Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

PARALLEL SIMULATION FOR AVIATION APPLICATIONS

Frederick Wieland

Center for Advanced Aviation Systems Development
The MITRE Corporation, MS W282
1820 Dolley Madison Blvd.
McLean VA 22102-3481, U. S. A.

ABSTRACT

The Detailed Policy Assessment Tool (DPAT) is a widely used simulation of air traffic control that incorporates advanced technology for user-friendly operation. DPAT computes congestion-related air traffic delays, throughputs, traffic densities, and arrival/departure schedules while incorporating ground delay and ground stop programs, in-trail restrictions, historical, current, or future traffic demand, a fixed or free-flight route structure, and other relevant parameters. DPAT's advanced capabilities include parallel discrete-event simulation technology and easy access through Web-based simulation.

1 INTRODUCTION

The purpose of this paper is to introduce the Detailed Policy Assessment Tool (DPAT) as an important system for aviation analysis, concentrating on its software architecture, its uses in aviation decision analysis, and its implementation as a parallel discrete-event simulation. DPAT incorporates two important technologies absent from other air traffic simulations, namely, parallel simulation technology and a Web-based interface. Parallel simulation (using multiple CPUs simultaneously) allows DPAT to simulate an entire day's worth of air traffic (roughly, 40,000 flights or 500,000 events) in less than one minute. Its use of a Web-based interface allows aviation analysts, with little knowledge of the subtleties of the computer science behind parallel processing, to access the model remotely, using their own desktop computer from any internet-connected location on the planet.

The MITRE Corporation's Center for Advanced Aviation Systems Development (CAASD), a federally-funded research and development center associated with the Federal Aviation Administration (FAA), developed DPAT over the past three years with its internal research funds. Particularly noteworthy is DPAT's ongoing use as an aviation analysis tool; it is not "vaporware" or merely a paper-based system. DPAT is a world-class aviation simulation, and its novel use of advanced technologies is likely to be emulated in future FAA and non-FAA simulation systems.

2 DPAT ARCHITECTURE

DPAT models the National Airspace System (NAS) as a sequence of capacitated resources that an airplane uses to travel from its origin to its destination airport. The resources are runways, terminal-area sectors, enroute sectors, and route waypoints (called "fixes") at which flow restrictions can be imposed. Airplanes may use any set of available resources. DPAT does not assume a particular route structure or the presence of restrictions; as such, it is useful for analysis of free flight routing, wind-optimal routing, or a restricted route structure (or any combination of these).

The basic inputs to DPAT are air traffic demand, routes for origin-destination (O-D) pairs, resources (airports, terminal-area sectors, enroute sectors, and fixes), resource capacities, and restrictions (both ground delay/stops and miles-in-trail). All of these inputs can dynamically change with time, for example, airport capacities may change as a function of the evolving weather. Although the weather is not explicitly modeled, its effect on resource capacities can be directly simulated to capture weather-induced delays.

2.1 External Architecture

The basic external architecture of DPAT is shown in Figure 1. DPAT obtains much of its data from other models, in particular models of route structure, resource capacity, and traffic demand. The top half of the diagram illustrates the generation of air traffic demand, which consists of commercial air traffic (flights that have a published schedule) and General Aviation (GA) flights, which includes corporate jets and pleasure aircraft that lack any published schedule. The commercial air traffic is obtained from on-line data sources, while the GA flights

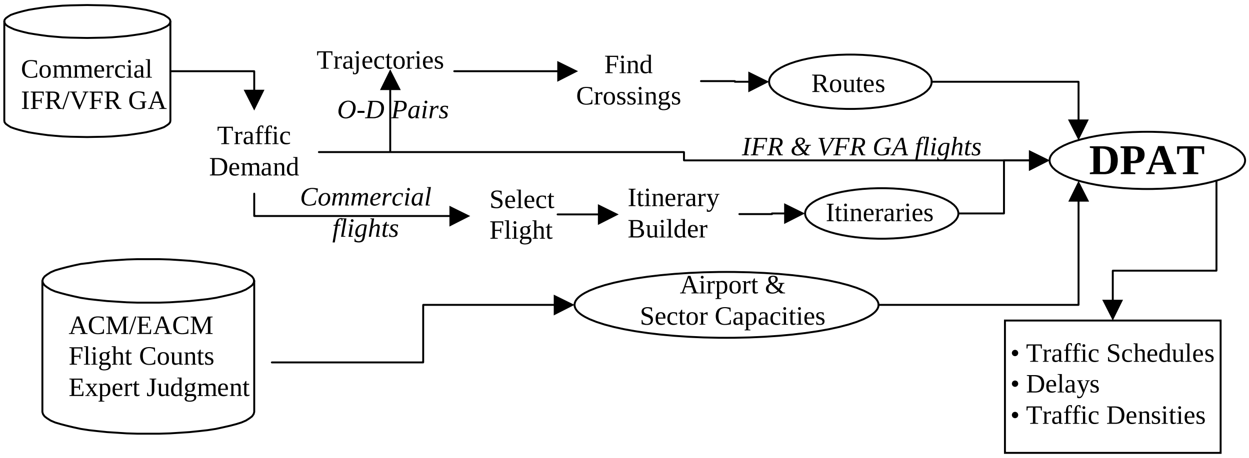

Figure 1: External Architecture of DPAT

are generated via a nonhomogeneous Poisson process whose input (frequency of GA traffic by airport by hour of the day) is derived from analyzing 220 days of actual air traffic.

Origin-Destination (O-D) pairs are extracted from the scheduled air traffic, and routes are generated for those O-D pairs by modeling the climb, cruise, and descent portion of the flight given the aircraft type, wind speed and direction, and routing structure. The routing structure can be an FAA-imposed preferred route, a "direct" route (great circle), a weather-avoiding route, or a wind-optimized route. The latter two require a database of weather/wind conditions at different altitudes. These different routes can all be generated simultaneously, and the selection of a route is done by DPAT prior to the departure of a flight.

GA flights are directly used by DPAT, while the commercial flights are bundled into itineraries prior to the simulation. An itinerary is a sequence of flights by a specific airframe, and they help DPAT model the propagation of a local disruption to other regions of the NAS. As DPAT is often used for NAS-wide analysis, itinerary generation is necessary.

Traffic demand can be derived from historical (electronic) data, from the current situation via a live FAA-provided data feed, or from an estimate of future air traffic. Future traffic demand is generated from estimated growth rates using the Future Demand Generator (FDG). The FDG applies worldwide, regional, country, or airport-specfic growth rates to a user-specified base level of air traffic, and provides DPAT with estimated future itineraries. These three air traffic data sources allow an analyst to simulate a past historic day, the current live situation, or an estimated future day.

The lower half of the diagram illustrates how resource capacities are derived. Airport capacities depend upon the number and configuration of runways, fleet mix, separation standards, and weather conditions. An FAA-sanctioned model called the Airfield Capacity Model (ACM) can be used to generate airport capacities for any airport in the NAS. Occasionally, airline analysts develop their own estimates of capacities, which can be derived from tower counts, expert judgement, or previous experience.

Sector capacities are derived from FAA-provided Monitor Alert Threshold (MAT) values. MAT numbers represent the point at which the sector controllers are reluctant to allow more traffic to enter their sector. The default MAT numbers can be changed by an analyst for specific studies, and, like all other inputs to DPAT, can be dynamically changed during the simulation to model disruptions, weather-induced or otherwise.

There are many other input sources not shown on the diagram, such as ground delay/stop program parameters, miles-in-trail restriction at fixes, provisions for adding extra demand as a percentage of commercial traffic, and more. With this input DPAT is able to compute the estimated schedule for each flight, which accounts for congestion-induced delay as well as random delay such as late passenger or baggage loading; equipment troubles; late clearance delivery; and so on. The resulting computed schedules are then used to derive flight delays, both passenger-experienced and congestion-induced, as well as traffic densities in enroute airspace, arrival and departure flows, and the impact of local disruptions on remote regions of the NAS.

2.2 Internal Architecture

Internally, DPAT is structured as a parallel discrete-event simulation. Such simulations are decomposed into Logical Processes (LPs) scattered among the different processors. The LPs communicate by passing time-stamped messages triggering an event at a remote LP. For DPAT, there are three types of LPs: an airport, an enroute (or terminal) sector, and a fix. Airplane information, consisting of all

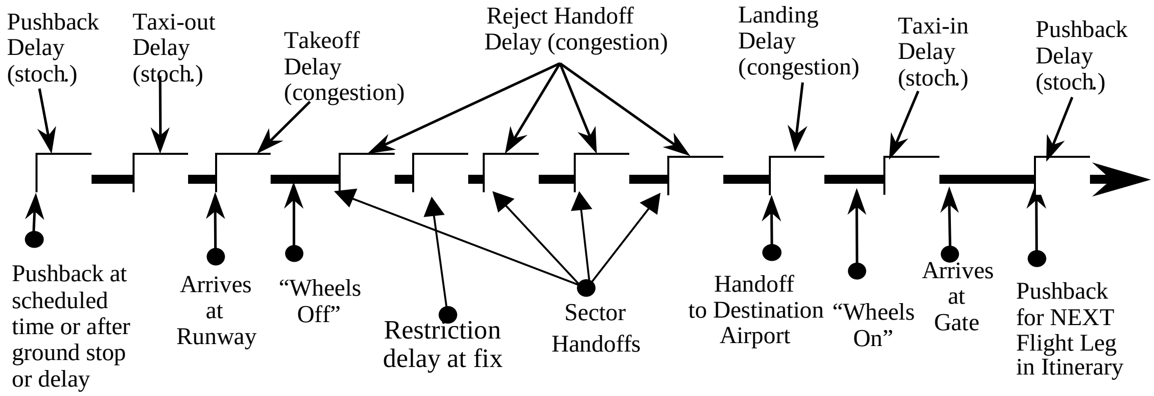

Figure 2: Delays Computed by DPAT During the Evolution of a Flight

the data required to process a flight from its origin to destination, is passed as a message among the LPs used by the flight.

The airport LP implements the takeoff and landing events, which handle all taxiing and runway queueing for both departures and arrivals. The sector LP implements a set of events corresponding to the NAS handoff protocol. These events include requesting a handoff, accepting a handoff, rejecting a handoff, and the later re-acceptance of a previously rejected handoff. The fix LP implements one event which computes the restriction-imposed delay at the fix.

DPAT computes a variety of different delays, as shown in Figure 2. These delays are of three types: delays due to congestion, delays due to random events, and passenger-experienced delays. The congestion-induced delays are a function of the traffic demand and the resource capacities. As traffic demand approaches the capacity limit, delays begin to mount. The delays due to random events include late aircraft loading, equipment problems, and so on. Passenger-experienced delays are a function of how late the airplane pushed back from the gate or arrived at the destination as compared to the published flight schedule. An interesting and intuitive result from DPAT is that passenger delays are often unconnected with congestion delays. If airlines expect congestion delays at particular airports or at particular times of the day, then flight schedules are inflated to account for these delays. A flight for which DPAT has computed fairly significant congestion delays may arrive on time as far as the passenger is concerned.

The delays shown in Figure 2 are at the level of resolution of the DPAT model. Most of the delays are air traffic control related, as DPAT is designed to study the efficiency of air traffic management. With clever use, an analyst can compute delays that are not explicitly modeled by DPAT. For example, delays due to excessive winds causing an aircraft to deviate slightly from its flight plan are not modeled. However, such route perturbations can be included in the trajectory design, which will then be folded into the flight path used by DPAT.

3 AIR TRAFFIC ANALYSIS

Because all resource and traffic demand data are provided to DPAT through text files, DPAT is easily configurable for any region of the world. To date, the model has been configured for the continental United States (CONUS), the Asia-Pacific region, Latin America, Canada, Taiwan, and Egypt. In CONUS, DPAT has been applied to a number of studies, including the effect of increasing arrival acceptance rates during moderate weather conditions, limits to aviation growth implied by current NAS resources (Wieland, 97), historical replay of equipment outages during a holiday traffic rush, and many others.

One of the strengths of DPAT is its ability to quickly compute traffic conditions for an entire airspace region. This strength was important during a study of future traffic growth in the Asia-Pacific region (Wojcik, et. al. 1997). Over 440 runs of the model were conducted for a sensitivity study combining future expected traffic growth with future expected increases in airport capacity. Figure 3 shows the 440 results for one of the seventeen airports modeled.

The Figure shows the passenger experienced delays in Shanghai, which measure how late airplanes arrive at the gate relative to their published schedule. The chart reveals that, if certain planned improvements in airport capacity are not made by the year 2004, delays will be substantial. By analyzing the network of Asian airports in this manner, key infrastructure improvement projects can be identified.

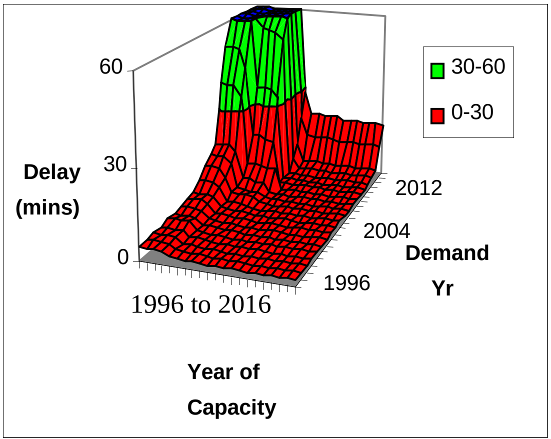

Figure 3: Passenger Delay, Shanghai Airport

Another example of DPAT analysis is shown in Figure 4. Here we have computed the overall system delay as a function of overall system capacity, to determine what happens to delays if the overall system capacity is increased. The results indicate that a doubling of system capacity will reduce average system delays by roughly a factor of 10, which is not unexpected as delays are generally a nonlinear function of system capacity.

In this example, the overall system capacity was increased. DPAT can be used to change only certain resources, or to degrade resources as a function of time. Such studies have been done in a variety of contexts, most notably in the analysis of the effect of atmospheric disruptions on the propagation of airborne delays to surrounding regions.

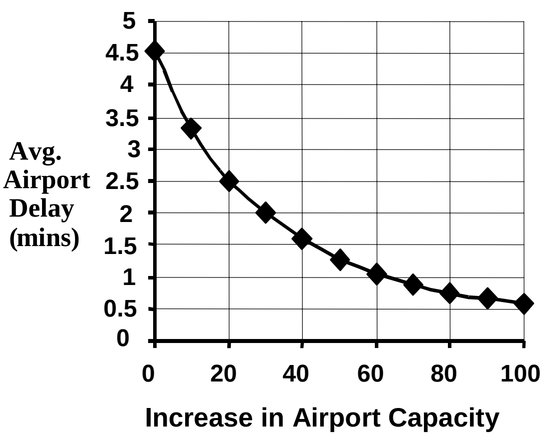

Figure 4: Delays vs. System Capacity

4 PARALLEL PROCESSING

The key to DPAT's fast processing time is its use of parallel discrete-event simulation (PDES) technology. An overview of the concepts involved in synchronizing a parallel simulation can be found in (Fujimoto, 1990). The specific parallel simulation engine used by DPAT is the Georgia Tech Time Warp (GTW) parallel simulator (Fujimoto 1989). GTW is a product of years of research into various synchronization techniques required to properly execute on multiple processors.

Although discussion of parallel simulation synchronization is beyond the scope of this paper, GTW incorporates a number of useful features that will appeal to the PDES cognoscenti. In particular, GTW employs memory-based throttling of optimism; fast computation of Global Virtual Time through shared memory; support for incremental state saving; disciplined I/O; global simulation initialization and termination; and many other features. One disadvantage of GTW is the absence of simultaneous event resolution logic, which had to be built into the DPAT simulation itself.

4.1 Spatial Decomposition

We will now focus on a question that is rarely discussed in the open literature: how, exactly, does a simulation model such as DPAT exploit PDES technology effectively?

DPAT employs a spatial decomposition in its use of parallelism. The various LPs mentioned earlier—airport, sectors, and fixes—are all spatial in nature. The moving objects are the airplanes, and they are modeled as messages passed between the stationary, spatial LPs. Decomposition of a simulation in this manner is rarely done: most simulations model moving objects as LPs as well. By decomposing the system in this manner, code complexity, software verification, and logic traces become much simpler.

Proximity detection involves determining what aircraft are close to what other aircraft, for spatial density computations and conflict detection/resolution resolution logic. Much work has been done in the PDES community concerning parallel proximity detection (see, for example, Steinman and Wieland, 1990). The basic problem is that the positions of LPs are scattered around different processors, so the joining of the information necessary to compute proximity becomes a computationally challenging task. The problem is even more vexing because in a sequential simulation all spatial coordinates are located in a centralized segment of memory, so the proximity problem is worse in a parallel simulation than in a sequential one.

DPAT solves this problem in part because of its spatial decomposition. Each sector contains a list of airplanes currently residing within it, and those airplanes that will enter it within a specified time period. These lists, plus the predictable nature of airplane trajectories, allow proximity to be computed without much difficulty.

4.2 Attention to Lookahead

In the context of the optimistic PDES technology used by DPAT, lookahead refers to how far in advance, in simulation time, an LP can schedule an event. Scheduling an event as far ahead in simulation time as possible enhances parallelism, because the probability of a synchronization event (a "rollback") is significantly reduced, and, even if such a rollback occurs, the number of "events undone" is likely to be minimal. Programming a simulation to take advantage of lookahead, however, is up to the simulation designers and implementers: parallel simulation engines merely respond to scheduled events, and (with some notable exceptions) do not schedule events themselves.

DPAT provides an acceptable level of lookahead by simulating, rather than emulating, the operation of air traffic controllers. As a simple example, consider the handoff protocol between sector A and B as an airplane passes between them. The controller of sector A will typically initiate the handoff request as the plane approaches the boundary of the two sectors, but while it is still well within sector A. DPAT initiates the handoff request when the plane *enters* sector A, thus maximizing the lookahead. If a subsequent event renders the handoff obsolete, which rarely happens, then the handoff request is cancelled.

By simulating the system in this manner, DPAT can accurately compute the entry and exit times for each airplane for each sector, can estimate controller workload, and can correctly handle both accepted and rejected handoffs. However, because the lookahead is increased, parallel efficiency is enhanced and the resulting system runs faster. Such techniques are used throughout DPAT to increase lookahead.

4.3 Other Efficiency Considerations

DPAT incorporates a number of commonly used optimizations that enhance its performance. For example, the reading of all scenario input files, traffic demand files, resource files, and other input files is performed prior to simulation execution; the results are stored in shared-memory segments. This increases DPAT's memory requirements but avoids costly reading of data files during simulation execution. DPAT can run comfortably in 256 Megabytes of memory.

DPAT also minimizes the size of messages by using them to store only the data that an event will change. Data that remain constant between events are stored in main memory, with references to them in the messages. Minimizing the size of messages decreases the amount of logic needed to format and interpret the messages, allowing for faster scheduling in the user code. While the user code runs faster, the GTW simulation engine uses shared memory to pass messages among LPs, so its internal overhead is independent of the size of the messages.

For a large DPAT simulation of the entire CONUS for one day, these and other optimizations, result in a run time of less than one minute on a four processor shared memory 300 MHz Sun SPARCstation, with 512 Mbytes of memory. For smaller studies, such as those done in the Asia-Pacific region or CONUS studies in which only a few hours of air traffic are modeled, DPAT will complete in less than 30 seconds. These run times include all the time required to initialize the simulation as well as to write data files subsequent to simulation completion.

5 WEB-BASED INTERFACE

While the use of PDES technology impressively improves the run time of DPAT, the credit for its widespread use lies mostly with its use of a Web-based interface. DPAT was one of the first simulations to use Web-based technology; in the future, interfacing in such a manner will be commonplace.

DPAT uses the Common Gateway Interface (CGI) protocol to interact on the Web. A series of UNIX shell scripts and Perl programs build Web pages dynamically. The user interacts with DPAT by specifying the basic inputs (traffic demand, system capacities and how they change with time, restrictions, delay programs, and so forth) on one page, running the simulation on a second, and inspecting the output on a third page.

The input page consists of host-interpreted Java programs as well as server-executed programs and scripts. Together, these utilities allow the user to specify and change the most commonly used inputs to DPAT. The run page consists of a summary of the input supplied to DPAT, allowing an analyst to go back and change any input that might have been entered erroneously. If all input is correct, the analyst then executes DPAT by clicking on a button. The system writes the Web-collected input to DPAT's data files, executes DPAT, and then routes the output files to the analyst's browser.

Because of DPAT's PDES technology, an analyst typically waits one minute plus network overhead before receiving the output. Network overhead can be substantial, especially at remote sites during busy periods of the day. For sites with fast internet service, the network overhead adds about 10% to the run time; for sites with slow service at busy times of the day, the network overhead can double the analyst's wait time.

The output page provides the capability of graphing the most commonly requested variables computed by DPAT on an airport-by-airport or sector basis. In addition, the raw schedule data is provided in a format that can be easily downloaded to a PC-based spreadsheet. This creative use of Web technology has led to DPAT's rapid acceptance both within CAASD and with external organizations.

6 VALIDATION

A critical question for any simulation such as DPAT is its degree of validation: how much can we trust the answers provided by the system? There are two ways to validate the system, theoretically and empirically. In theoretical validation we determine whether the conceptual model makes any sense given what we know about the real-world system. In empirical validation, we compare the actual results from the simulation to those obtained in the real-world system.

Theoretically, DPAT is valid up to a point. Modeling the NAS as a sequence of capacitated resources makes sense given that we are interested in delay, throughput, and density statistics. Airplanes do, indeed, compete for resources with each other: runways, controllers, airspace sectors, terminal gates, and so on. To the extent that we are interested in computing resource contention delays, the model is theoretically sound.

However, it lacks some random elements that are important in a real-world system. For example, DPAT does not attempt to cancel flights if they are excessively delayed, and it does not reroute airplanes unless the analyst has provided alternate routes in the input data file. Flight cancellation/substitution and rerouting are all important tools used by airlines to manage NAS disruptions. Some work is being done on modeling these effects in DPAT, however, the work is in a preliminary state (Callaham, et. al. 1997). DPAT is useful, therefore, for computing *worst-case* delays, which would happen if all intended flights actually flew. When DPAT is reading a live data feed, such as the FAA-operated Enhanced Traffic Management Feed, reroutes and flight cancellations/substitutions are available to DPAT and can be directly modeled.

Because of these theoretical concerns, empirical validation is more difficult. To compare real-world results with DPAT results requires a traffic day where rerouting and flight cancellation are at a minimum. Typically such days occur during periods of good weather. One such validation study focused on how close DPAT could get to predicting throughput for three major airports on a good weather day: Atlanta, Dallas Ft. Worth, and Chicago (Burke and Li, 1997). The results indicate that, with appropriate choice of airport capacities, DPAT correctly predicted the hourly throughput about 50% of the time; the remaining 50% of the time the throughput was generally within 30% of actual. Because there is much that is not modeled in DPAT, this level of accuracy is surprisingly high, and may indicate that the approximation of the NAS as a set of capacitated resources is closer to reality than originally suspected.

Validation of a simulation such as DPAT is an ongoing process. It is insufficient to validate it once, and then forever use its results. Rather, detailed DPAT studies carefully validate their input and test the output under conditions whose results are known and are similar to the scenario under study. Then the model is applied to the actual scenario.

An often-asked question is how valid DPAT results are given that some of its input sources, such as airspace capacities, are uncertain. The answer is that simulation, as a general computational technique, is *best* used when some of the input is uncertain. A skilled analyst will carefully perform a sensitivity experiment over all variables whose values are poorly known and may affect the result in a dramatic manner. The Asia-Pacific study mentioned earlier is an example of such a technique. The resulting distribution of output yields an insight into the behavior of the system that few other computational techniques can provide. This capability is particularly enhanced for fast-time simulation, where hundreds or thousands of runs are possible in a short amount of time.

7 PLANNED EXTENSIONS

The conceptual model of DPAT is flexible enough that it is easily extended beyond its original purpose. We shall concentrate on two important extensions, one dealing with real-time decision analysis and the other with non-delay metrics.

7.1 Real-time Decision Aid

The miles-in-trail (MIT) feature of DPAT allows it to compute the delays resulting from FAA-imposed restrictions due (most commonly) to bad weather, although equipment outages and other problems can also trigger the imposition of restrictions. Restrictions control the flow of aircraft into and out of the regions or airports where the problem lies. Regulating the miles between aircraft is the most common form of enroute restriction, however, regulating the time between successive aircraft is also used.

To configure DPAT as a real-time decision support tool, it initializes itself using live data feeds from the FAA. The live feeds provide information on currently airborne flights and flights that intend to depart within the next eight hours. Combining this information with the proposed MIT restrictions entered by the decision-maker and the expected future capacity constraints, DPAT can quickly assess (in less than a minute) the expected magnitude of the MIT delays. The decision-maker can assess multiple different options in a few minutes' time. If a particular input variable is thought to be uncertain, the decision-maker can run the system over the range of anticipated values, thus gaining insight into the probable behavior of the NAS in the presence of the uncertain variables and the MIT decision.

DPAT is currently being validated in this configuration. The system employs a Java-based interface for the decision-maker to enter and view the impact of restrictions. The real-time feed and the DPAT simulation all run on a multiprocessor Unix-based workstation.

7.2 Metrics other than Delay

Although delay and its related statistics (throughput, traffic density, etc.) are important metrics, many analysts wish to configure DPAT to measure other quantities of interest. In particular, environmental impacts (such as air particulates) and economic statistics (such as cost or revenue) can be extracted from the basic model.

Economic statistics such as the cost to operate flights or the revenue from operating an air traffic control system can be computed by DPAT (although it is not currently configured to do so). Flight operating costs depends mostly on fuel burn, which in turn is driven by aircraft type, engine type, weight, altitude, time spent in the air vs. time on the ground, and so on. Although currently DPAT does not compute any cost statistics, adding this metric is straightforward.

8 LESSONS LEARNED

The most important lesson learned so far in designing, implementing, and using DPAT is the value of fast-time simulation. While it is theoretically possible to run a sensitivity study with a simulation that takes hours to complete, in practice this is rarely done. Hundreds of hour-long runs often take days to complete, and in most cases the runs have to be done multiple times because of input errors, uncertainties in parameters, or because the understanding of the original decision question is evolving. Practically, it can take months to complete a study with a long-running simulation.

DPAT's fast run time, however, allows rapid assessment of input variable sensitivity as well as thorough exploration of the input space, all in a relatively short time. As such, it is much more practical to complete a thorough study with a fast-time simulation than with one that runs much slower. Although this result may seem intuitive, it has been our experience that the slow acceptance of parallel simulation technology is often motivated by the belief that a slow, sequential simulation is just as useful; our experience has shown the opposite to be true.

A second lesson learned concerns building fast-time simulations. To build a fast simulation, it helps to begin with a fast simulation engine. The GTW parallel simulator has proven its usefulness time and again as an efficient engine for parallel simulation. While it contains fewer features than many sequential engines, its efficient kernel makes building fast simulations easier.

A third lesson concerns simulation acceptance. Acceptance is abetted when the domain experts can easily understand the conceptual model, and when access to the simulation is made easy, such as through a Web-based interface. While a good user interface has been a goal of most simulation projects, it has only been within the last five years that providing distributed access to a centralized simulation core over the Web has been possible. Distributed access allows analysts to use the simulation from their own desktop, enhancing convenience and building acceptance.

A fourth lesson concerns the tradeoff between abstraction and emulation. Many simulations attempt to directly emulate the real-world system that they are representing. While in some cases direct emulation is critical, it is not true in all cases. Abstracting the system to a level where important questions can be analyzed is often more useful, and less costly, than building a detailed emulation of the physical world.

DPAT has proven its utility as a world-class aviation simulation in numerous studies of the CONUS as well as of international regions. DPAT continues to evolve as an analysis product, and its pioneering use of these technologies is likely to be emulated by other simulation systems in the future.

ACKNOWLEDGEMENTS

Over twenty CAASD people have contributed to DPAT development over the past three years; we would like to thank each one. We would also like to thank Dr. Richard Fujimoto of Georgia Tech, as well as Dr. Chris Carrothers and Maria Hybinette, for their help with GTW over the years. DPAT uses the GTW parallel simulation system. For further information on GTW, contact Dr. Richard Fujimoto, College of Computing, Georgia Institute of Technology, Atlanta, GA 30332-0280, fujimoto @ cc.gatech.edu, or on the Web at www.cc.gtech.edu/fac/Richard.Fujimoto.

The contents of this paper reflect the views of the author. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty or guarantee, or promise, expressed or implied, concerning the content or accuracy of the views expressed herein.

REFERENCES

David P. Burke, Wei Z. Li, "NAS Model Airfield Capacity Calibration," The MITRE Corporation, MTR WN97W63, McLean, VA, August 1997 (not for public release).

Michael B. Callaham, Jason H. Goodfriend, Debra A. Moch-Mooney, "Modeling Airline Response to NAS

Delays," The MITRE Corporation, MTR 97W102, McLean, VA, September, 1997 (not for public release).

R. M. Fujimoto, "Time Warp on a Shared Memory Multiprocessor," *Transactions of the Society for Computer Simulation*, 6(3):211-239, July, 1989.

R. M. Fujimoto, "Parallel Discrete-Event Simulation," *Communications of the ACM*, 33(10): 31-53, October 1990.

Jeff Steinman and Frederick Wieland, "Parallel Proximity Detection and the Distribution List Algorithm," *Proceedings of the 8th Workshop on Parallel and Distributed Simulation*, Edinburgh, Scotland, 1994.

Frederick Wieland, "Limits to Growth: Results from the Detailed Policy Assessment Tool," *Proceedings of the 16th AIAA/IEEE Digital Avionics Systems Conference*, II:9.2.1 ff. Irvine, CA October 1997.

L. A. Wojcik, F. A. Amodeo, D. N. Buza, J. A. Nardelli, F. P. Wieland, "World Regional Air Traffic Modeling with DPAT," MTR 97W70, The MITRE Corporation, McLean, VA, September 1997.