Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

USING COMPUTER SIMULATION FOR RAPID TRANSIT OPERATING STRATEGIES

John S. Carson

Carson/Banks & Associates
4279 Roswell Road
Suite 604-173
Atlanta, Georgia 30342

Onala M. Atala

Parsons Brinckerhoff Quade & Douglas, Inc.
3340 Peachtree Road, Suite 2400
Atlanta, Georgia 30328

ABSTRACT

In this paper, we describe the development and use of a discrete-event simulation model of the central subway portion of MBTA's Boston Green Line light rail rapid transit system. The model was used to investigate a number of operational and scheduling issues; initially, it was used to study the effect on the train schedule of unexpected delays at station platforms for individual trains. The model was developed in GPSS/PC™ and animated using Proof™.

1. INTRODUCTION

It is now within the capability of the state-of-the art Centralized Traffic Control (CTC) and Automatic Train Operation (ATO) technology to make many of the decisions traditionally performed by dispatchers and supervisors on rail properties. Particularly when unusual events such as special events traffic, delays due to line or car problems, or accidents occur, decisions on when and where to hold or express certain trains, routing choices, and insertion of extra service in the line can be programmed into computers using an "expert systems" approach wherein the computer examines the symptoms, makes a diagnosis, and prescribes a remedy.

A very basic first step in installing sophisticated train control, however, is testing the remedies. The conventional wisdom, for example, holds that a good way to get a line back on schedule is to express some of the first trains to arrive after a problem occurs. While this is often true, it may not be so in a particular set of circumstances. Similarly, short-turning trains on a high-traffic line may accomplish less than intended. A simulation model recently developed by Parsons Brincherhoff's Transit Systems group has the potential to test the impact of delays in different locations on a transit network and to evaluate the effectiveness of various strategies in overcoming the adverse effect of delays. This paper presents a description of the simulation program's capabilities and a case study of application of the program to the MBTA's Boston Green Line.

2. THE MODEL

The simulation model was initially written in GPSS/PC, but later was converted to GPSS/H. An animation was developed using Proof. (GPSS/PC is a product of Minuteman Software. GPSS/H and Proof are products of Wolverine Software. Both GPSS/PC and GPSS/H are descended from IBM GPSS/360 and GPSS V, but each has its own unique implementation and has added different (non-compatible) extensions to the basic language.) In addition, a user interface was developed to allow a non-expert to change system parameters, and control simulation runs and animations.

The GPSS simulation model has given Parsons Brinckerhoff (PB) the capability to:

- Run multi-train, multi-start simulations
- Perform operational analyses at interlockings and branching points
- Analyze headways and passenger capacities under varying operating assumptions
- Analyze yard operations
- Analyze single-tracking operation
- Analyze temporary speed restriction operations
- Analyze normal and abnormal (i.e., scheduled and non-scheduled) operations

2.1 Model Inputs

The following inputs were required for the Green Line simulation:

- Signal block length
- Maximum permissible speed for each block
- Station platform location, length, and number of berths
- Train performance characteristics; tractive effort curves
- Operating rules relating to approaching and passing signals displaying various aspects
- Routing for each service line
- Headways at portals for each service line
- Nominal dwell times for each station
- Station dwell function relating actual dwell to headway and nominal dwell

This information was input directly at the PC terminal using data supplied by the MBTA and PB's field observations.

2.2 Model Operation

Trains on each line are generated in accordance with the headway information input to GPSS/PC. At the user's option, trains can be generated strictly in conformity with the headway (e.g., exactly three minutes apart) or using a randomization function whose distribution conforms to the observed distribution of arrival or departure times.

The model moves from one "event" to the next, where an event is defined as a change in status of a train. Passing from one block to another, stopping at a station, and moving through a switch are examples of events. In the MBTA study, trains were generated at the inbound portals of the Central Subway. They moved through the system following the appropriate routing for their service line and left the simulation at the outbound portal.

Train operations are in accordance with the operating rules input to GPSS, which prevent two trains from attempting to occupy the same space at the same time. Thus, for example, the first train arriving at Copley Junction will get the right-of-way, and trains whose moves through Copley Junction would interfere with the first train will be held until the track is clear following passage of the first train.

If the simulation is allowed to run without intervention to test strategies or to generate interim reports, a two-hour time period can be simulated with the GPSS/PC Model in 15 to 20 minutes on an IBM PS/2 Model 60, depending on complexity of the scenario. Interactive intervention in GPSS/PC slows the process somewhat, but at worst we have only found it necessary to schedule an hour of 80386-based PC time to simulate 2.5 to 3.0 hours of system operation.

When the model was converted to GPSS/H (for ease of use in animating with Proof), we found that the same two-hour time period could be simulated in less than five minutes on the identical PC. In general, GPSS/H was approximately 10 times faster on compile times, and about five times faster on run times, than GPSS/PC. (The model consisted of about 800 GPSS blocks and 1,000 lines of code total.)

2.3 Model Outputs

The simulation records train arrival and departure times at station platforms, green, yellow, and red signals encountered, and other parameters indicating overall system performance in the specified configuration. The movement of individual trains are also plotted on a CADD system as string lines (time-space diagrams). Summary tables are prepared showing the amount of time each train on each service spent in the portion of the system being simulated, enabling analysis

Table 1. Example of Summary Run Time Table

DELAY DETAILS: LOC'N 1 BOYLSTON
TRAIN # & RTE. LOC'N 2 COPLEY
28 B TIME 1 300 SEC. TIME 2 300 SEC.

MBTA GREEN LINE SIMULATION DELAY6 FILE
DELAYS: YES. HEADWAY VARIATION: NO. DWELL FUNCTION: YES

TRAIN NUMBER	SERVICE LINE	IN TIME	OUT TIME	ELAPSED TIME	TRAIN NUMBER	SERVICE LINE	IN TIME	OUT TIME	ELAPSED TIME	TRAIN NUMBER	SERVICE LINE	IN TIME	OUT TIME	ELAPSED TIME	TRAIN NUMBER	SERVICE LINE	IN TIME	OUT TIME	ELAPSED TIME
3	1	291	2478	36.45	104	2	6254	8925	44.52	96	3	5780	7825	34.08	95	4	5616	8824	53.47
86	1	5091	7390	38.32	79	2	4801	7516	45.25	89	3	5360	7577	36.95	92	4	5436	8652	53.60
110	1	6531	8843	38.53	84	2	5205	7948	45.72	2	3	320	2542	37.03	80	4	4716	7934	53.63
73	1	4371	6692	38.68	4	2	472	3309	47.28	93	3	5579	7825	37.43	72	4	4356	7623	54.45
90	1	5331	7664	38.88	74	2	4432	7309	47.95	112	3	6642	8907	37.75	70	4	4176	7444	54.47
15	1	1011	3354	39.05	97	2	5881	8787	48.43	85	3	5150	7434	38.07	66	4	3996	7272	54.60
77	1	4611	6954	39.05	67	2	4072	7028	49.27	13	3	950	3262	38.53	88	4	5256	8552	54.93
94	1	5571	7924	39.22	55	2	3361	6359	49.97	75	3	4520	6832	38.53	76	4	4536	7835	54.98
65	1	3891	6283	39.87	61	2	3734	6781	50.78	107	3	6410	8735	38.75	63	4	3816	7123	55.12
81	1	4851	7245	39.90	47	2	2992	6069	51.28	71	3	4310	6652	39.03	62	4	3636	7010	56.23
106	1	6291	8700	40.15	91	2	5512	8655	52.38	82	3	4940	7309	39.48	87	4	5076	8452	56.27
102	1	6051	8484	40.55	29	2	1912	5164	54.20	68	3	4122	6531	40.15	58	4	3456	6841	56.42
69	1	4131	6601	41.17	41	2	2685	5953	54.47	103	3	6200	8655	40.92	54	4	3276	6670	56.57
11	1	771	3285	41.90	36	2	2281	5612	55.52	9	3	740	3201	41.02	39	4	2376	5822	57.43
60	1	3651	6190	42.32	10	2	841	4202	56.02	64	3	3890	6416	42.10	51	4	3096	6546	57.50
52	1	3171	5732	42.68	23	2	1552	4985	57.22	56	3	3470	6012	42.37	83	4	4896	8351	57.58
99	1	5811	8484	44.55	18	2	1213	4915	61.70	59	3	3680	6273	43.22	8	4	576	4039	57.72
22	1	1491	4210	45.32						53	3	3260	5900	44.00	37	4	2196	5696	58.33
49	1	2931	5657	45.43						6	3	539	3201	44.37	48	4	2916	6443	58.78
7	1	531	3285	45.90						100	3	5990	8655	44.42	5	4	396	3938	59.03
57	1	3411	6190	46.32						78	3	4370	7036	44.43	45	4	2736	6289	59.22
40	1	2451	5234	46.38						25	3	1602	4272	44.50	33	4	2016	5597	59.68
19	1	1251	4109	47.63						46	3	2840	5540	45.00	1	4	216	3842	60.43
35	1	2211	5098	48.12						38	3	2420	5146	45.43	30	4	1836	5486	60.83
44	1	2691	5617	48.77						50	3	3059	5848	46.48	42	4	2556	6231	61.25
32	1	1971	4928	49.28						34	3	2210	5032	47.03	26	4	1656	5386	62.17
28	1	1731	4796	51.08						20	3	1370	4202	47.20	24	4	1476	5283	63.45
										43	3	2630	5540	48.50	17	4	1116	4953	63.95
										31	3	2000	4915	48.58	14	4	936	4829	64.88
										16	3	1160	4202	50.70	21	4	1296	5203	65.12
										27	3	1790	4915	52.08	12	4	756	4826	67.83

STATISTICAL SUMMARY:	ROUTE B	ROUTE C	ROUTE D	ROUTE E
TOTAL TRAINS	27	17	31	31
AVG. RUN TIME	42.80 MINUTES	51.29 MINUTES	42.52 MINUTES	58.38 MINUTES
STD. DEV.	4.01	4.59	4.40	3.76

FILE ID DELAY6

of performance under various assumptions about delays and intervention strategies. Table 1 is an example of a run-time summary by service.

2.4 Model Animation

We developed an animation depicting train movement and signalling using Proof, a newly developed animator for the IBM PC from Wolverine Software. The animation also dynamically created the string line (time-space diagrams) referred to in the section on Model Outputs and exhibited in Figure 1.

Proof is a post-processing animator that runs from three input files: a layout file, an animation trace file, and an optional script file. The layout file describes the static background. This background is a virtually unlimited two-dimensional plane, with the screen acting as a viewpoint onto all or a portion of the layout. The simulation itself crates the animation trace file. The trace file controls all movement on the screen of dynamic objects, and all updating of dynamic text (e.g., statistics being displayed dynamically). The optional script file allows a user to put together a presentation consisting of one or more "slides" (usually, static text and/or tabular data), snapshots from previously run animations, and portions of animations. Being CAD-like and geometric-based, Proof allows unlimited zooming, rotating, and panning, plus an isometric view (from the side and above) in addition to its standard orthogonal (top-down) view. Proof also offers very smooth motion with any number of objects moving simultaneously, while maintaining a constant (although user adjustable) ratio of simulated time to real time. Finally, since Proof is file driven, and the files are in human-readable ASCII format, Proof can be used with any simulation language that can write to a file. In fact, we have developed a version of the animation to run with both GPSS/H and GPSS/PC.

A Proof layout file was developed to represent the total track network for the central subway portion of the Boston Green Line, with all stations, interlockings (switches and crossovers), and traffic control signals. Since GPSS/H has built-in statements to write output to files, it was quite easy to add statements to the model to produce the animation trace file. In general, when a train began moving into a track circuit (signal block), the simulation model would write a command to the trace file telling Proof to place the train object at the beginning of the corresponding "path" and to start moving at a specified speed. While the GPSS transaction was in an ADVANCE block representing travel time to the next signal, the object would move to the end of the associated Proof path (representing the signal block) and either stop (if the signal was Red), slow down (if the signal was Yellow), or continue to full speed (if the signal was Green). More complex logic was needed at station platforms, which might have multiple, dedicated or non-dedicated, load/unload points. In general, very little change to model logic was required to animate the simulation.

In contrast to GPSS/H, with GPSS/PC a modeler must resort to calling a FORTRAN routine to write the trace file. In addition, with its integer clock, GPSS/PC may present problems with objects "jumping" on the screen, unless the time unit is chosen sufficiently small or care is taken to artificially adjust (increase) train speed so that the GPSS transaction representing a train completes its ADVANCE time (travel time, necessarily integer valued) simultaneously with the train object reaching the end of its path in the Proof layout. Otherwise, the train transaction would begin its time advance representing travel time on its next signal block before the train object in Proof reached the end of its current path (representing its current signal block). (The newest release of Proof removes this problem by allowing a user to specify either speed or time duration for an object moving on a path.)

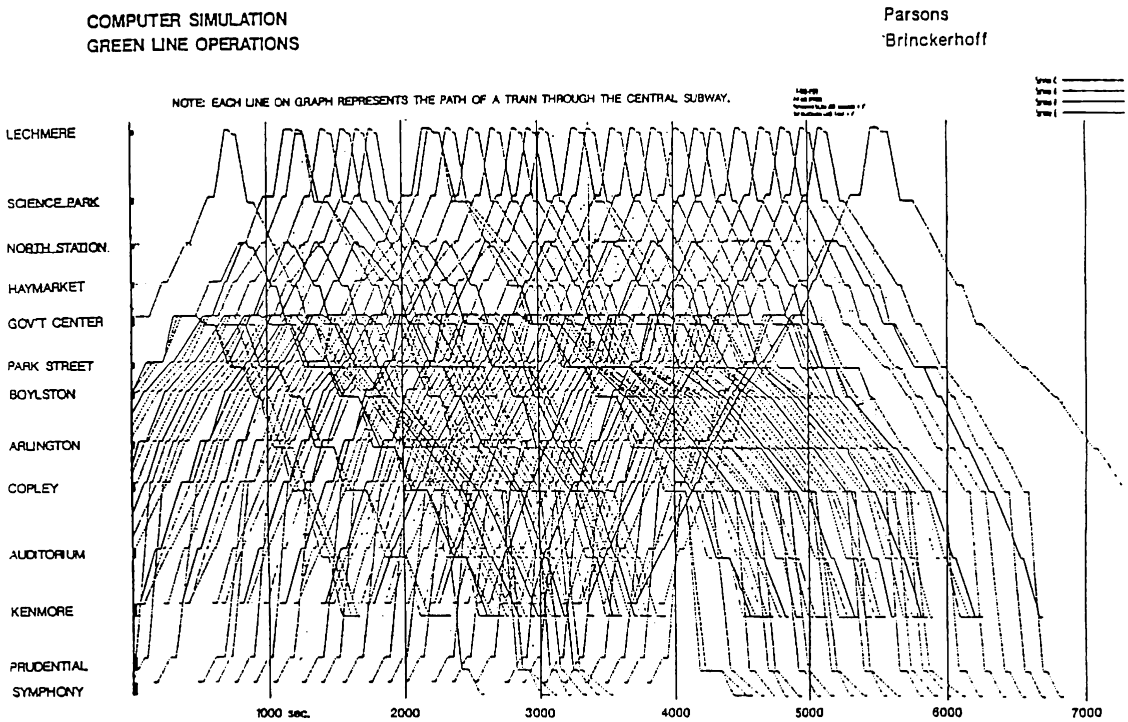

Figure 1. Train Position versus Time in Seconds

The animation was used to debug and verify the model, especially train movement at interlockings (switches and crossovers) and in stations. The animation was also used to convince model users of the model's validity, and to dramatically portray the results of the simulation.

3. CASE STUDY

Possibly the most complex light rail line operating in the United States, the Central Subway portion of MBTA's Green Line currently carries four distinct routes, with four outer termini (two of which operate some scheduled turnback service) and three scheduled in-town termini. Figure 2 is a schematic of the Green Line. Extra trains are also inserted in the schedule at the discretion of supervisory staff. Peakhour headways now are on the order of 70 seconds. Our micro-computer based simulation accurately reflected such aspects of the system as:

- train performance
- speed limits by block
- block lengths
- signal system characteristics
- platform lengths and number of berths
- interlocking plant performance
- dwell times as a function of headways (if a train is delayed, it will make a longer stop when it arrives at a station)

The simulation was run for a baseline case of scheduled headways without variation, to establish the Central Subway's performance under "no problem" conditions. Figure 1 shows an example of the basic, no-delay string line. The scenario showed the bunching of trains and occasional headway gaps that occur in reality as a function of delays at interlockings and similar system-related factors. It also showed that over time, the system had a tendency to self-regulate. Several delay scenarios were then run to study the impact of abnormal headway gaps on the various routes. One, two-and-a-half, and five-minute delays were tested at selected stations. Simulations were also run to test the impact of overall dwell reductions such as might be achieved with more doors per car or high-platform loading. Results were analyzed for realism and potential responses.

3.1 Results

Table 1 exhibits a Summary Run Time Table from one simulation run. For this particular run, train #28 on route B (service line 1) experienced abnormal delays of 300 seconds, beyond the normal station dwell time, at each of two stations, Boylston and Copley. The train number and service line for the train to experience an abnormal delay, the length of any abnormal delays and delay location were all specified as user input data. Note that in Table 1, for each service line, the train data has been sorted from smallest to largest elapsed time, where elapsed time is the total time a train spent in the underground portion of the network being simulated. Note also, that in Table 1, in time, out time and elapsed time are in minutes.

Four basic scenarios were simulated: no delays, 1 minute, 2.5 minutes, and 5 minute delays, at each of two stations. In addition, the delay locations were varied, and headway entering the portals to the underground portion was simulated with and without randomization, to produce a large number of other scenarios on which to base the following conclusions.

A number of interesting results emerged from the simulation runs. First, even with strict schedule adherence for inbound trains and no added delays, the range of Central Subway travel times was quite large:

Route B...36 to 50 minutes
Route C...44 to 57 minutes
Route D...35 to 54 minutes
Route E...56 to 65 minutes

The difference between routes reflects the different distances traveled to the various inbound terminals, but the key point is the variation in running times for an individual route. Clearly, it is necessary to compare the effects of delays and strategies with the "normal" run of events reflected in the "no delay" scenario. Nominal running times may be non-representative of actual conditions.

Second, a one-minute delay for a train at each of two stations, such as might be associated with a minor door problem, made no appreciable difference in the operation. In fact, subway running times were actually slightly better. However, these tests were simulated with randomization in train start times, and it proved difficult to distin-

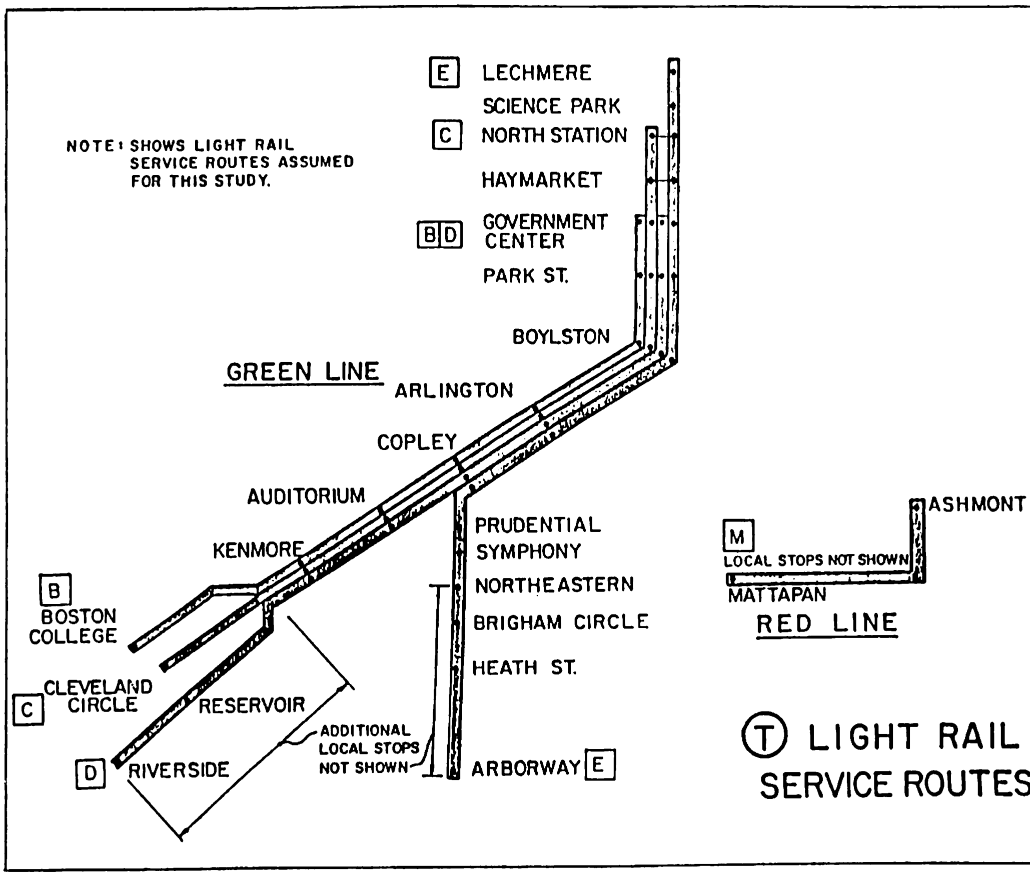

Figure 2. Schematic of Green Line

guish between effect of the specific delays and normal variation in train start times.

Third, and somewhat surprisingly, more trains were actually in operation with the five-minute delay scenarios than with shorter delays. This may be a reflection of a phenomenon also observed on freeways, where the greatest capacity is achieved at speeds significantly slower than free-flow.

Finally, within the limits of the delays studied, the system's ability to recover after one train experiences an unusual delay does not seem to vary as a function of the length of the delay. Within 45-60 minutes after a pair of delay events of either 2.5 or 5 minutes, the system appears to have recovered in the sense that the spacing of following trains on each route resumes a regular pattern with no major gaps. However, maintenance of scheduled next trip headways may require operation of meeting crews or extra cars from the outer ends of the lines.

4. CONCLUSIONS

In addition to the Green Line work, PB has applied this program in simulating operations on CTA's Loop Elevated under various route combinations, primarily in the context of accommodating the Southwest Line when it begins operations. We have also used GPSS/PS and GPSS/H to simulate the effects of single-track operations at MARTA. In all cases, it provided a realistic set of answers to questions about system configuration and operations. We have found it to be a very cost-effective tool for analyzing transit systems in both planning and operational stages. Its flexibility allows one to describe the transit line being analyzed in the simplest terms that will suffice, or to insert as much detail as necessary to analyze service performances. The fact that it provides multi-train, multi-route, multi-stop capability makes it especially powerful.

In the hands of a team of experienced transit systems analysts, GPSS is an excellent adjunct to the system concept planning and design process. Since its operations are transparent to the user, it is very easy to identify the assumptions that cause particular phenomena in system performance. For example, a chronic bottleneck in MBTA operations is caused by the track configuration at Government Center station. A series of "what-if" runs incorporating different track configurations or different run-through patterns can assist in determining whether service can be expedited effectively by changing the downtown terminals of some routes, or whether nothing short of a major rebuild of the Government Center turnaround area could possibly improve the situation measurably. Simple modification of some of the input parameters will provide answers in short order.