*Proceedings of the 1999 Winter Simulation Conference*
*P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.*

# A FLEXIBLE INTERFACE AND ARCHITECTURE FOR CONTAINER AND INTERMODAL FREIGHT SIMULATIONS

Beth C. Kulick
James T. Sawyer

Automation Associates, Inc.
512 Via de la Valle Suite 209
Solana Beach, CA 92024-2714, U.S.A.

## ABSTRACT

SIMCAP (Simulation-based Capacity Analysis Platform) is a simulation-based software tool designed to support analysis of intermodal terminal operation, specifically with respect to track capacity and yard capacity. The application was designed with a modular, extensible software architecture with an emphasis on capturing the primary complexities that exist in an actual intermodal terminal. SIMCAP is not just a simulation model, but a *modeling system* that comprises several interacting software components. Each component represents a functional aspect of facility operations allowing for future implementations of terminals with different equipment, layout, and procedures to be more readily adapted without major coding revisions.

The ability to analyze terminal designs with alternate track layouts, equipment configurations, and varying demand requirements was of paramount importance when designing and developing SIMCAP. The first implementation of the SIMCAP paradigm was developed for Burlington Northern Santa Fe railroad; it was successfully used to analyze capacity during a peak period at an existing intermodal terminal. It is planned to further use the tool to compare terminal performance as proposed infrastructure changes are made.

## 1 ARCHITECTURAL OVERVIEW

The primary objectives of any simulation effort often determine the complexity of the tools needed. For example, if the project goal were a quick, one-time analysis of a relatively well-defined system, a single stand-alone model might serve as a sufficient tool. Inputs could be implemented directly in the model itself, and output performance measures could be inferred by a simulation analyst studying the model's behavior and its low-level, unformatted output. Such an approach limits the usefulness of that model to the precise set of conditions pre-programmed into its structure. Any changes desired by the analyst therefore mandate changes in the model itself.

On the other hand, in projects where rapid answers to terminal infrastructure investment questions are required, the flexibility of the simulation effort is of primary concern. Key objectives include the reusability and adaptability of the modeling system for ongoing analysis as a terminal operates within a changing business environment and with different facility constraints. In addition, changes to input scenarios for analyzing terminal performance need to be accessible to a variety of users—many who have little or no expertise with simulation software. For these reasons, Automation Associates, Inc. (AAI) has developed a multi-component application framework, called the Simulation-Based Capacity Analysis Platform, or SIMCAP.

SIMCAP represents an entire "modeling system," of which the simulation model is just one component. The various components and their relationships are illustrated in Figure 1. The various components that support the simulation model each embody a particular set of functionality, including historical data import, train manifest generation, terminal scheduling, simulation, and animation.

For ease of maintenance and flexibility, the non-simulation model components are all implemented using the common programming framework of Microsoft Visual Basic for Applications (VBA). The implementation of the major supporting components within VBA not only allows for a powerful implementation using a variety of Microsoft products, particularly Microsoft Access, but it also allows for direct links to those objects that are exposed within Systems Modeling Corporation's Arena simulation software, version 3.5. The Visual Basic object library within Arena was used to incorporate significant flexibility into the model—especially for dynamically creating alternate terminal configurations that encompass major rail infrastructure changes.

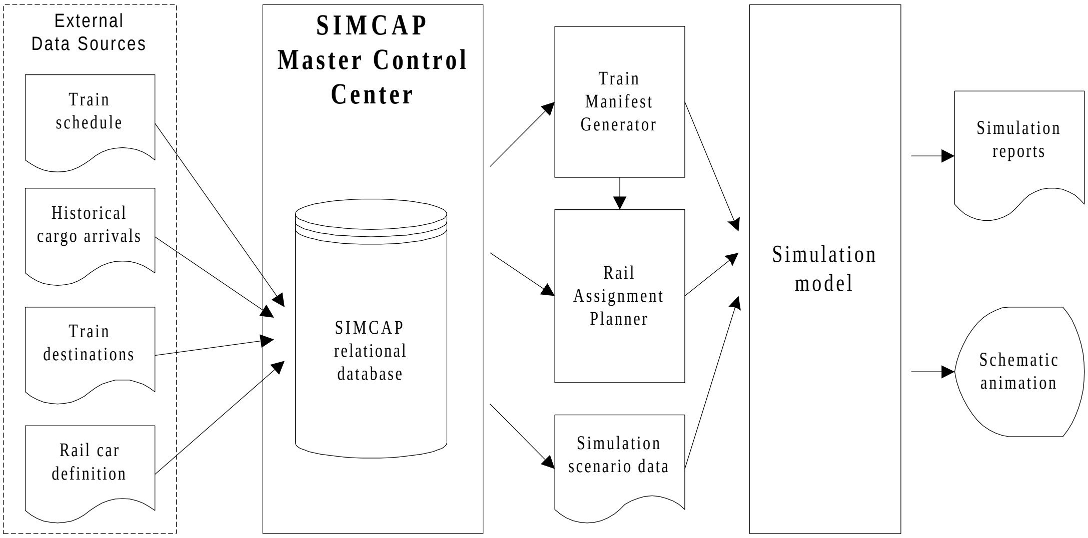


Figure 1: SIMCAP Architecture Overview

A key advantage to utilizing the VBA architecture within SIMCAP is its widespread availability in industry applications, not to mention the ease of use of its development environment. Readily available, off-the-shelf software products were used to provide the base functionality required within the modules of SIMCAP without reinventing or duplicating the significant development momentum provided by Microsoft and other software vendors.

## 2 DESCRIPTION OF SIMCAP COMPONENTS

The major components within SIMCAP are described briefly in the following sections.

### 2.1 SIMCAP Control Center

A master control center, implemented in Microsoft Access, that is used for defining input scenarios, historical data import and filtering, launching of the supporting Visual Basic components, initiating simulation runs, and viewing simulation output results.

### 2.2 SIMCAP Database

A relational database implemented in Access that is used for storing and organizing multiple information sources needed by the simulation model. The database structure provides an environment that allows for efficient data storage, enforces data integrity, and provides data validation. In addition, custom import routines allowed the utilization of corporate data sources outside of SIMCAP itself. Future enhancements to SIMCAP may attempt to make this connectivity more seamless.

### 2.3 Train Manifest Generator

An application developed in Visual Basic that creates detailed train container and trailer manifests based on user inputs and analysis of historical and other imported files. The manifests produced are model-readable and create task demands for the terminal resources in the model (strip tracks, switch engines, cranes, etc.). Even though the primary input to this program is historical data, it has been designed to generate new manifests when it is desired to test the ability of a terminal to respond to increases in volume of cargo movement traffic.

### 2.4 Train Schedule

A text file of a train schedule (arrivals and departures) that is imported from an actual terminal operational database. The SIMCAP database provides a schedule editor such that adjustments (adding of new trains, etc.) can be made after import. The train schedule is an input file to the train manifest generator and to the rail assignment planner.

### 2.5 Train Destinations

A text file of the terminal destinations within each train's route that is imported from the actual terminal operational database. This information is an input to the train manifest generator. This file is necessary for the train manifest generator to dynamically assign cargo to appropriate trains based on their specific destinations.

### 2.6 Historical Cargo Arrivals

A text file imported from the terminal operational database. The historical information includes a record of information for each cargo unit that arrives to the facility to be loaded on an outbound train. This includes time of arrival, priority, type (container/trailer), and destination terminal. This file is an input to the train manifest generator. It is used to construct a time-dependent arrival profile of outbound cargo arrivals such that the train manifest generator can determine how much cargo has arrived and is available for loading onto a particular train. A time-dependent profile is constructed to allow for the user to conveniently create scenarios that include volumetric increases of cargo movement.

### 2.7 Rail Assignment Planner

A Visual Basic application that uses a heuristic algorithm to determine which rail tracks (strip tracks) to assign for train loading and unloading activities. This planning routine roughly mimics the thought process that an actual "trainmaster" may use when allocating strip tracks to trains. The output of the train manifest generator is necessary to determine the quantity (length) of rail cars that need to be loaded and unloaded for each train. The simulation model directly uses the output of the rail assignment planner.

The algorithm has several objectives; a primary goal is to reuse vacated cars from previous inbound trains for subsequent loading of outbound trains.

### 2.8 Simulation Scenario Data

Model-readable scenario definition information in addition to that provided by the historical and other input files from the actual terminal operational database. This data is edited and organized within the SIMCAP database. Examples of entered data include terminal track and switch configurations, types and quantities of train loading equipment, yard storage configuration, business rules, etc. The model reads the input data from the Access database.

### 2.9 Simulation Model

A discrete-event model developed in Arena that represents the major functional areas within a single intermodal rail terminal. The model includes numerous logic modules implemented in Arena and/or embedded in VBA code. These include: 1) train action controller that produces requests for terminal resources in the appropriate sequence, 2) dynamic terminal rail network generation, 3) switch engine tasking, 4) switch engine routing through terminal network, and 5) container/trailer storage yard operations. For dynamic terminal rail network generation, the model uses Arena's VBA object library to automatically create the terminal rail layout and configuration from the user inputs.

### 2.10 Simulation Reports

Detailed event logs and other outputs that are written to the SIMCAP database directly from the simulation. The SIMCAP control center includes a report manager to create both tabular and graphical outputs. All reports have the same look and feel; the report manager is designed to be general such that different families of simulation outputs can be grouped independently, yet viewed in the same reporting format.

### 2.11 Animation

A real-time animation of terminal activities including the rail and the trailer/container storage yard. The dynamic terminal network generation routine within the simulation model uses Arena's VBA object library to automatically create the animation based on the entered rail layout.

## 3 BENEFITS OF A MODULAR APPROACH

The organization of SIMCAP into a set of components provides significant benefits especially as it is reapplied to new terminals where there are physical layout differences as well as variations in business and operating conditions.

### 3.1 Reusability for Multiple Projects

The current components are all designed to accept parametric inputs such that it is possible to reuse them for other facilities. Of particular interest is the ability for SIMCAP to automatically rebuild new physical track networks (in the simulation and animation) based on an input description of physical track locations and connections within the SIMCAP database.

### 3.2 Easy Replacement of Components

If a particular component is not applicable for a particular facility without the need for significant change, it can be replaced with a component that more suitably represents the objectives for that project. The SIMCAP master control center can be easily reconfigured to appropriate launch control to the new component as necessary.

### 3.3 Cost-Effective

The ability to add, replace, or modify single components rather than restructure the entire modeling system allows for adaptations of SIMCAP for other intermodal terminals (and potentially related types of facilities) to be built at a fraction of the cost of the original implementation.

### 3.4 Ability to Select Application Environment

The open architecture of the simulation model and its supporting modules allows for new components to be built using an application environment that best suits the required functionality. At the current time, all applications chosen for SIMCAP share common links with Visual Basic. This is not an absolute requirement for application selection, as custom routines could possibly be written to enable basic information exchange between non-VBA applications. However, an application environment that supports a Visual Basic object library can offer significant flexibility.

### 3.5 Distribution of Development Effort

The separation of SIMCAP into a set of components facilitates the separation of coding efforts to one or more programmers. This distribution of code across multiple components also helps in isolating problems during testing and analysis.

### 3.6 Scalability

The architectural design of SIMCAP makes it relatively easy to add and test new modules while minimizing the effect of existing code. The SIMCAP master control center is also extendible allowing for additional utilities and programs to be added and managed within a common user interface.

## 4 FUTURE ENHANCEMENTS

The first implementation of SIMCAP was directed towards representing the scope of a particular intermodal terminal of interest. This terminal encompasses most of the functionality shared by other intermodal facility operations, but there are enhancements and features that can be added to further increase usability and flexibility as it is reapplied. Some of the known enhancements that are to be added to SIMCAP are described as follows.

### 4.1 Additional Equipment Classes

The current version of SIMCAP allows for scenarios to be developed with a given set of equipment types. The current set includes switch engines for moving rail cars to and from strip tracks, rail mounted gantry cranes for loading and unloading rail cars, yard hostlers with chassis for moving containers, etc. Other terminals may use sideloaders for loading and unloading rail cars, hostlers with “bombcarts” for moving containers, or locomotives instead of switch engines to move rail cars, etc. It is planned to expand the simulation model logic such that it can include a larger range of different equipment types and options.

### 4.2 Development of a Graphical Terminal Layout Tool

The current version of SIMCAP uses tabular formats to provide information to the dynamic terminal rail network generator within the simulation model. This approach provides needed flexibility; however, it would be more convenient to define layouts using a graphical layout tool that could be automatically translated into the desired tabular format. A graphical application that supports Visual Basic is planned for implementing this capability within future versions.

### 4.3 Interactive Rail Planning

The existing rail assignment planner algorithm is executed at fixed time intervals which is currently each twelve-hour period, representing a typical trainmaster shift. From observing an actual trainmaster while performing rail planning activities, it was noted that he would start with a daily plan and then make adjustments to accommodate problem situations that occur throughout a typical day. Many of the problems that the trainmaster manages are beyond the planning level scope of this model; however, it is desired to also reinitiate rail planning activity when there are certain situations detected within the model. An example of a problem situation could be when planned loading or unloading occurs late due to terminal resource contention problems.

The simulation model can be designed to detect a when a given problem situation occurs and then reinitiate the rail planning routine for subsequent trains. This interactive approach would better mimic the trainmaster and his role of continually determining the best rail allocation strategy based on current facility conditions.

### 4.4 Connectivity to External Databases

The current design of SIMCAP relies on custom routines to import external data such as the train schedule from the actual terminal operational database. These data sources are currently in the form of text files, and as such, require additional programming outside of SIMCAP to extract the data from the operational database itself. As advances in data access technologies are becoming more available within VBA, we may begin to explore more direct methods of connecting to the operational database and acquiring the relevant data.

## 5 CONCLUSION

The modular approach used for constructing SIMCAP offers a great deal of flexibility that is difficult or impossible to achieve if a single application approach is used. A significant strength of this approach is the ability to select the application environment that best suits the objectives of the module as well as the needs of the users. Furthermore, Visual Basic provides a common link across all of the components within SIMCAP such that it is possible to extend functionality and facilitate ongoing use of the tool.

This scope and objectives of this project provided an opportunity to explore and apply new technology during implementation. Among these are the ability to utilize real-time communication between the simulation model and an Access database both before and during model execution. Additionally, portions of the model structure and animation were dynamically built from information provided within an Access database providing an automatic method of model construction that has never been previously possible with stand-alone models.

**ACKNOWLEDGEMENTS**

Special thanks to Kevin Bennett (AAI), Nora Ryan, and Kurt Reichelt for their contributions to the design and development of SIMCAP.

**AUTHOR BIOGRAPHIES**

**BETH C. KULICK** is Vice President of Automation Associates, Inc. She joined Automation Associates in 1988 as a Simulation Analyst and has created customized simulation models for a variety of industries. She received a M.S.I.E. from Texas A&M University and a B.S.M.E. from Purdue University. Her interests include creating advanced business analysis tools that are based upon simulation models.

**JAMES T. SAWYER** is a Simulation Analyst for Automation Associates, Inc. He joined AAI in 1996 and has over seven years experience in designing and developing simulation-based software solutions across a wide variety of application domains. James received his M.S. in Industrial and Systems Engineering from Georgia Tech in 1996, and his B.A.S. in 1) Mathematical/ Computational Sciences and 2) Music Technology from Stanford University in 1992.