HUB SIAASHING: A Knowledge-Based System for Severe, Temporary Airline Schedule Reduction

Trish Dutton, American Airlines

HUB SIAASHING is a knowledge-based system that recommends contingency plans for American Airlines System Operations Control (SOC) during inclement weather or other airport disruptions where severe schedule reductions must be made. The system evaluates the current situation to determine flight cancellation, delay, and overfly candidates that will provide relief at the hub airport with minimal impact on systemwide operations. HUB SIAASHING provides an expedient method for reducing the schedule by locating and ranking all possible candidate plans with explanations for these suggestions.[1]

Problem Description

SOC is the organization chartered by senior management with managing the daily operational events that have the potential to affect the safety, efficiency, or profitability of the airline. One of the major tasks for SOC is to ensure that the diverse requirements of individual departments are resolved to the best overall benefit of the airline and its customers when operational compromises are indicated by weather, mechanical problems, airport issues, air traffic control problems, fuel shortages, or other situations. The *operations coordinator* performs a control function that focuses on the macro level to ensure that American Airlines can operate as much of the schedule as possible. These tasks include monitoring the system to identify potential problem areas, identifying operating options, and coordinating irregular operations by selecting and implementing contingency plans (SABRE 1990).

When the situation is such that schedule reduction is necessary (for example, bad weather might force a 25-percent reduction in all inbound flights into Chicago for the next 2 hours), the operations coordinators devise and implement contingency plans that enable the airline to operate a maximum number of flights, provide the relief necessary, and minimize any negative impact on the airline and its customers. Prior to the deployment of HUB SIAASHING, SOC operations coordinators used printouts of several flight operations system (FOS) transactions to manually locate candidates for cancellation, delay, or overfly. They used colored markers to highlight candidates and proceeded to weed through the possibilities, ranking them manually. This process was labor intensive, taking as long as 12 hours in some cases.

FOS is a transaction-processing system containing information necessary to operate the airline, such as flight schedule data, aircraft information, and crew assignments. Because FOS is a near–real-time system, the printouts tended to become more and more outdated with each passing minute; therefore, the operations coordinators interfaced with several other departments within the airline (figure 1) to determine the current, actual situation at hand. For example, the crew scheduling area was contacted to make certain that crew legality issues could be handled if a particular plan was implemented or simply to check that the data on the printouts were up to date. The coordinators analyzed the historical market situation to ensure that they would not be canceling flights that had already been canceled. Passenger information, connections, and the possibility of reaccommodating people on later flights were researched. Discussions with maintenance took place to guarantee that aircraft could make scheduled maintenance if these aircraft were rerouted.

HUB SIAASHING automates many of the routine tasks. The search for the appropriate candidate patterns is done programmatically, saving time and allowing the operations coordinators to handle the more difficult situations. HUB SIAASHING locates the candidates for schedule reduction and assimilates much of the information necessary to facilitate educated decision making, thus minimizing the negative impact in

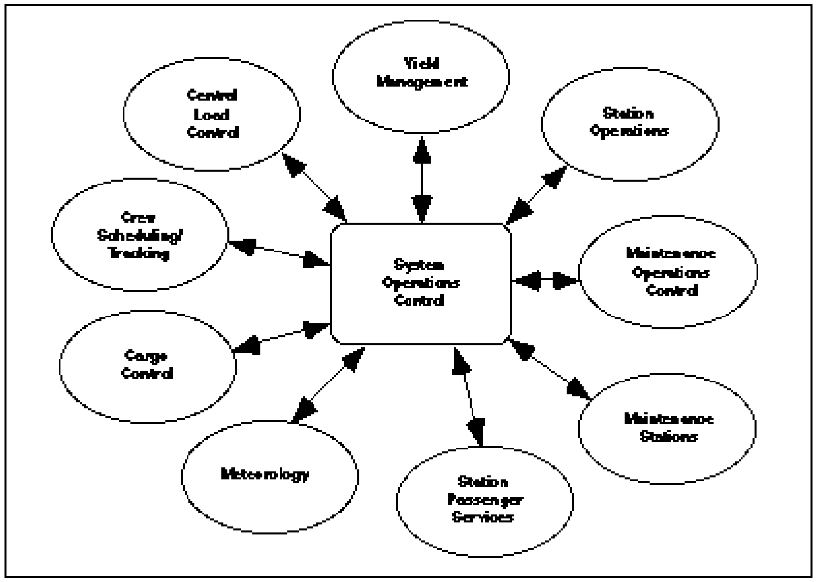

Figure 1. SOC Process Flow.

situations where it is impossible to operate all flights.

Application Objectives

The primary objective of the system is to provide a planning tool that assists the operations coordinators with decision making during irregular operations where temporary schedule reductions need to be implemented. Most importantly, HUB SIAASHING was to provide consistent planning for future events. Users are responsible for the safe and efficient operation of the airline and will not implement any plan that they do not fully understand. Therefore, the application was required to emulate the domain expertise to gain user acceptance.

This application had several additional objectives: The tool must vastly reduce the amount of time needed to implement contingency plans. It must provide explanations for the plans it suggests. The application must minimize system disruptions with regard to crew and aircraft routings. It must assist with minimizing passenger inconvenience. Any tool used in this dynamic, operational environment must be flexible and tailorable to the situation. Because most of the users are not computer literate, the application had to be intuitive to the user community. All applications deployed in SOC must use the existing hardware platform.

Importance of an AI Solution

An AI solution fit extremely well for this problem type. HUB SIAASHING uses knowledge-based technology integrated with a user interface and the corporation's computer facility. The problem is solved using rules to locate the patterns that the experts could find manually. Airlines are scheduled optimally and operated as best as possible, accounting for the weather, airport restrictions, equipment problems, and a number of other variables. A knowledge-based solution is the best technology to provide the flexibility to handle the dynamism, emulate the experts' search techniques, and provide the type of consistency necessary.

An AI solution was chosen because this technology would allow the knowledge engineers to focus on knowledge acquisition and use any number of expert system shells to provide the inferencing and knowledge representation that the domain required. The engineers were well versed in several shells on various hardware platforms, enabling them to spend more time with the experts and use rapid prototyping methodologies to quickly produce a working system. This application was successful because the use of an AI solution allowed it to meet all the project objectives in a short time frame and at little cost.

Previous Solution

An attempt was made to implement a similar system using traditional operations research techniques. The system did not suggest contingency plans; rather, it estimated the cost of canceling a particular flight segment. The users would still need to search for patterns that minimized the negative impact of reducing the schedule. The application was not well received because there was no explanation of results, it was inflexible, it used nonstandard hardware, and it did not reflect actual airline policies and decision making. The application developers did not interact with the users once requirements were defined and delivered a system that was perceived to be out of date.

Traditional approaches are inadequate for this problem because they do not allow for the flexibility required in this dynamic environment. Each event is different, and although airline policy drives decision making, the experience of senior operations coordinators is necessary to ensure that the choices selected for each situation are plans that can actually be implemented.

Traditional approaches require a longer time frame from initial concept to deployment than a rapid prototyping approach. The amount of interaction that knowledge engineers had with their users ensured that a more satisfactory solution had been found to their problem than the traditional method used in the failed attempt described earlier.

Application Description

HUB SIAASHING is meant to be used as a planning tool. It is typically used prior to actually executing schedule-reduction plans. The operations coordinators provide the application with pertinent input (for example, the hub station, time frame for the reduction, situational variables such as willingness to ferry airplanes.[2] They use the resulting suggestions and make final decisions regarding plan implementation. The users need a planning tool, not a reactionary tool. They are experts at reacting to problems and implementing appropriate corrective action but need assistance with timely planning for possible future events. Because they are tasked with making the minute-to-minute operational decisions, they tend to focus on the present, not on future problems. HUB SIAASHING assists them by providing this previously non-existent planning tool.

Environment: Hardware and Software

HUB SIAASHING is deployed on one MACINTOSH IIX with 16 megabytes of random-access memory and an accelerator card on an ETHERNET local area network (LAN) that contains over 350 individual workstations, 10 file servers, and 270 printers. The system interfaces with FOS for flight, crew, and passenger data and with MVS-TSO for market information.

SUPERCARD was used for development of the user interface. The knowledge-based system was entirely C based, written in THINK C and CLIPS (C-language integrated production system). CLIPS is a public-domain tool written by the National Aeronautics and Space Administration that was originally modeled on Inference Corporation's automated reasoning tool (ART-LISP). The shell provides a C implementation of the RETE algorithm and is portable across several hardware platforms (NASA 1991). Because the SOC hardware standard was MACINTOSH, the developers were limited to a couple of shells. Knowledge acquisition determined that the problem required a classic pattern-matching solution, easily managed with the RETE algorithm or other forward-chaining inferencing technique. Time precluded building the entire system in C; the developers were intimately familiar with CLIPS, so there was no learning curve, and using a shell simplified the development life cycle.

Architecture and Design

The application was designed with project objectives in mind, and the architecture had to ensure that it would be reliable and maintainable. It combines a knowledge base with traditional techniques such as parsing and report generation. The top-level design of the system is straightforward: a user interface and a knowledge-based system (figure 2).

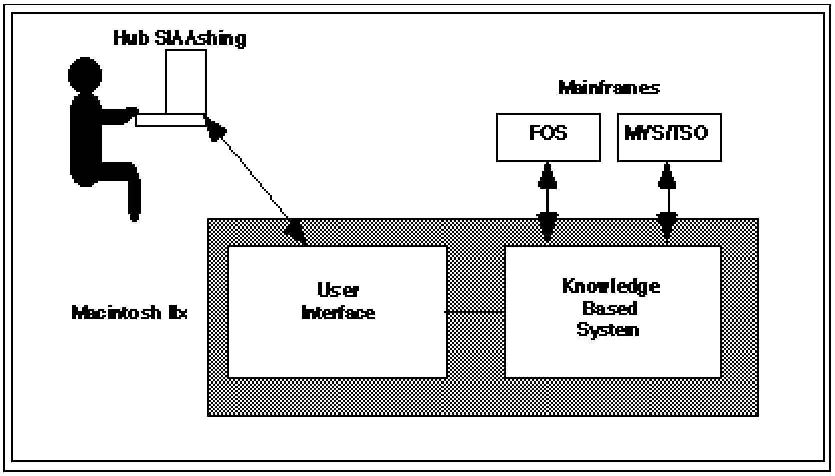

Figure 2. HUB SIAASHING *Top-Level Design.*

The user interface gathers the user's input parameters and launches the knowledge-based system. The knowledge-based system performs all other functions, including interacting with the corporation's computer facilities (specifically FOS and MVS-TSO), parsing and preprocessing data obtained from the various sources, locating the contingency plans through CLIPS rules, and formatting the results for display to the user. The knowledge-based system is divided into four major modules: the driver, the preprocessor, the knowledge base, and the formatter (figure 3). A library containing situational operands is loaded at run time, thus allowing for the flexibility so crucial to the success of the application.

The *driver* controls the processing. It makes calls to all other knowledge-based system modules, provides error-condition information, displays a status progression bar to the user, launches and controls the embedded CLIPS tool, and archives results to a file server.

The *preprocessor* performs the FOS and MVS-TSO downloading. It handles obtaining flight, crew, and passenger data from FOS, parses and abstracts the data, and builds CLIPS objects (templates). The preprocessor handles obtaining market files from TSO data sets. The functions were difficult to complete because the SOC environment uses Tri-Data Systems, Inc., NETWAY 3270 for TSO emulation, and the application programming interface for the MACINTOSH had not been completed by the vendor. The HUB SIAASHING developers were striking new ground with each line of code written.

The *knowledge base* contains rules to locate cancellation, overfly, and delay patterns in the current operating schedule and provides explana-

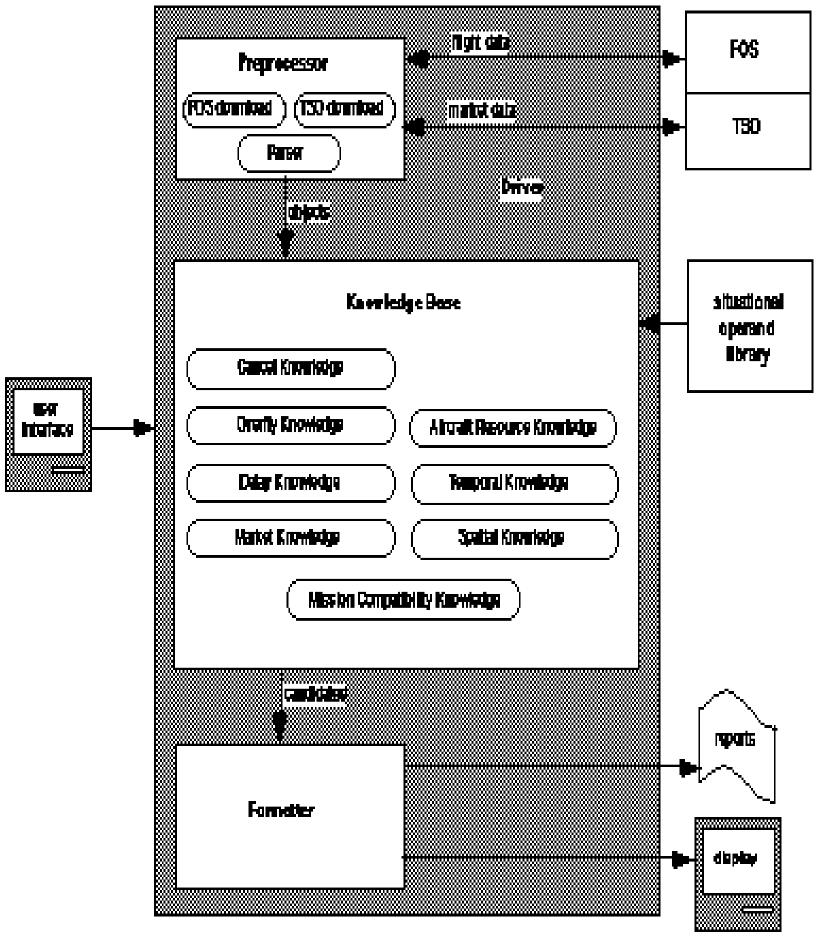

Figure 3. HUB SIAASHING *Knowledge-Based System Functional Diagram.*

tions for these suggestions. It uses airline policy, senior operations coordinator expertise, and situational operands to select and rank possible candidate plans. The rules are designed with ease of recovery from the irregular operation and minimal negative impact on systemwide operations built in. A *candidate plan* consists of flights that might be removed from the schedule, ensuring that the availability of aircraft and crews is balanced (figure 4) at all airports. For example, a basic scenario is to cancel one inbound and one outbound flight that are of the same equipment type, balance the schedule, and meet all situational constraints (figure 5).

The *formatter* prepares the results of the knowledge base for output

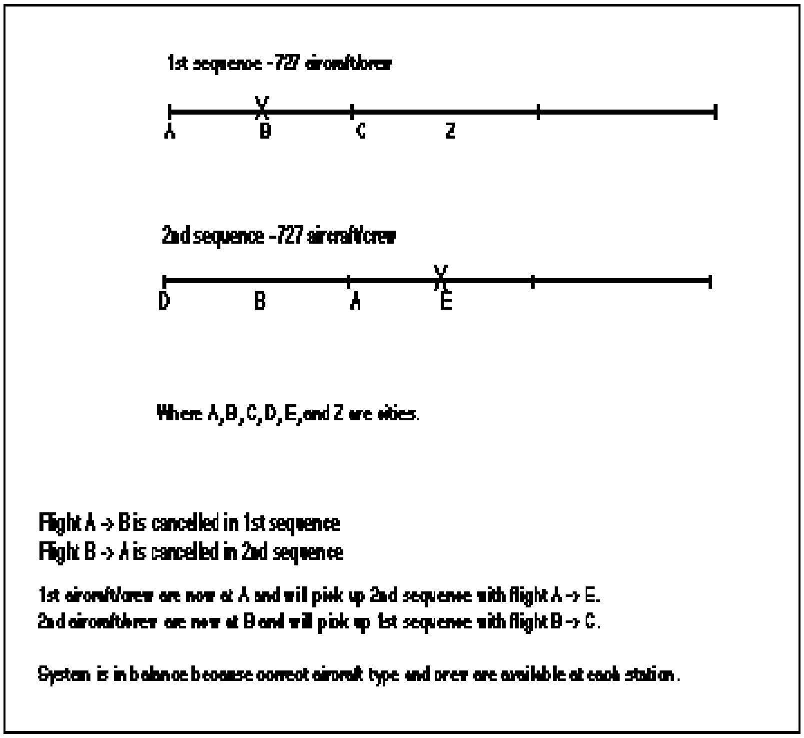

Figure 4. Example of System Balance.

to the user. Various user-tailored reports can be produced as an online display or as hard copy.

Innovations

HUB SIAASHING is an innovative application of an AI solution. Several reasons for making this statement are outlined in this subsection.

Traditional approaches used to solve similar problems at American Airlines had limited success and were not cost effective. In particular, a solution using operations research techniques was deployed with low user acceptance.

This knowledge-based system is the first for the SOC operations coordinators. It enhances the user's decision-making abilities and saves time during irregular operations by using pattern-matching and constraint-based reasoning to consistently locate schedule-reduction candidates and find a greater number of candidates than previously possible.

The system proves that a rapid-prototyping, phased approach using knowledge engineering techniques produces a quality system with high

```
IF
    sequence pattern
        <inbound flight + outbound flight>
    meets passenger constraints AND
    meets aircraft constraints AND
    meets crew constraints AND
    meets temporal constraints AND
    meets spatial constraints AND
    meets market constraints
THEN
    possible cancellation candidate of rank X
    with because-factors A B C ...
```

Figure 5. Typical Cancellation Rule.

user acceptance. Its development introduced a formal life-cycle methodology into SOC, where none had previously existed.

HUB SIAASHING demonstrates a novel integration between a MACINTOSH IIX workstation, the American Airlines FOS, and MVS-TSO. To my knowledge, interfacing a knowledge base on a MACINTOSH with the MVS-TSO environment had not been done before, anywhere in industry.

Application Justification

HUB SIAASHING is used by over 60 SOC personnel, including management, operations coordinators, crew scheduling personnel, and flight dispatchers. It is used on the average of one or two times each week, except in the winter when it is used almost daily.

Quantitative Benefits

The cost of a severe schedule-reduction day is measured in more than financial terms and differs with each event. It is widely known to be expensive, as much as $51,000 for each cancellation; therefore, any tool that enhances decision making is inherently beneficial. The cost of an unplanned cancellation (short notice) is three times that of a planned cancellation. HUB SIAASHING provides a tool for planning reduction in a more timely manner, thus reducing the cancellation costs.

The average length of time for implementation of schedule-reduction plans has decreased significantly. Locating plan patterns manually took an average of 8 hours; HUB SIAASHING accomplishes this task in less than 30 minutes. Each time the application is exercised to assist with an event, it saves the company over seven hours of labor costs alone. In the past, most irregular operations solutions were found while one or more operations coordinators was on overtime pay. This reduction in overtime is another labor expense savings.

The system locates candidates that would not otherwise be found because of time constraints during schedule-reduction sessions. Because the rules were designed with ease of recovery built in, it minimizes the number of cancellations required. As many as 30 percent more cancellations were required in the past to ensure the airline remained in balance.

HUB SIAASHING was developed in a short time for minimal cost. Total development cost was limited to hardware upgrades and labor. The software tools were already in house, so no additional expense was incurred.

Several application modules were slightly modified, at no cost to the user, to deploy a system for SOC crew coordinators to locate specific crew patterns. This type of code reusability is of great benefit at a time when most traditional software development costs continue to rise.

Applications deployed at American Airlines must provide return on investment within one year of implementation. HUB SIAASHING has fulfilled this obligation.

Qualitative Benefits

There are many qualitative benefits provided by HUB SIAASHING. This system allows SOC to plan at a closer temporal interval to the actual irregular operations without sacrificing the planning approach.

The system is intuitive to the user and looks and feels similar to other information sources he/she uses. The user knows exactly what the system provides and how to leverage the tool to facilitate better, well-informed decisions. By using situational operands, it is possible to manipulate the rules for optimal performance for the current situation.

Knowledge system developers forced a more formal implementation approach onto the application development group. Procedures are now in place at SOC that did not exist prior to this implementation. For example, no user sign-off procedures existed prior to the installation of the first prototype.

Application Development and Deployment

The application was developed using a rapid-prototyping, incremental-building methodology. The first version was deployed in four months, with two additional versions installed in 1991. Thus, the entire project was complete in approximately one year, each version taking three to four months to develop and deploy.

The development staff consisted of two knowledge engineers, one concentrating on the knowledge base and one focusing on communications, data preprocessing, and report generation. An application development programmer was on the staff to produce the user interface. A system analyst developed the user documentation and assisted with project management.

Knowledge Acquisition

Much of the development effort focused on knowledge acquisition and the determination of requirements. Prototypes were produced quickly and used in later knowledge-acquisition sessions for rule verification and as a tool to discover new knowledge that could be added to the knowledge base.

Engineers used interview techniques to learn of airline policy for schedule reduction. They employed bad weather simulation exercises with the expert operations coordinators to obtain actual operating rules. Crew scheduling personnel were interviewed to obtain crew situation knowledge. Yield management and operations analysis professionals were queried about market information. The knowledge engineers applied survey techniques to obtain operational commonsense information from the entire user community.

Validation and Verification

Complete validation of the application was done prior to deployment. Testing was performed throughout the development life cycle, with emphasis on close teamwork to ensure easy integration of all modules. Regression testing was performed for each version to ensure enhancements did not corrupt those functions already installed. Thorough system and user acceptance testing was performed on a test LAN that completely duplicated the production LAN environment, including use of actual data sources.

Training and User Acceptance

Because of the critical operational nature of SOC, training was accomplished one on one, with knowledge engineers training each of 66 users at the HUB SIAASHING workstation during users' regular work hours. Users had to remain on duty during training sessions. This training included the theory behind knowledge-based systems, description of each rule, concepts used in the application, and hands-on practice.

The airline policy rules were well known among the users. Rules based on the expertise of senior operations coordinators had, with few exceptions, complete user concurrence. User acceptance has been extremely good for applications deployed at SOC. The time taken to explain the application concepts to each individual has enabled them to trust the results. Now, when an event threatens a hub, the users reach for HUB SIAASHING to give them a head start at handling the situation.

Deployment

The system has been deployed formally since February 1991; although a prototype was installed in December 1990 for limited use. The current version was installed in December 1991. The application was deployed in two major steps. The first version was written almost entirely in CLIPS, containing seven basic sets of rules to locate candidates. It located simple cancellation patterns and was not situationally flexible. The final version contained 49 metarules to locate the candidate plans. The preprocessing and report-generation functions were rewritten in C rather than CLIPS. It selects more cancellation patterns as well as overfly and delay patterns. It is extremely flexible and can be tailored to the situation at hand.

The major problems encountered were system performance issues. The amount of data to be searched (American Airlines operates over 2300 flights each day) was enormous. Each flight has a flight template, at least two crew templates, market information, and passenger attributes associated with it. The RETE algorithm tends to use large amounts of memory, and it is time consuming to build the RETE network (pattern-join net). The number of possible plans, dependent on the number of flights at the hub during the time period in question, can be large. Even with the increase in functions, the second version of HUB SIAASHING was optimized to operate 45 percent faster than the first version.

Maintenance

HUB SIAASHING was designed to facilitate maintenance. The rules are generic in nature and will not require major modification. All pertinent operands are placed in tables that are input at run time to reduce the necessity of changes that require recompilation. These tables facilitate a vast range of application flexibility and have virtually no negative impact on execution speed. In the year since the first deployment, the only system maintenance has been two table updates. The users have control over many of the situational operand tables. Because the operation tends to change over time, it made good business sense to design this feature into the application.

It is anticipated that some maintenance might be required for the functions that interface with FOS and MVS-TSO, and the code was developed with this need in mind. All C code is in ANSI format and follows programming procedures outlined in the project documentation.

Conclusions

HUB SIAASHING is an innovative application that assists American Airlines with severe, temporary schedule reduction and minimizes the negative impact on our customers and systemwide operations. It is the first knowledge-based system for the operations coordinators, proves the approach is solid, and demonstrates a novel integration between the corporation's MVS-TSO facility and the MACINTOSH workstation. All project objectives were met: consistency and enhancement of decision making in this domain, flexibility, system performance, minimization of airline disruptions and passenger inconvenience, and development timeline and cost criteria.

Acknowledgments

Many thanks go to our experts Roger Beatty, John Schaefer, and Paul Prizzi; SOC managers Don Kneram, Kyle Phelps, Ed Gallagher, Doug Brown, and Paul Behm; Knowledge Systems engineers Judy Self and Bruce Kimball; Knowledge Systems managing director Lynden Tennison; and application developers Debbie Seals, Kevin Blackwell, and Virginia Dalfonso.

Notes

1. Airport disruptions are termed *irregular operations*. For example, an ice storm in Chicago forces the closure of a runway, and only 50 percent of the schedule might be operated. A *hub station* is an airport at which passengers, crews, and aircraft come together in complexes or banks to make connections (American Airlines currently has one international and six domestic hubs). A *cancellation* is a flight that cannot be operated. A *delay* is a flight that is operated later than its scheduled departure time. An *overfly* is a flight sequence that removes a stop at a hub. For example, the original sequence might be Las Vegas to Dallas–Fort Worth to Denver. An overfly would fly directly from Las Vegas to Denver, removing the stop at Dallas–Fort Worth.

2. A *ferry* moves an aircraft and crew(s) to a destination where needed. No passengers are aboard the flight.

References

NASA. 1991. CLIPS Reference Manual, Volume 1: CLIPS Version 5.0. Cape Canaveral, Fla.: National Aeronautics and Space Administration.

SABRE Computer Services SOC Operations Model, Working Paper, SABRE Computer Services, Dallas, Texas.