Towards an Integrated Modeling and Simulation Framework for Freight Transportation in Metropolitan Areas

Qunzhi Zhou*, Amol Bakshi*, Viktor Prasanna*†, Ramakrishna Soma†
*Ming Hsieh Department of Electrical Engineering, University of Southern California
†Department of Computer Science, University of Southern California
{qunzhizh, amol, prasanna, rsoma}@usc.edu

Abstract

Freight transportation at distribution nodes such as marine ports, airports and rail yards has been putting tremendous environmental pressure in metropolitan areas. A prerequisite for proposing any solution that would make the existing systems more efficient is an accurate analysis and understanding of freight movements. A single model cannot fully capture aspects of freight transportation which interact and affect each other in a complex manner. Rather, integration of a variety of legacy simulation and analysis tools along with holistic optimization is a necessity for freight transportation system design. This paper proposes an integrated modeling and simulation framework for freight transportation using semantic web technology which offers benefits of modularity, extensibility and reusability of both code and design to the applications. We discuss the implementation strategies and methods to achieve these goals and identify some of the key research challenges in realizing our framework vision.

1. Introduction

Globalization, international trade and economic developments have led to an increase in the volume of freight flows by all transport modes. Increased freight transportation puts great pressure in metropolitan areas that have to deal with issues such as congestion on roadways and air and noise pollution. Freight movements inside and outside maritime or inland container terminals form connective flows with different behaviors due to the differences between supporting infrastructure. The state-of-the-art method for evaluating complex traffic scenarios is simulation modeling which affords the opportunity to evaluate traffic control and design strategies without committing expensive and time-consuming resources to implementing the alternative strategies in the field. Major efforts have been invested in the use of engineering and computer technology to model, simulate and predict the behavior of both terminal and out-terminal roadway traffic systems during the past few decades.

Terminal simulators model special infrastructure inside a container terminal – ports, yards, gates, logic of freight flow, etc. – in an attempt to simulate traffic flows inside terminals accurately. These include marine port simulators [1] and air port simulators [2]. Simulators that model traffic on roadways can be classified as macroscopic, mesoscopic or microscopic. Macroscopic models [3,4] tend to model traffic as a continuous flow, often using formulations based on hydrodynamic flow theories. Mesoscopic models [5,6] model individual vehicles, but at an aggregate level, usually by speed-density relationships and queuing theory approaches. Microscopic models [7,8] capture the behavior of individual vehicles and drivers in great detail, including interactions among vehicles, lane changing, and behavior at merge points. Increasing the modeling detail and hence the granularity of simulation improves accuracy of the simulation result but also increases the complexity of computation and sensitivity to modeling errors.

A holistic evaluation of the impact of different freight transportation policies requires a comprehensive model and simulator that includes terminals and roadway networks. For instance, a model of a marine port may be used to predict an increase in terminal throughput as a result of a change in cargo handling strategies or introduction of new technology. However, unless this evaluation includes a simulator that models the entire transportation chain, such an evaluation may easily ignore the fact that improved terminal throughput could cause an increase in congestion on the traffic network used by trucks to serve the terminal. More important, this increased congestion on the roadways could nullify most of the expected increase in terminal efficiency if trucks are not able to service the terminal at the desired rate. The problem becomes far more complex if one takes into account cost, safety, competition, legal, socioeconomic, environmental and other factors. Hence, to facilitate the introduction of new technologies and methods to improve

IEEE IRI 2008, July 13-15, 2008, Las Vegas, Nevada, USA

freight transportation efficiency and keep pace with rising demand, a framework for integrated modeling, simulation and analysis of the overall transportation system is needed. In this paper, we propose such a framework and discuss its architecture and technology components. The implementation of this framework for various use cases is in progress.

The rest of the paper is organized as follows. We review related work and challenges of integrated transportation simulation in Section 2. Section 3 describes sample use cases of our integration framework. In Section 4, we describe the architecture of the framework, examine individual components and present the key research and implementation issues in each of them. We conclude in Section 5.

2. Related work and challenges

Integrated use of traffic simulation packages and other software applications are among the new trends in solving engineering problems in traffic scenarios. Integrated simulation systems can be categorized as homogeneous or heterogeneous based on whether they integrate simulators of the same type (e.g., traffic simulators), or whether they integrate simulators of heterogeneous components (e.g., roadway networks and marine terminals). A few examples of homogeneous integration are MEZZO-VISSIM [9], Paramics-DYNASMART [10] and the VISSM-TerSim [1] test-bed. TraNs [11] is an example of heterogeneous integration system.

MEZZO-VISSIM simulates different areas of a traffic network at different levels of detail. It combines the strengths of meso simulation of large scale networks with less sensitivity to modeling errors and less calibration with those of micro simulation of interested areas in greater detail. The integration architecture consists of a new module outside the meso and micro modules that contains some components shared by all the tools: a database with information of the entire network graph, the travel behavior component with route choice models and path generation algorithms. Each time a simulated vehicle makes a route choice, the common module is consulted. Both simulators also update the network database regularly with link conditions in their sub-network.

Paramics-DYNASMART combines two simulators in an embedded structure. The primary objective of the integration is to provide a better route-choice capability for the micro simulation of Paramics by using the path dynamics of DYNASMART. The framework consists of several additional modules such as data communication and routing decision modules to transmit vehicle route decisions processed at the abstract network to the detailed network.

VISSIM-TerSim allows joint modeling and simulation of traffic in marine terminals and roadway networks. It consists of three modules: TermSim, VISSIM and TermCost. An external program is used to execute VISSIM COM commands to access its simulation data, allowing VISSIM to work as an automation server to export objects. TerSim employs a client program to collect data from VISSIM through the server interface and converts it into its inputs. Also TerSim writes its outputs as the corresponding input object of VISSIM. The TermCost is an offline cost model to analyze the simulation data.

TraNS, as a heterogeneous integration platform, combines the capabilities supported by a network simulator ns-2 and those supported by a traffic simulator SUMMO for realistic evaluations of VANETs (vehicular Ad-Hoc networks) applications. TraNS employs the Traffic Control Interface (TraCI) module for interlinking road traffic and network simulators. TraCi module translates the information exchange within VANET simulated by ns-2 into atomic mobility commands and feeds the signal to SUMMO to manipulate the mobility of individual vehicles.

The key difference between our effort and the ones mentioned before is that our framework is not targeted for a single problem and a single class of end users. In the above systems, a fixed set of tools are tightly coupled through data exchange modules and reconfiguration of the system for different simulation workflow needs extensive reimplementation. The framework for integrated transportation modeling and simulation proposed in this paper offers benefits of modularity, extensibility and reusability. Researchers developing different simulation tools, optimization and control algorithms, data analysis models, etc., would be able to insert components into the overall simulation environment with little effort and without affecting other modules. Therefore the system will be an evolving framework allowing additions, improvements, changes on a modular basis while maintaining the integrated nature. As end users, domain experts can define and simulate a variety of operational scenarios, and view the associated forecasts as an aid to decision making.

There are many challenges in realizing the above framework vision, some technical and some organizational. One of the key technical challenges is building a loosely coupled architecture which allows independent simulators, databases, real-time measurement and business components to communicate with each other during simulation time. Complex issues can arise such as data heterogeneity, tool interoperability, etc.

Inspired by a wealth of work on model/ontology based information and tool integration [12-13], we have adopted semantic web technologies to design the integrated simulation system. Modeling schemas are used to capture the freight transportation domain concepts in an ontological framework. Target applications are modeled by workflows of integrated tools. The ontology describes the semantics of the high level workflows and also provides a query model for legacy tools to communicate. To the best of our knowledge, our work is the first to use model/ontology technique to address integration problems for freight transportation modeling and simulation.

3. Use cases of the integration framework

In this section, we discuss illustrative examples of use case scenarios that justify our integrated modeling and simulation framework for freight transportation systems. In addition to highlighting the utility of the framework, the types of use cases we address also influence the software architecture and user experience design.

Our framework will allow a domain expert (e.g., a city planner or a port supervisor) to evaluate the system-wide impact of local changes – e.g., the impact of a change in container handling strategy at the maritime terminal on the peak congestion on a particular roadway segment. We are using model data from the Los Angeles metropolitan area that includes the Long Beach port and the roadway network that serves freight traffic to and from the port.

A number of relevant use cases can be defined in this setting. For instance, the port management could be interested in understanding how the port operations and the nearby roadway network will be affected if key pieces of infrastructure (e.g., loading cranes) cease to operate temporarily. In addition to understanding the impact of breakdown, there are other use cases that relate to proactive planning of port capacity and operational strategy to handle projected increase in the amount of freight handled by the port. For example, city planners will need to understand potential bottlenecks in the system as freight traffic continues to grow – will an increase in the amount of cargo moved by rail reduce the roadway congestion in the areas surrounding the port? Where should inland container storage facilities be located to maximize truck utilization and reduce the average time a truck has to wait in traffic and within the terminal? How many additional incoming shipments can the port handle before the terminal infrastructure becomes a limiting factor? If the freight transportation supply chain has to be de-bottlenecked, which section of the infrastructure should be overhauled first?

Clearly, no single model or simulator can capture the entire freight transportation system. Also, it is unreasonable to expect the end user to be proficient in the domain and also be an expert in using the entire set of simulators that need to be configured and executed in concert to answer questions such as those listed above. To allow users to rapidly and intuitively specify and evaluate the above scenarios, we focus on defining application-independent abstractions that capture the intent of a use case in terms of input and output. The scenarios specified through these abstractions are then translated into specific configurations and commands to integrated simulators.

The input data set for a typical simulation can be divided into two main categories: model information and control events. For the examples, model information may consist of parameters describing the attributes of terminal infrastructures, road networks, etc. Control events like the "crane failure" need to be mapped to the corresponding set of time/model parameter settings. The simulation outputs for a given scenario are values of evaluation functions of modeling variables, which could be the port throughput or the average traffic flow rate near the terminal. Graphs can be plotted based on the output data as per user's requirements. The configuration of our integrated modeling and simulation framework can therefore be divided into two steps: preparing model inputs, and defining control events and outputs.

Prepare model inputs: Users provide model information of a complex target system by composing a set of base models. For the crane failure use case, users can choose TerSim to model Long Beach port and VISSIM to model the Los Angeles roadway network. The base set of models for the port and the roadway network are 'published' into our framework to form a *model inventory*. Users can select the models from the *model inventory*, enabling the reuse of base models for composing a variety of system-wide models.

Define control events and outputs: End users configure a set of "what-if" simulation scenarios to explore the relationships between control events and simulation outputs. Each scenario is bound with a workflow or target application model that is composed using the model inventory. For the crane failure use case, users should be able to create a scenario control labeled "crane failure" and associate it with a set of time/model configuration information for the TerSim simulator. E.g., 100% capacity could indicate normal operation till, say, 10am, 30% capacity from 10am-5pm indicates crane failure, 150% capacity from 5pm-7pm indicates repair of the failed cranes and also extra capacity made available to handle the backlog. Users are able to create various such scenarios at high level to examine the input-output relationship and aid rapid decision making.

A significant advantage of the two step inventory-scenario development flow is the reduction in cost of application definition. If each scenario has to be constructed from scratch, the number of scenarios that can be defined and analyzed in a given time becomes significantly lesser compared to our approach where a bulk of the definition and source data already exists in the system. The inventory also acts as a single version of the "truth" and prevents the errors caused by data duplication and inconsistent modification.

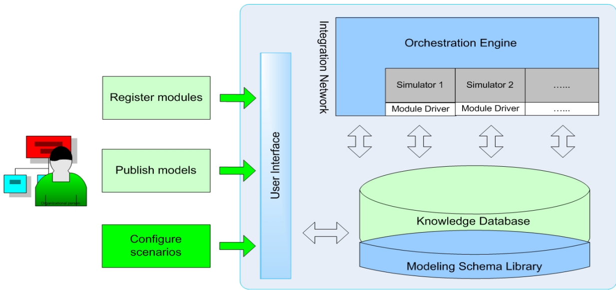

Figure 1. Architecture of the integration framework

4. Architecture of the integration framework

Our framework enables the integration of various simulation tools into a unified environment. Some of the key design objectives are a) Intuitive interface for creating applications in the context of decision making; b) Fine-grained integration of simulation runs conforming to user-defined workflows; c) Easy integration of new tools into the system, which does not require building of special components that facilitate point-to-point communications among them.

We adopt a model-based approach for designing our framework. In this approach, a modeling language representative of the freight transportation domain is used as a primary enabler of data and system integration. There are four major components in the framework as shown in Figure 1. In the following subsections, we describe each component in detail.

4.1 Modeling schema library

A modeling schema library contains schemas defining a common vocabulary across the system and forms the conceptual basis for integration. It contains a *domain model*, a *data model* and a *module model*.

The *domain model* captures the elements of traffic systems, their attributes and relationships. Some elements of domain model are shown in Figure 2. In general, we classify the domain objects into physical and nonphysical objects. Physical objects include traffic network, traffic flow, traffic units etc. Nonphysical objects include property descriptions like lane width, ship schedule, driving behavior etc.

The *data model* is a set of schemas providing the integration framework with various data models employed by legacy tools. It consists of three schemas which identify: a) the legacy data objects; b) the base system models and c) the data types that are used to define high level workflows.

The *module model* captures the functionality of legacy tools. It has the information necessary for the framework to control and coordinate the execution of legacy tools at simulation time such as the load/unload status, data objects and minimum execution time step etc.

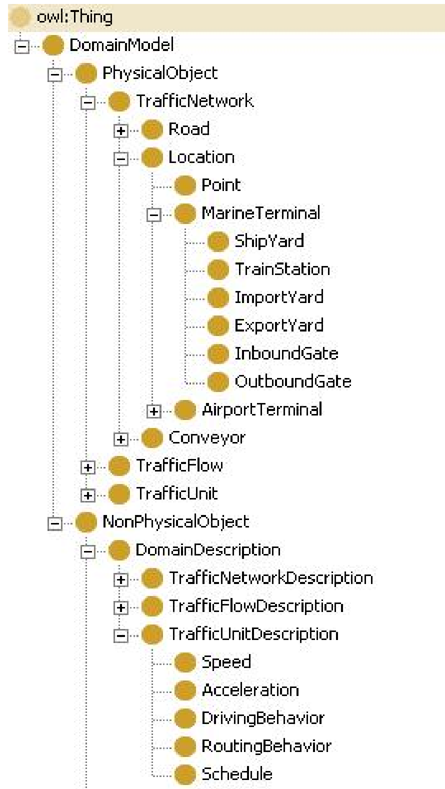

Figure 2. Domain model

Based on our previous experiences [12], semantic web technologies are particularly attractive for representing these models because they are built on open standards, provide the requisite level of expressiveness to capture the rich relationships in the application domain, have precisely defined semantics and are accessible through flexible APIs. Therefore we created the modeling paradigms in the form of an ontology definition using the OWL language and expect to continuously refine these models.

4.2 User interface

The user interface provides a high-level, domain oriented view for users to specify applications using system resources. In the architecture view, it links end users and the system's OWL data-store. It consists of three components: the *module register*, *model publisher* and *scenario manager* to support the use cases defined in section 3.

The *module register* provides facilities for users to manage legacy tools. Users insert a new tool into the system by providing a small set of configuration information which helps the framework to access and take proper control of it. The information conforms to the module model schema and is populated to the module catalog of the knowledge database (section 4.3).

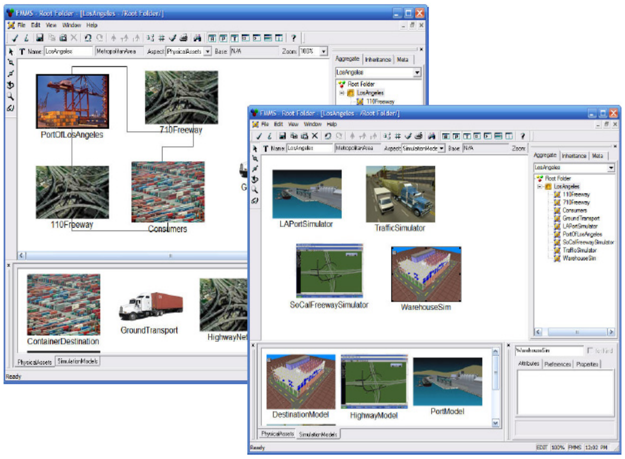

Figure 3. Workflow composer

The *model publisher* is used by domain experts to publish base simulation models. These base models can be retrieved later to define target applications or simulation workflows. As we known, it is not practical to provide a general interface for users to model complex target systems because some types of data objects defined by simulation tools may be only accessible through proprietary APIs which are usually tightly coupled to their visual interfaces. Instead, the publisher interface allows users to provide meta-information of base models which are developed by various legacy tools. The publisher stores the base models in the file system and indexes this location as meta-information in a *model inventory* (section 4.3).

The *scenario manager* provides an intuitive interface for "what-if" scenario configuration and design space exploration. It consists of two sub-components: the *workflow composer* and the *scenario editor*. The workflow models a target application by composing proper base models and also defines the execution flow of corresponding simulation tools. The *scenario editor* binds a workflow to a labeled scenario. It enables users to configure modeling parameters of the target system and explore the simulation input/output relationships in the context of decision making.

We developed the *workflow composer* UI using the Generic Modeling Environment (GME), which is a configurable graphical tool suite supporting model-based system design. Using the composer, users merely drag published base models and drop them into the definition palette to specify a simulation workflow. Figure 3 shows a simple workflow created in this environment, in which two roadway network models, two terminal models and one cost model are included. Additional user required interface modules such as the web service for remote application access are also expected to be implemented in the future.

4.3 Knowledge database

The knowledge database consists of three components including the *model inventory*, *module catalog* and *simulation data pool*. It uses paradigms defined in the modeling library as data schemas. In addition to store meta-information for base models and legacy tools, it also plays the key role for enabling an extensible integration architecture.

The *model inventory* provides universal addressing for the base models published by users. A target system can be modeled by referencing inventory elements in the workflow definition using the *scenario manager*. Inclusion by reference is valuable because any change made to some components of the model inventory can be instantly reflected in each scenario that contains that component. For example, if a highway network is redesigned and new segments are added or existing segments are removed, the change has to be made only in the inventory via the publisher interface but not in each scenario invoking that model.

The *module catalog* records meta-information of legacy tools which conforms to the module model schema (section 4.1).

The *simulation data pool* contains data produced by legacy tools during simulation. Traditional traffic simulation integration systems adopt a point-to-point data exchange method. Different from them, our framework enables the shared-network communication between legacy tools using the common database as a data exchange medium. Simulation modules only read from and write to the common data pool, integrating new tools does not affect neighboring modules. The indirect coupling of disparate tools hence leads to a modular, scalable and extensible integration framework.

We used the OWL data schema to prototype the knowledge database and the Jena API, SPARQL querying language to interface with the OWL data store. The OWL ontology data model can naturally capture the relationships between heterogeneous data objects in the integrated system and provides rich querying and inference capabilities which facilitates high level designs.

4.4 Integration network

The integration network manages the fine-grained execution of the workflow. It consists of two components: the *module driver*s and the *orchestration engine*.

Each legacy tool is registered with a *module driver* which enables the legacy tools to be accessible and controllable by the framework. The *module driver*s provide data channels between legacy tools and the database and control channels between tools and the orchestration engine. Implementations of *module driver*s are tool-dependent. For example, VISSIM driver should maintain a VISSIM COM interface at the tool side and a standard web service interface at the framework side.

The *orchestration engine* is the component that executes the simulation workflow. At runtime, it schedules and controls the execution of underlying simulation tools. In our framework, the high level workflows defined using the GME interface are translated into lower level, executable workflows. Various workflow paradigms like BPEL, XScufl, Windows WF have been proposed and successfully used in workflow applications. Many characteristics of the Windows WF make it an attractive choice to be used in our system. These include the ability to embed workflow engine/runtime in a host application and to control and monitor the workflow status makes it possible to write advanced workflow management environment which builds on the basic functionality framework. Finally it provides a declarative representation of workflows through a XML based representation. This feature makes it possible to generate workflows corresponding to the high level descriptions specified by users and simplify the procedure of translate our user defined workflows in GME into the work flows that can be executed by the Windows WF.

5. Discussion and conclusions

In this paper, we have highlighted the problems addressed in an integrated transportation modeling and simulation framework, and proposed its architecture. The architecture consists of four main components, the modeling schema library, the user interface, the knowledge database and the integration network. The modeling schema library contains the domain model, data model and module model, which provides a common vocabulary and conceptual basis for integration. The common database provides model addressing and indirect data exchange mechanisms to reduce the complexity of passing heterogeneous data among legacy tools. The user interface provides a domain oriented view for specifying high level simulation workflows which are then mapped into executable workflows. The workflow orchestration engine component of the integration network is responsible for actually controlling the legacy tools that constitute a workflow. Finally module drivers are responsible for extracting data from legacy applications and converting the data formats to be compatible with the system's data store. We have highlighted the main ideas, useful techniques, research challenges and the implementation status for each of these components.

The key requirements of the framework that we identified were a) fine-grained simulation integration, b) Usability and c) Extensibility. We looked at the components to support integrated simulation, including components to capture, manage and execute workflows. The user interface layer makes the system usable by providing a high level view of applications while hiding the low level implementation details of integration. Extensibility is achieved due to the modular, non point-to-point communication architecture of the system.

6. Acknowledgement

The work described in this paper is supported by the Metrans Center at the University of Southern California, Los Angeles, and its funding agencies, the US Department of Transportation and California Department of Transportation (Caltrans). The authors are grateful to Professor Petros Ioannou, Yun Wang and Hwan Chang (USC) for domain guidance, helping us identify relevant use cases for the framework, and sharing a wealth of information about VISSIM and TerSim simulators.

References

[1] Ioannou P., Chassiakos A., Valencia G. and Hwan C., Simulation test-bed and evaluation of truck movement concepts on terminal efficiency and traffic flow, *Metrans Project 05-11 Final Report*, 2007.

[2] http://www.atac.com/Products_Airports-a.html

[3] Buisson C., Lebaque J.P. and Lessort J.B., A discretized macroscopic model of vehicular traffic flow in complex networks based on the Godunov scheme, *Proceedings of CESE'96*, 1996.

[4] Elloumi N., Haj-Salem H. et al, METACOR: A macroscopic modeling tool for Urban Corridor, *Proceedings of Triennial Symposium on Transport Analysis*, 1994.

[5] Jayakrishnan, R., Mahmassani H.S. and Hu T.Y., An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research C*, 1994.

[6] Ben-Akiva, M., et al., Real-time simulation of traffic demand-supply interactions within DynaMIT, *Transportation and network analysis: current trends*, 2002.

[7] http://www.english.ptv.de/cgi-bin/traffic/traf_vissim.pl

[8] http://mctrans.ce.ufl.edu/featured/tsis

[9] Burghout W., Wahlstedt J., Hybrid traffic simulation with adaptive signal control, *Transportation Research Record*, 2007.

[10] Sahraoui A., Jayakrishnan R., Microscopic-macroscopic models systems integration: a simulation case study for ATMIS, *Simulation*, 2005.

[11] Piorkowski M., Raya M. and Lugo A.L., et al, TraNS: joint traffic and network simulator for VANETs, *Proceedings of the 13th ACM MobiCom*, 2007.

[12] Ramakrishna S., Bakshi A., Prasanna V.K., A Semantic Framework for Integrated Asset Management in Smart Oilfields, *Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid*, 2007.

[13] Wache H., Vogele T., Visser U., et al, Ontology-based integration of information: a survey of existing approaches, *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*, 2001.