AI Magazine Volume 23 Number 2 (2002) (© AAAI)

TALPS
The *T-AVB* Automated Load-Planning System

Paul S. Cerkez

■ Because of military drawdowns and the need for additional transportation lift requirements, the United States Marine Corps developed a concept that enabled it to modify a commercial container ship to support deployed aviation units. However, a problem soon emerged in that there were too few people who were expert enough to do the unique type of planning required for this ship. Additionally, once someone did develop some expertise, it was time for him/her to move on, retire, or leave active duty. There needed to be a way to capture this knowledge. This condition was the impetus for the T-AVB AUTOMATED LOAD-PLANNING SYSTEM (TALPS) effort. TALPS is now a fielded, certified application for Marine Corps aviation.

Historically, one of the most difficult problems facing marine aviation logistics planners was finding an affordable, flexible, and rapid means of providing intermediate maintenance capability for forward-deployed aircraft. To overcome these challenges, in the mid-1980s, the Department of the Navy purchased the *T-AVB*s, and United States Marine Corps (USMC) aviation simultaneously introduced the MARINE AVIATION LOGISTICS SUPPORT PROGRAM (MALSP) (figure 1).

MALSP incorporates a flexible building-block concept, known as *contingency support packages* (CSPs), that follows a prearranged deployment and employment scenario for assembling the right mix of marines, support equipment, mobile facilities, and spare parts within a Marine aviation logistics squadron (MALS) to support deployed aircraft. The key word is *flexible.* CSPs can rapidly be configured to support the contingency aircraft mix and marshaled for movement. CSPs comprise the fixed-wing or rotary-wing common support and/or the peculiar intermediate maintenance activity (IMA) and supply support for the various deploying aircraft. Initial support packages (30 days of spare parts) called *fly-in support packages* (FISPs) are flown into the operational theater as part of the fly-in echelon (FIE); the balance of the Marine Air Ground Task Force (MAGTF) commander's tailored aviation logistics support arrives in theater aboard the *T-AVB*. Without the *T-AVB*, it would require more than 140 *C-141* cargo aircraft flights to deploy an MALS with an IMA-level capability to a crisis area.

The *T-AVB* ships were acquired as a result of the USMC (1983) "Feasibility Study of the Aviation Logistics Support Ship." Two ships have been modified for use by USMC I-Level aviation maintenance and supply organizations. The Department of Transportation Maritime Administration (MARAD) maintains the ships in a five-day reduced readiness status using a civilian, commercial U.S. Merchant Marine retention crew stationed aboard each ship to monitor equipment conditions and conduct vessel maintenance and repair. The *T-AVB*s are part of the maritime prepositioning force (MPF).

The mobile facility work centers used by the Marine Corps conform to the standard commercial International Standardization Organization (ISO) container dimensions, which are 8′ x 8′ x 20′. Figure 2 shows a typical work center mobile facility being prepared for loading. Figure 3a shows mobile facilities "complexed" at a shore base, and figures 3b and 3c shows part of the same capability on the ship. *Complexing* is the process of moving mobile and assembling facilities into functioning work spaces. A complete work space is said to be *complexed* when it is capable of doing its assigned job(s). A collection of functional work centers is called a *complex*. Figure 3c shows a

Figure 1. SS Curtiss (T-AVB 4) on an Exercise off the Coast of California.

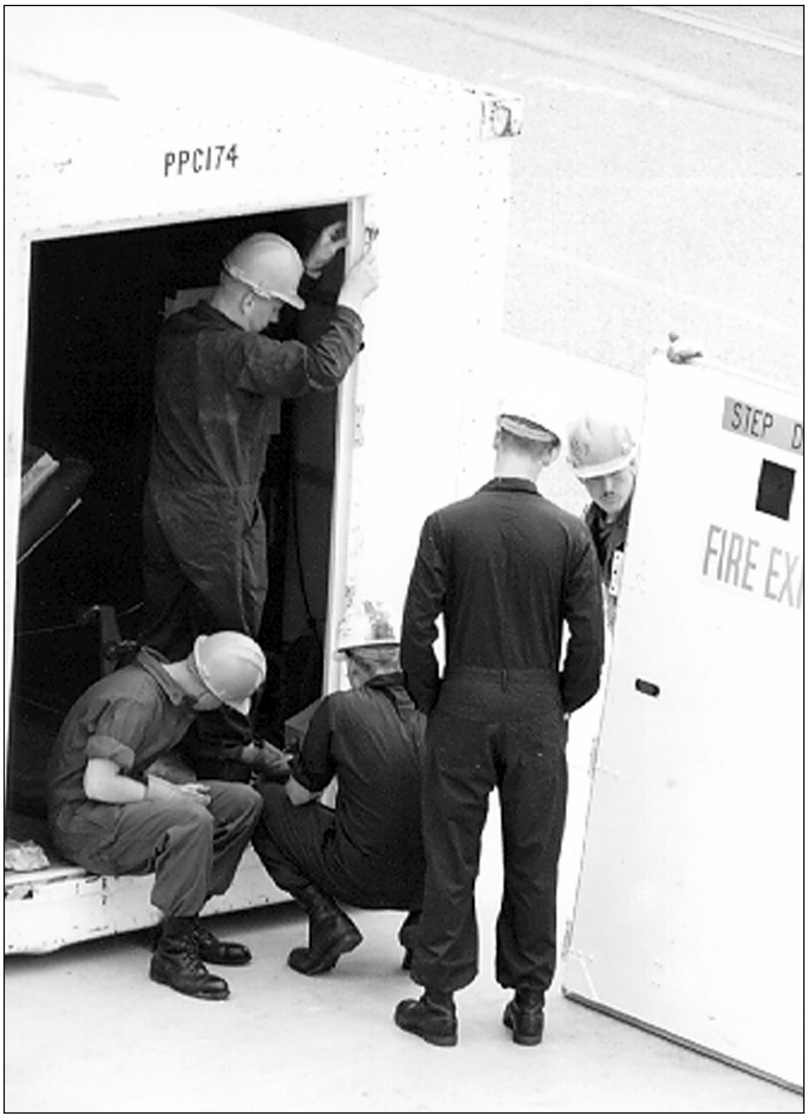


Figure 2. Mobile Facility Being Prepared for Loading.

special doublewide arrangement, scaffold ladders, and additional maintenance mobile facilities. Access modules are used to access second- and third-tier mobile facilities that are complexed below decks in support of IMA-level repair capability. Figure 4 shows a typical access module.

The modifications to the ships to support a mobile facility setup allows a MALS to operate fully functional work centers on board a ship, in an expeditionary mode ashore, or both. Two basic load-out configurations exist for each ship: transport mode and operational mode.

In the *transport* configuration, the ship is loaded for maximum capacity. In this mode, mobile facilities are not accessible, and the equipment contained therein cannot be operated. In this configuration, more than 650 twenty-foot equivalent unit (TEU) containers can be loaded. In this mode, the ship is a standard container ship and supports resupply operations and missions. The function of resupply is the secondary mission of the *T-AVB*.

In the *operational* configuration, the ship is loaded such that mobile facilities can be placed in a functional, operating condition. What you have in effect is a tailorable, floating, aviation repair facility. Officially, in this configuration, 300 mobile facilities and 42 access modules can be loaded, or 342 TEUs. This configuration allows the embarked work centers to process and repair defective or broken aircraft components while en route to an operational theater or, should the concept of operations in theater dictate, continue operating until finally moved ashore (referred to as operating *in stream*). This mode describes the primary mission of the *T-AVB* and the most difficult area of load planning. The views in figures 3b and 3c show the operational mode.

A third mode exists called *combination.* As you might guess, it is any time the ship is in any condition other than operational or transport. As mentioned earlier, flexibility is the key. Some holds of the ship might be fully loaded in transport mode, yet others are in operational mode. Combination mode is usually planned for when more cargo than is used in operational mode must be embarked, but some capability must be available at all times. Typically in this mode, the ship is offloaded at the destination and then "back loaded," if necessary, to provide full capability. The term *back loading* (or *back loaded*) refers to putting mobile facilities back on the ship in operational mode that might have been nonfunctioning prior to the offload.

Load-Planning Overview

The embarking Marines responsible for a particular ship must develop the load plan for this ship. The civilians manning and loading the *T-AVB* will load the ship any way the Marines tell them as long as it does not put the ship in an unsafe condition. *Unsafe* is defined as any condition that would "hazard" the vessel. For example, if the ship were loaded so that it was top heavy or too heavy on one side, it would put the ship at risk of capsizing. "Most significantly, a ship's officer is concerned to keep his ship from capsizing. Without sufficient stability in a rolling motion, this goal would be in jeopardy" (La Dage and Van Germert 1990). A top-heavy vessel is said to be tender or cranky. The ship's roll is slow and tends to lag behind the changes in sea-surface inclinations. The ship tends to not return to a vertical position.

T-AVB load planning is a time-consuming, inflexible process made more so by the high tempo of operations and pressure to execute operational orders in the time allotted in a time of war. The manual system of load planning is not responsive (in a timely manner) to modifications in the force structure, concept of employment, or both. There is no formal training, and on-the-job training (OJT) opportunities for implementing and exercising load-planning considerations are scarce. The lack of this experience and training was abundantly evident when the *T-AVB*s had to be loaded for Desert Shield and Desert Storm. At this time, the *T-AVB* concept was still new, and there were no experts. It took 5 full 24-hour days to load one of the ships for deployment to the desert. With all the changes, the actual manifest and inventory had to be validated manually after the ship set sail (figure 5). Changes were being made until the end. Despite the problems, the value of the *T-AVB*s was fully realized.

The following facts are true: First, load planning is complex and tedious. Two, no formal training is available. Three, attrition of experienced personnel occurs regularly (orders, retirement, force reductions, and so on). Fourth, if the load plan is found unstable or modified after being presented to the ship's first mate or master, it must be redone. Fifth, for a variety of reasons, the *T-AVB*s will not be exercised often enough to maintain a knowledge base readily available to plan loads and deploy.

To develop a load plan, the planner must have a list of all mobile facilities and cargo to be loaded. Mobile facilities embarked include not only maintenance work centers but also supply department mobile facilities, bulk cargo, and rolling stock. This list must identify mobile facility-container power requirements; mobile facilities needing air or water hookups; mobile facilities-containers needing access; mobile facility interconnection requirements (shop integrity); ownership of the mobile facilities (rotary wing, fixed wing, work center, and so on); type of mobile facility; projected off-

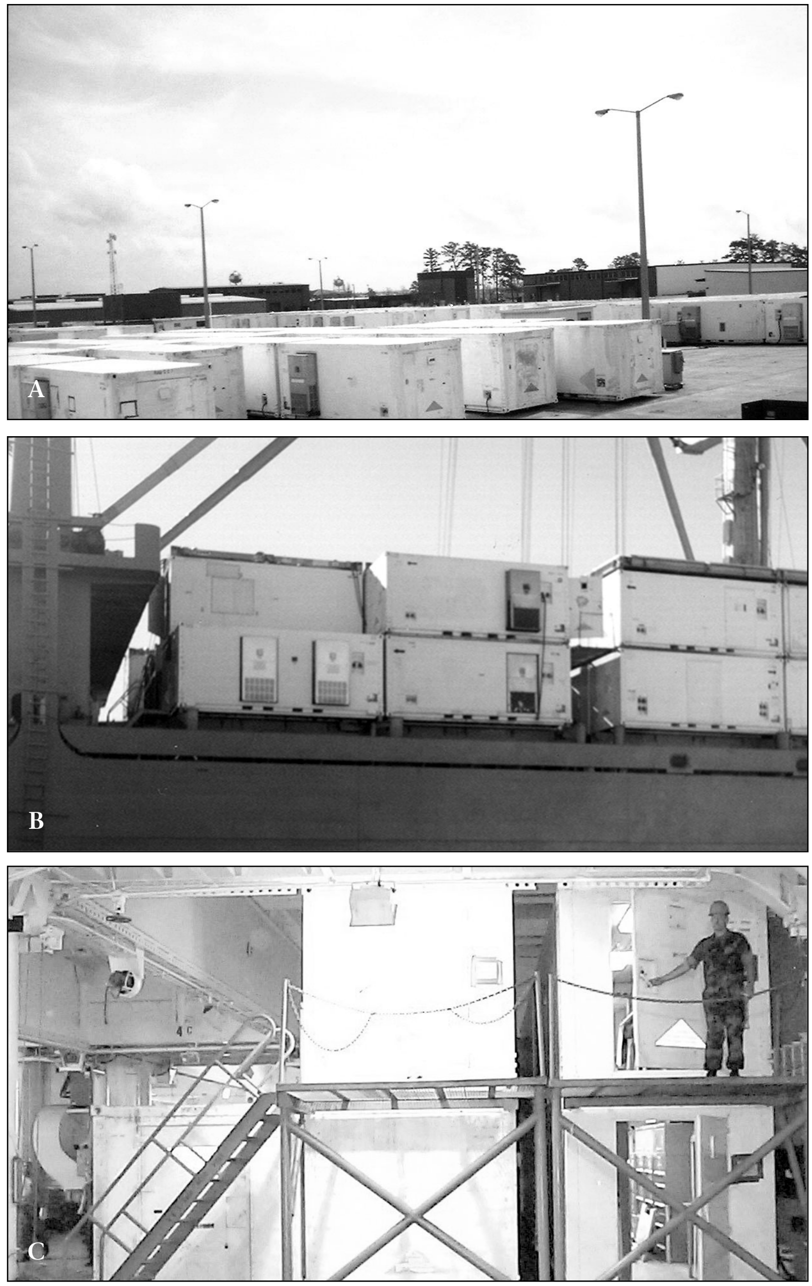


Figure 3. Mobile Facility Complexes.

A (top). Mobile facilities "complexed" ashore in operational mode. B (middle). Mobile facilities complexed on ship in operational mode. C (bottom). Mobile facilities complexed inside the ship.

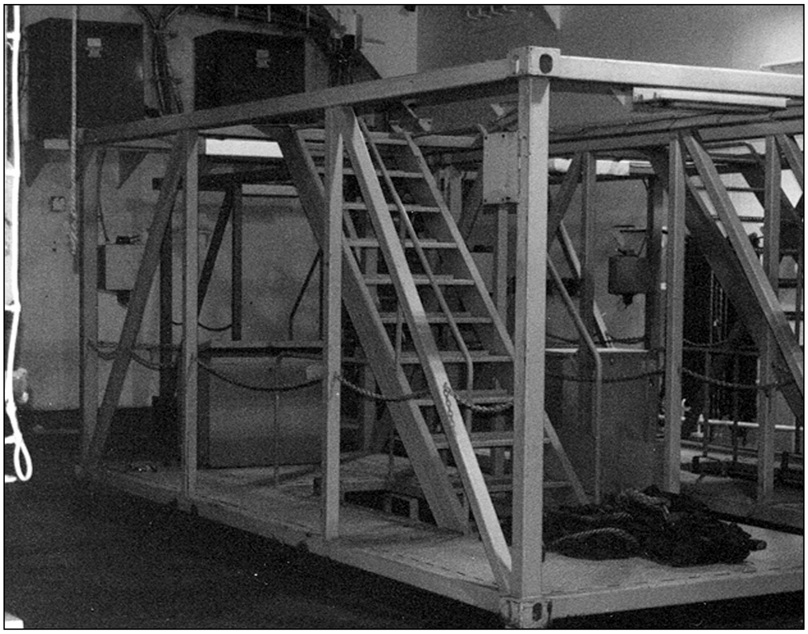


Figure 4. Access Module.

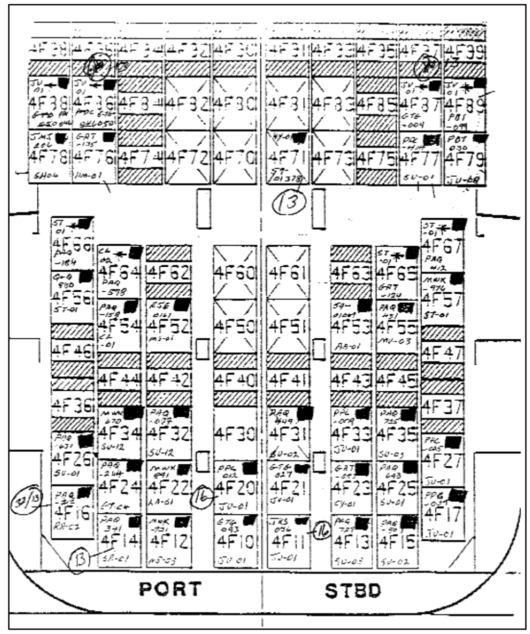


Figure 5. Manual Load-Planning Sheet.

load priority; the availability and locations of facility assets on the ship (air, power, and so on); special limitations on locations or mobile facilities; types of additional cargo (rotor blades, nose cones, rolling stock [mobile motor generators (MMGs), mobilizers, and so on], petroleum, oils, lubricants, and so on); pier-side facilities at both departure and destination ports; and status of the ship's cargo-handling equipment, mobile facility support systems, and ship access points (hatches, ramps, and doors).

Once the load planner has all the requisite data in hand, he/she must compare what is needed against the ship's facilities and develop a proposed load plan. After the load plan is completed, it must be presented to the ship's first mate or master for approval. If the generated plan is found to be unsafe (that is, "…the ship floats upside down"), it must be redone. Any modifications to an approved plan also require resubmission and approval.

Taking into account the earlier conditions, assume it takes the load planner only 1 minute for each item of cargo to identify where to place it in the ship, with more than 350 mobile facilities and access modules, it will take over 5 hours to develop a load plan for just these items (figure 6). Now, add to this rolling stock and other bulk stores-cargos that might take two minutes for each item because of irregular shapes and sizes and ability to stack (or lack of). Assuming no changes in what is to be loaded, an experienced, seasoned load planner can develop a load plan to present to the ship's first mate or master in about 8 to 10 hours. In reality, it takes anywhere from 1.5 to 2.5 days to develop the initial load plan. In private conversations I have had with experienced load planners, a common theme is "a twenty-foot box in a twenty-foot hole, throw me in that briar patch…. You want the *T-AVB* in the operational mode? Goodbye. Call me when you figure out where you want everything and then I'll load it for you."

The problem presented here has been likened by some to an extension of the classic bin-packing problem. Others have called it a scheduling problem. In reality, I see it as both. When you add the temporal aspect of the ordering that cargos must be loaded (in which of five ports will it be loaded on and which of five it will come off), the system must be "aware" of the spaces above and below when it is identifying a cargo item for storage. For example, if a cargo item is not going to be loaded until port 3, then cargo items loaded in ports 1 or 2 cannot be designated for the space above the one being held for the port 3 item. You would end up having a "container in space." Prior to TALPS, *T-AVB* load planning had always been done manually, (the "stubby pencil" method). Figure 5 is an actual planning sheet used for one load out). The particular problems presented by this unique situation made it an ideal candidate for automation.

Automation of the Load-Planning Process

The purpose of the T-AVB AUTOMATED LOAD-PLANNING SYSTEM (TALPS) is to automate the *T-AVB* load-planning process. The TALPS program uses AI to follow the same logical steps that an expert uses in completing complex tasks associated with load planning.

The ability to develop load plans with a Prolog-based expert system was proven in the early 1980s when SRI International developed the AUTOMATED AIR LOAD-PLANNING SYSTEM (AALPS) for the U.S. Air Force using Quintus Prolog. AALPS was constraint based, but like a number of other load-planning programs, it requires the user to place the item of cargo. The aircraft cargo loading system then validated the load against all constraints. Stanley and Associates developed a ship loading program called COMPUTER-AIDED EMBARKATION MANAGEMENT SYSTEM (CAEMS) using a Paradox database driving an AUTOCAD user interface, interfaced using the C language, for the USMC in the late 1980s. CAEMS was used to help load the ships coming back from Desert Storm, and a much improved, updated version is still in use by the USMC embarkation community today. AUTOSHIP (Autoship Systems Corporation) is another software tool available to commercial shipping companies that supports loading containerized cargo. AUTOSHIP is a ship-type, class-specific tool and is configured at purchase time for the vessel(s) it will support. The U.S. Army Military Traffic Management Command had an application called CODES that is in the process of being upgraded, modernized, and renamed to ICODES (INTEGRATED COMPUTERIZED DEPLOYMENT SYSTEM). ICODES is being developed by the CAD Research Center, California Polytechnical Institute in San Luis Obispo, California, and has seen limited fielding (as of this writing). This listing is by no means exhaustive. There are a number of other applications that are similar to the ones mentioned here.

CAEMS, ICODES, and AUTOSHIP operate primarily the same way for ships that AALPS does for aircraft loads: The user loads the cargo, and the system validates the load against constraints. These programs are designed to be extremely flexible in that they never know what kind of ship or load they might have to develop. All three are template based. CAEMS does have an AI module that does *autoproration* (a term used by the developers to describe how the module computes the flow of cargo into a location), but ICODES is an AI agent-based application (originally built using CLIPS [C LANGUAGE INFERENCE PRODUCTION SYSTEM] and now being developed in C++) that will automatically place cargo items in a template developed by the user. These routines analyze the cargo to ensure it can get to its designated cargo storage location (that is, can it fit through the hatch, make the turn onto a vehicle ramp) and assign specific cargo items to the template locations. The templates act as "greedy attractors" (locations trying to pull certain types of cargo to them) to specific cargo types, and individual serialized items are then stowed. For example, if the template shows a position for an M1A1 tank for Unit A, any one of Unit A's tanks could end up there unless the user designates a specific one. Developing these templates is the most time-consuming operation of load planning; it is, in effect, manual load planning.

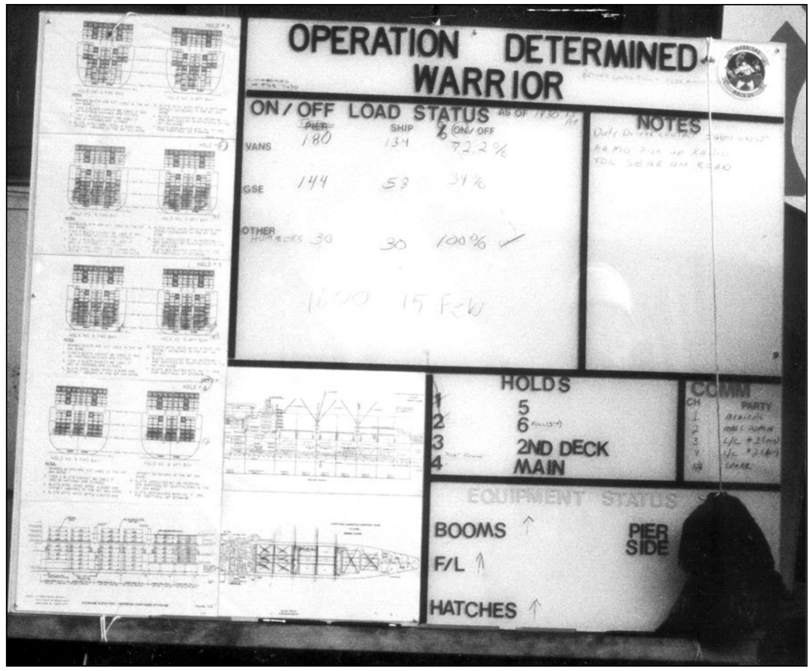


Figure 6. Manual Load Tracking.

Although TALPS will also support this manual cargo-placement method of operation, the significant difference with TALPS is that it can also place the cargo automatically. With most of the other systems, a domain expert is doing the template and load-plan development. Because of the unique mission of the *T-AVB*, all the template knowledge for any type of load the ship is capable of carrying is in the TALPS fact and rule bases. Because of the unique functioning provided by the ship, there are extremely few people with *T-AVB* load-planning expertise. The problem is that there are domain experts for the ship (the *TAVB* itself), there are domain experts for the cargo (mobile facilities), and there are experts in container ship loading, but there are extremely few experts in all three domain areas. TALPS combines the expertise from all three domains for this application.

The PROLOG development environment chosen for this expert system is PDC's VISUAL PROLOG (latest version used for development is VIP 5.2). Prolog was chosen early on primarily because it allowed me to work directly toward a solution by taking advantage of its inherent backtracking mechanisms, built-in string-handling features, built-in list handlers, goal-seeking structures, and internal and external fact base storage. Prolog allowed me to work out the logic of the problem without regard to how to work out the implementation. I didn't have to worry about sequences as much as identifying the rules as they applied. I had considered VISUAL BASIC and C++ as the implementation language, but with each, it seemed I spent more time trying to figure out how to code the solution rather than work on the problem itself. Also, it seemed as though I had to generate a lot more code in the other languages than Prolog to do the same job. A side benefit of the environment I chose was that I could develop the graphic user interface (GUI) directly in the same language without the need for a multilanguage application, thus avoiding the problems associated with that.

The Advantages of TALPS

One important feature of TALPS is that it automatically considers the ship's load and stabilization requirements. As such, the ship's first mate or master will not reject a load plan as being unsafe. CAEMS and ICODES must export the load to another application for trim, stress, and stability (TSS) verification, representing one of the single most significant benefit of TALPS: time savings. With a manual load-planning time of 8 to 10 hours a session (that could be rejected as unsafe, thus restarting the 10-hour clock), the time to develop a load plan can be significant. In actual planning exercises, the time to complete a load plan with TALPS from start to finish has been under one hour. (**Note:** During the actual load-planning process, ICODES does compute the TSS of the load as part of its agent processing, but it does not certify the load.)

Additionally, TALPS provides cargo-preparation schedules, load-team assignments, cargo-flow schedules, power-distribution plans, and scaffolding and access plans. These additional items are by-products of the load-planning process within TALPS that normally would have to be prepared manually after the plan is approved. Each of these products would normally take hours by themselves to produce. All these products increase the efficiency of the loading evolution. CAEMS and ICODES do not provide these additional capabilities. (**Note:** ICODES has been designated as the Department of Defense [DOD] ship-loading software system and as such will be replacing both CAEMS and TALPS. Many of TALPS functions are being incorporated into ICODES. All functions that ICODES will absorb are expected to be completed by 2005; however, there are functions that ICODES will not pick up, that is, cargo preparation schedules, load-team assignments, cargo-flow schedules, load tracking) Figures 4 and 5 show some of the old ways of managing the whole load-planning and -execution effort.

The Evolution of TALPS

The TALPS efforts began in 1992 with a proposal from the Naval Aviation Maintenance Office to Headquarters Marine Corps, Department of Aviation, Aviation Support: Logistics Office. From 1992 through 1997, the TALPS development team participated in every *T-AVB* training exercise as observers, interviewed all load planners involved with each exercise, and extracted knowledge from the few load-planning manuals that existed for the ships. From that effort, a *T-AVB* load-planning manual was written, and the TALPS software was produced.

During the initial development efforts, the load-plan generation routines went through a couple of revisions. As more and more knowledge was gathered about the process of developing load plans for the *TAVB*s, the system was modified accordingly. Because the system was rules and facts based, handling most new conditions was simply a matter of adding new facts and rules to cover the situation.

Early on, we knew that the system would need to be flexible. A couple of different approaches were tried to give us the flexibility we needed. The most notable was the attempt to use a genetic algorithm (Goldberg 1989) to generate a load plan and then have it evaluated against the fact bases for fitness. This effort was attempted, but the genetic algorithm would never advance beyond 50-percent fitness; it would reach that point in about 10 to 20 generations. I tried many different approaches to get past the 50-percent problem (mutations; small, medium and large generations; different representations; different population sizes; fixed-length and variable-length chromosomes; single- and multiple-crossover points; Monte Carlo selections, roulette, random), but I could never seem to find out why 50 percent was the best it would do. I even rewrote the fitness routines from scratch. During the troubleshooting efforts, I even manually created a full generation of 100 entries, each having a fitness value in excess of 85 percent (at least 25 percent had a fitness of 100). For about 5 generations, it appeared as though the whole generation's population was approaching 100 percent, but then in a few more generations, it plummeted to close to what I previously had been generating randomly (about 10 to 20 percent). Then, in less than 15 generations, I was back at about 50 percent. After 6 months, the effort was abandoned because of product-delivery requirements and budgetary limitations. The lessons learned from developing the fitness function for the genetic algorithm were then applied to the rule and fact bases. (Someday I would love to go back and really find out what was going wrong).

In May 1997, TALPS 1.03c was certified by the American Bureau of Shipping (ABS) as a safe loading instrument, and the software was distributed to MALS. Even though target users were involved during the development cycle, after distribution to the users, some negative feedback was experienced.

During the development of TALPS 1.0, the primary guideline for development was that the system had to support and be traceable to Marine Corps doctrine, which meant that it had to be able to support the operations planner doing deliberate planning using *notional assets*. Simply put, load plans using *generic assets* (that is, notional) are generated to test concepts and develop "on-the-shelf" plans that could be used in an emergency. TALPS 1.0 supported this (notional planning), but a big problem noted after fielding was that is not the way the load planners worked most of the time. In doing deliberate planning TALPS worked fine, but it was tedious to use for exercises where load plans needed to be developed for training missions. For loads less than a full ship, TALPS was not very efficient. Planners had to manually place most cargo items. Part of the issue here was that during exercises, the ship is rarely loaded to capacity, whereas in deliberate planning, or "real-world operations," the ship is fully loaded. The TALPS algorithms were developed to support the fully loaded operational scenarios. In training exercises, load planners would plan loads based on all the free space available. The planners would place the cargos manually. In essence, for training exercises, TALPS was little more than an electronic "stubby pencil" that did TSS calculations.

In 1998, DCS Corporation was contracted to update the load-planning manual (USMC 1998), and in 1999, DCS was again contracted to update the TALPS software, the user interface, and the rules and fact bases to account for additional modifications made to the ships after the initial release of TALPS. TALPS is reviewed after the annual *T-AVB* exercise and updated or modified as required. TALPS 2.1 was fielded in November 2000 and was used to plan the 2001 *T-AVB* exercise. TALPS 2.x still supports doctrine, but now it also supports the smaller size load outs, as experienced in training scenarios. In December 2001, the annual user review of TALPS was conducted, and several modifications were suggested to further support *T-AVB* load planning. Most of the suggestions submitted were purely cosmetic, and a few submitted by ship personnel were to add functions that are available in other packages (or, in most cases, a process done manually) to ease their work. What follows is a discussion of the underlying methodologies of how TALPS works.

The Technology of TALPS

TALPS is primarily a constraint-based, expert scheduling system. TALPS is configured to recursively process all cargo items and assigns them to cargo locations. After each complete iteration of cargo assignments, the ship's TSS characteristics are evaluated. If any safety parameters are exceeded, the plan is rejected, the system backtracks (using the internal Prolog backtracking mechanism), and the system recalculates the load. By incorporating domain knowledge into the rules that process the cargo data, many of the conditions that would cause a plan to fail are avoided. By avoiding the known unsafe conditions, safe load plans are almost always generated correctly the first time.

As a result of the interviews during the initial TALPS development efforts, certain patterns emerged that later became ironclad. Certain mobile facilities will always be combined and colocated with particular other mobile facilities, and these "blocks" will almost always go into a select few ship locations. A *block* is normally made of two, three, or four mobile facilities. As a result, rules and facts were incorporated to take advantage of these heuristics. By building a fact base of these standardized blocks and their possible locations and adding rules to process them, blocks of mobile facilities can be assigned in seconds, leaving only the unattached mobile facilities to be dealt with by the system. One of the biggest challenges was to represent the knowledge and data so the PROLOG engines could process it (Cerkez 1995). In TALPS, the block's data are represented as facts, each containing a single paired list that represents a block. An example of the data representation of a single predefined block and three of the legal cell block sets is shown in figure 7.

The top-level clause (autoload) controls the

```
mf_blocks([“MV01B”,”2F32”,”MV02A”,”2F34”,”MV03”,”2A34”,”NA01”,”2A32”])
mf_blocks([“MV01B”,”2F33”,”MV02A”,”2F35”,”MV03”,”2A35”,”NA01”,”2A33”])
mf_blocks([“MV01B”,”3F32”,”MV02A”,”3F34”,”MV03”,”3A34”,”NA01”,”3A32”])
```

Figure 7. Data Representation of a Single Predefined Block.

```
cargo_item(1,”NONE”,”DRY006”,”TAVB”,”000”,”TAVB00”,”Z”,”0”,”DRY006”,”000”,ds,”AC”,”N
A,”TAVB only”)
cargo_params(“DRY006”,w(2500,2500,2500,2500,5000,5000,10000,”LBS”),d(240,”INCH-
ES1”,96,”INCHES2”,98,”INCHES”,1306.67,37),b(120,58.8,48))
cargo_info(“DRY006”,”Embark1”,”Debark1”,0,0,0,0,0)
```

Figure 8. Data Representation of a Single Cargo Item.

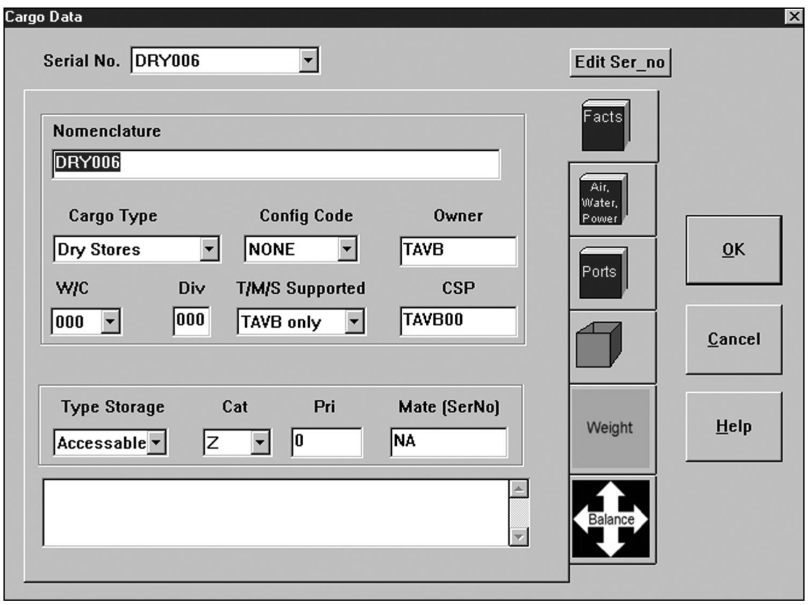


Figure 9. Data Representation Screen of a Single Cargo Item.

sequence of events, and cargos are assigned in an order that prunes the search space rapidly. Access modules are almost always placed in the same 42 specific locations on the ship. The autoload clause calls the subclauses that handle access modules early in the process, removing 42 cargo items from the search space. Dry stores and crew reefers (refrigeration containers) are handled the same way, removing another 10. Next come deep stow cargos. They require no access and are always put into the same 54 cargo locations, thus further reducing the count. The system then searches for predefined blocks, potentially removing another 50 to 60 mobile facilities from the search space. The previous steps are an application of the heuristics learned during the initial TALPS development efforts; the load planners always got these items out of the way first. By the time the system has to place individual cargo items, the search space has been reduced typically from 350 items to about 180, or about one-half the search space.

Another example of the heuristics involved is predefined blocks. The system reads in a predefined block, determines if all the necessary mobile facilities are available and awaiting location assignment (that is, not already assigned), and verifies that the cargo locations are available. If both are true, the block is then loaded, and system-stored parameters are updated to reflect the loaded cargo. If not, the system checks the next available set of preferred cargo locations. Once all the locations are exhausted, the current block is rejected, and the system backtracks and retrieves the next predefined block starting the process all over. Structuring the rules and data in this format allows the system to adapt to exceptions to the rules should there ever be a mobile facility block in excess of four mobile facilities. The exception is then handled in data without having to change any code.

Once all the predefined types are assigned, the system then starts assigning cargo items individually based on the parameters listed earlier. If any cargo item is placed that has a user-designated partner, they are then treated similarly as a predefined block. If both cannot be placed, then a new set of locations is searched for.

The three facts shown in figure 8 show the internal representation of a single cargo item. The serial number DRY006 is the link. The data represented carry all the data needed not just by the system but also by the user before and after loading. One task I had was to come up with a user interface that made it easy for the user to make changes in the data parameters and was easy for them to understand. Figure 9 is an example of the GUI supporting the data in figure 8.

Ship cargo locations are defined internally, as shown in figure 10. The cell number 6F13 is the unique identifier. The two lists at the end of the fact contain constraint data and preference data. By expressing a preference for a particular set of cargo types ([“DS,” “SEAC”]) into a specific location, the cell becomes greedy and tries to attract these types of cargo. The limitations list ([“TEU,” “ISO”]) prevents unwanted cargos. Various other data about the ship that affect load planning are also stored. As with the cargo, we also had to allow the user the ability to modify the cargo location data in easy-to-understand terms. Figure 11 is a sample of this GUI.

In all cases of cargo-to-location assignment, cargo-item and cell-location characteristics are evaluated. By structuring the facts and rules to account for cargo needs that matched cell facilities, I created a knowledge mapping that

```
define_cell(“6F13”,6,”FWD”,1,3,”G”,0,0,0,0,0,”
DSW”,1,[“TEU”,”ISO”],[“DS”,”SEAC”])
```

Figure 10. Data Representation of a Single Ship Location (Cell).

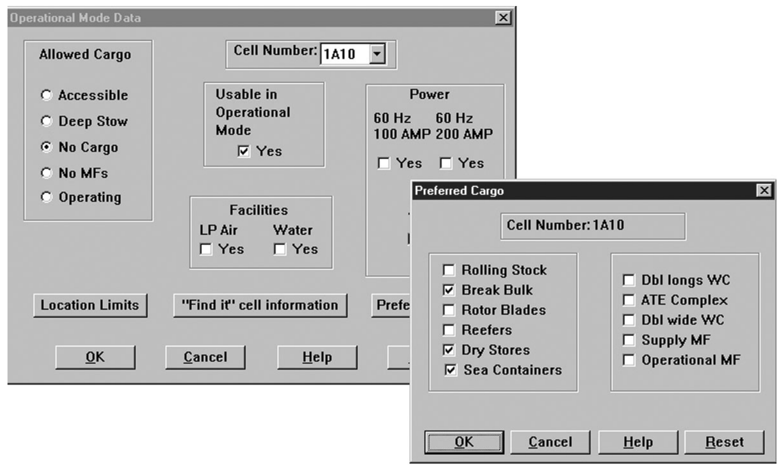


Figure 11. Data Representation Screen of a Single Ship Location (Cell).

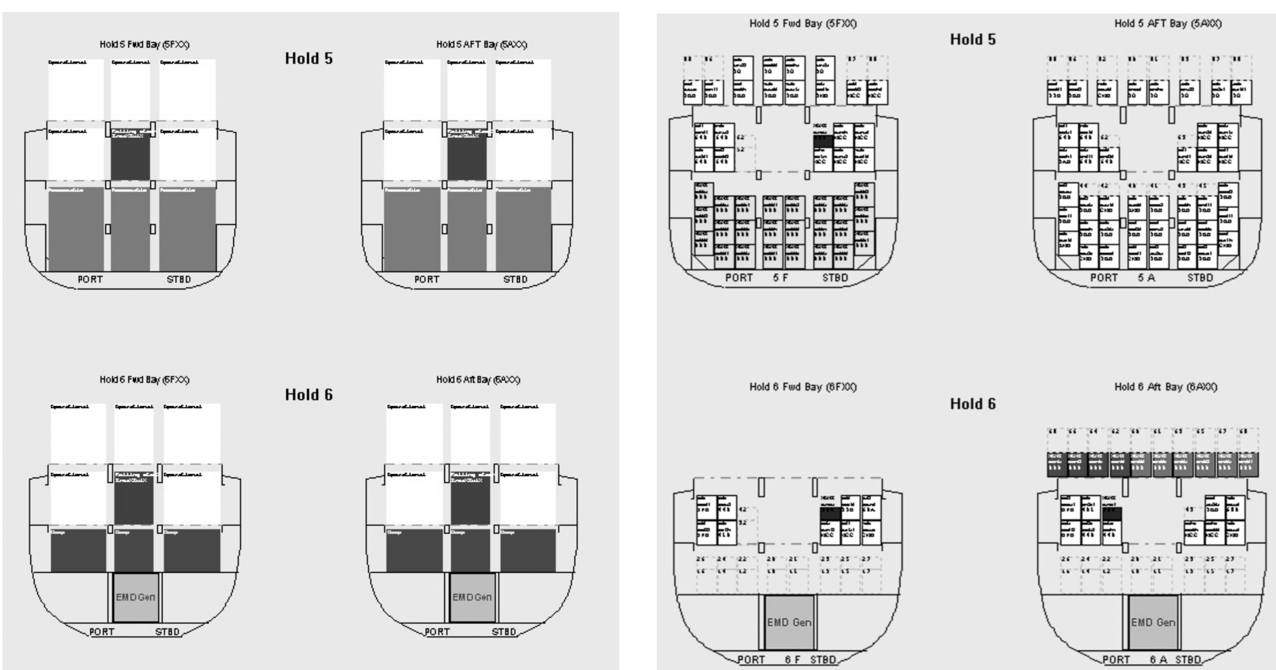


Figure 12. Hold Representations.

A (left). Hold schema screen. B (right). Hold load-planning screen.

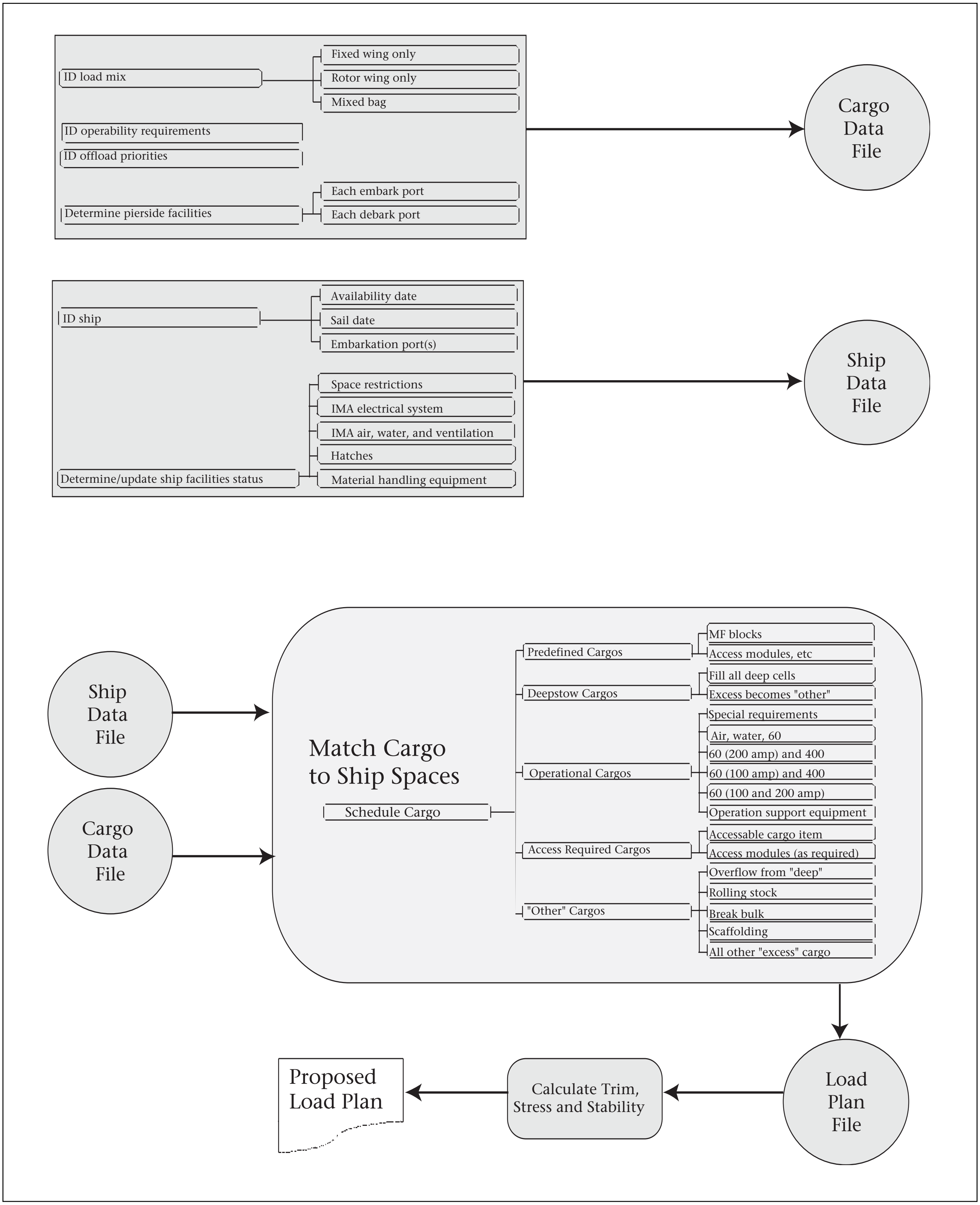


Figure 13. Simplified TALPS Flow Diagram.

allowed for direct pattern matches. The only thing I had to do extra was create a set of rules that handle the "don't-care" situations (for example, a cell provides 400-hertz power, but the cargo item does not need it). The five positions at the end of cargo_info (all zeros in this case) map directly to the five positions after the G in the cell location shown in figure 10. This particular cargo item is a dry store container (cargo_item fact, fourth field from the end: ds), which maps directly to the preferred cargo type in cell 6F13, DS. In this case, the cargo item and cell location would be a match.

Overall, the planning process allows the user to define the general concept of the ship's load, which is accomplished by setting a hold's parameters. A schema was developed to maintain the knowledge of the hold's capabilities as well as attributes of the hold in various configurations. Figures 12a and 12b show the hold schema screen representation and the resulting load-planning representation with cargo loaded.

In addition to the holds themselves, a separate schema was developed to maintain data about the cargo-handling equipment, access ports and hatches, and location usability based on the status of same. The user sets these parameters at any time during the planning evolution to reflect the current condition of the ship. Rules within the system act on these conditions and subsequently modify, as necessary, the cell-location parameters.

TALPS rules process all the facts, within constraints set by the user and imposed by the system, and rapidly produce a certified safe load plan. Figure 13 is a highly simplified drawing of the load-planning process. All the cargo data are entered or updated, all the ship data are updated, and then the cargo is scheduled into cargo locations. After the scheduling of the cargo, the load is validated for TSS. (The TSS summary screen in figure 14 is one of the dialogs used by the ship's first mate to verify the TSS of the vessel and to allow him/her to make fuel and ballast corrections.) At any point of failure, the system backtracks and starts the process again. The output is shown as a proposed load plan only because the one constant in *T-AVB* load planning is change!

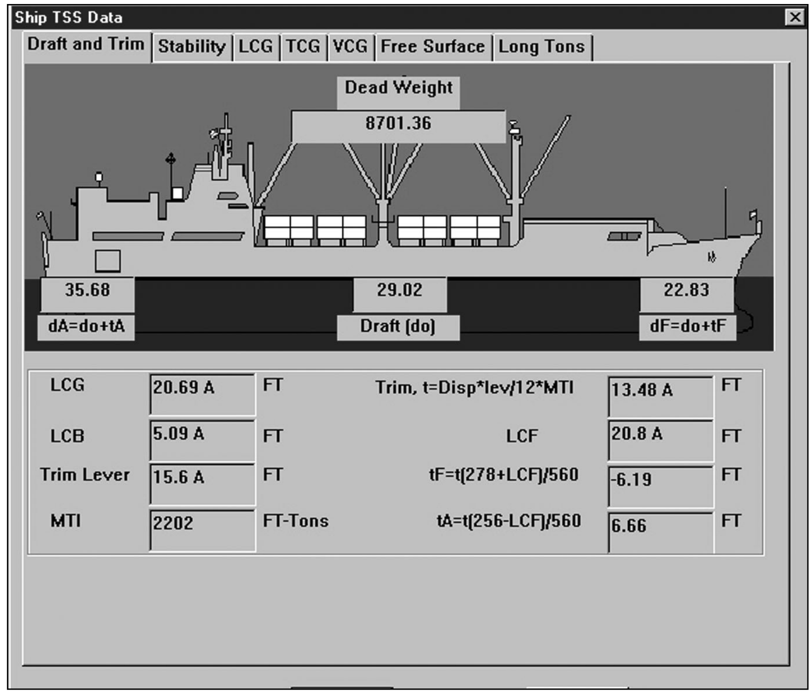


Figure 14. Trim, Stress, and Stability Summary Screen.

The Future of TALPS

TALPS was originally conceived as a tool to help the harried planner develop load plans for the *T-AVB* class of vessels. Over time, it evolved into a repository to maintain the volatile corporate knowledge of the *T-AVB* load-planning process. TALPS has currently planned existence until 2005 at which time a DOD ship-loading tool (ICODES) will be fielded. This new tool is designed to incorporate the knowledge and expertise that currently resides in TALPS as well as other ship-loading applications.

Acknowledgments

This project was sponsored by the Department of Aviation, Logistics Support (ASL-34), at the Headquarters, United States Marine Corps, Washington, D.C.

References

Cerkez, P. 1995. Knowledge Representation: An Explanation for "Buddy," Technical Report, CS5374, Department of Computer Science, Florida Institute of Technology.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Mass.: Addison-Wesley.

La Dage, J., and Van Germert, L. 1990. *Stability and Trim for the Ship's Officer.* 3d ed. Centerville Md.: Cornell Maritime.

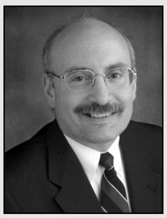
USMC. 1998. Aviation Logistic Support Ship (*T-AVB*) Logistic Planning Manual, Revision A. DCS Corporation, Lexington Park, Maryland.

USMC. 1983. Feasibility, Study of the Aviation Logistics Support Ship. Washington D.C.: United States Marine Corps.

Paul S. Cerkez is currently a software developer and technical adviser (in AI) for DCS Corporation, with 22 years of naval aviation avionics experience in the areas of general aviation electronics support and automated test equipment (ATE) and 10 years computer experience in the development of custom, knowledge-based software. In addition to being a U.S. Marine (master sergeant, retired), he holds a B.S., in electronics management, from Southern Illinois University at Carbondale, and an M.S., in computer information systems (AI), from the Florida Institute of Technology. Cerkez is the recognized subject matter expert for the Marine Corps *T-AVB* ships and performs training on *T-AVB* loading and load planning and TALPS software operations and provides expert advice to Marine Corps planners on *T-AVB* use. His e-mail addresses are pcerkez@acm.org and pcerkez@dcscorp.com.

Congratulations to the 2002 AAAI Fellows!

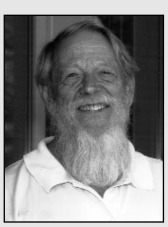
Each year a small number of fellows are recognized for their unusual distinction in the profession and for their sustained contributions to the field for a decade or more. An official dinner and ceremony will be held in their honor at AAAI-02 in Edmonton, Alberta, Canada.



Kevin D. Ashley
University of Pittsburgh
For significant contributions in computationally modeling case-based and analogical reasoning in law and practical ethics.

Michael Gelfond
Texas Tech University
For significant contributions to the development of stable model semantics, answer set semantics, and work in cognitive robotics, logic programming, and nonmonotonic reasoning.

Eric Horvitz
Microsoft Research
For significant contributions to principles and applications of probability and utility in computing, including advances in the control of problem solving, decision making under limited resources, human-computer interaction, and machine learning.



Henry E. Kyburg, Jr.
University of Rochester and University of West Florida / Institute for Human & Machine Cognition
For significant contributions to the study of probabilistic, statistical, and nonmonotonic inference.

Michael I. Jordan
University of California, Berkeley
For significant contributions to reasoning under uncertainty, machine learning, and human motor control.

Sarit Kraus
Bar-Ilan University and University of Maryland
For significant contributions to modeling of negotiation, collaboration, and non-monotonic reasoning, including theoretical advances and applications in various computational domains.

Stephen H. Muggleton
Imperial College of Science, Technology and Medicine
For significant contributions to the theory and practice of inductive logic programming, especially applied to the discovery of new biomolecular theories from observational data.

Katia P. Sycara
Carnegie Mellon University
For significant contributions to case-based reasoning, autonomous agents, and multiagent systems.