largest network to use the system is that of the Gothenburg region, which has a population of 0.7 million and 11 different transit operators. One of the problems in Gothenburg was the distribution of costs and revenues among these operators and the distribution of deficits among the municipalities concerned.

In most cases, Volvo has participated in planning origin-destination surveys on public transit. In one application, optimization of route connections reduced the number of transfers by 24 percent and at the same time reduced wait time by 12 percent and vehicle requirements by 11 percent. Optimization of departure times in another city reduced existing transfer times by 20 percent. In that city, Volvo also printed the departure times from each stop, which were to be posted at the stop.

REFERENCES

1. I. Andréasson. A Method for the Analysis of Transit Networks. Presented at the Second European Congress on Operations Research, Stockholm, Nov. 29 to Dec. 1, 1976.
2. D. Hasselström. Connecting Bus-Routes at a Point of Intersection. Presented at the Second European Congress on Operations Research, Stockholm, Nov. 29 to Dec. 1, 1976.

Computer-Animated Simulation of Taxi-Dispatching Strategies

Ezra Hauer, Ronald M. Baecker, and Paul D. Bunt, University of Toronto

A conversational, animated modeling environment implemented on a minicomputer is used to build and experiment with models of taxi-dispatching systems and strategies. The implications of each strategy are portrayed by the production and playback of an animated movie that depicts the simulated system as if it were real. The modeler can observe the system, modify the strategy, and immediately see the effects of the change. Use of the technique to assist in verifying the correctness of the simulation program, to enhance human intuition and aid strategy formulation and testing, and to facilitate technical communication about the model is described.

The primary task of the taxi dispatcher is to match taxicabs and passengers. The focal point of the dispatching process is the decision as to which taxi is to serve which call. The dispatcher's choice is made in a more or less systematic manner, governed by a set of rules that can be called a dispatching strategy. Obviously, many such strategies exist. They may be designed to (a) be simple, (b) ensure an equal share of business among taxis, (c) minimize the taxi travel distance, or (d) provide a high level of service to passengers.

Present taxi-dispatching practice has evolved over years of experience. Each company has custom tailored its procedures to a specific area and fleet size. Two elements, however, are common to most: exchange of information between the dispatcher and taxi drivers by means of radio communication and division of the service area into zones for referencing vehicle position. By using only a pencil and paper, a skilled dispatcher can oversee the operation of 100 or more taxis.

Still, it is natural to ask whether the efficiency or the service provided by the taxi industry could be improved by developing better strategies. The question cannot be answered in general terms; the answer must be in terms of the costs of implementing a dispatching strategy and expected benefits such as fewer vehicle kilometers of travel, more equitable distribution of fares, and less waiting time for passengers.

The only practical way of providing quantitative evaluations of proposed dispatching strategies is through computer simulation modeling. By using this technique, the essential features of a taxi system and the rules of a dispatching strategy are expressed as a computer program. The computer then derives the implications of adopting the strategy. There are, however, several major problems with this technique:

1. Simulation models are often so complex that there is little assurance the computer is doing what it was intended to do.
2. Simulation models are only as good as the assumptions on which they are based so that, even if the model is doing what was intended, it may have little relevance to reality.
3. Developing good strategies is difficult. The taxi dispatcher may be thought of as playing a "game" that even beginners can play. The beginner may be content to plan one move ahead and only consider a few passengers and taxis at a time, but this is not very good game-playing. To play the dispatching game well, one must think a few moves ahead, considering all pieces in the game. Not only must immediate mistakes be avoided; the pieces must also be well placed for future developments.
4. In most simulation environments, implementing and testing new strategies are difficult. Cards must be punched, complicated and error-prone job-control conventions overcome, and slow batch turnaround endured.
5. Finally, simulation modeling is a mystery to many transportation planners and practioners—a technological black art—partly because explaining its content and convincingly demonstrating its results to the uninitiated are so diffcult.

This paper describes the application of a relatively new simulation technique—conversational, animated simulation modeling—to the development of taxi-dispatching strategies. The technique helps to verify the correctness of a simulation program by visually tracing the process and thus assuring the user that the computer is doing what it was intended to do.

Animation provides the user with the ability to identify situations in which the existing dispatching strategy leads to poor-quality decisions. The strategy can then be modified and its performance tested again. In other words, the animated taxi-dispatching "game" is an efficient tool for the enhancement of intuitive understanding and for the improvement of dispatching strategies. The implementation and testing of these strategies are aided by the conversational modeling environment, which also makes possible technical communication about models among researchers and with practitioners.

CONVERSATIONAL, ANIMATED SIMULATION MODELING

The use of computer animation in simulation studies is called dynamic modeling. A dynamic modeling environment consists of a computer language for expressing simulation models and their animations and a programming system in which these programs can be written, verified for correctness, and executed. The initial dynamic modeling environment is described elsewhere (1). In the project that is the subject of this paper, a new and superior environment was used—a conversational implementation of the process-oriented simulation and animation language SLOGO (2).

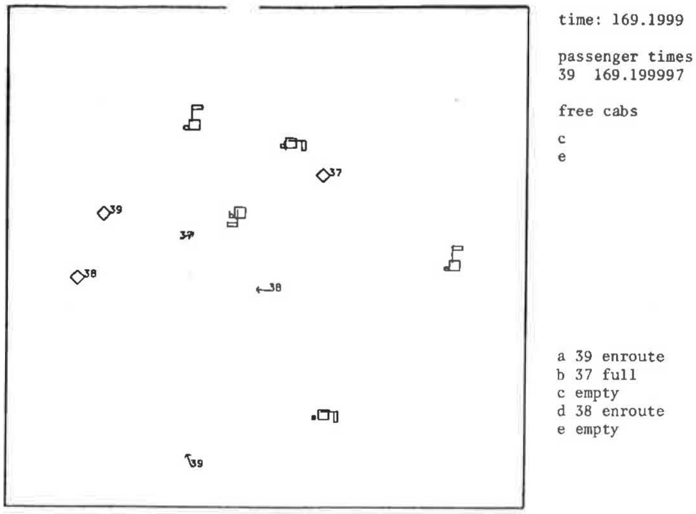
Because the implementation of SLOGO is conversational, the modeler can input, revise, and test simulations directly at a computer terminal. Because the simulation language is process-oriented, models can be expressed in a natural manner, the entities in the program mirroring the entities in the situation being modeled. Thus, with a few keystrokes, the modeler can alter a dispatching strategy or change a model parameter. Seconds later, a new simulation begins to execute and the results of the changes can be observed.

SLOGO contains animation capabilities, which means that the execution of the models is visualized as an animated movie that depicts the evolution of the simulated system. Passenger origins and destinations are represented by arrows and diamonds that appear on the screen when a call is received and disappear when the passenger has been delivered to the destination. Taxis are represented by squares with flags, which are upright when the cab is free, sideways when the cab is en route to pick up a passenger, and down when the cab is delivering a passenger to a destination. As the model executes, the taxis move to their various fare origins and destinations. The movie, which is produced as a by-product of the model execution, can be viewed at the display terminal as it is produced and can later be reviewed at any speed as often as desired. Frames from the movie may be recorded on an electrostatic printer-plotter. An example of this graphical form of model documentation is shown in Figure 1 (the detailed legend shown in this figure is applicable to the other figures as well).

Simulation Verification

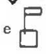
Programs that simulate real systems are invariably complex, and the process of developing a complex program is highly idiosyncratic. No matter how competent

Figure 1. Electrostatic printer-plot animated model and legend.



```
time: 169.1999

passenger times
39  169.199997

free cabs

c
e

a 39 enroute
b 37 full
c empty
d 38 enroute
e empty
```

LEGEND

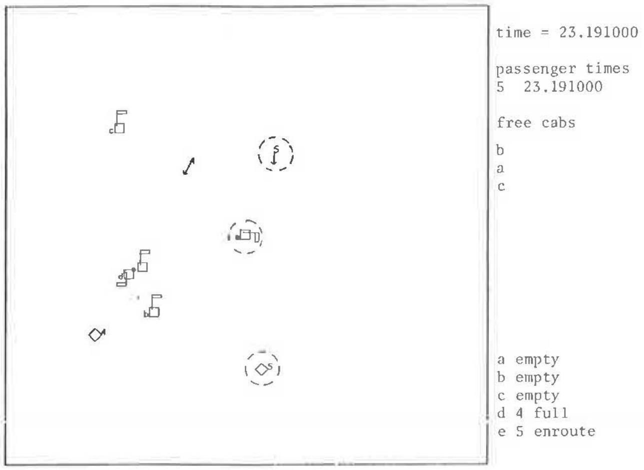
↖39	LOCATION OF ORIGIN OF PASSENGER CALL NUMBER 39 IS AT TAIL OF ARROW.
◇ 39	LOCATION OF DESTINATION OF CALL 39 IS CENTRE OF DIAMOND.
e 	CAB NO. e IS LOCATED AT CENTRE OF SQUARE. FLAG IN UPRIGHT POSITION INDICATES CAB IS UNENGAGED.
a 	CAB NO. a IS CURRENTLY ENROUTE TO PICK UP A PASSENGER.
b 	CAB NO. b IS CURRENTLY FULL, DELIVERING A PASSENGER TO HIS DESTINATION.
time = 169.1999	CLOCK INDICATING ELAPSED TIME FROM BEGINNING OF SIMULATION.
passenger time 39 169.199997	LIST OF PASSENGERS WAITING TO BE SERVED, INDICATING CALL NO. AND ARRIVAL TIME FOR EACH.
free cabs c e	ORDERED LIST OF EMPTY CABS AWAITING DISPATCH.
a 39 enroute b 37 full c empty d 39 enroute e empty	SYSTEM STATUS INDICATING PRESENT ENGAGEMENT OF ALL CABS IN SYSTEM INCLUDING ASSOCIATED CALL NO. OF FARES BEING SERVED.

the programmer may be, when programs exceed a certain complexity it is difficult for the human mind to exercise strict control. Even after the program appears to be working properly, there is always a lurking suspicion that some possibilities may have been overlooked. Yet the machine will continue to churn out numbers, some of which may be wrong. They may not be completely out of the range of reason but sufficiently so to lead to erroneous conclusions. Uncertainty about the correctness of results obviously hampers the use of simulation as an analysis tool.

A variety of measures may be used to increase confidence in the correctness of the output of a simulation program. One technique is to print out intermediate results for the first few events and to examine the correctness of each step. This tedious procedure reduces the change of error, but it is time consuming, cumbersome, and does not ensure that the results will be any more correct than those that have simply been double-checked.

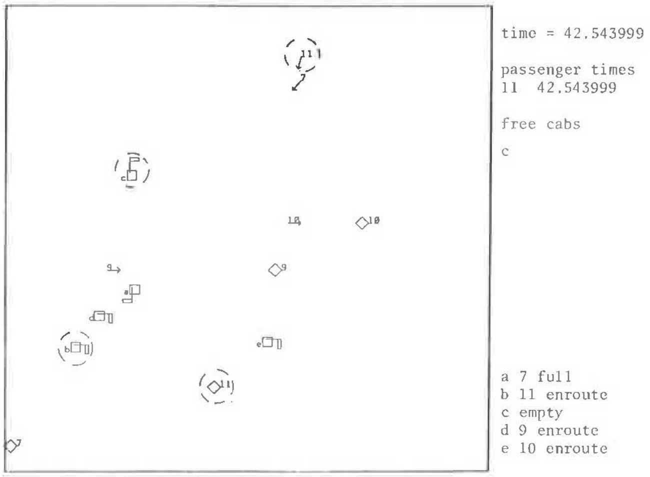
A second method is to replicate the simulation with two different programs in two different languages and then run the two models with the same sequences of random numbers. The two programs should produce identical results at each step. This was done in this project by using FORTRAN and SLOGO. In two instances, discrepancies were detected. Neither discrepancy proved to be the result of an error. Both were caused by slight differences in the assumptions and approaches of the two formulations. Uncovering and clarifying these differences led to a clearer understanding of the programs and greater confidence in the validity of the results.

Figure 2. Dispatching strategy S1.



```
time = 23.191000

passenger times
5  23.191000

free cabs

b
a
c

a empty
b empty
c empty
d 4 full
e 5 enroute
```

Figure 3. Shortcoming of strategy S1.



```
time = 42.543999

passenger times
11  42.543999

free cabs

c

a 7 full
b 11 enroute
c empty
d 9 enroute
e 10 enroute
```

A third technique is to write down in precise mathematical terms the assumptions embodied in the program and the properties of the result it is supposed to compute and then to prove that the result of the program does satisfy those properties. This technique of proving programs correct is in its infancy and has apparently never been applied to a simulation model or a program of comparable complexity.

The fourth verification method is visual process tracing, a capability unique to model animation. In principle, simulation translates the main features of a real process into computation commands and then collects and outputs numerical information about the process. Animation of this computation reproduces an image of the real process. Thus, it is now possible to confirm, through visual clues, that the real process and the simulation procedure are homologous.

In the simulation of a taxi system, the program is supposed to follow a certain dispatching rule. For instance, taxis are listed in order of priority, and calls are assigned to the first taxi on the list. When the passenger is delivered and the taxi is free again, that taxi is placed at the bottom of the list. It is essential that the simulation program follow this strategy exactly; otherwise, its results are useless. In the writing of the program, however, many things can go wrong. The typing of one incorrect letter may mean that calls are assigned to the last and not the first taxi on the list. If a command is neglected, taxis may join the list in the incorrect position. Errors of this nature are difficult to detect because the program will not abort in execution. In an animated process, such errors are immediately evident.

In a conversational, animated modeling environment, visual verification is a constant and continuous process that happens every time the researcher sits in front of the display and tries to examine the performance of a dispatching strategy.

Strategy Development

Use of the conversational, animated modeling environment enhances human intuition about the process being simulated and therefore contributes to the development of new dispatching strategies. This can be illustrated by examining the evolution of some simple dispatching strategies that use the animated model.

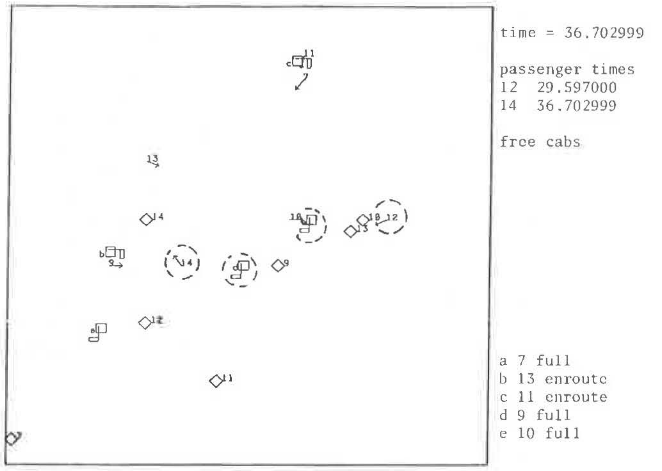
One common element in current taxi-dispatching practice is reliance on a "priority list." When a customer calls, the dispatcher assigns the call to the taxi at the top of the list. The same taxi is placed at the bottom of the list when it delivers its fare. Thus, a natural rotation and a demonstrably equitable distribution of fares among taxis are established. This strategy (S1) was selected to be animated first to serve as a benchmark against which other strategies could be compared.

Once the program was debugged and its correctness was verified, the performance of strategy S1 could be examined. Figure 2 shows the state of affairs at simulation time = 23.2 min. The list of free taxis appears in order of priority in the upper right corner. Thus, the next call to come in will be assigned to taxi b. At this point taxi d is carrying passenger 4 from origin (arrow labeled 4) to destination (diamond labeled 4). (As explained in the legend of Figure 1, the table in the lower right corner lists the status of each taxi and the passenger to whom it is assigned.) Passenger 5 has just called in. His or her origin and destination are given by the arrow and diamond labeled 5. Taxi e, which was on top of the priority list, has just been assigned to the call and its flag lowered to indicate that it is now en route. This assignment seems to be appropriate because taxi e happened to be the free taxi closest to the origin of passenger 5.

It takes only a few more frames to identify a situation in which adherence to the priority list is not in the best interest of either the taxi company or the traveling public. Figure 3 shows the situation at the instant at which passenger 11 calls in. At this time taxicabs b and c were both available, and b had priority. The call has thus been assigned to b, which is now en route to pick up the fare. But taxicab c was obviously in a better position to serve the call. Assignment of taxi c to passenger 11 would have resulted in shorter travel distance for the taxi and less waiting time for the passenger.

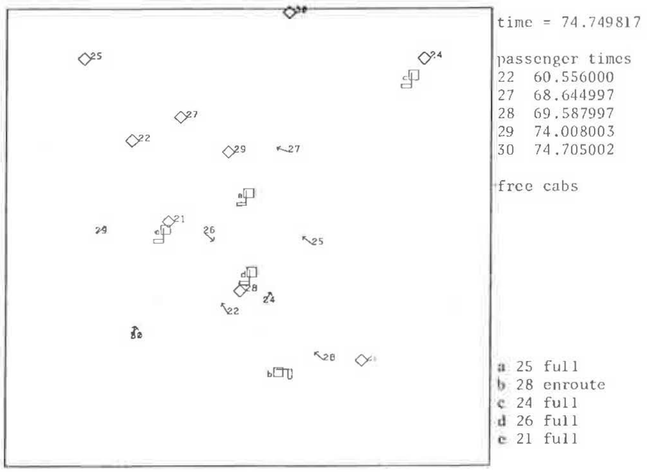
The identified deficiency of priority dispatching occurs only if more than one taxicab is available for service when a call comes in. Thus, significant improvements may be hoped for only when the demand is less than capacity. If only two taxis are free when the call comes in, the simple priority strategy will make the correct assignment in half of the cases; if three taxis are free, the right assignment will be made in one-third of the cases. Thus, the larger the number of idle taxis is at a given instant, the less desirable strategy S1 appears to be.

Figure 4. Shortcoming of strategy S2.



```
time = 36.702999

passenger times
12  29.597000
14  36.702999

free cabs




a 7 full
b 13 enroute
c 11 enroute
d 9 full
e 10 full
```

Figure 5. Shortcoming of strategy S3.



```
time = 74.749817

passenger times
22  60.556000
27  68.644997
28  69.587997
29  74.008003
30  74.705002

free cabs




a 25 full
b 28 enroute
c 24 full
d 26 full
e 21 full
```

In the conversational, animated modeling environment, modification of dispatching strategies is often a simple task. Changing only a few lines in the dispatcher module of the program results in the derivation of S2, a strategy that always assigns the closest free taxi to a call. Moments later, the new model begins to execute. Now taxi c and not b will pick up fare 11. All goes well with S2 until a temporary shortage of free taxis occurs, that is, until the time at which more than one passenger is waiting when a taxi becomes free.

If strategy S2 is used, the taxi that just became free is ordinarily assigned to pick up the passenger who called in first. Thus, in Figure 4, taxi d is about to deliver passenger 9 and become free. At that point in time, passengers 12 and 14 will be waiting. Because passenger 12 called in some 7 min before passenger 14, taxi d would be assigned to serve passenger 12. But the origin of passenger 14 is closer to the location at which taxi d will become free. Furthermore, taxi e is also about to become free and could be assigned to passenger 12. So it seems beneficial to assign taxi d to the service of passenger 14 and taxi e to pick up passenger 12. This switch will reduce the distance traveled en route for both taxis. It will somewhat increase the waiting time of passenger 12 but reduce waiting time for passenger 14.

Strategy S2, therefore, needs to be improved to deal more effectively with congestion. This results in the derivation of strategy S3, which will do for passengers what S2 did for taxicabs, that is, consider which of the waiting passengers is closest to the taxi that is about to become available.

Strategy S3 completely abandons the concept of a priority list. It is based on the following two rules:

1. If more than one taxi is free when a call comes in, assign the call to the taxi closest to it.
2. If more than one passenger is waiting to be assigned when a taxi becomes free, assign the taxi to the closest call.

Strategy S3, however, is not perfect either. One of its deficiencies can be vividly demonstrated by means of animation. In Figure 5, passenger 22, who called in 14 min earlier, is still waiting to be assigned to a taxi in spite of the fact that five of the passengers who called in later have already been assigned and some have even been delivered to their destinations. This inequity is a result of the use of the closest taxi-closest passenger rule. Calls 23 to 26 effectively screened passenger 22 from being assigned to taxicabs as they became free. The screening process may happen in central as well as outlying areas.

Several devices may be used to increase the equity of strategy S3 for both passengers and taxi operators. For example, one can limit maximum waiting time or the number of deviations by using a background priority list. Strategy S3 also suffers from other shortcomings. For example, only the free taxis are considered in making the assignments. Anticipating the time at which additional taxis will become free could probably eliminate some inefficiencies.

An example of this is shown in Figure 6, where taxicab a was the only free taxi and was thus assigned to a new call from passenger 15. It would probably have been better to dispatch taxi b to pick up passenger 15 after it had delivered passenger 13. Figure 6 also shows another deficiency of strategy S3. Taxicab a is clearly in no position to serve future calls. Its next dispatch order should thus be formulated so that the overall constellation of taxicabs is improved with respect to anticipated demand. One way in which this problem is alleviated in practice is by dividing a city into zones or areas and circulating taxis only within these areas. Dispatching systems based on zones could also be portrayed and examined by using the animation technique.

Figure 6. Failure to anticipate future developments (strategy S3).

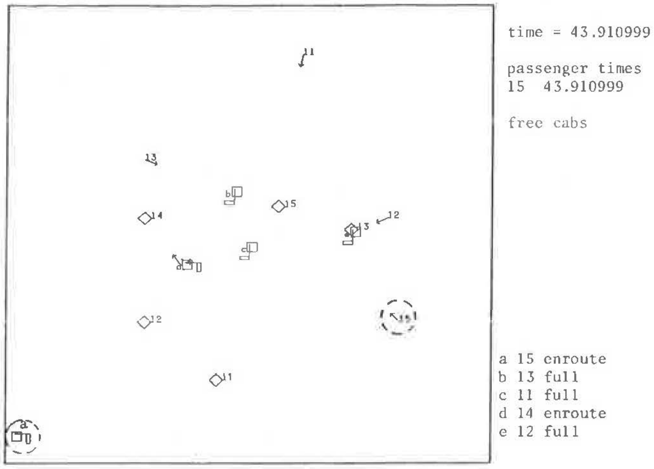


Table 1. Statistical results of three model runs for passenger level of service.

	Dispatching Strategy		
Item	1	2	3
Demand rate, calls/h/taxi	4.8	4.8	4.8
	2.4	2.4	2.4
	0.6	0.6	0.6
Mean passenger wait time, min	159.6[a]	159.6[a]	19.4
	9.5	7.6	7.0
	7.5	4.5	4.5
Maximum value of passenger wait time, min	312.5[a]	312.5[a]	100.8
	36.3	34.2	32.0
	18.4	19.7	19.7
Standard deviation of passenger wait time, min	84.8[a]	84.8[a]	18.5
	5.8	5.3	4.6
	3.8	2.8	2.8

[a] Demand rate over system capacity.

Table 2. Statistical results of three model runs for overall system performance.

	Dispatching Strategy		
Item	1	2	3
Demand rate, calls/h/taxi	4.8	4.8	4.8
	2.4	2.4	2.4
	0.6	0.6	0.6
Total kilometers traveled	4834	4834	3731
	4912	4477	4403
	4861	3901	3901
Percentage of kilometers of travel in which taxis were engaged[a]	51	51	66
	50	55	56
	51	63	63
Revenue kilometers per taxi hour	24.2	24.2	31.5
	16.0	16.0	16.0
	4.0	4.0	4.0
System load factor[b]	1.20	1.20	0.93
	0.61	0.55	0.54
	0.15	0.12	0.12

Note: 1 km = 0.62 mile.

[a] Percentage of total kilometers traveled with the meter turned on.

[b] Ratio of demand rate to service rate. A value of >1.0 indicates the system is overloaded.

The intuitive concepts gained by use of animation need to be tested quantitatively in order to evaluate the relative merits of different dispatching strategies. Statistics were collected for performance measures such as passenger waiting time, taxi revenue, and taxi load sharing under a variety of demand rates. Deficiencies and strengths of each strategy that were observed in the animation are supported statistically (Tables 1, 2, and 3). For example, at low demand rates, the mean passenger waiting time is reduced from 7.5 to 4.5 min by switching from strategy S1 to strategy S2.

Communication About Models

The animation of a simulation helps the model builder to communicate with those who have an interest in the model's operation. Although conventional simulation models produce pages and pages of individual and aggregate data, it is difficult to explain how the models actually work in terms of the output they produce. Through visual tracing, animation provides a simple yet effective method of demonstrating the internal workings of the model. Animation can thus become a vehicle for effective communication with those who are totally unfamiliar with simulation models. In this project, for example, a taxi broker and one of his dispatchers were invited in for consultation. The visual display made it very easy to communicate about the model, and the resultant exchange of ideas had several beneficial effects. The practitioners were given some ideas about how current dispatching strategies might be improved. The modeler obtained further insight into the nature of the dispatching game. A valuable communication channel was opened by means of which undocumented real-world expertise was made available.

The relative ease with which the model could be revised also proved beneficial in the demonstration of the model for the practitioners. Many inevitable "what if" questions can be answered quickly by making simple modifications to the model. Conversational, animated simulation modeling is also a valuable tool for improv-

Table 3. Statistical results of three model runs for business-sharing equity.

	Dispatching Strategy		
Item	1	2	3
Demand rate, calls/h/taxi	4.8	4.8	4.8
	2.4	2.4	2.4
	0.6	0.6	0.6
Average number of fares per taxi	80	80	80
	80	80	80
	80	80	80
Number of fares, maximum/minimum[a]	83/77	83/77	83/78
	82/77	85/77	87/72
	81/79	95/67	95/65
Kilometers full			
Average per taxi	493	493	493
	493	493	493
	493	493	493
Maximum/minimum[a]	315/285	315/285	351/301
	318/299	335/278	329/275
	322/292	351/265	315/265
Average kilometers en route per taxi	478	478	253
	490	403	389
	480	288	288

Note: 1 km = 0.62 mile.

[a] Maximum divided by minimum values give range to indicate distribution.

ing technical communication among a problem-solving group of scientists, engineers, and planners.

COST-EFFECTIVENESS OF THE TECHNIQUE

Is the animation technique cost-effective? It is difficult to quantify the value of the detection of a subtle bug in a simulation, of the extra insight into model behavior that may or may not have been obtained without this technique, or of more effective communication within the research community or with the public. However, some rough cost estimates can be made based on the experience gained in this project.

SLOGO currently runs on a medium-scale minicomputer, a PDP 11/45 that has a minimum of 48K words of memory. Sustained use requires that the system have one disk pack unit for movie storage and playback. The total equipment configuration is valued at \$150 000, which includes a \$30 000 graphics terminal on which the movies are viewed. The UNIX time-shared operating system used by SLOGO costs \$20 000 and is commercially available. SLOGO and its underlying graphics support package (3) took roughly 2 person years to design and implement.

If such a system were implemented again in a commercial environment in the late 1970s, the hardware would cost about \$100 000. The UNIX software would still cost \$20 000. A new version of SLOGO with integrated graphics capabilities could be constructed in a year by a very skilled systems programmer. A workable installation would probably sell modeling time at the graphics console for between \$20 and \$50/h. A planner could experiment with several runs of the model in an hour at the terminal.

It is also possible to use the system remotely. SLOGO drives a Tektronix direct-view storage tube over phone lines. Images are painted slowly, depending on the bandwidth of the phone line, and animations must be viewed by superimposing multiple frames and then clearing the screen and starting over. Although this technique is awkward, it is adequate for the initial stages of model construction and debugging. A planner located in Montreal, for example, could carry out most of his or her work remotely and only travel to Toronto for the strategy exploration production runs.

SUMMARY AND CONCLUSIONS

The pilot project described here, in which computer-animated simulation modeling has been applied to the investigation of strategies for taxi dispatching, provided evidence that the technique is valuable for verifying the correctness of simulation models, for augmenting researchers' intuition about the effects of various strategies, and for improving technical communication about models among researchers and with practitioners. This is particularly important when the system is complex for it then becomes increasingly difficult to guarantee the absence of bugs in the model, to predict all the effects of presumed improvements in strategy, and to discuss them without recourse to appropriate still and moving pictures.

Figures 1 through 6 show only one kind of picture by which the simulation can be visualized. Many other kinds of graphical representations of model behavior could be produced during or after execution of the model. Distributions of waiting times and histograms of taxi distance traveled can be displayed and updated as each request is serviced. Means and variances can be computed and displayed in relation to corresponding values for other strategies. Such pictures are easily computed in SLOGO.

The major problem encountered was the need for the user to learn a new language. Planners with an engineering education generally learn only one programming language—typically FORTRAN—whereas researchers developing new simulation and animation systems regard FORTRAN as archaic and awkward and base their new systems on modern computer languages. There are only two solutions: The planner must either (a) invest several weeks of intensive effort in becoming fluent in the new langauge or (b) work through an "interpreter" who translates the planner's intentions into programs.

In conclusion, the results reported here, in an accompanying film (4), and in the earlier pilot study (1) seem sufficiently promising to warrant further work. In spite of the numerous shortcomings of SLOGO, many of which are too technical for this report and will be dealt with in the design of a subsequent simulation and animation system, the results justify the undertaking of a longer range and larger scale use of this technique on some significant transportation planning problem.

ACKNOWLEDGMENTS

Tom Duff developed SLOGO and did the bulk of the SLOGO programming. The Transportation Development Agency and the National Research Council of Canada provided financial support. The views expressed are ours.

REFERENCES

1. R. M. Baecker and T. R. Horsley. Computer-Animated Simulation Models: A Tool for Transportation Planning. TRB, Transportation Research Record 557, 1975, pp. 33-44.
2. T. D. S. Duff. Simulation and Animation. Department of Computer Science, Univ. of Toronto, MSc thesis, 1976.
3. W. T. Reeves. A Device-Independent, General-Purpose Graphics System in a Minicomputer Time-Sharing Environment. Department of Computer Science, Univ. of Toronto, MSc thesis, 1976.
4. R. M. Baecker, G. Hill, and M. Tuori. Computer Animated Simulation of Taxi Dispatching Strategies. Dynamic Research Group, Univ. of Toronto, 10-min, 16-mm, black-and-white, sound film, 1976.