ANL/DIS/CP-102061

ELIST8: A SIMULATION SYSTEM FOR TRANSPORTATION LOGISTICS PLANNING SUPPORT

Mary D. Braun, Gordon R. Lurie, Kathy L. Simunich,
Charles N. Van Groningen, Hanna J. Vander Zee, and Mary Ann Widing
Decision and Information Sciences Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne IL 60439
e-mail: duffy@dis.anl.gov

KEYWORDS

Discrete simulation, military, transportation, Java

ABSTRACT

Planning for the transportation of large amounts of equipment, troops, and supplies is a complex problem. Many options, including modes of transportation, vehicles, facilities, routes, and timing, must be considered. The amount of data involved in generating and analyzing a course of action (e.g., detailed information about military units, logistical infrastructures, and vehicles) is enormous. Software tools are critical in analyzing these plans.

ELIST (Enhanced Logistics Intra-theater Support Tool) is a simulation-based decision support system that assists military planners in determining the logistical feasibility of an intra-theater course of action. The scenario data used by the ELIST simulation includes the units to be moved, their required movements and activities, an infrastructure database, a vehicle database, and a set of simulation parameters and constraints. Analysts can use the simulation to evaluate the overall success of a plan (determined by required delivery dates), as well as to view detailed information on such things as over- and under-utilized resources, infrastructure bottlenecks, and individual trips. The simulation can be run for as many days as desired or stopped at any time. The state of all queues and resources, and the location and activities of all entities, can be examined using many options for reports and graphs.

The current version of ELIST (v.8) employs a discrete event simulation developed in Java. Java fulfills a primary requirement for multi-platform execution. In addition, the object-oriented framework has greatly facilitated entity and process development within the simulation. This paper describes the problem domain and the modeling approach taken, and presents the representations used for entities, events, facility resources, network links, and other objects. The benefits and problems involved with developing and executing the simulation in Java are discussed.

INTRODUCTION

ELIST (version 7) has been successfully used in the military logistics community for a number of years to assist in planning analyses and training exercises (Macal *et al.* 1995). Ongoing use of ELIST7 has led to requests for more detail, more capabilities, more flexibility, and more speed. ELIST7 runs at a fairly aggregate level, in which individual transports are not explicitly modeled; instead transportation capabilities are represented in short-ton-miles/day. ELIST7 is primarily a C language tool and uses formatted files for input.

ELIST8 was conceived to model the transportation of military cargo at the individual vehicle level, requiring a much more detailed simulation than in ELIST7. This version was to more easily integrate with other models and tools in the fort-to-foxhole logistical arena. It uses the Oracle DBMS for data persistence (rather than flat files) and to facilitate data sharing with other models. In addition, the new system was required to run on multiple platforms, specifically Windows machines and Sun workstations. Because the desired changes, to the simulation in particular, were so significant, modifying the existing ELIST7 would not have been efficacious. ELIST8 is a completely new system.

We decided to develop the new version in Java for a few reasons. First, Java easily provides the platform independence we required. Second, Java provides native support for a number of required and potentially useful capabilities. The JDBC (Java Database Connectivity toolkit) supports multiple DBMSs. Graphical user interface, networking and internet, and multi-threading capabilities are all available. In addition, we wanted the advantages of an object-oriented programming language. Finally, the team was experienced with other object-oriented languages and was eager to try out Java. We did have two concerns. First, we were concerned that the immaturity of the language and changes in the ongo-

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ing development of Java itself might impact our effort. Second, the execution speed of this semi-interpreted language might hamper acceptance of the system. We did assume that execution speed would be a universal issue and would lead to faster virtual machines in time. Our findings on these issues are discussed later.

MODEL

The model underlying the ELIST simulation is based on many discussions with military logisticians. The simulation is based on their requirements, experiences, and observations. ELIST models the intra-theater movements of personnel, equipment, and supplies by limited transportation assets over a constrained transportation infrastructure. The purpose of the system is to aid the planners in evaluating the potential success of a theater leg of a transportation plan, analyzing vehicle and facility requirements for a given plan, and identifying possible bottlenecks in infrastructure or process. Because users often require trade-offs between speed and accuracy, the model has been designed so that different degrees of detail may be applied to different portions of the scenario data. The experienced analyst thus has choices for focusing on specific aspects of the plan when time is limited. These detail/aggregation choices will be discussed in the following subsections.

Six modes of transportation are modeled in ELIST: road, including line haul operations, rail, water, air (fixed-wing), helicopter, and pipeline. (At this time air and pipeline are not yet implemented in ELIST8.) ELIST can model either the home theater or a foreign theater.

The main inputs to the model are:

- Unit data (the entities to be moved),
- Vehicle data,
- Transportation infrastructure data for the theater, and
- A set of model parameters.

Unit Data

The military units and their required activities originate from the Time Phased Force Deployment Data (TPFDD.) This format is used by the joint forces to represent deployment plans. Additional data were needed for ELIST as well as for other models, so the Expanded TPFDD (ETPFDD) was developed. The ETPFDD contains an operational hierarchy, as well as a compositional hierarchy, of unit data. Each unit may be composed of other units as well as quantities of commodities. ELIST is primarily concerned with the commodities, which are the entities to be moved through the simulation.

Commodities can be represented at multiple levels of aggregation. In Figure 1, for example, the unit ULN 3AG9C is composed of amounts of the following commodities: Personnel, Aircraft Tonnage, Oversized Tonnage, and Outsized Tonnage. At level 2, only aggregate weight is known (the numbers shown are measurement tons and short tons.) The Oversized Tonnage is also shown at the more detailed levels 3 and 4. At level 4, individual vehicles or containers are represented along with their weights, lengths, widths, heights, and areas.

The analyst can specify the level of detail to be modeled either by individual units or for the entire scenario. Aggregate unit components start out as single entities, but as they are transported through the simulation, they can be split into multiple smaller entities. Often the components are split so they can fit on transports. Discrete entities, on the other hand, must fit dimensionally onto their transports and cannot be divided.

The set of commodities used in a scenario is dynamic. Each commodity is classified in one of five commodity categories: PAX (personnel), POL (liquid fuel), RORO (Roll On Roll Off equipment), Breakbulk, and Container. The commodity categories are used throughout the model to determine how the cargo can be loaded on and off transport vehicles, how it should be stored, and various other activities. The actual commodity is used to determine possible modes of transportation and to select vehicle types.

In addition to unit compositions, initial unit locations and all subsequent movement requirements are included in the ETPFDD. The simulation is driven by the scenario units and their movement requirements. A movement requirement includes a destination node and a required delivery time. A mode of transportation may also be specified. Possible requirements have been expanded from those in the original

Figure 1. Sample Compositional Unit Hierarchy

ULN 3AG9C										
1 3AG9C PAX P951	Personnel	10								
2 3AG9C BLK D1632	Aircraft T	11.0	5.00							
2 3AG9C OVR D1633	Oversized	766.0	307.30							
3 3AG9C A2D D279	Oversized	766.0	307.30							
4 3AG9C A2D D595	Oversized	2	11.0	2.80	166	83	55	96	TRAILER CARGO	
4 3AG9C A2D D602	Oversized	2	29.5	8.30	265	95	81	175	TRUCK CARGO 2-	
4 3AG9C A2D D609	Oversized	2	4.0	0.50	109	62	44	46	TRAILER CARGO	
4 3AG9C A2D D616	Oversized	3	6.3	1.20	132	64	53	58	TRUCK UTILITY	
4 3AG9C A2D D623	Oversized	17	38.7	16.50	249	111	97	191	ARMORED RECON	
2 3AG9C OUT D1634	Outsized T	52.0	24.70							

TPFDD to include any number of intermediate or follow-on destinations, staging and other types of delays, marry up and assembly with other entities, and transitions into vehicles that in turn can be used as transportation assets.

A large set of units may be involved in activities all over the world, so ELIST allows the analyst to select the country or countries of interest. ELIST is an intra-theater model, and does not focus on strategic (overseas) deployment. Only those activities in the theater of interest will be modeled. If no other information is available, all units will start out as available and ready to move at their first location in the scenario. Alternately, another (external) model can be used to generate strategic arrivals of loaded ships and aircraft into the theater.

Vehicle Data

ELIST has data on every vehicle type that might be used. The model uses information such as payload, cargo area dimensions, curb weight, average speed, and average on-load and off-load times.

ELIST uses a vehicle grouping called an *asset* in order to specify preferences of how to move specific commodities. An asset is a set of vehicle types that will be used in the same ways in the plan. Examples are a set of similar heavy trucks or a set of tanker railcars. Commodity-asset rules list (in order of preference for each commodity) the assets that may be used to transport that commodity. Conditions based on lateness and distance may be included in these rules.

A specific collection of vehicles is defined for a scenario. There are multiple ways that the analyst may choose to represent vehicles in the simulation. First, a vehicle can be modeled as *fully tracked* or as a *capability asset*. A fully tracked vehicle is a discrete entity that makes loaded and empty trips and that always has a current location in the network. A capability asset, on the other hand, is a vehicle (or set of vehicles) considered to be generally available. Such a vehicle is only specifically modeled when it is being used to transport cargo. When its trip is finished, the vehicle designated as a capability asset goes back into the pool, and it can be used immediately in an entirely different location. No empty vehicle trips are modeled for a capability asset.

Second, the locations where vehicles can be used may be specified by setting up *asset pools*. Asset pools are collection of vehicles that serve specific locations within the theater network. An asset pool can also be set up to serve all locations. Line haul operations may be set up using asset pools with trailers that can serve only two nodes. Asset pools also have *home nodes* where all fully tracked vehicles congregate when they are idle.

Last, the analyst may specify time periods during which vehicles are available. By default, an asset is available throughout the scenario. However, specific intervals, initial times, or final times can be entered.

Infrastructure Data

A theater infrastructure is required for the area of operations and must include all the nodes (i.e., ports, staging areas, and destinations) referenced in the ETPFDD. Links — roads, railways, waterways, and pipelines — connect the nodes. All nodes have geographic coordinates for mapping and distance calculations. Infrastructure resources are modeled in four main ways within ELIST: as rate resources, gate resources, capacity resources, or discrete resources.

A rate resource is characterized by its ability to process a given amount of cargo in a given time; for example, the capability to load containers on railcars in containers/hour. The resource represents the men and equipment involved. The time consumed by an activity using a rate resource will be at least as long as the amount being processed divided by the rate. However, the activity may take longer than the time dictated by one resource, as in the case where multiple resources with different rate values are required.

A gate resource also represents a capability specified as a rate. However, this type of resource does not have any duration associated with its use. Road capacities, for example, are represented using gate resources; some number of vehicles may travel onto the road per hour.

A capacity resource is a nonconsumable resource that is used in some fraction of the whole for the duration of an activity. Its use, however, has no impact on the activity duration. An example is storage area at a port.

Some equipment is represented discretely. Cranes at a berth, for example, are acquired for a ship loading activity and released when the activity is complete.

The main node types modeled are seaports, airports, and intersections. The basic node, an intersection, can represent a staging area, a rail terminal, or any kind of intersection of links. At this type of node, trucks, railcars, and helicopters can be on- and off-loaded. Capabilities for loading cargo to and from vehicles are all modeled as rate resources. These capabilities are broken into three parts: infrastructure, MHE (Materials Handling Equipment), and personnel. This approach gives analysts flexibility in storing the static aspects of a terminal, such as ramp and dock capability, separately from more dynamic capabilities such as personnel and equipment. From these values, a rate resource is constructed for the simulation that has a capability that is the minimum of the three components. Attributes for POL, containers, breakbulk cargo, and RORO cargo loading for rail

Figure 2. Sample Unit History

30GB UTC:1E777 UIC:WAB7AA

Elem	Commodity	Cargo ID	Amount	Sq Ft	Time	Event	Location
3	Personnel	P2041	14.0	0.0	2 0:00	Available TOZEUR-NEFTA -TPAL-IAP	TOZEUR-NEFTA -TPAL-IAP
					2 0:00	Waiting for SAFAQIS DD Asset Buses	TOZEUR-NEFTA -TPAL-IAP
					2 0:00	Waiting for BIZERTE DD Asset Buses	TOZEUR-NEFTA -TPAL-IAP
					2 8:33	Road trip from TOZEUR-NEFTA -TPAL-IAP to UNKN TUNISIA -UNQU-RPA	TOZEUR-NEFTA -TPAL-IAP
					2 8:33	Road Onloading (Buses)	TOZEUR-NEFTA -TPAL-IAP
					2 9:03	Waiting for Serial for TOZEUR-NEFTA -TPAL-IAP to UNKN TUNISIA -UNQU-RPA	TOZEUR-NEFTA -TPAL-IAP
					2 9:03	Road Departure in Serial (Buses)	TOZEUR-NEFTA -TPAL-IAP
					2 18:36	Road Arrival (Buses)	UNKN TUNISIA -UNQU-RPA
					2 18:36	Road Enter Node (Buses)	UNKN TUNISIA -UNQU-RPA
					2 19:05	Road Offloading (Buses)	UNKN TUNISIA -UNQU-RPA
					2 19:20	Final Delivery	UNKN TUNISIA -UNQU-RPA
4	Oversized Ton...	D3306	207.3	2073.0	2 0:00	Available TOZEUR-NEFTA -TPAL-IAP	TOZEUR-NEFTA -TPAL-IAP
					2 0:00	Any mode trip from TOZEUR-NEFTA -TPAL-IAP to UNKN TUNISIA -UNQU-RPA	TOZEUR-NEFTA -TPAL-IAP
					2 0:00	Road Onloading (Commercial Trucks)	TOZEUR-NEFTA -TPAL-IAP
					2 1:59	Waiting for Serial for TOZEUR-NEFTA -TPAL-IAP to UNKN TUNISIA -UNQU-RPA	TOZEUR-NEFTA -TPAL-IAP
					2 1:59	Road Departure in Serial (Commercial Trucks)	TOZEUR-NEFTA -TPAL-IAP
					2 11:33	Road Arrival (Commercial Trucks)	UNKN TUNISIA -UNQU-RPA
					2 11:33	Road Enter Node (Commercial Trucks)	UNKN TUNISIA -UNQU-RPA
					2 12:02	Road Offloading (Commercial Trucks)	UNKN TUNISIA -UNQU-RPA
					2 12:02	Final Delivery	UNKN TUNISIA -UNQU-RPA

and road vehicles are all represented. Basic nodes also include resources for storage areas for different cargo types, as well as other resources.

Seaports have all the data associated with an intersection node, plus the data for ship berthing and loading. Berths are represented as separate entities with their own attributes and resources, which include physical dimensions, number and types of cranes at the berth, and RORO and POL load capabilities. Some resources, such as personnel, are considered to be port-wide, others are assigned to a single berth.

In order to collect or deliver cargo at a seaport, a ship must first be berthed at an appropriate length of dock. If needed, cranes are assigned to the ship. Offload takes place in fixed time intervals, with as much cargo off-loaded as possible in that time (e.g., 1 hour). In this way, aggregate cargo can be divided, off-loaded, and moved forward in a timely manner, rather than off-loaded in large quantities before moving forward. On-loading ships does not require this artificial parameter because the cargo waiting on the ship is not being held back — when the ship is fully loaded, it will embark.

Aircraft parking areas at Airports are explicitly modeled. Only a limited number of aircraft can be processed at a time because of tarmac area limitations. All aircraft loading resources can operate anywhere within the airport.

The primary attributes of infrastructure links that are used by the simulation are length, rate of march, and rate of entry. Limited link capacity is modeled by the rate at which vehicles can start onto the link (rate of entry.) Road and rail links can also be limited in the size and weight of vehicles that may traverse them.

Scenario Parameters

The analyst can specify a number of options and parameters to direct and tune the simulation; a few examples are given. The simulation can enforce the mode of travel specified in the ETPFDD, or it can select a mode on the basis of the situation. The maximum amount of time to wait for additional cargo and the percentage of capacity preferred prior to departure can be specified for various vehicle types. The analyst can decide whether the storage area at a node should constrain movement or the cargo should be allowed to simply overflow. The preferred number of railcars in a train can be input, and the analyst can decide whether trucks must travel in serials (groups of vehicles). Scenario parameters are stored in the database as are all other ELIST data.

SIMULATION

The simulation is driven by discrete events. Initially scheduled events are: unit availability, ship arrivals, and plane arrivals. These events all have participant lists that contain unit component entities. Whenever an activity is completed, the newly idle unit components look in their required activity lists and create and schedule new activities as appropriate.

Unit components keep detailed histories of each completed activity as well as time spent queuing, and this information is available to the analyst. Figure 2 shows sample histories of two components of ULN 30GB from the movements report. The first element, 14 personnel, became available to move on day 2 at 0:00 and waited in two queues (for two asset pools) for the asset Buses. At 8:33, a bus was acquired, a trip was scheduled, and on-loading began. At 9:03, the bus was ready to depart. It was required to travel in a serial, which was immediately available. The people arrived at their destination at 18:36 and off-loaded. They had

no other movement requirements. The bottom section shows the activities of 207.3 short tons of oversized tonnage.

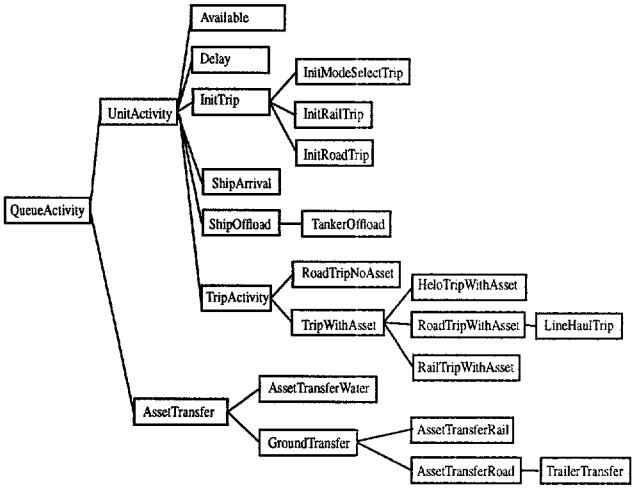
Activity Objects

Activities are implemented as objects in Java, each with an "execute" method. Some activities can be considered compound activities and are represented as a single object with multiple states. A trip with an asset, for example, actually consists of a fixed series of subactivities — cargo onload, travel, node entry or parking, and cargo offload — with the asset, route, and participant data remaining the same throughout. Different methods are executed on the basis of the state of the trip.

Planning a trip is also a scheduled activity. This activity consumes no time (except for queue time) but is required whenever travel occurs. Planning a trip occurs in a trip initialization event; this event's main function is acquiring transport vehicles. It also verifies that a route exists. If a movement does not have a required mode, and more than one mode is indeed possible for the movement, this activity selects the trip mode. If appropriate assets are immediately available, the highest preference asset/mode for the commodity type is selected. Otherwise, the activity waits until at least one vehicle is available, then commits to that mode. If no asset exists that can carry the cargo, or no route exists, the cargo will go into an error queue.

The subclass capability and interface construct of Java were found to be very effective in implementing the different activities of the simulation. This success is primarily because there are many similarities among the transport activities for different modes, different cargo types, trips with or without cargo, and trips with or without assets. Figure 3 shows the classification hierarchy of some activities to show the level of inheritance.

Figure 3. Partial Activity Hierarchy



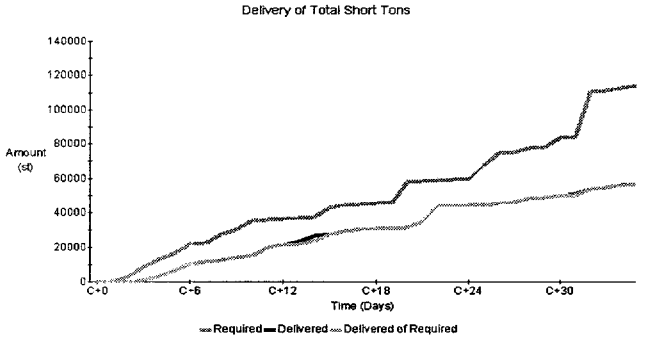
Event and Asset Managers

An event manager object handles the scheduling and execution of events. Events generate priorities on the basis of their participants' required delivery times. The simulation can be run to a specific time or run and stopped as desired. An asset manager object oversees the allocation of assets to activities. Vehicles are chosen on the basis of the commodity-asset preference rules, specific cargo dimensions, current location of the cargo and the vehicle (if it is fully tracked), and service area of the vehicle's asset pool. When an asset vehicle completes a trip, and there is a queue for the asset, it is either assigned immediately if it is needed at its current location, or the asset manager sends it to the location of the highest priority event in the queue. If there is no queue, the asset is sent to its home node.

RESULTS

Many reporting options are available to assist the analyst in evaluating the success of a plan, identifying bottlenecks, and analyzing asset usage. Plan success depends on the on-time arrival of all units at their final destinations. Summary arrival results can be viewed in total short tons for the entire scenario or by cargo type or destination. Figure 4 shows the total number of short tons required to be delivered to all destinations over time, the amount delivered in the simulation, and the short tons (of those delivered) that were also required at the time.

Figure 4. Total Delivery Graph

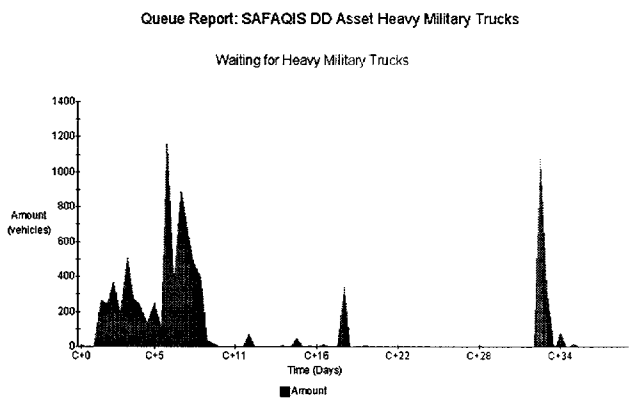


A query facility allows groups of units to be specified on the basis of name, location, delivery, and many other factors. These groups or individuals can then be reported and graphed. For example, all units traveling to a certain destination or all units not yet arrived could be reported.

Usage-over-time data are saved for all network and asset resources and all queues. The user interface allows easy selection (for reporting) from among all queues, destinations, seaports and airports, road links, routes, assets, asset pools, etc. Figure 5 shows a sample graph of the amount of cargo (in truck loads) waiting in the queue for Heavy Mili-

tary Trucks. It is apparent that, although there was a backup of cargo waiting for this type of truck, the backup was not persistent throughout the scenario. Note that some of the cargo waiting for Heavy Military Trucks may actually have been moved by other assets. Other graphs are available to view the actual usage of each asset.

Figure 5. Queue Graph



CONCLUSIONS

The ELIST8 simulation is a critical piece of an extensive system that includes import, export, visual editing, and error checking of vehicle data, asset and commodity data, network data, ETPFDD data, and scenario parameters. Mapping and query capabilities are included and continue to be enhanced. Simulation results can be saved to "projection" tables in the database. These tables are accessible to other tools to allow simulation and animation of an entire plan from U.S. bases to forward locations in a foreign theater. The capabilities provided by this expanding collection of tools will assist military logisticians with both their data management and analysis tasks. By providing greater ease and speed for modifying and analyzing the plans, these tools enable more "what if" scenarios to be examined and facilitate the logistical planning process.

The overall feeling of the team toward Java has been positive, but there have been problems. We have not found a development environment that satisfies everyone. The favorite seems to be VisualAge for Java (IBM). It has many useful features that facilitate development, a good visual debugger, and handles the scale of our system (over 680 classes not including third party). However, its method-based editors (as opposed to file-based) are not always desirable, especially when working with user interface classes. Also, VisualAge for Java is available on Windows NT but not Sun Solaris, our main development platform. The file-based Java environment Forte (Sun Microsystems) also works well on the PC. It is available for Solaris and Windows platforms, and it is free. We ran into numerous compile or path errors, and were unable to get our system running with JBuilder (free, Foundation version from Borland/Inprise.)

The immaturity of Java has led to time spent on "work-a-rounds," especially in relation to the Swing user interface classes. The Bug Parade at the Java Developer Connection site (http://developer.java.sun.com/index.html) has been beneficial in identifying Java problems and solutions, and the work-a-rounds have not consumed a significant portion of the development time.

Speed of execution continues to be a concern. It is easy to measure but difficult to quantify because of the large number of variables associated with the size of a scenario: the number of items being moved, the number of movements being made, how constrained the network and assets are, etc. However, as an illustration, a scenario with 717 units and approximately 140,000 short tons of men, supplies, and equipment, all moving to an average of one destination and using three modes (road, rail and water) was simulated in slightly more than 6 minutes on a Sun Ultra 10 workstation using JDK1.2. This is a relatively small scenario. Future work will involve thoroughly exploring the speed issue.

ACKNOWLEDGMENT

This work was supported under a military interdepartmental purchase request from the U.S. Department of Defense, Military Traffic Management Command Transportation Engineering Agency (MTMCTEA), through the U.S. Department of Energy contract W-31-109-ENG-109.

REFERENCE

Macal, C., C. Van Groningen, and M. Braun. 1995. "Simulation of Transportation Movements Over Constrained Infrastructure Networks." In *Proceedings of the 1995 Simulation Multi-Conference* (Phoenix, AZ, April). 27(4):97-102.