Dynamic Movement Prediction: An On-Line Railroad Simulation Model

Michael G. Tashker, Peter J. Wong

SRI International, Menlo Park, California

ABSTRACT

The Dynamic Movement Predictor is a FORTRAN event-step railroad simulation. It is initialized with yard inventories and consists of trains currently running from an on-line data base (i.e., RAILS: Railroad Automated Identification and Location System). The user inputs block definitions, train schedules and marshalling instructions as well as expected traffic for nonautomated network points. The simulation generates yard and train reports as well as system cost statistics. The simulation system allows the user great flexibility in editing the input data and allows him to make tactical operating decisions, due to fast turnaround time.

DMP OVERVIEW

The Dynamic Movement Predictor (DMP) is a computer model design to take advantage of the enhanced real-time information capabilities of the car movement data base of the Grand Trunk (GT) railroad's Railroad Automated Identification and Location System (RAILS). The DMP provides a computer tool for investigating improvements to the operation of the GT system. Therefore, evaluations of various train schedule alternative can be conducted rapidly with minimum costs and with no disruption to GT operations*.

The DMP is an event-step simulation. In an event-step simulation, the occurence of events in time is scheduled in an event calendar; the simulation clock proceeds at nonuniform time steps from one event to another by appropriately looking up the scheduled events in the event calendar. Although the logic for an event-step simulation is more complicated than that for a time-step simulation, the nonuniform occurrence of events in the DMP indicated that an event-step simulation would be the most efficient approach.

The section of program that actually performs the simulated movement is the DMP executive, which commences when the event calendar has been formed by initialization and ceases when the end of the specified simulation period is reached. The general flow of the executive is to find from the event calendar the next event, perform the appropriate activity to the train involved, and place a new event in the calendar (except immediately after the end of a train run, when the event calendar is compressed). Subroutines are called by the executive to perform each type of activity.

In addition to this general flow, the DMP executive contains a conflict avoidance logic that approximates the actions of trains near each other. This logic can slow down or stop trains in an overtake situation or delay trains near track switches. Although the corrective action of this logic is executed infrequently, its presence allows the network to be made quite complex.

The simulation logic provides the mechanism by which cars are transferred between the yard file and train file, thus simulating the movement of cars on trains, the setting off and picking up of cars from the yards, and the accummulation and storage of cars at yards. Although the logic is designed to move supercars (a group of cars moving together from the same system origin to the same system destination), it can be used to move individual cars, depending on the use of DMP.

The simulation package consists of the DMP executive program; a number of programs by which the network, train schedule, and block definition are input; a preprocessor program that creates yard inventories and train consists from the RAILS car movement data base; and a set of programs that produce output reports after simulation run. Because of the modular nature of the programs, initialization data for a particular day can be stored and rerun with different schedules. In a similar manner, different schedules can be stored and used as needed.

PROGRAM MODULES

The DMP is a collection of several jobs to be run in a specified sequence. The jobs are:

- Block definition input
- Network configuration input

*See References 1 and 2 for details

CH1437-3/79/0577-0583$00.75

1979 Winter Simulation Conference

- Train schedule input
- RAILS car movement data processing
- Interchange and industrial traffic input
- Optional train schedule editor
- Simulation executive
- Train, yard, and system report production

Figure 1 shows the sequence of running these jobs. The output from each job is stored on disk, so a job need not be rerun unless its input parameters have changed. Various system parameters may be saved and run later under different conditions if desired.

Block Definition

DMP uses a blocking concept similar to the classification concept commonly used by railroads. A block is defined as a set of destinations and off-going roads for which cars are grouped. A block of cars can be carried by a train to one setout point. At any given pickup or setout point, however, a train may pick up or set out more than one block. Blocks can consist of cars going to single yards or groups of yards.

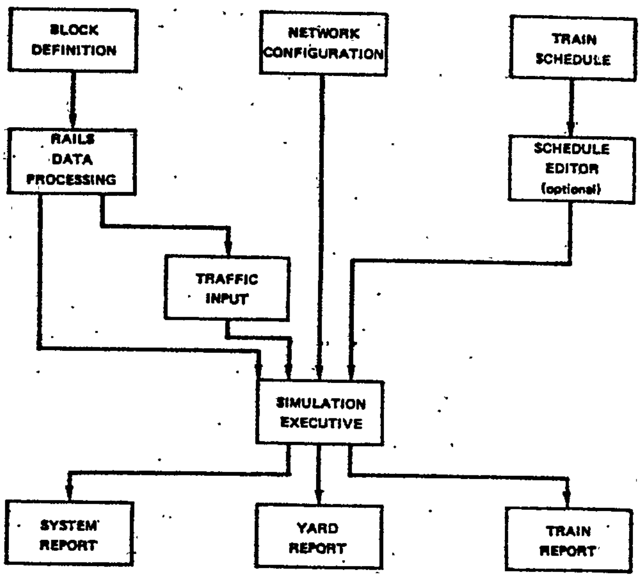

FIGURE 1 SEQUENCE OF JOBS RUN IN DMP

Network Configuration

The DMP model of the GT track network is shown in Figure 2. Unlike most maps, which are composed of sections of road connecting nodes, the DMP network model is composed of track sections, or arcs. Each arc has a unique number. Both yards and interyard stretches are modeled as arcs. Each arc has associated with it a length, transit time parameters, and both constant and time-dependent delays. The system network is defined by describing the arcs connected to the two ends of each arc, as well as the direction of each arc. The junction of three arcs is a switch and is considered to have both a preferred and an alternate direction. Our modeling of the junction of three arcs is not meant to model real switches, although it could if the network mode were complex enough. The idea of preferred and alternative connecting arcs is a simple aid to routing trains; trains will travel along preferred arcs unless an arc switch activity is scheduled. Thus, for the majority of cases, trains can run along their routes without user input to control them.

Train Schedule

The train schedule instructs the simulation abou activities performed by each train. This includes information commonly found in the GT freight train schedule and marshalling instructions as well as classification definitions and routing instructions. The data for a given train consist of a variable amount of information, depending on the complexity of activities the train performs. In addition to some general information, a train schedule consists of a list of yards at which activities take place, as well as codes for the activities and parameter lists. The primary activities are described below.

Arc Switch - The train is switched from a preferred to an alternative arc at its next junctio

Turnaround - The train leaves its current arc at the junction from which it entered.

Train Makeup - The makeup activity starts a train on an arc at a specified time with specified number of power units (and horsepower) and a caboose. A flag may be inserted if the train's makeup is to be triggered by the termination of another train (turn trains).

Delay - The train is delayed a specified number of minutes at its present arc.

Car Setout - All cars in the specified blocks are set out into the yard inventory. The current time is stored with each car to be used in conjunction with pickup cutoffs.

Car Pickup - The pickup delay, nominal cutoff, and maximum number of cars allowed to be picked up are specified. In addition, for each block or group of blocks to be picked up, a cutoff and pickup limit is specified if it is not governed by a default value.

The yard inventory is scanned for cars in the specified blocks meeting the cutoff requirements: a car is picked up if it entered the yard before the current time minus the cutoff. Cars are picked up in the block order specified by the input list up to the pickup limit specified. In no case, however, will cars be picked up such that the train car limit will be exceeded.

If the train does not have a setout scheduled, or if no cars actually are set out, a pickup will be effected only if the train can accommodate a certain minimum number of additional cars. If a setout is accomplished, the actual car limit is used to determine the maximum number of cars the train can pick up.

Train Termination - Train termination implies setout of all blocks on the train.

Scheduled Departure - This activity causes a scheduled departure from a yard at a particular time. A departure time is calculated by adding the times of all the activities performed at the yard to the yard to the time a train enters the yard. If this departure time is earlier than the scheduled departure time, the train waits until the scheduled time; if departure time is later than the scheduled time, no action is taken, and the train departs.

RAILS Car Movement Data Processing

The RAILS data processing job produces current yard inventory, train consist, and predictive traffic files acceptable to the simulation executive from RAILS car movement data. This "front-end" job also screens car movement data for average movements and movements on unmodeled trains involving unmodeled yards and checks the data for logical consistency.

Interchange and Industrial Traffic

This interchange and industrial traffic job accepts free-form input for industrial and interchange traffic entering the road and merges it with traffic records obtained from RAILS data processor. Traffic that can be entered at any mechanized yard at any time is described by numbers of cars (loaded, empty, or unknown), by block, or by destination yard.

Optional Train Schedule Editor

The train schedule editor is useful when making a simulation run with only a few changes in the schedule. With this editor it is possible to add a train to the schedule, delete a train, replace all data for a train, or modify parts of the data for a train. Modifications of a train's data include changing the horsepower, maximum number of cars, start time, or frequency of a train (e.g., a train can be put out of service, or a special can be added); adding data for a new activity arc; and replacing data for an activity arc. Activity may be added, deleted, or replaced within an activity arc's data. Specialized subcommands are available to add and delete blocks for setout and pickup, and to change the pickup cutoff. The editor commands are field free; shortened forms of commands may appear.

Simulation Executive

The simulation executive, the logic of which is discussed in the next section produces a simulation log--a record of all train and car movements--for use by the output report programs.

Train Reports

The train report* lists the arrival and departure times of each train at each station, as well as the number of cars arriving and departing these stations. At each station where there is a pickup or setout, the train report program lists the number of cars in each block that are set out or picked up. A sample train report is shown in Figure 3.

Yard Report

The yard report* for a single yard details the departure of cars from the yard aboard simulated trains and the arrival of cars aboard trains and from interchange and industrial sources. The times of these activities are specified, and the effect on individual block counts and the yard total is shown. Block counts are indicated as positive (arrival of cars) or negative (departure of cars) followed by the new block inventory. Block identifiers may be simply the block number itself or a name assigned to a block or group of blocks (which may be different for each yard).

System Reports

There are two system reports: a car hire cost report and a train cost report. The car hire cost reports lists, for each destination, the total number of hours that cars needed to reach their destination and the accrued car costs based on an hourly car hire cost rate. The train cost report tabulates car miles and associated car hire costs by train as well as train miles and total train costs.

SIMULATION

The simulation executive initializes various internal arrays and the event calendar with data from files produced by the input program modules. Two arrays are of particular interest. Train Equipment (TEQUIP) and Yard Equipment (YEQUIP) provide storage for information on cars on trains and in yards, respectively. Cars on all trains share TEQUIP, which contains a linked list for the cars on each train currently in the system as well as a list for unused space. YEQUIP is organized similarly for all modeled yards, with the addition of a reserved information area for each yard. The movement of cars to and from trains and yards is accomplished by transfer of data between TEQUIP and YEQUIP. This physical transfer of data is required on small systems in which there is not enough core for all the cars in the system. In such cases, YEQUIP is segmented into several data overlays; unused overlays are stored on disk. Without a core constraint (and on virtual systems), records for all cars can be kept in a single array, and transfer between trains and yards can be accomplished with pointers.

The simulation executive initializes the event calendar with data obtained from the RAILS car movement data processor. The event calendar is a set of one-dimensional arrays (see Figure 4). Once all the operating arrays have been initialized, the executive begins processing events.

A high-level functional flow chart of the executive is shown in Figure 5. On a more detailed level, however, a distinction needs to be drawn between events and activities. An activity (e.g., pickup, makeup, setout) is one that is planned and scheduled by the user and programmed in the train schedule. An event is an occurrence whose information is stored in the event calendar. There are only three kinds of events: an overtake, an end of delay, and an arc update. Scheduled activities are performed during the successful completion of an arc update.

*Train and yard reports can be teleprocessed to GT yard and dispatching offices.

ATCHN (I)	(Arc Train Chain) Index in the event calendar of the train immediately in back of this one on the same arc. The pointer of last train on arc is negative index of first train; 0 if the only train on the arc
CARC (I)	(Current Arc) Signed pointer to the train's current arc
EFLG (I)	(Event Flag) A flag indicating the nature of the train's next event 0 = arc update +J = index of train to be overtaken at this even -J = index of train for which it is waiting 32767 = 24-hr update of makeup events
ETIME (I)	(Event Time) Time at which this event is to take place
ETRN (I)	(Event Train) Pointer to the block of information about a particular train in TRAIN (i.e., the train that participates in the event)
LSTDST (I)	(Last Distance) Distance along current arc at time of last event
LSTIME (I)	(Last Time) Time at which last event occurred
NARC (I)	(Next Arc) Signed pointer to next arc of the train
TPLACE (I)	(Train's Place) Train's system entry number
TRNSPD (I)	(Train Speed) Speed of train

FIGURE 4 EVENT CALENDAR ARRAYS

The executive begins processing an event by fetching the following information from the event calendar: the train involved, the event flag (type of event), the train's current and next arcs, and a pointer to the next train behind it in its current arc (if any). The event flag shows whether the event fetched is a normal arc update, an overtake, or the end of a delay after an overtake (see Figure 6).

Overtake

If the event is an overtake, the train ahead is checked to determine whether it is moving or stopped. If it is moving, the train behind is slowed down, and the time at which it is to get off its current arc is recomputed for the new speed. At that time, all trains behind it are checked for overtake situations caused by the speed reduction. If there are any future overtakes, overtake flags are set, and the times of overtake are entered into the event calendar. If the train ahead is stopped, the current train is also stopped, a delay flag is set, and its next event is the expected time that the train ahead starts moving. Trains behind are checked for possible overtakes caused by the delay.

End of Delay

If the event is the end of a delay, the train's speed is restored, and the next event for this train is reaching the next arc. All trains behind this train are checked to determine whether they can have any overtake flags removed now that the train is moving.

ARC Update

Most events that do not involve interferences between trains are simple arc updates; that is, a new arc is to be entered and an activity is to be performed at that time if specified in the train schedule. If an update is to be executed, it is first determined whether the next arc is occupied. If the next arc is not occupied, the activities scheduled for that arc are performed, and the time at which the train will reach the end of that arc is scheduled as the next event.

If there is a train on the next arc, the direction of the occupying train is examined. If the trains are going in opposite directions, and the other train has as its next arc this train's current arc, there will be a conflict and an error message will be printed. If the occupying train is going in the opposite direction but will switch to another arc, the train in question will be delayed at the arc boundary until the switch is made (except in stations, where conflicts of this kind are not considered). A delay will also result if a train on an arc is too close for the train in question to cross the boundary. If these conflict situations are not present, the new arc is examined for overtake situations. If the train will overtake a train already on the arc, the overtake flag is set, and the next event for this train is the overtake. If there are no overtakes, the next event takes place when the train reaches the end of the arc.

COMPUTATIONAL CONSIDERATIONS AND EXPERIENCE

The DMP has been designed with a built-in data overlay structure to eliminate its dependence on the size of the host machine's memory storage when operated under IBM OS 360/370. The storage for cars in yards may be organized by the user into overlays, which are stored on high-speed disk and called into memory only when needed. In this manner, the memory storage does not have to be large enough to store data for all cars in all the yards of a network. If memory storage is sufficiently large, all data may be put into a single overlay, which is resident in memory; this of course saves internal disk read/write time. Organizing the data into a single overlay is the appropriate method when running under OS/VS1, OS/VS2, or VM/CMS, where the operating system performs paging and program management.

The DMP is currently operational on an IBM 370/135 under VS1, a commercial VM/CMS timesharing service, and has been run on a CDC 6400. For a 12-hour simulation with 23 yards and approximately 40 trains, the simulation executive takes 2 to 4 minutes under VS1 (with 362K real core) and 2 minutes under the relatively fast VM and CDC systems.

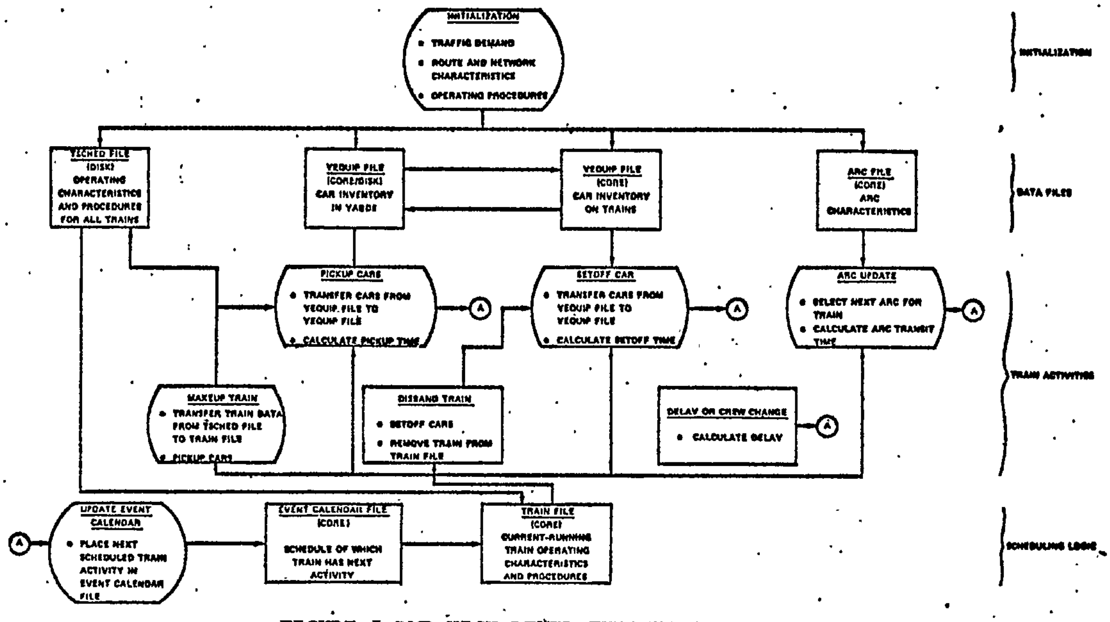

FIGURE 5 DMP HIGH-LEVEL FUNCTIONAL
FLOW CHART

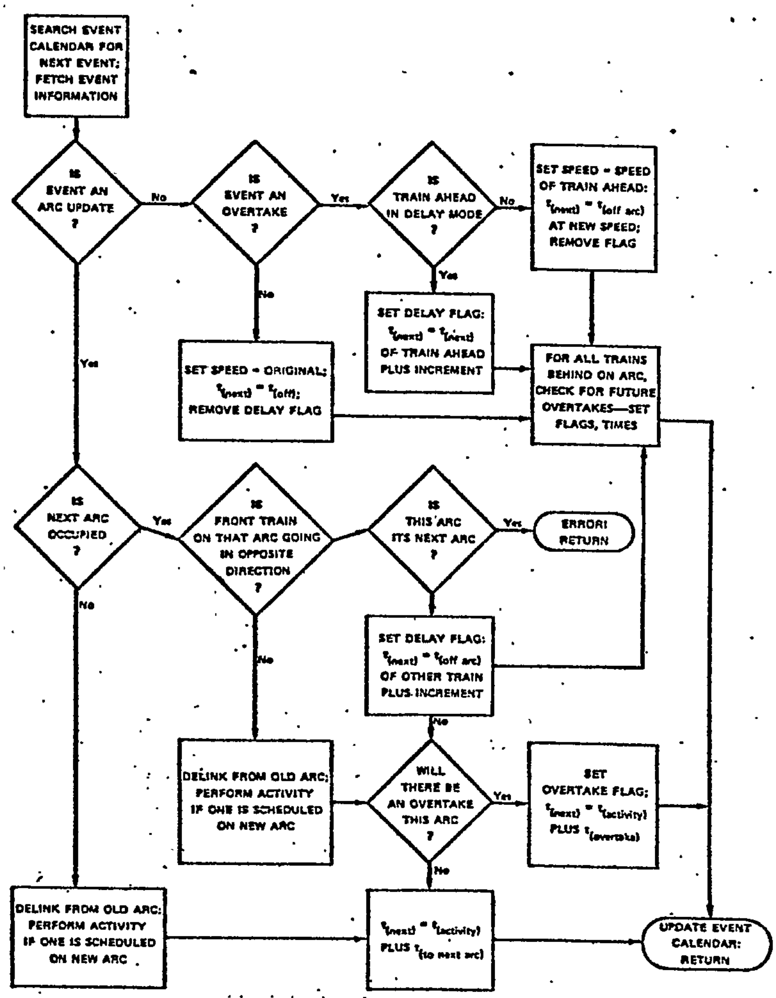

FIGURE 6 DMP EXECUTIVE

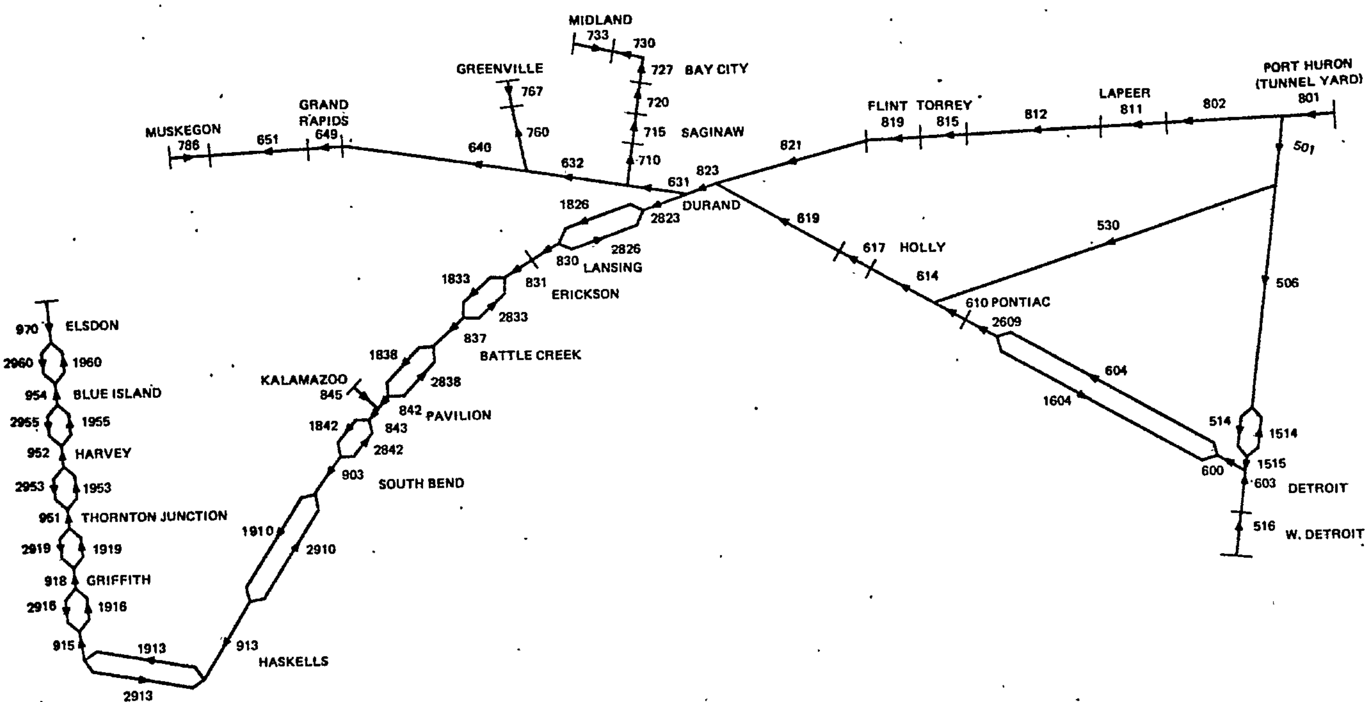

FIGURE 2 DMP NETWORK REPRESENTATION OF THE GT SYSTEM

*** TRAIN SUMMARY ***

TRAIN NUMBER	YARD NUMBER	ARRIVAL TIME	ARRIVAL DAY	WITH CARS	******** SETOUT ********* NO. OF CARS (BLOCK NO.)			******** PICKUP ********* NO. OF CARS (BLOCK NO.)			DEPARTURE TIME	DEPARTURE DAY	WITH CARS	GROSS TONS	CAR MILES
383	837	0	0		0(0)	0(0)	0(0)	47(11)	0(0)	0(0)	500	67	47	2910	0
	842	530	67	47	0(0)	0(0)	0(0)	10(11)	0(0)	0(0)	545	67	57	3246	[illegible]
	903	0	0	0							655	67	57	3246	[illegible]
	913	0	0	0							740	67	57	3246	6068
	918	0	0	0							814	67	57	3246	7652
	951	827	67	57	57(11)	0(0)	0(0)	0(0)	0(0)	0(0)	0	0	0	0	8274

FIGURE 3 SAMPLE TRAIN REPORT

ACKNOWLEDGMENTS

The authors would like to acknowledge the project technical direction provided by Mr. H.M. Tischler, General Manager-Information Services at GT, and the active participation of Mr. N. Lay, Manager-Operations Planning at GT, who was responsible for the field implementation and the training of GT personnel in using the procedures developed herein.

REFERENCES

1. P. J. Wong, et al, Grand Trunk Western Improved Railroad Operations Using a Railroad Automated Identification and Location System: New Procedures for Tactical Operating Planning Decisions, Vol. 1, Prepared for the Federal Railroad Administration by SRI International, Menlo Park, California, 94025, Report No. FRA-OPPD-77-18, October, 1977.

2. M. G. Tashker, et al, Grand Trunk Western Improved Railroad Operations Using a Railroad Automated Identification and Location System: Dynamic Movement Predictor (DMP), Vol. 2, Prepared for the Federal Railroad Administration by SRI International, Menlo Park, California, 94025, Report No. FRA-OPPD-77-19, October, 1977.