices are most frequently used in retrieval; which indices are most frequently changed; and what indices are synonyms. Additionally, ForMAT maintains an index dictionary that allows the user to make wildcard searches for defined indices.

The ForMAT Interface

ForMATs user interface includes multiple window displays with a host of function buttons and menus that enable planners to accomplish functions such as: load TPFDDs, create or modify FMs, load/save case bases, generate queries to the case base, etc. The top level window provides a workspace in which the user can load data from a TPFDD, create or load a new case base, perform FM queries, review FM information, and create new FMs. Another window is provided (the FM Edit window) that allows the user to view the ULN contents of the FM and associated indices. In this window, the user is provided with buttons and menus that support modification of the FM. As the ULN contents of a FM are modified, ForMAT will automatically update associated FM indices.

Users can view and modify the hierarchical composition of an entire deployment plan through the LINKS window. Through the LINKS window, the user can view the FMs of a TPFDD graphically. A given deployment plan may be made up of one large FM, a hierarchy of FMs, or some combination of single and hierarchical FMs. Options are also provided in the

LINKS window for changing the hierarchical organization.

Remote Usage While ForMAT was developed primarily to support the deployment planner in the mixed-initiative mode, ForMAT can be used autonomously, e.g., ForMAT queries can be generated and executed by another system. To illustrate, after a mission planning session using BBNs TARGET system, the operator can estimate force requirements by passing mission specifications to ForMAT such as:

- MISSION = DESERT-BLAST
- THEATER = PACOM
- GEOGRAPHICAL-LOCATION = KOREA
- FUNCTION = MISSION-AIRCRAFT
- SERVICE = AIR-FORCE
- DEST-CC = UNKNOWN
- UIC = "WALOAA"

These specifications are automatically converted to queries for execution in ForMAT. As the queries are executed, a listing of the retrieved FMs is compiled in a query report. Each FM is prioritized by the degree to which it satisfied the individual and cumulative query statements. During JWID-94, ForMAT successfully received information from TARGET and generated relevant lists of potential mission forces (Maybury 1995).

Modification Challenges

All user interaction, including user-guided FM modification is recorded by ForMAT's "history mechanism". To date, the trace provided by the history mechanism has been used by a knowledge engineer to study how users create, retrieve, modify, and incorporate FMs into a plan. Additionally, several of the reports generated by ForMAT has been successfully used during a recent military exercise to identify the changing data fields and FM indices of the deployment plan as it was built and refined (Mulvehill 1995).

Although ForMAT can be used to detect what fields are modified, there is currently no reasonable way available to determine "why" the fields were modified. This problem is further complicated by the observation that experienced planners tend to use ForMAT quite differently than novice users. For example, expert planners who have had a good deal of exposure to the methods provided by the current operational system, i.e., the Joint Operations Planning and Execution System (JOPES) understand the structure and language of the TPFDD. The expert planner will automatically know that if the "transport-type" is a ship, than the "origin" needs to be a seaport. The novice user might change the "transport-type" but not realize that the value of "origin" may also need to be changed accordingly. If the novice user is not using a system such as TPEDIT (which supports constraint checking on TPFDD data fields) to make such a change, the plan will become inconsistent. In addition, experienced planners tend to reuse their own plans, while novice operators tend to use template FMs or copy FMs from any plan that "appears" to satisfy their requests.

While ForMAT currently supports consistency checking on mundane bookkeeping values that the planners do not necessarily want to pay attention to, ForMAT does not suggest what FM to use or how a force module from a past plan should be modified for use in a current situation.

Future ForMAT Enhancements

Several enhancements to ForMAT are currently being investigated. These include:

- Joint work between MITRE and the University of Maryland is focusing on the potential to use the case-retrieval subsystem of CaPER (a case-based planning system implemented using high performance AI computing techniques) as support to the ForMAT system. Current work focuses on identifying specific queries and query classes that take too long to perform on the serial MITRE system. The development of parallel algorithms for these query classes, and the testing thereof, are the focus for this research (Hendler & Mulvehill 1996).
- ForMAT currently interfaces with GEs TACHYON system, which is a temporal reasoning tool. Using data from ForMAT, TACHYON can be used to display and reason about the temporal relationships among the ULNs of a FM. To date, ForMAT can pass the ULN information from one or more FMs to TACHYON and TACHYON can then be used simply to generate gantt displays and pert charts, or TACHYON can be used to provide "what-if" support for plan sequencing, which may include both force and extra-force constraints. We are investigating better integration of ForMAT and TACHYON to provide feedback to the planner on the potential for temporal constraint violations, and to enable the planner to build contingency plans to offset detected potential temporal constraints.
- Providing advice to the user on how to modify retrieved FMs and their constituent ULNs so that they can be used to satisfy the requirements of the current

situation has been a major challenge for ForMAT developers. To date, ForMAT has been used to track the user as the plan changes and report mechanisms have been used to inform the user of plan inconsistencies. In addition, ForMAT's index augmentation mechanisms (FARs) have been used to support plan integrity – which is one type of modification. While ForMAT can detect and display inconsistencies in a plan, ForMAT has little or no knowledge of why the current plan is being built or why a previous plan was built; although some knowledge is available in the description text associated with a FM, and the initial mission guidance that necessitated the development of a plan is available through other systems.

Work with CMU's Prodigy-Analogy system (Veloso 1994) is being pursued to investigate and to provided more sophisticated methods for providing support to the user in determining how best to modify the past elements of a plan to fit the current situation. One method that is being explored involves the manual parsing of the mission guidance into a set of goals and subgoals which can then be associated (as indices) with the FMs of a given plan. ForMAT's history mechanism is also being modified to provide information to Prodigy-Analogy indicative of how: (a) the user builds goals, (b) the goals are associated with and used to build deployment plans, and (c) other operators, e.g., "create-fm" are used during plan development and modification.

Conclusion

ForMAT is a research tool, developed as part of the RL/ARPA Knowledge Based Planning and Scheduling Initiative (PI), to acquire information on, and to support the development of force deployment plans. ForMAT employs case based reasoning (CBR) techniques to store, index, and retrieve FMs; and knowledge acquisition/engineering techniques to facilitate classification and to monitor how the system is being used (Cross et al. 1994).

Within ForMAT, a FM is represented as a case within a case base. Through ForMAT the user can create, view, retrieve, modify, and manage FMs; as well as import and export the TPFDD to/from other systems. Currently, ForMAT contains 319 cases derived from 16 plans (TPFDDs). 9 of the 16 plans were built over a two year period either during user training, or during military exercises and demonstrations. Each of these 9 plans were built through the reuse of FMs in the other plans.

Using ForMAT, a user can quickly construct a new plan by reusing parts of stored plans. ForMAT provides user interface capabilities to facilitate the search for force modules and data consistency mechanisms to support the user as he/she modifies the plan.

ForMAT embodies many of the functions required to support an operational environment. ForMAT has been used repeatedly and successfully in a number of military exercises, and is considered to be sufficiently stable and complete (along with its associated documentation) to serve as guidance for the development of an "operational" system.

Acknowledgments

This work has been funded by ARPA and Rome Laboratories (RL) as part of the ARPA/RL Knowledge Based Planning and Scheduling Initiative.

References

Armed Forces Staff College, 1991. *The Joint Staff Officer's Guide, AFSC PUB 1.* Norfolk, Va.

Cross, S. E., Roberts, D., Mulvehill, A. M., and Sears, J. A. 1994. Case-based Reasoning Applied To A Force Generation Decision Aid. In Proceedings of the 8th International Symposium, Methodologies for Intelligent Systems, 174-182. Charlotte, North Carolina, Springer-Verlag.

Hendler, J. and Mulvehill, A., Jan. 1996. *High Performance Support for Case-Based Planning Applications.* Israeli Conference on Artificial Intelligence, Tel Aviv, Israel.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, Ca.: Morgan Kaufmann Publishers.

Maybury, M. T. November 6 - 17 1994. Distributed, Collaborative, Knowledge Based Air Campaign Planning. In Proceedings of the NATO/AGARD Lecture Series 200 on Knowledge-Based Functions in Aerospace Mission Systems, 2-1 - 2-13. Torrejon, Spain; Chatillon, France; and NASA Ames, CA.

Mulvehill, A. 1995. Reusing Force Deployment Plans. In Proceedings of the AAAI 1995 Fall Symposium on Adaptation of Knowledge for Reuse, Cambridge, MA.

Veloso, M. 1994. *Planning and Learning by Analogical Reasoning.* Lecture Notes in Artificial Intelligence 886, New York: Springer-Verlag.

Walker, E. and Cross, S. eds. 1994. *Proceedings of the ARPA-Rome Laboratory Planning Initiative Annual Workshop.* San Mateo, Ca.: Morgan-Kaufmann Publishers.