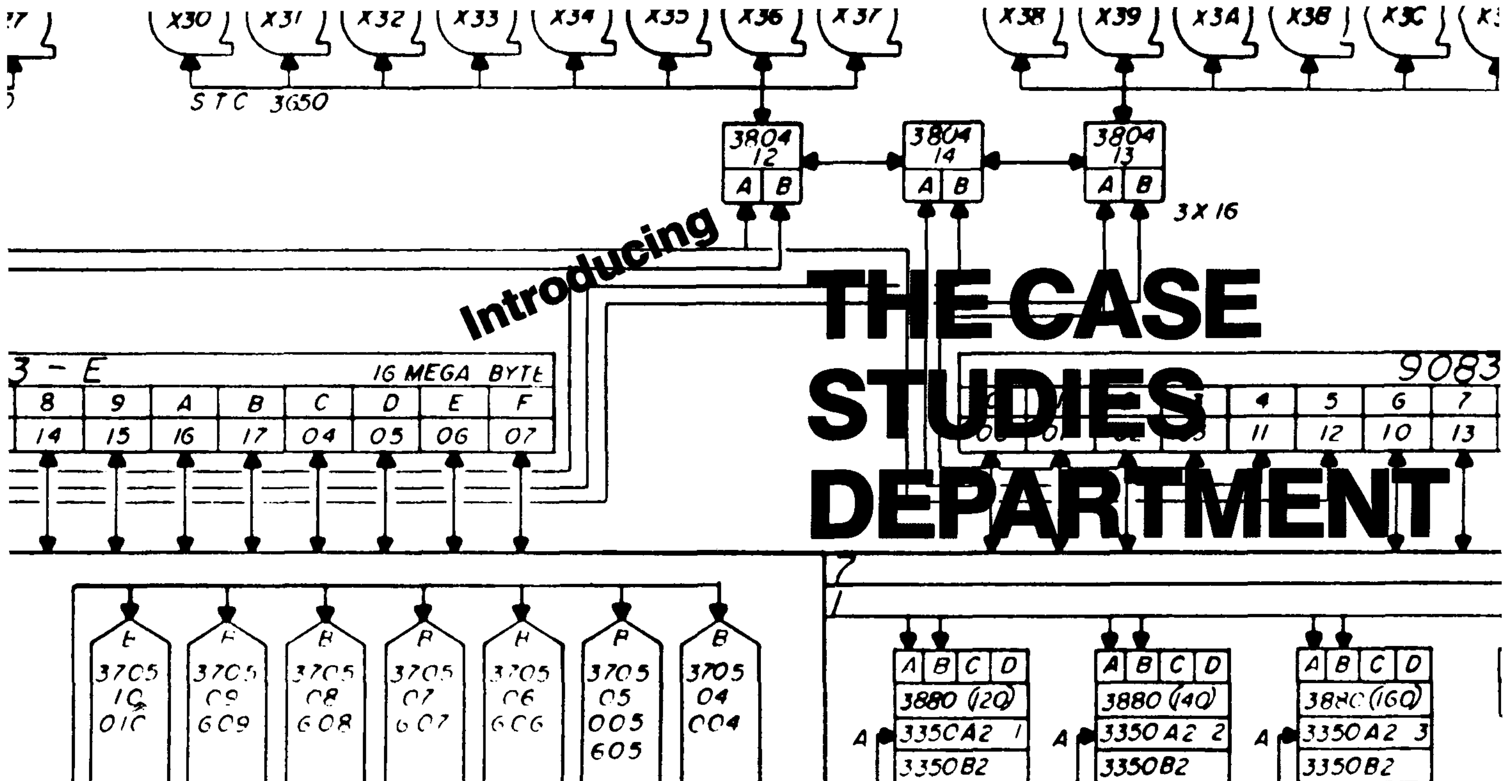


Introducing THE CASE STUDIES DEPARTMENT

Where can you find a solid, forthright overview of the computer systems and management behind airline reservations? NASA's space shuttle? Or any of the multitude of other large computer systems that support important projects or national activities? It's hard, sometimes impossible: partly because the people who worked on such systems often do not have the time to write about their experiences; and partly because many professional journalists who interview these people do not have the technical background to ferret out answers to the fundamental design questions addressed in these systems.

Through its Case Studies Department, the *Communications* is attempting to do something about this. Case studies are articles that report experience in constructing and using major computer systems. They aim to transfer knowledge about the capabilities and limitations of contemporary computer systems and their design processes. They report experiences both positive and negative. They provide a blend of broad information and important technical detail. The case study format provides access to information that might otherwise be inaccessible.

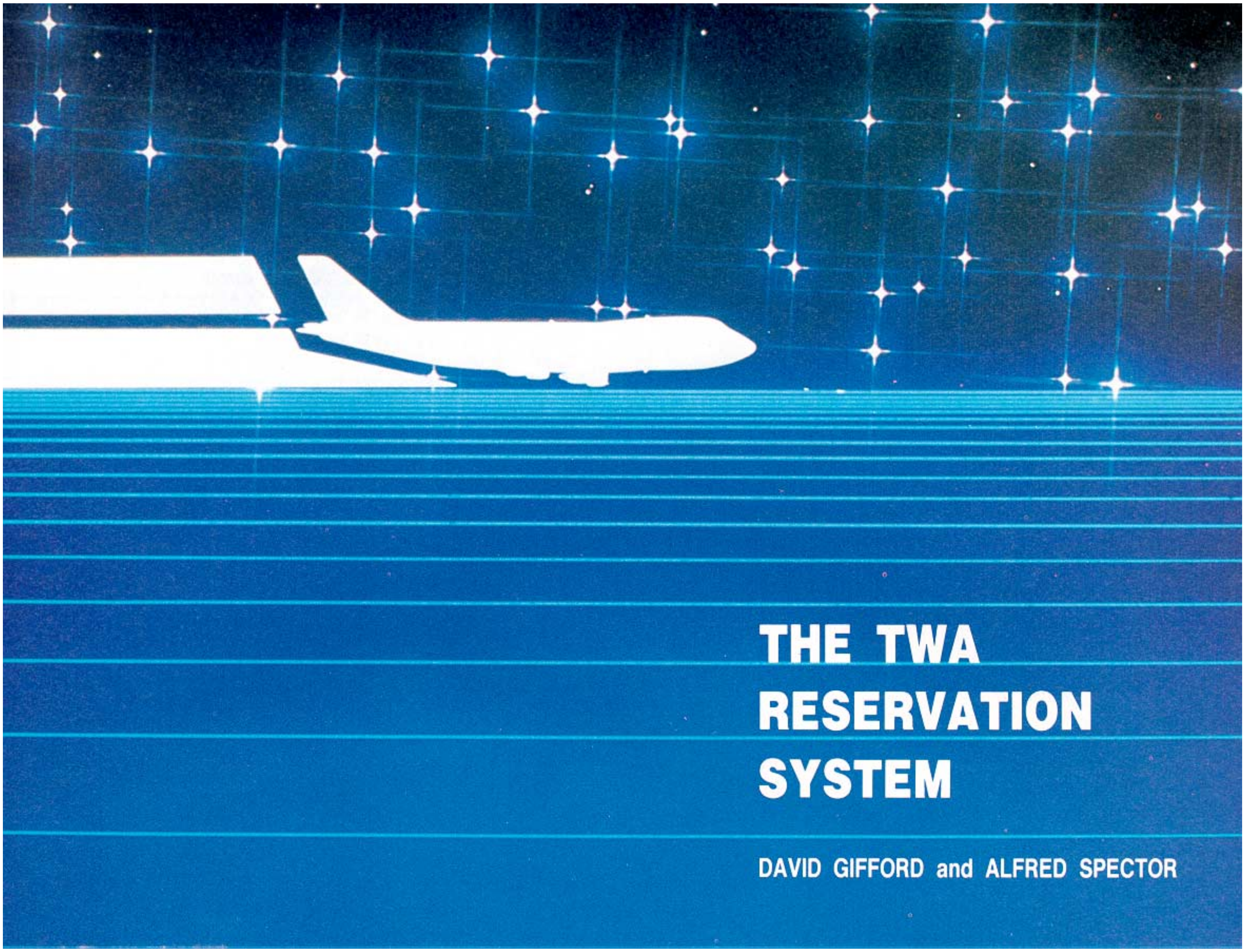
The principal type of case study will be an edited interview consisting of an introduction by the interviewer, an edited series of questions and answers with figures and citations as needed, and a conclusion by the interviewer. Each question and answer is likely to be the condensed form of a longer discussion at the interview. Material may be added or deleted during editing. The interviews for the first few case studies will be conducted by the Case Study Department editors; eventually, the editors will have enough experience to delegate this responsibility to other interviewers. In all cases, the interviewers will have experience and expertise with the type of system under discussion. A case will be considered seriously only if the interviewees can speak relatively freely about its merits and shortcomings. The final, published material will be agreed to by the editors and interviewees.

A secondary type of case study is a regular article submitted by an author in the normal way. Although we welcome and encourage them, we do not expect many submissions of this kind.

The concept of case study articles was proposed in 1982 by David Gifford during the planning for the new *Communications*. The editors were fascinated by the proposal. I encouraged Gifford to provide details. Gifford wanted his colleague, Alfred Spector, to participate in the efforts. By February 1983, the three of us had worked out a policy and procedure for Case Studies; that document set forth the responsibilities of the editors, interviewers, interviewees, and professional staff at Headquarters. I appointed Gifford and Spector as co-editors of the Case Studies Department in Spring 1983. They selected the TWA ticket reservation system as their first case and conducted the interview in April 1983; the transcriptions, editing, and approvals were completed in April 1984 and the result is published here. They conducted an interview about NASA's space shuttle in May 1983 and the results will be published within a few months. Other interviews are being planned now.

The Case Studies Department is still an experiment. After we have published a few cases, we will evaluate the effort required to do one study and the success we've had. The Case Studies Editors and the Editor-in-Chief appreciate your comments.

Peter J. Denning
Editor-in-Chief



THE TWA RESERVATION SYSTEM

DAVID GIFFORD and ALFRED SPECTOR

This is the first in a series of case studies on contemporary computer systems. Each article in this series will present an in-depth look at a single operational system and will consist of a set of interviews with the individuals responsible for its design and implementation. The interviewers will ask questions about system goals, design, implementation, performance, and operating experience; but they will also be looking for insights into the ideas that a system embodies, how it has evolved, and how it will meet the challenges of the future.

The case study presented here considers the TWA airline reservation system. It is drawn from a trip that editors David Gifford and Alfred Spector made to TWA in April 1983.

You will first be reading their interview with Carl Flood, director of system software for TWA. Flood discusses the goals and scope of TWA's system and the extent of the airline's dependence on it. He also explains the idea behind TWA's Coverage Group, an innovative approach to system updating and modification.

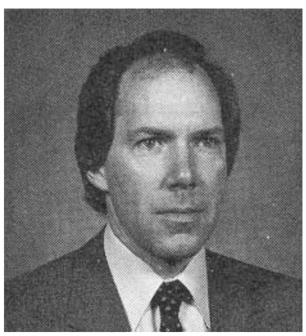
Ted Celentino then discusses the services provided by the system, the structure of the database, the user interface, means of connection with other airlines, system security, and building applications.

Neil Buckley provides a description of the system's hardware and gives a thorough account of the Airlines Control Program (ACP) system. He also covers such features as crash recovery, system performance, transaction processing and recovery, and multiprocessors.

Finally, Dave Lewis discusses network structure and considers the problems that continued expansion will present, along with some of the solutions that might evolve.

Table I presents an outline of the article. Interviewers David Gifford and Alfred Spector are identified by their initials; the interviewees are identified by their last names.

TABLE I. Outline of the Interview



CARL FLOOD—SYSTEM OVERVIEW

Carl Flood started with TWA in 1965, when he helped put together their first computer reservation system. He went on to become an application programming manager, manager of TWA's Operating System Group, and then manager of the Coverage Group. In 1978, Carl was promoted to director of system software.

GOALS AND SCOPE OF THE SYSTEM

AS. **Carl, let's talk about the goals and scope of the system first. What can you tell us about the present goals of TWA's reservation system?**
FLOOD. When we first implemented our current system in 1970, it was basically a name reservation and inventory system. Since then, we've added virtually every other function that's performed in providing service to the passenger. Ticketing and fare quotation were two of the initial major add-ons to the reservation system. They were added between 1972 and 1973. We also use the system for some services that aren't passenger related. For example, we use the system for cargo and for functions like plane weight and balance. The overall goal of the system is to provide an automated and cost-effective way of doing business.

DG. **Do you provide service on your system to anyone besides TWA employees?**
FLOOD. Yes, to travel agencies. About 50 percent of the bookings come from travel agencies as opposed to TWA reservation offices or airports. All of the major airlines provide automation services to the travel agencies.

AS. **How have you seen the requirements on your system grow over the past 13 years?**
FLOOD. No matter how much or how quickly we automate, we never seem to make much headway against the demands. It's unbelievable. When we first implemented the system, we had some obvious applications that needed to be added: fare quotation, ticketing, seat assignment, cargo. We thought that once we handled those we would be able to shift from a developmental phase into a maintenance phase. Well, that hasn't happened. We have a tremendous backlog of applications. We're also redeveloping existing applications from time to time. For example, we had "completely developed and implemented" our schedule change function. With time, though, it became clear that we needed to be able to update our schedules on a much more dynamic basis. We took the batch and on-line load schedule change application and translated it into a totally on-line function.

AS. **What is the approximate size of your system right now?**
FLOOD. There are about 11,000–12,000 communications terminals out in the field—worldwide. The system runs on a 9083 CPU, which is a high-performance 3083 uniprocessor. The size of the on-line database is 144 spindles of 3350s. It's fully duplicated, which means there are 72 modules duplicated on 72 others. We duplicate for the supportive availability as well as for performance.

AS. **How much data is in your system?**
FLOOD. We have between a million and a million and a half passenger records at any one time, each of which amounts to 1.2–1.5 Kbytes. That means our passenger record database is about 2 Gbytes. Passenger records are by far the largest part of the database, although there are many other types of records.

DG. **What sort of transaction load do you process on your system?**
BUCKLEY. A typical daily volume runs close to 7 million transactions. The message rate at a peak time is around 170 transactions per second, although we've gone up to in excess of 200.

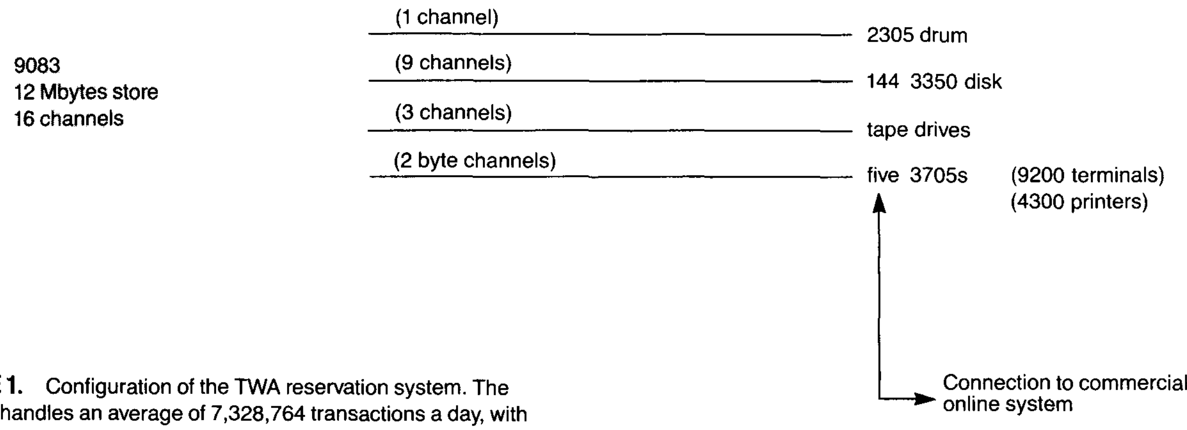


FIGURE 1. Configuration of the TWA reservation system. The system handles an average of 7,328,764 transactions a day, with a peak load of 196 transactions per second.

AS. **What are your response-time goals?**
FLOOD. We would like to be able to guarantee that 90 percent of the transactions are responded to in 3 seconds or less with an average response time of 1.5 seconds. You obviously can't afford to leave a reservation agent or an airport agent who has to deal with passengers waiting much longer than that. So, part of the justification is tied in to the efficiency of services to our passengers. On the other hand, our system is by its very nature a high-performance system. Independent of the needs of the user, we have to provide low response time in order for our system to handle the volume of processing that it does. In the early days, the systems that existed were taxed to provide for the processing that was required. We decided that in order to process several million messages a day we had to process every message in a couple of hundred milliseconds.

AS. **Has the response time significantly changed over the years?**
FLOOD. We are working to improve our methods of tracking response time, although the complexity of our network has been making that very difficult. It is not as straightforward as calculating internal processing time plus line time of communications to equal response time, because there are now concentrators on the network and intelligent terminals.

AS. **Has traffic increased in terms of the number of bookings you take per day and the number of transactions per day?**
FLOOD. Yes. For example, the number of transactions per boarding has increased dramatically. When our system first started operating, the only functions that were automated dealt directly with making a reservation and selling a seat. We knew that each boarding would create about 10 or 12 transactions. As we have added functions to the system that are indirectly as well as directly related to boardings, that ratio has gone up. It's now at something like 80 or 90 transactions for each boarding. The cost of automation or the number of entries into the system that have to be processed for each passenger boarding is quite high.

TWA's DEPENDENCE ON THE SYSTEM

DG. **I imagine that the system is really crucial for TWA. Can you estimate how much revenue is lost when it's down?**
FLOOD. I saw that question on the list that you sent me, and I've given it some thought. We have looked at that issue and we've done some calculations, although none very recently. There was a presentation that I developed a number of years ago that actually had some real numbers in it. We used a model that tried to estimate the extent to which outages upset our customers and translated that into potential lost ticket revenue. We had to make some assumptions about how many of the outages occurred in prime time, because, if an outage occurs at one in the morning, it's only going to affect a few passengers on the west coast and in Europe. Using the model we could compare years. For example, in 1972 we had 548 outages, and, according to the model, those outages could have meant something in excess of 2 million dollars in lost revenue. When the same calculations are performed for 1976, when we had 131 incidents, the outages could have meant 250 thousand dollars in lost revenue. Thus, if you believe all of the numbers and assumptions, the system improvements that resulted in the increased availability were worth 1.75 million dollars.

The problem with trying to nail down lost revenue is that the effect of an outage is very subjective. It depends on the day of the week, the time of day, and the length of the outage. There are certain airline functions that are not affected very much by a one-minute outage. For example, a one-minute outage does not affect the people in the reservation offices as much as it affects someone trying to board passengers on a plane when there's only five minutes left before flight time. In order to estimate lost revenue we had to make many assumptions—what to use for the cost of a ticket, how many people would actually switch to another airline, and so forth.

The extent of TWA's current dependence on the system has really rendered the question of lost revenue moot, though. It isn't so much a matter of how much a minute or two of downtime costs—the point is we can't operate the airline unless the system is up.

AS. **That last point is an interesting one. The airline is critically dependent.**
FLOOD. Absolutely.

AS. **Does a power failure usually interrupt service?**
FLOOD. It shouldn't. We've been working to make our backup power systems 100 percent reliable.

DG. **Some banks keep a duplicate paper flow of all the transactions that go through their system. Do you have any paper backup, or do you rely completely on your automated system?**
FLOOD. The only paper backups are people's tickets.

AS. **Do you store any data in another physical location?**
FLOOD. Yes. There's outside storage of data. We take dumps of the files on a daily basis and store them off-site for backup purposes.

DG. **What about contingency plans? Could you take your backup tapes to a remote location and operate your system from there if your primary facilities were inoperable?**
FLOOD. Emergency backup facilities are an issue that we have investigated and continue to look at. There are major expense/risk trade-offs that must be considered. In addition, there's the practical matter of assuring that the strategy will, in fact, allow for quick and effective recovery. It's unlikely that an entire facility would be absolutely and instantaneously destroyed. Starting with that assumption, the best approach to backup from a practical and economic standpoint is to reestablish an operating system, even if it only provides part of it's normal function. For example, we would not like to run with only half of our disks, but we could if we did not replicate information. If we had equipment failures, we would depend on IBM to locate some replacement hardware, and we could also use equipment that we normally have dedicated to our test and batch systems.

THE COVERAGE GROUP

AS. **Can you describe the functions of the Coverage Group?**
FLOOD. I'd describe it as an on-site programming support group. It's responsible for maintaining the availability of our on-line system. It works as a buffer between the programming staff and the on-line system. The programming staff doesn't have direct access for program changes or database updates—any change has to be submitted and processed by the Coverage Group. Database changes are requested on certain forms and may require a certain level of authority or approval. Programming changes may go through further checking before they are entered into the on-line system. If a program is updated and then creates a problem, the coverage people make the decision whether to fix it or to take it out.

I think one of the major advantages of having a group like this is that it maintains some objectivity relative to the modifications that are being made to the system. Coverage Group people are not faced with the responsibility for programming deadlines that the programming staff itself has. They can look at things a little more objectively. They can ask for further documentation or testing. They give us the kind of control function that we think is essential for maintaining high levels of availability.

TED CELENTINO—APPLICATIONS
Ted Celentino has been working with reservation systems since he joined American Airlines to work on the Semi-Automatic Business Environment Research (SABER) System in 1962. In 1969 he joined TWA and managed the installation of the airline's Programmed Airlines Reservation System (PARS). During this period, he was manager of the Coverage Group as well. After the system was installed, Ted managed the System Software Group for two years and was then promoted to director of commercial applications. Two years later, he moved on to his present position: director of PARS applications for the on-line reservation system.

SERVICES PROVIDED

DG. **Ted, how have applications changed in the 20 years you've been in this field?**
CELENTINO. As Carl was saying, the applications have changed because of all the refinements that have been introduced. We've added a lot of error checking and recovery software, too, which also complicates things.

AS. **What other sorts of applications do you run?**
CELENTINO. We have additional applications that

support our passenger traffic, such as baggage tracing and control. This application keeps track of lost bags and helps reunite passengers with their bags. We have a cargo-control application that's completely separate from passenger reservations, although it provides the very same service for the booking of freight. There's a certain amount of freight capacity, which is like seat inventory, that you can book on an aircraft. Freight reservations are reflected in our cargo air way bill control system database.

We have a credit-check-verification system, as well as numerous interconnections with hotels and other airlines. The U.S. domestic airlines have an organization called ARINC, which stands for Aeronautical Radio Incorporated, that provides communications services between airlines. We're also developing direct connections to other airlines.

We're also automating travel agencies by providing software specifically designed for them. We can help large companies that make extensive bookings for their employees, too, in much the same way.

AS. **What kind of applications would a travel agency need?**
CELENTINO. They might want to purchase foreign currency through our system, to give you one example. They would enter the request in our system, and we would send a message to a foreign-currency-exchange service.

DG. **Do you support electronic mail?**
CELENTINO. Yes, in the sense that we have a database that agents all over the world can access. Messages are put in the database from one agent to another. For example, a message might be that passengers are arriving on flight such and such who will need to be connected to some other flight. However, we don't provide general-purpose electronic mail.

AS. **Is there any way you can give travel agents important general information about flight changes and things?**
CELENTINO. There's a bulletin board called the direct reference system (DRS) that holds text that's organized into pages. The pages can be accessed by category, subject, day, or page number. We have cohost agreements with other airlines that allow them to use a certain number of DRS pages so they can communicate with travel agencies that use our system. We also provide travel agencies with a certain number of DRS pages for their own use.

DATABASE STRUCTURE

AS. **Let's move on to the structure of your database. Can you describe your reservation database?**
CELENTINO. The primary database we keep is a complete seating inventory for all of our flights. That means we have a schedule database that describes the flight schedule: the cities it flies to, the equipment that flies on it, and the capacities that are on those airplanes. For each flight for each date we keep track of the seats that are available for sale and the seats that are already sold.

The secondary database that interacts with the inventory consists of the reservations we create for customers. A reservation sells a seat out of the inventory, and so we have a reference from the reservation to flight inventory. The reservation also gives us additional information about the customer, such as phone contact, the number in the party, the spelling of the names, hotel and car reservations, and perhaps credit-

card numbers. We also keep information associated with the history of a reservation. For example, we keep track of the flights that have already been flown and changes that have been made to the reservation.

Flight and reservation information is kept one day in the past and 331 days into the future. Every night at midnight, we purge reservations that are more than 24 hours old. We keep requests for reservations that are more than 331 days into the future on file in the computer until they cycle into the date range; at that time a full reservation record is made and the inventory count is updated.

DG. **How is all this information indexed? For example, let's take the case of a particular flight on a particular day that has a certain number of seats available. How many different ways are there that I can get at that flight?**
CELENTINO. All the inventory data are accessed by date, departure time, and city. To look at inventory

data, an agent can do a schedule or an availability display. The difference between schedule and availability is that schedule just tells you the routing that you fly: what flights, what day, and what cities. Availability tells you what seats are available on those flights and days. To display a schedule, agents would input an "s", date, on point, off point. We also allow agents to compress the amount of data they are shown by putting in an approximate departure time rather than a specific one.

AS. **How are reservations indexed?**
CELENTINO. By the passenger's name, flight number, and departure date. All three are needed. If a passenger forgets his or her flight number, the agent can try to find a record of it by looking through all flights to the appropriate destination at that particular travel time. It's rare that a passenger doesn't know at least a couple of pieces of information that lead to his or her record.

AS. **Say that my associate Gifford, here, has made simultaneous reservations for New York to San Francisco and San Francisco to New York. Is there some way for you to catch that?**
CELENTINO. If he calls at different times and makes two separate bookings, it's hard to catch. I say no way, but there are some off-line programs we run. We run a duplicate-booking-analysis program that does comparisons by flight, date, and name. It prints out suspected duplicates if the name is exactly the same. It doesn't catch much because duplicate bookings aren't usually within the same airline.

USER INTERFACE

AS. **Let's consider the user interface. Exactly who are your users?**

CELENTINO. Mostly reservation agents, cargo agents, baggage agents, airport agents, and travel agents. There are other users, though, working on what you might call "aeronautic" functions, like weight and balance calculations or weather display.

AS. **How do you train agents?**
CELENTINO. The training they give out in the reservation office for a new reservation agent is probably on the order of 40 hours. We also have on-line computer agent instruction. Our training department creates lessons for all of our users—not only reservation agents, but cargo agents, airport agents, and baggage agents.

DG. **What sort of user interface do you provide to the system?**
CELENTINO. The basic system requires very precise input, which means that we need highly trained, highly skilled people. That was the way our system was set up: It was originally designed for an airline to use with its own people. For efficiency of use, the design required that they learn the precise formats of input messages. It didn't give them a great deal of flexibility. For example, to display a passenger record, you type asterisk for display, the flight number, the date, a dash, and the name. That's a very precise input format.

Since we started to bring travel agencies into the system in 1975, we've obviously had to make it more flexible. We can't give everybody the kind of training that our own agents get. Today it's by no means completely user friendly, but as we add things they tend to be more friendly. We have a few functions now where you can input information through a mask that shows what the input should look like. There is an intention to limit that because the most important goal of the system is to provide fast response to a large number of people. In order to do that, we have to keep the processing work load per entry to a minimum. However, we are working on some things that will allow a less sophisticated user to access us via viewdata- and videotext-like systems. Over a period of time, it's going to be more flexible.

It's no longer a matter of how much a minute or two of downtime costs—the point is we can't operate the airline unless the system is up.

AS. **Would you say that the user interface is a relatively large and complex part of the entire system?**
CELENTINO. No; again, because the basic design of the system assumes that agents are highly trained. If an agent does something wrong, it should be sufficient to tell him and he can start over again.

DG: **Have you done any studies or solicited suggestions on how to improve your user interface?**
CELENTINO. We've done both, actually. We don't have to solicit much because our users are always telling us what enhancements they would like.

CHANGING THE DATABASE

AS. **Airline deregulation has made the airline business environment more dynamic. How has this affected your system?**

CELENTINO. Our fares and our schedules are changing much more rapidly than they used to, which has created an urgent demand for application programming. We've got 18 people handling fares and pricing, where we used to have 4. We've gone from loading fares twice a year to where we have an on-line link with ATP to give us new fares that we can load every day.

DG. **What's ATP?**
CELENTINO. Air Tariff Publishers. It's another subsidiary of the airlines. It assembles the fares from all

the airlines and then supplies each with a consolidated database either by magnetic tape or over a communications line. It serves as a clearinghouse for fares.

AS. **Do you have to bring your system down to load new fares?**
CELENTINO. No, we use some off-line processes to support the on-line, real-time system, but we don't bring the system down. What we do is capture data from the on-line system, process this data off-line along with updates we have received, and then load the result of the off-line process back into the on-line system. The off-line process is a huge sort of millions of fare items.

CONNECTIONS WITH OTHER AIRLINES

AS. **Can you describe how TWA interfaces with other airlines through the ARINC network?**
CELENTINO. ARINC (for Aeronautical Radio Incorporated) is a message-switching center between airlines for flight-availability-status messages, flight-schedule-change messages, and booking messages. I'll start by giving you a concrete example. Let's say I have a customer that wants to book a flight on TWA and on the second segment of his trip he wants to book a flight on United Airlines. In my system I have information reflecting the flight schedules of United Airlines and the availability of their flights. If I book a United flight for my customer, I create a record in my system of the United flight and send United a booking message via ARINC advising them that I sold one of their seats. They use that incoming message to adjust their inventory.

AS. **Does United create a reservation record in response to your message?**
CELENTINO. Yes, but United's reservation record doesn't have the same information that I have in mine. Suppose that I have 10 segments of flight in mine, and one of those is United. The United system doesn't know anything about the other 9 flight segments that the customer is going to fly. When United creates a reservation record for their segment, they get the customer's name, the fact that the reservation came from TWA, and a contact for the customer in the event that United has to change the reservation.

AS. **How do you keep the inventory count consistent with all these copies flying around?**
CELENTINO. Hold on now. There's only one count of United's inventory and that's their count. The information that I have in my system about United reflects their flight schedule and the availability status for each flight. When United sells out a flight, they send out an availability-status message over ARINC saying that the flight is closed.

AS. **Is it really a binary flag that says "open" or "closed"?**
CELENTINO. Originally it was. There have always been agreements that said that if a flight were open you could sell a certain number of seats. For example, if it's open, sell four. If I don't want you to sell four, I'll send you a close. However, there are now agreements where I can sell seats on another airline's flight until they tell me that they only have a certain number of seats left. Beyond that point, they keep me advised of exactly how many seats they have left.

DG. **What happens if you sell another airline's seat but they don't have any seats left?**
CELENTINO. That's pretty rare but not unheard of. What happens is that both airlines try to find the best alternate booking available.

AS. **What can you tell me about the proposed interconnection between Eastern and the TWA system?**
CELENTINO. The interconnection already exists—it's called Direct Access. Right now we're only providing information display; eventually we're going to be sending other information. In fact, we're working on a direct access link for booking messages. The need for faster, more accurate information exchange is motivating us to interconnect systems.

DG. **Doesn't that raise a lot of policy issues? Do you have any problems, for example, with data flow across international borders?**
CELENTINO. Yes and no. The airlines as an industry are very cooperative in terms of addressing common problems. There's an international air transport association (IATA) that all of the international carriers belong to. It establishes standards for data exchange. The international carriers have established another organization, SITA (for Société Internationale de Télécommunications et Aèronautique), that actually provides international-communications service. At times, when we want to use SITA for certain purposes, we run into differences with governmental authorities. "You can't send this kind of data across that line, or if you want to install another telephone line, you have to get it from this authority"—that sort of thing.

PROTECTION

DG. **We'd like to know what you're doing to protect the system. What about authorization?**
CELENTINO. Every agent has a sign-in code and a duty code, and must log off when leaving the system. There's a whole hierarchy of authorization levels and duty-code levels. The duty code identifies the functions that an agent will be allowed to access. A reservation agent, for example, is able to create bookings, work queues, and display lists, but can't create tickets. There are 16 different duty codes.

AS. **How private are the data in your system?**
CELENTINO. Most of our data are not very private. The fact that Mr. Jones has a reservation on flight 659 is not known to the whole world, so if somebody calls in and says, "I'm Mr. Jones, and I have reservations on flight 659," pretty generally that's Mr. Jones. The information that we keep on the customer isn't very private anyway.

AS. **If I called TWA, said I was Mr. Smith, and asked about Mr. Jones' reservation, what would you tell me?**
CELENTINO. If you wanted to know when Mr. Jones was arriving, I'd probably tell you, if I could determine that you were trying to meet him. If you wanted to know his credit-card number, I definitely wouldn't tell you. Agents are taught what data not to give out as part of their training.

BUILDING APPLICATIONS

DG. **What would make it easier to build applications? It seems as if there's an immense amount of applications code in your system.**
CELENTINO. A very high-level applications-oriented language would really help. We have used a language called SABERTALK, which is based on PL/1. It's not very high level. If we had something that was very flexible, easy to learn, and capable of generating efficient code, that would be extremely helpful in terms of turning out applications. Right now we're still coding most of our applications in assembly language, because we need the efficiency we get that way.

DG. **Is there anything you'd do differently if you had to build the system again from scratch?**
CELENTINO. I don't think I would do anything differently. In fact, the system has evolved just as it should have. I would like to see somebody take a very broad perspective view of all the functions of our system and redesign it to be more flexible. We could use more table-driven applications and more general as opposed to specific routines. I don't think the basic processing architecture should change, because I don't think anything else has come along yet that can process our work load. However, the type of data that we have processed over time has changed considerably.

NEIL BUCKLEY—SYSTEM INTERNALS
Neil Buckley joined TWA in 1968 to work on reservation systems and became involved in ACP database generation for ACP applications in 1971. In 1973 he became a senior systems analyst working on ACP internals. After that assignment, Neil was project leader of the Coverage Group and then project manager for the installation of TWA's 3033 processors and 3340 disks. Neil is now responsible for the overall performance of TWA's reservation system.

HARDWARE CONFIGURATION

AS. **Neil, could you describe the hardware configuration of your system?**
BUCKLEY. Sure. It runs on a single 9083 CPU, which is a 3083 with a slight modification for ACP. We have nine channels of 3350 disks, 144 devices in all. We have one channel with a 2305 drum and three tape channels, although most of the time only one is in use. We need the three tape channels for a special procedure we do every night that involves capturing the entire system. We also have two byte channels, one for four 3705 communication controllers running NCP, the other for an EP 3705.

DG. **What's the difference between ACP and PARS?**
BUCKLEY. ACP is an IBM operating system specialized for high-volume real-time transaction processing. PARS is an application that runs on top of ACP and implements our reservation system.

DG. **How much memory is in your 9083?**
BUCKLEY. The 9083 has 16 megabytes with 12 megabytes in actual use. The other four are not used right now because our backup 3033 processor only has 12 megabytes. Four megabytes wouldn't really increase our throughput much. ACP is not driven by core requirements as much as a commercial system like MVS is.

DG. **How many megabytes of storage are used as disk buffers and how many for code?**

BUCKLEY. It's really split right down the middle. ACP currently has six megabytes. That includes some buffers. The control program itself is about 400 Kbytes. Then we have four pools of core. The 128-byte pool is used for communications. The 381-byte pool is used for small data records. The 1055-byte pool is used for large data records. Data records are either 381 or 1055 bytes long. We have another pool that's also 1055 that's used for entry control blocks. An entry control block is created for every input message and persists for the entire time the message is being processed. There's a large segment of core set aside for highly active applications programs. That's probably another quarter of a megabyte.

The second six megabytes are used to buffer disk records. This is called VFA, for Virtual File Access. One half is used for 381-byte records, the other for 1055-byte records.

AS. **Based on your performance measurements, do you ever reconfigure the layout of data on disk?**
BUCKLEY. Very rarely. The way ACP's database is designed, every disk gets equal access. That's hard to believe, but it's almost always true. The algorithm for dispensing space is one record on a cylinder on a module, the same record on the same cylinder on the next module, and so on to the end. Then it cycles back to the first module. By defining where the high-access

data are and putting them in the middle of a disk, we can produce a nice bell-shaped access distribution. That minimizes the queue length on the device. Because we have the same data on two devices, we can go to the one with the shortest queue when we're reading. The drawback with this approach is that, when we want to expand our system, things become rather complicated.

TABLE II. Glossary of Special Terminology

2305: A fixed-head drum that holds 11 million bytes of information

3083: A high-performance 370-series central processing unit

3340: A disk drive that holds 70 million bytes of information

3350: A disk drive that holds 317 million bytes of information

3705 (EP 3705): A network control processor

9083: A high-performance 370-series central processing unit

ACP: Airlines Control Program—a real-time operating system that was designed for airline reservation systems

ALC: Airlines Line Control—a synchronous, full duplex communications protocol used in the airlines industry; each character is six bits in length

ARINC: Aeronautical Radio, Incorporated—an organization run by U.S. domestic airlines that provides a message switching network

ATP: Air Tariff Publishers—an organization run cooperatively by all airlines as a clearinghouse for fare information

DASD: Direct-access-storage device; commonly known as a disk-storage unit

entry: A transaction

IMS: An IBM database system

IPL: Initial program load

message: A transaction

module: A disk drive

MVS: An operating system for large-scale IBM computer systems that supports on-line and batch processing

NCP: Network control program

PARS: Programmed Airlines Reservation System—applications software that runs on ACP to provide an airline reservation system

PNR: Passenger Name Record

SABER: Semi-Automated Business Environment Research—an early reservation system built for American Airlines by IBM; the forerunner and foundation of today's airline reservation systems.

SITA: Société Internationale de Télécommunications et Aèronautique—an organization created by international airlines to provide comunications service

spindle: A disk drive

VFA: Virtual File Access—a buffer set aside in main memory to avoid accesses to disk storage

AS. **How do you reconfigure your database when you move to larger disks?**
BUCKLEY. A good example is when we moved from 160 3340 disks to 144 3350 disks. The way we accomplished the move was to start a physical capture of the data on the 3340 disks and a restore of the same data on the 3350s. The capture was running 40 minutes ahead of the restore. When the capture completed, the on-line system was shut down, and the remainder of the captured data were restored onto the 3350s. While the on-line capture was running, some of the records that had already been captured were updated; remember that the system was still on-line. These records were written on an exception tape that was applied to the 3350s before the system was brought up on the new configuration. The whole process took about four hours.

DG. **Isn't there a way to avoid all that trouble by logically reorganizing your database?**
BUCKLEY. As a total database, no. Individual record types usually get reorganized in place by having some application software that says the new ones shall be this and the old ones shall be that. For example, at one time a PNR was a 381-byte record. We found that it always overflowed so there were always two records. That increased accesses. To reduce accesses, we made it a large record. We had 3.9 million old format PNRs and we did not convert them. What we said was, all new ones shall be large and we will handle the old format ones for as long as they are in our database. That time in our case was one year. So we migrated from the old format to the new format.

CRASH RECOVERY

AS. **On the subject of crash recovery, how do you utilize the 3033 backup when there's a failure?**
BUCKLEY. When we switch from the 9083 to the 3033, we swap peripherals and restart the system on the 3033. We have a peripheral switch that takes all of the peripherals on the 9083 and interchanges them with those on the 3033. The total procedure takes about 3.5 minutes. The peripheral switch takes less than a minute. Then we actually cycle the system up, initialize buffers and core, and bring applications programs into memory. A cold boot IPL takes about two to three minutes—considerably longer than it did before the system got larger. We could do it in less than a minute at one time.

AS. **Do you usually bring the 3033 into play when there's trouble?**
FLOOD. The tactic that we use for outages is to recover the system first. The primary objective is not to solve the problem but to keep the system up. Quite often the most expedient approach is to bulk-switch. If it's obviously a hardware problem with a CPU or something that appears related to the CPU, the operator will bulk-switch. However, we've noticed that CPUs are getting more reliable. and we don't bulk-switch as much as we used to. When we do bulk-switch to the 3033, we don't take any action to preserve the MVS or VM work load that it was processing.

AS. **How do you track outages of your system?**
BUCKLEY. The three major categories of outages that we track are hardware, software, and other. We don't

let someone sweep an outage into the unknown category, because that means that you're not going to track it and resolve it. The breakdown across the three categories of hardware, software, and other strangely enough comes out at the end of the year to be very close to being a one-third, one-third, one-third split.

DG. **How reliable is ACP?**
BUCKLEY. On the whole, very reliable. We have a lot more trouble with the applications that we run on the ACP operating system than we do with the system control program itself. Applications are constantly changing because we have a tremendous pressure from our various users to react to our competition. There are a lot of individual applications that go up, don't work, and are patched because they're needed immediately. Quite frankly, the loss of data in ACP is more associated with application problems than it is with the system control program itself. Even at that, I think one of the points to note is that the combined contribution of ACP and applications to outages is still less than one-tenth of one percent.

SYSTEM PERFORMANCE

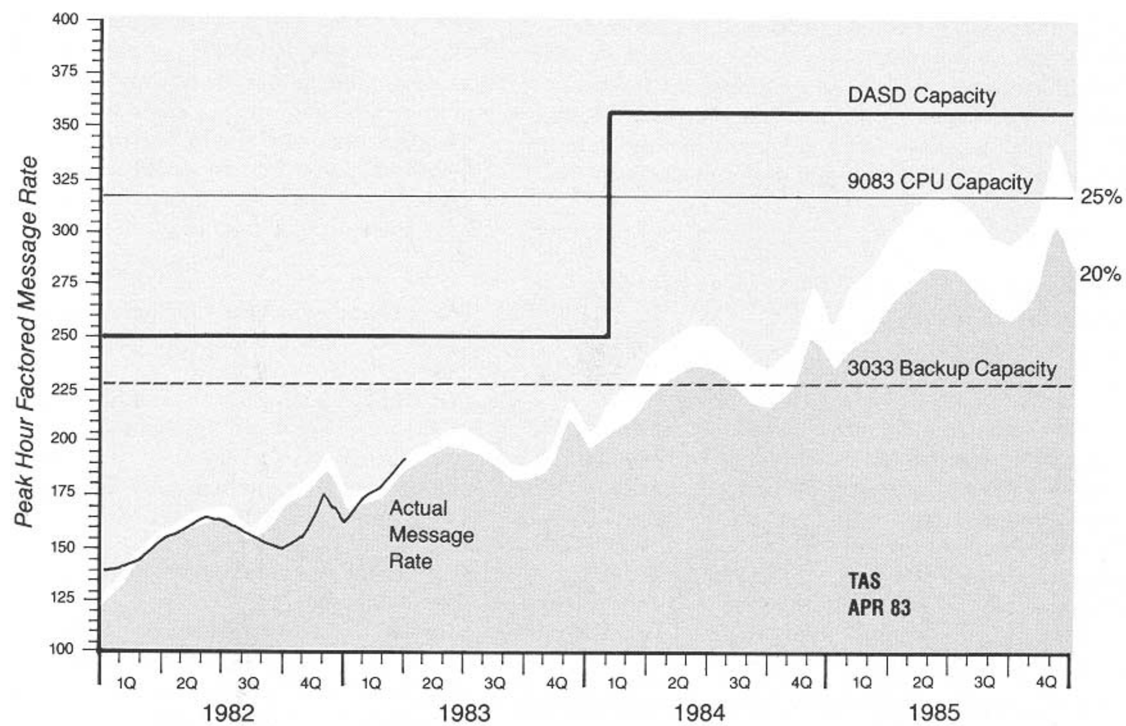
AS. **That's pretty impressive. Let's move on to system performance. To what extent is your main system utilized?**
BUCKLEY. We're running 50 percent utilization right now on the 9083; if we were still on the 3033, we'd be somewhat over 80 percent. That's why the 3033's days as a viable backup are numbered.[1] We really don't want to run above 90 percent CPU utilization. The 9083 has enough capacity to take us through 1984. In an ACP environment, we know precisely that the 3033 is a 5.2 MIPS machine. The hardware monitor that we currently utilize has been unable to get a MIPS reading on the 9083, but just by extrapolation it should be about 7.4.

DG. **How do you measure your work load?**
BUCKLEY. We plot our growth in terms of messages. We keep track of our current utilization per message, and we project where we are going to be. We alternate between having disk or CPU capacity as our limiting resource. Each input message requires about 20,000–25,000 instructions and 12.5 secondary storage accesses. Of those 12.5 accesses, about 2 go to the drum, 3 to VFA, and 7.5 to disk. With our current disk configuration and message load, that means that each disk performs on average 10 accesses per second.

DG. **The next logical question, then, is how do you measure disk capacity?**
BUCKLEY. That's the number of accesses our disks can support per second. Right now, we're using 70 percent of the storage capacity on a 3350, and we plan on getting 14 accesses per second for our 3350s. At 15 accesses per second, our disk I/O queues start getting a

[1] As of May 1984, TWA is using a second 9083 as a backup.

FIGURE 2
PARS System Capacity and Message Volume Projections



PARS system capacity and message volume predictions. The graph shows the actual transaction rate (solid line) and the predicted transaction rate (dotted line) for the *TWA* system. The *DASD* capacity line indicates the maximum transaction processing rate that can be sustained by *TWA's* disk configuration. Note the anticipated disk configuration change in the first quarter of 1984. The 9083 and 3033 capacity lines indicate the maximum transaction rates that can be sustained by each of those processors.

little longer. We're thinking of installing a 32 Kbyte buffer in each disk control unit to help with disk channel contention. The buffer would eliminate the need to synchronize the control unit to its channel and a disk. This would allow us to run our nine disk channels at higher utilization.

DG. **Do you know what the optimum size is for your VFA buffers? Have you run any studies?**
BUCKLEY. Yes, we have. In our environment, four megabytes reaches the point of diminishing return. Of course, to get the most return on VFA, we have to select what records are eligible to be held in it. For example, our city-pair record database is too large to be a good candidate for VFA. A good candidate for the cache would be output that is waiting to go to an agent: It's updated several times during a transaction, and we can afford to lose it. When the system crashes, it's not always possible to recover the contents of the cache. We usually don't write through the cache since that would cause extra disk accesses.

AS. **What's the multiprogramming level in the system?**
BUCKLEY. For 200 messages a second, we would expect no more than 40 entries in the system at any one time. Our average start-to-finish processing time for a request is 195 milliseconds. You have to keep moving them out just as fast as they're coming in.

AS. **How do you monitor system performance?**
BUCKLEY. IBM supplies a data-collection package with the system that summarizes system performance. It tells you how many messages you've done, how busy the CPU was, etc. We've also developed several data-collection packages of our own. One tracks entries by their first three characters and keeps a history on how every package is performing. If performance is off, we look at those data first and usually find that something that we've implemented recently has performance problems. We had one program that was doing a linear search through a very large piece of core almost 100 times a second. As the network grew, the average CPU time per message was up 0.3 milliseconds, which is a huge jump for us. We traced the problem to the linear search, changed the application program to use a binary search, and the problem went away.

We're going to stay with the uniprocessor for as long as it's feasible.

TRANSACTION PROCESSING AND RECOVERY IN ACP

DG. **How does transaction processing in ACP differ from that in a traditional database system?**
BUCKLEY. In ACP, a transaction consists of a single input and a single output. It may take several transactions to complete a process. Consider a reservation. The agent uses a set of transactions to sell a flight from the inventory and enter data about the passenger. When the agent is done, there is a final transaction called "end transaction" that builds the passenger name record from the data that the agent has input, associates the record with a seat, and indexes the record. A reservation involves anywhere from 5–15 transactions. The information that ties these transactions together is held in a control block called the Agent Assembly Area, or AAA.

DG. **Isn't the one-input-one-output philosophy somewhat restrictive?**
BUCKLEY. It's a trade-off. We'd like more flexibility, but this way we've got a lot of control over our system. There are some cases where an input creates two outputs, like when an output produces both a ticket and a boarding pass. Whenever we can combine transactions to increase efficiency, we do so. But we don't combine transactions if it means losing track of the load that's going to be placed on the system.

DG. **Does the information that's held across transaction boundaries go into the VFA cache?**
BUCKLEY. No. For the most part, that information resides on our drum. Some ancillary information that we can lose might be put in VFA. The AAA is an important record because it keeps track of inventory that has been decremented but that does not have a passenger reservation associated with it yet. The drum is stable enough that we don't write through it. If the drum had a catastrophic failure, then we would lose the contents of the AAAs, and thus we could lose inventory. If an alert agent realized that he had decremented inventory before such an event, he would release it when the system came back up. If an agent doesn't do that, we do have some utilities that look for inventory discrepancies.

DG. **That means that your database isn't completely consistent. How often do you check for inconsistencies?**
BUCKLEY. It depends on whether or not we're having problems. There was a time when we would check the system's inventory every night. If we noticed any kind of a trend change, there usually was a bug somewhere or we were having a lot of system crashes.

DG. **What else do you do to ensure the consistency of your database?**
BUCKLEY. We can reconstruct reservations. We log on tape the creation of passenger-name records and their corresponding reservations. Every night the log tapes from the on-line system are run through a batch program that maintains a master file of all of the active passenger-name records in our system. If we think that data on a certain flight are damaged, we can reload it from the master file that we keep off-line. We use the

log tapes when somebody calls up and says that they are absolutely sure they created a reservation on our system, despite the fact that we have no record of it. We then search the file for any data on the incident.

DG. **What you're saying, then, is that you're willing to tolerate a certain amount of inconsistency if you can get a large transaction volume and high availability in return.**
BUCKLEY. It's the only way we know to maintain the transaction rate and availability we need. There are very few alternatives with current technology because of the volume that we have to sustain. A 90 percent utilization rate would make it impossible for us to log all our transactions the way IMS does.

DG. **Why couldn't you do extensive logging like IMS?**
BUCKLEY. We just couldn't sustain our load. We're running 196 messages per second today. If you take the processing time and the accesses that are required per message and you then decided that every time you updated a piece of data you were going to log it somewhere, obviously you would more or less have to double the size of the system. Since ACP currently really only runs on a single CPU, we would never be able to accomplish that. At one time we used to log physical

records to tape as they changed. We were logging 20 reels of tape a day at 6250 bpi. The initial concept was to use logging to guard against hardware failures. When we got to a fully duplicated system, we discontinued logging physical records. We do enough logical logging to be able to recreate PNRs.

DG. **How does ACP ensure that you're not allocating disk records twice?**
BUCKLEY. ACP has a checkpoint mechanism that lets us preserve the integrity of file allocations, pool record allocations, etc. It checkpoints the current status of space allocation once every second. If it crashes, it will come back up with a file allocation status that is no more than one-second old. ACP may "lose" free storage, but it won't allocate records twice.

DG. **How do you control concurrent access to data in ACP?**
BUCKLEY. Locks are set at the physical record level. If an application program wishes to update a record, it "holds" the record, which sets a lock. After it is finished, it "unholds" it and the record is unlocked. An application program that is trying to hold a record can request to be queued if the record is already held. With so many records and only about 40 entries in the system at any one time, however, there's almost never a "hold" queue for a record.

DG. **Can you give an example of what would cause a record to be locked?**
BUCKLEY. Consider an agent who has done seven or eight transactions and now performs an "end transaction." The end transaction creates a new PNR record and files it away. The PNR has to be indexed in one of the six alpha groups on a given flight/date. Working from the individual's name, the end transaction application completely holds the corresponding index record and inserts an index item that points to the new PNR.

DG. **How do you avoid deadlock?**
BUCKLEY. All locks are acquired in the same order. When we see a queue building on a record, we can anticipate deadlock. If the activity on a record is low enough and the system doesn't crash, we output a message to the console and our Coverage Group can try to patch up the problem by aborting the offending transaction.

MULTIPROCESSORS

DG. **Have you ever considered switching to a multiprocessor ACP configuration?**
BUCKLEY. Yes, and the bottom line is, of course, that we may have to some day. Some of our competitors have already done so. Beyond 325 messages per second, our load can't be sustained on a uniprocessor.

But think of all we'd lose with a multiprocessor configuration. We wouldn't be able to efficiently utilize the VFA cache. Record locks would have to be stored in disk controllers with a special option, which makes the system more complex and prolongs outages. Many of our applications would have to be substantially modified. We would need more disk capacity to store things that are in global storage right now, which means more work. The result would be a larger configuration, with more interdependent components, that would be harder to control and operate.

DG. **Wouldn't you be getting additional system availability from a multiprocessor?**
BUCKLEY. Hypothetically we would, but in reality that rarely happens. If you lose a processor, it leaves locks in the disk controllers that are not cleaned up. In actual fact, there have been instances where the failure

of a control unit caused the entire system to go down. A lock for a specific record is only kept in one control unit, and thus the failure of a control unit is a fairly serious event that requires manual intervention.

DG. **Can you think of any architectural features that might help to support multiprocessing on ACP?**
BUCKLEY. I think that I/O to a disk should be done by a single processor; otherwise, it's difficult to add and delete processors. Of course, there can be different processors for different disks. Some of the features of the IBM XA architecture will probably facilitate this, like floating channels, multiple paths, and the EXDC, which will do a lot of the work.

ACP SERVICES AND STRUCTURE

AS. **Do you think you could summarize the services that ACP provides to application programs?**
BUCKLEY. Assembly language programmers call on ACP services by using macros. There are macros to get a block of core, to get a file address for data storage, to return such things, to defer processing, and so on. These macros tend to be very efficient. The I/O path length is about 600 instructions from beginning to end. That's probably one of the most expensive events that an application can do. There are other events that are an order of magnitude smaller. Applications run in problem state to protect ACP.

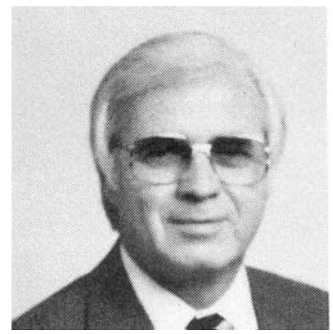
DG. **How does scheduling work in ACP?**
BUCKLEY. The scheduler has three lists. The ready list is filled with entries that are in the system and ready to run. The input list contains messages that are just arriving for processing. The ready list has priority over the input list. The third list is the deferred list. Entries on this list are not processed until the other two lists are empty.

DG. **Looking back on your experience, how would you sum up the good and bad points of ACP?**
BUCKLEY. The main good point is, of course, that we can process such a large volume of transactions. We haven't found anything that can do that as fast. By way of criticism, I can start by saying that ACP is a very costly environment for developing applications. It's expensive to write applications in assembler because it involves a lot of testing and debugging. A major improvement would be an efficient high-level language for applications development. SABERTALK isn't bad; it's just that we need more functions in the system software—support for program segments larger than 1055 bytes, for instance.

Another problem is that it's very hard to expand your database with ACP. It's complex, time-consuming, expensive, and difficult. As databases get bigger, things can only get worse. Something has to be developed that can respread databases and expand them in place, without bringing the system down.

ACP is also not well suited to multiple processors. You cannot just install it and have your current applications base run on it. It's also limited by direct-access storage devices. When you're running multiple processors, record hold contention gets pushed into the disk controllers and is less manageable as a result.

DG. **Do you think that the demands of your industry will cause you to switch to a new system in the next 10–15 years?**
BUCKLEY. No. I think that ACP will evolve. It would be very hard to justify throwing away the effort and manpower that has been devoted to it and just switching to something new. I don't foresee that happening.



DAVE LEWIS—COMMUNICATIONS
Dave Lewis started working for TWA in 1970 and has been involved with the reservation system on many levels. Dave is currently director of data communications.

NETWORK STRUCTURE

AS. **Dave, could you tell us something about the organization of your communications network?**
LEWIS. It's a star network with leased lines from our main site going out to our remote facilities. Terminals are connected through controllers that connect to multidrop lines that run at 2.4 Kbits per second. On a single 2.4 Kbit/second line, we put a maximum of either 10 different locations or 50 terminals. We concentrate four 2.4 Kbit per second lines onto a 9.6 Kbit per second line that runs back to the central facility in Kansas City and connects to the reservation system. We use remote line concentrators to reduce the number of ports needed at the central site and to reduce the number of lines. The network standard we use is called ALC, for Airlines Line Control. In addition to ALC, we also support BISYNC connections to our commercial system and the weather bureau, and an SLC connection to ARINC. The commercial system processes transactions having to do with crew scheduling, flight planning, and so forth.

AS. **What percent utilization would you make of a 2400 baud line during peak periods?**
LEWIS. We have a fairly good model of how much communications load to expect through a given CRT,

and thus we plan to run our lines close to 50 percent. If you plan for more than 50 percent, you end up with response-time problems. Response will be erratic, and there will be periods during the day when it will be too slow.

AS. **What kinds of network operations problems do you run into?**
LEWIS. Our biggest problem is getting AT&T to install new lines.

DG. **What sort of backup do you have if one of your communication lines fails?**

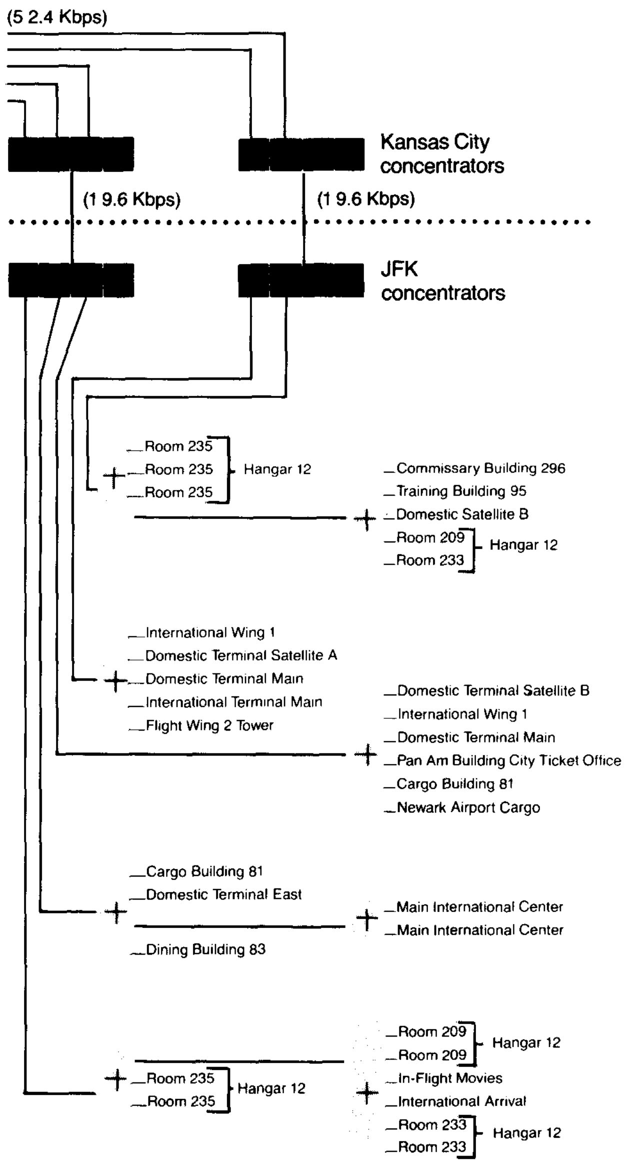


FIGURE 3. Block diagram of *TWA's* central site communications facilities. There are a total of 176 high speed lines that run from the 3705 communication controllers to *TWA* locations and travel agencies around the world.

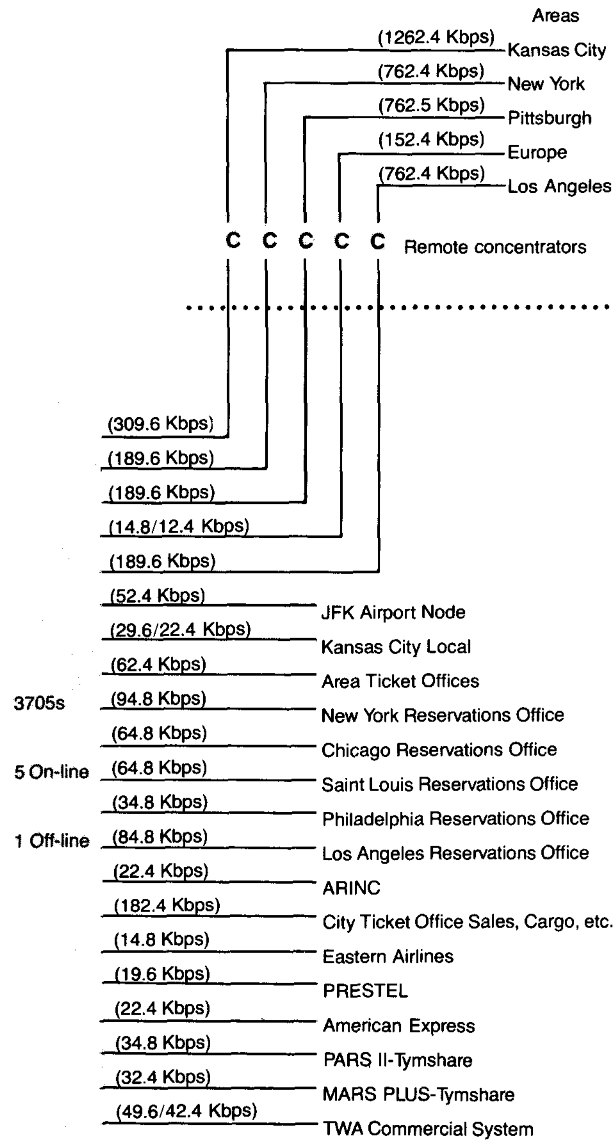


FIGURE 4. Block diagram of Kennedy Airport Network Node. The two 9.6 Kbit/sec data circuits from Kansas City are demultiplexed back into five 2.4 Kbit/sec data circuits. These 2.4 Kbit/sec circuits are multidropped to serve various terminal buildings, cargo agents, and nearby Newark airport.

LEWIS. At all of our major locations, we can dial up a 9.6 Kbit/second line for backup to replace a line that has problems.

AS. **Do you encrypt your data communications?**
LEWIS. We don't use encryption. We've got some security in the form of user codes, sign-on, and so forth.

AS. **Have you considered using any of the public data networks?**
LEWIS. We do use them, in fact. We connect to the TymShare network through gateways in Dallas and San Francisco. This allows us to make our system available

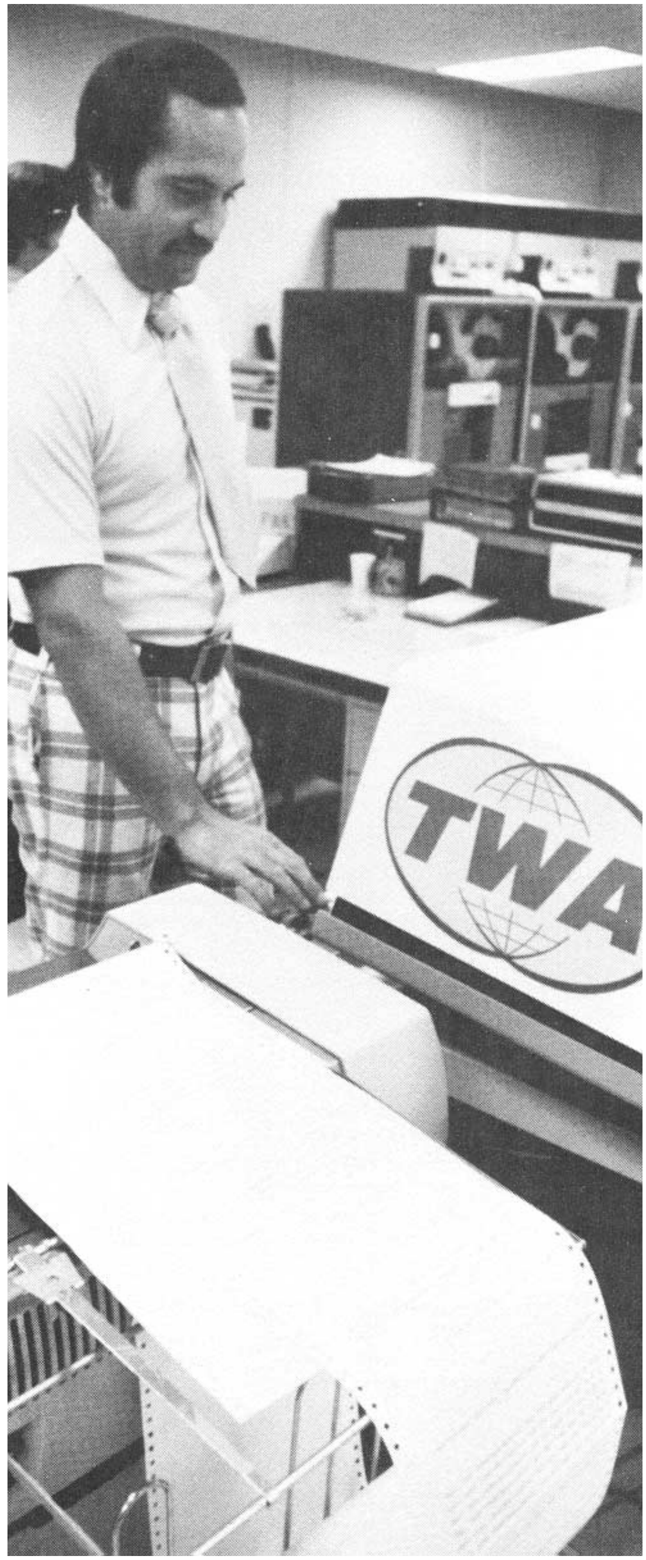
to travel agencies that don't have dedicated access. About a thousand agencies are involved this way.

DG. **Do you use packet switching?**
LEWIS. Packet switching and public networks are great if you're moving a small amount of data to scattered locations. In Europe, we use X25. The London Cargo System is an X25 packet switched network that all airlines interface to. Similarly, we use SITA to get to 100-odd locations throughout the world where we have less than 10 CRTs at a great distance from our central facility. It's much cheaper, even at SITA rates, to do this than to extend your own system that far. If you have a concentration of equipment and you're moving a lot of data, there's no way you can ever do that cheaper through a public network. You can always build your own for less money.

DG. **Your interairline protocol, ALC, was designed to use 6-bit characters for channel efficiency. Is it still important to heavily optimize messages in order to keep costs down?**
LEWIS. I don't think so. Lines to our reservation centers are limited by their data rate, but most of our lines are to travel agents. The number of travel agencies that I can handle on one line is limited by the number of drops that I can have on a single line and still have a reliable circuit. Thus, my guess is that you still have to have all the mileage to service all of your clients, so the cost of your network includes a large constant component that is necessary to reach all of them.

DG. **Have you considered using satellite channels?**
LEWIS. In fact, we have used satellite channels. We have an arrangement with RCA, the company that provides our transatlantic cables. If a cable fails, RCA provides a backup. If that backup fails, the third line is satellite. We've tried this, and we know how to do it. We're not crazy about it, but as long as you're in a C band you can have successful data transfer. As you probably know, most of the available capacity is in the K band, where the wave length is shorter than a rain drop—if it rains, you lose data. I don't believe the average corporation can afford that.

DG. **If you installed a satellite channel between your central facility and, say, Los Angeles, would the long propagation delay cause any problems?**
LEWIS. We have a concentrator and host system in Los Angeles, so I can get around the propagation delay. The concentrator polls the 760 users at the center and has their data sitting waiting for a poll from the reservation system. That means that the reservation system won't have to poll Los Angeles as often, and the propagation effects are considerably less than what they would be if the system had to poll each terminal.



DG. **What kinds of terminals do you connect to?**
LEWIS. We've put in quite a variety of different kinds over the 13 years we've been operating, and we still

support them all. The reservation system sees only a particular type of terminal. It looks like an IBM 2946/4505. The conversion from a given terminal type to our network standard virtual terminal is done in the front end communication processors.

AS. **We've heard lots of availability statistics about the central facility. How do they compare to the availability of your field terminals?**

LEWIS. Certainly it's not as good as at the central site. It never will be. There are too many additional components between the user and the central site, like concentrators, which are fairly reliable, terminal controllers, the terminal itself, the printers, and the phone line. The system is available to a typical terminal user 98 percent of the time or better. We have a guarantee to travel agencies that if they don't see 95 percent total they don't pay, and we normally collect our money. You have to consider the way they do their accounting too: They will have several pieces of equipment in their office; if one CRT is down, they count that as down time.

AS. **Wouldn't you be better off putting more redundancy into your system instead of going in for costly enhancements to the on-site system for making it more available?**
LEWIS. You've got to have the availability at the central facility because that's where all of your users can be affected. Any given line supports a very small percentage of our total users. Our average up time is over 99 percent, and that's for over 10,000 users.

AS. **What are the communication costs for your network?**
LEWIS. By the end of 1983, our bill—just the data communications bill to AT&T—will reach a million dollars a month. That's a substantial amount of money. Voice communications cost about twice that much. So we are talking about between 35–40 million dollars on the communications side versus perhaps 10–15 million for the entire central site. Central site expenditures include the programming efforts for the applications programming, for ACP modifications, for new software, and so forth. Roughly half of our money is spent for data communications and half for central site work.

FUTURE OF THE NETWORK

DG. **It seems appropriate to conclude by talking about what you see in the future for the TWA network. Are you considering adding direct ticket purchasing or boarding pass generation?**
FLOOD. Well, it's pretty complex when you get into human nature. I'll give you an example. We had two self-service ticket machines at Laguardia in the summer months, and we promoted them. There was somebody there to direct people to them for a while. They had an incentive—a free drink or something—and it still wasn't used very much. After they stopped manning the machine, it fell into more disuse. A lot of people who were either one-time travelers or who hadn't heard about the machine didn't know what it was, so consequently they didn't use it. I think the reason that automated bank tellers caught on is that they're the only alternative if you want money during off hours. If you go to the airport and you're willing to wait in line, there's always somebody there to issue you a ticket. The incentive is lacking.

DG. **What do you think the future holds for your communications network?**
LEWIS. The network has been changing and it's going to continue to change. We're going to see a lot of personal computers on the networks, which should mean less demand on the system, since users will be able to keep data in their PCs and work on them there.

DG. **Carl, David, Neil, Ted, thanks for your time, and thanks for working so hard to come up with all these good answers.**
FLOOD. Our pleasure.

Acknowledgments. The editors would like to thank Carl Flood, Neil Buckley, Ted Celentino, and Dave Lewis of TWA for their time and assistance. We would also like to thank Betty Brannick for transcribing the interviews.

BIBLIOGRAPHY
IBM Corporation. ACP system: Concepts and facilities. Order No. GH20-1473-1. White Plains. New York, Mar. 1975. A detailed examination of the facilities and implementation of ACP.
IBM Corporation. ACP system: Message routing concepts. Order No. GH20-1693-0. White Plains, New York, May 1975. Describes the way that messages can be routed between ACP systems and how terminals can be shared by ACP and another operating system.
Knight, J.R. A case study: Airlines reservations systems. *Proc. IEEE 60*, 11 (Nov. 1972), 1423–1431. Reviews the development of early systems and discusses the early evolution of reservation systems in the 1960s.
McAvoy, R.A. Airline reservation system. *Datamation 3*, 8 (Oct.–Nov. 1957), 9–14. A look at the requirements and design of an early on-line system for Eastern Airlines. The system was designed to service 125 terminals for 22 hours a day.
Perry, M.N., and Plugge, W.R. American Airlines SABER electronic reservation system. In *AFIPS Conference Proceedings*. Western Joint Computer Conference, vol. 19. AFIPS Press, Arlington, Va., May 1961, pp. 593–601.
Siwiec, J.E. A high-performance DB/DC system. *IBM Syst. J.* 2 (1977), 169–195. Describes how ACP evolved from early reservation systems and provides a nice description of the workings of ACP.

CR Categories and Subject Descriptors: C.2 [**Computer Systems Organization**]: Computer-Communications Networks; C.4 [Computer Systems Organization]: Performance of Systems; D.4 [**Software**]: Operating Systems; H.2 [**Information Systems**]: Database Management; J. 0 [**Computer Applications**]: General; K.6 [**Computing Milieux**]: Management of Computing and Information Systems
General Terms: Design, Human Factors, Management, Performance
Additional Key Words and Phrases: Airline reservation system, PARS, ACP

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

Some photos and slides courtesy TWA.