Towards Modeling and Predictive Analysis of Emergent Behavior of Intelligent Transportation System of Systems

Kendra Cooper[a], Lawrence Chung[a], Ben Amaba[b], Scott Darlington[c], Laurel Dickson-Bull[d]

[a]Eric Johnson School of Engineering and Computer Science
The University of Texas at Dallas
Richardson, TX, U.S.A.

[b]Worldwide Executives
IBM Complex Systems - Rational
Miami, FL, U.S.A.

[c]Rational software
IBM Software Group
Ottawa, ON, Canada

[d]Intelligent Transportation Solution
IBM Rational
Littleton, MA, U.S.A.

ABSTRACT

Transportation systems are at the foundation of economies – cost effectively, safely, and sustainably moving people, goods, and ideas locally, regionally, nationally and globally. The current U.S.A. national transportation system is composed of land, water, and aerospace services. The individual systems supporting these services are large scale and complex, involving numerous stakeholders (e.g., citizens, planning, operations, maintenance, manufacturers/developers, regulators, public safety) and technical standards (e.g., Traffic Management Data Dictionary, National Transportation Communications for ITS Protocol). The demands on these systems are growing rapidly; to support these impending needs a vision for the future of transportation has been proposed: an integrated, international, intermodal, inclusive, intelligent, and innovative system of systems (SoS), with multiple interfaces and shared information and infrastructure elements. In this collaborative research project, we propose to define a standards-based approach that supports their modeling, simulation, and predictive analysis, with particular focus on the emergent behavior of the SoS. We view the emergent behavior problem from a component-based systems engineering perspective: how to represent, select, compose, and analyze the individual systems and the overall composition of the SoS. The research is being conducted in the context of an Intelligent Transportation Simulation Game, to make it feasible for the academic environment and appealing to students. Here, we present our preliminary work to establish a well-structured prototype SoS for the simulation game, which will be used to explore the research questions with a concrete, example system. The prototype SoS currently includes the game system and four small, independent transportation systems (rail, car, bus, pedestrian), providing multiple routes for a commuter to travel to work. Although simple, the prototype provides a foundation to build upon in the next steps of the project.

Keywords: System of Systems, Modeling, Simulation, Predictive Analysis, Intelligent Transportation

Computing, Software, and Systems Engineering (2018)

https://openaccess.cms-conferences.org/#/publications/book/978-1-4951-2096-1

INTRODUCTION

Transportation systems are at the foundation of economies – cost effectively, safely, and sustainably moving people, goods, and ideas locally, regionally, nationally and globally. The current U.S.A. national transportation system is composed of land (e.g., pedestrian, highways, transit, motor carriers, rail, pipelines), water (e.g., ports, waterways, pipelines), and aerospace (e.g., aviation; commercial, civil and military space) services. The individual systems supporting these services are large scale and complex, involving numerous stakeholders (e.g., citizens, planning, operations, maintenance, manufacturers/developers, regulators, public safety) and technical standards (e.g., Traffic Management Data Dictionary, National Transportation Communications for ITS Protocol).

The demands on these systems are growing rapidly; to support these impending needs a vision for the future of transportation has been proposed: an integrated, international, intermodal, inclusive, intelligent, and innovative system of systems (SoS), with multiple interfaces and shared information and infrastructure elements (FTAG, 2001). A SoS has characteristics that distinguish it from very large, complex, monolithic systems (Maier, 1996), including:

- Operational Independence of the Elements.
- Managerial Independence of the Elements.
- Evolutionary Development.
- Emergent Behavior.
- Geographic Distribution.

In this collaborative research project, our goal is to define a standards-based approach that supports the modeling, simulation, and predictive analysis, with particular focus on the emergent behavior, of the SoS. We view the emergent behavior problem from a component-based systems engineering perspective: how to represent, identify, select, compose, and retire the individual systems and analyze the overall composition of the SoS with respect to its emergent behavior. The research is being conducted in the context of a Transportation Simulation Game, to make it feasible for the academic environment and appealing to students.

Our preliminary work has focused on establishing a prototype SoS for the simulation game to explore the research questions with a concrete, example system. Our example system is a distributed, web-based simulation game for an intelligent transportation system. Key feature of this application are the modularization of independent systems for various transportation options within a city. The prototype SoS currently includes the game system and four small, independent transportation systems (rail, car, bus, pedestrian), providing multiple routes for a commuter to travel to work. The routes take different amounts of time, money, and vary in the environmental impact. Players can challenge another player, spending and earning coins and awards for their choices.

The remainder of this paper is organized as follows. The results of the preliminary prototype development effort are presented in the next section, followed by a discussion of related work and conclusions.

INTELLIGENT TRANSPORTATION SYSTEM OF SYSTEMS SIMULATION GAME: PROTOTYPE DEVELOPMENT

As part of a long term project, the prototype development work uses a systematic, model-based approach which includes requirements models (goals, system requirements), design models (architecture, component level), implementation, and testing. To represent the models in our project, we use a Vision Document (IBM 2014) from the Rational Unified Process (Krutchen, 2003) that we have tailored for SoS to capture the high level goals, and SysML (OMG, 2012) to capture the system requirements and design models. The requirements and architectural concept of a preliminary prototype SoS have been proposed; a development team has implemented the SoS using java and standard libraries as a graduate level capstone course project. The focus of the results in this paper is on the architecture concept and its realization.

The architectural concept includes a middleware communication infrastructure (CORBA) and transportation systems; each has a SoS adapter to support the composition of the independent, heterogeneous systems. The requirements and architectural concept need to be refined in the next steps of the research to consider the representation and dynamic, automated selection, composition, and analysis of the emergent behavior of the SoS (functional, non-functional).

System of Systems Architecture Concept

The SoS architecture concept provides a component-based solution, in which the components are independent, distributed systems that communicate using a standard CORBA middleware infrastructure (Fig. 1). For our intelligent transportation simulation game, the independent systems may include, for example, helicopter, rail, bus, road, and ferry transportation systems. The simulation game is also an independent system that can run without any other systems present; it has a rudimentary knowledge of alternative transportation routes (cost, starting location, ending location, time to travel, and so on). To enhance the game play experience, additional systems may be added to the SoS; here the emergent behavior is the additional route options made available for the player to choose from. This architecture concept provides a well-modularized, flexible solution that can adapt to changes, such as the launch of a new system or removal of an existing system.

In order to support the identification, selection, composition, and retirement of systems in the SoS, a SoS Adapter component is proposed in the architecture. In order to select a "good" system or collection of systems to include or retire the capability to provide predictive analysis (what happens if system X is added? X and Y are added? System W is removed? System W is modified?) is needed.

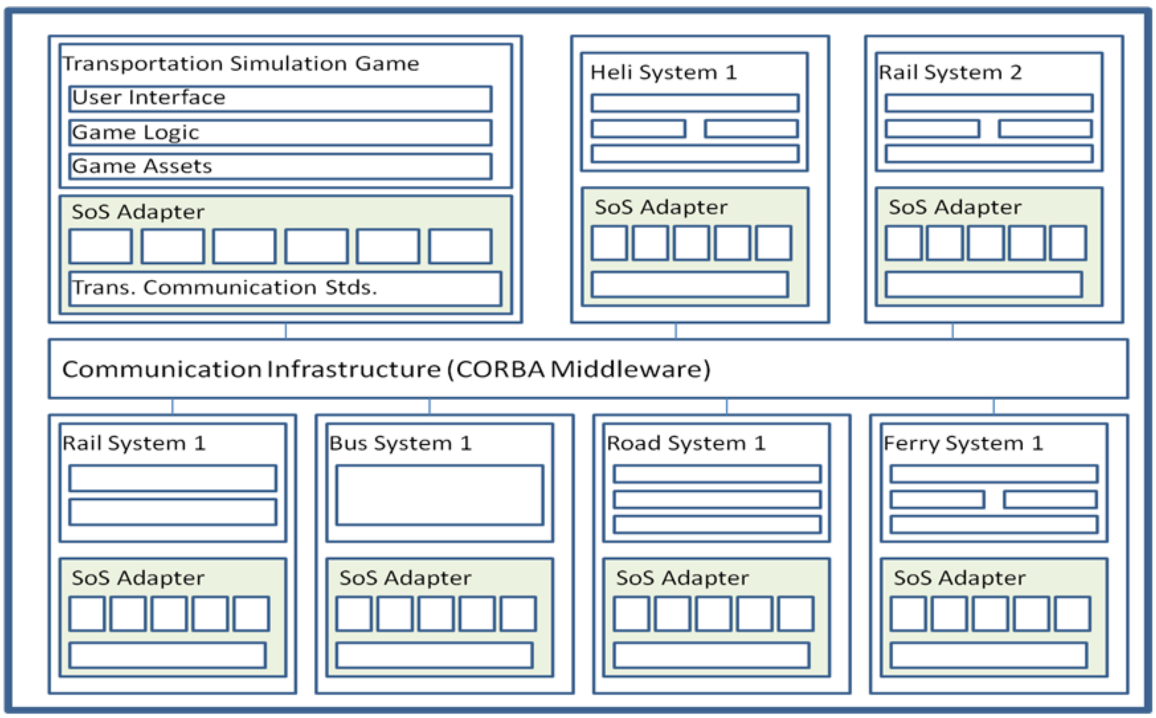
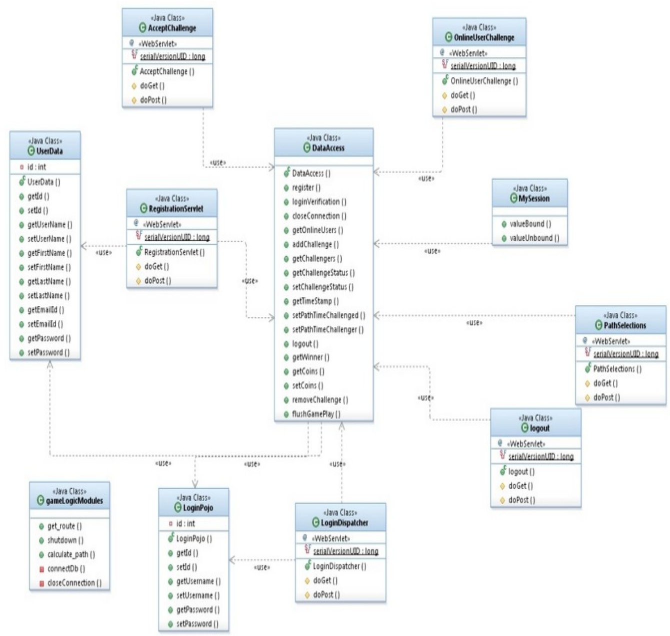


Figure 1. System of Systems Architecture Concept

The Realization of the Architecture Concept

The architectural components were refined into component level designs, represented in static and dynamic views with class diagrams and sequence diagrams respectively; the data in the systems were modeled using entity-relationship diagrams. The designs were implemented using well-established technologies (e.g., Java, MySQL); the system was manually tested using a collection of 26 test cases.

A sample class diagram representing part of the Transportation Simulation Game and a sample of the table structures for this system are presented in Fig. 2. The tables store data for the Players (identifier, e-mail, first name, last name) and the Game Play (Challenges, coin rewards, level achieved).



```
mysql> use gamedb;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_gamedb  |
+-------------------+
| gameplay          |
| online_users      |
| user              |
| user_credentials  |
| user_gameaccount  |
+-------------------+
5 rows in set (0.03 sec)

mysql> desc user;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int(11)     | NO   | PRI | NULL    | auto_increment |
| email     | varchar(50) | YES  |     | NULL    |                |
| firstname | varchar(20) | YES  |     | NULL    |                |
| lastname  | varchar(20) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
4 rows in set (0.16 sec)

mysql> desc user_credentials;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| id       | int(11)     | YES  | MUL | NULL    |       |
| username | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql> desc user_gameaccount;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| id    | int(11) | NO   | PRI | 0       |       |
| coins | int(11) | YES  |     | NULL    |       |
| level | int(11) | YES  |     | NULL    |       |
+-------+---------+------+-----+---------+-------+
3 rows in set (0.04 sec)

mysql> desc gameplay;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| challenger          | varchar(20) | NO   | PRI |         |       |
| challenged          | varchar(20) | NO   | PRI |         |       |
| accepted            | tinyint(1)  | YES  |     | NULL    |       |
| challengeTime       | datetime    | YES  |     | NULL    |       |
| PathTimeChallenger  | int(11)     | YES  |     | NULL    |       |
| PathTimeChallenged  | int(11)     | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

Figure 2 Sample Class Diagram and Table Structures for the Transportation Simulation Game System (Dharmadhikari 2013)

Screenshots of selected points in the game play are presented in Fig. 3 (Player Path Selection) and Fig. 4 (Player Winner Declaration). Each screenshot shows the players' web-based interfaces on the left and right side of the screen; the windows at the bottom of the screen are the individual transportation systems executing.

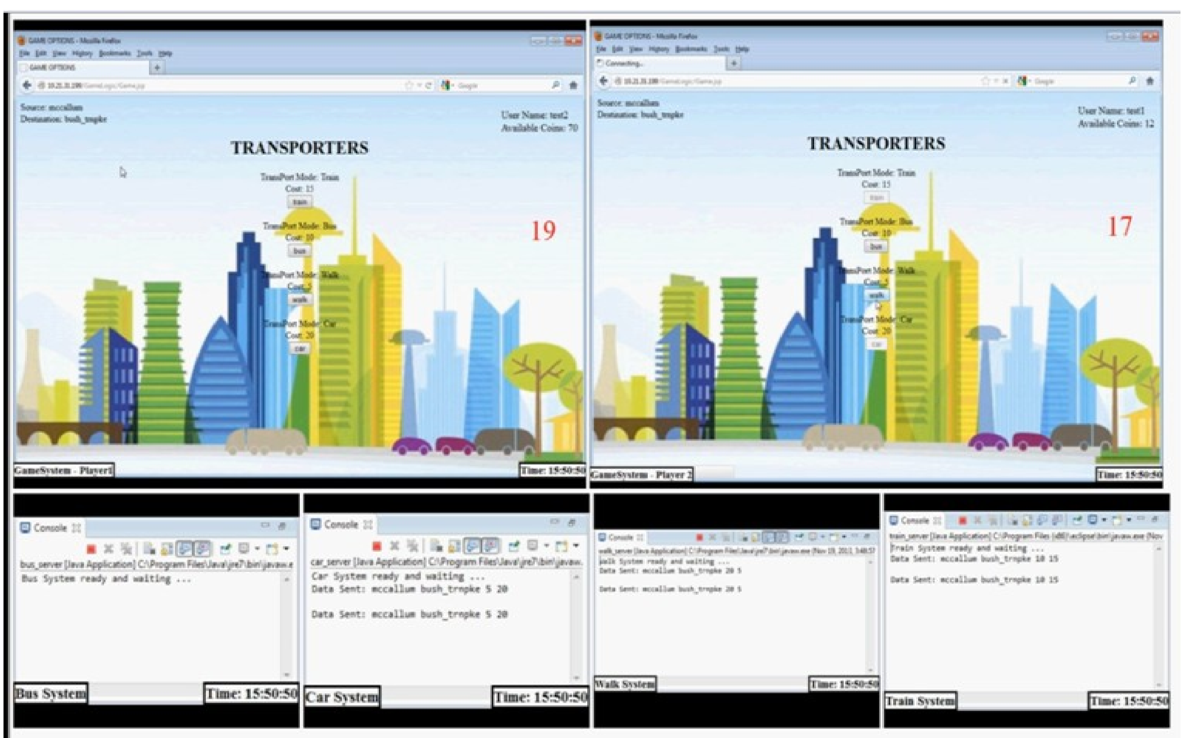


Figure 3. Player Path Selection (Dharmadhikari 2013)

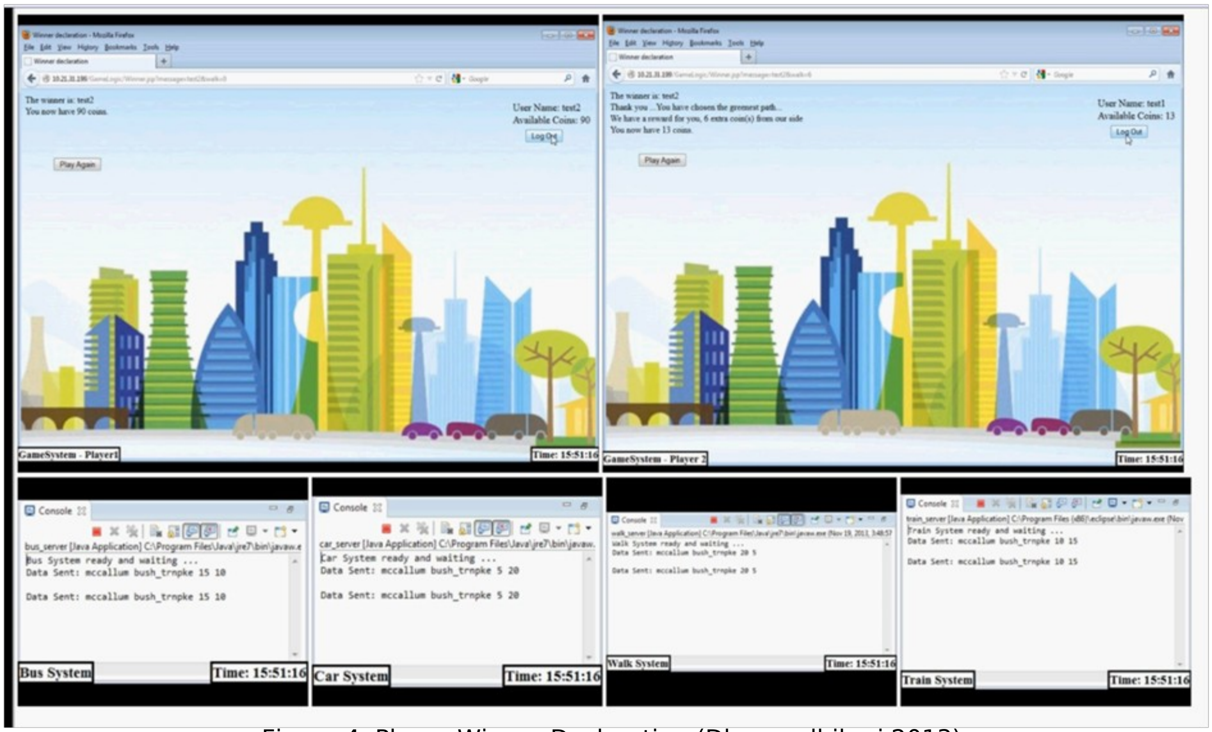


Figure 4. Player Winner Declaration (Dharmadhikari 2013)

The prototype provides a well-structured solution that can be extended. A key limitation of this work is the SoS Adapter component. Currently, it is very simple; exploring alternative solutions for this problem is the focus of the next step in our research.

RELATED WORK

The DEVS is a comprehensive, broadly applicable approach for modeling, simulating and testing SoS (Hu 2003) (Mittal, 2006) (Mittal, 2011) (Mittal, 2012) (Mittal, 2008). A key goal of DEVS research is to facilitate the integration of independent systems in a manner that helps to cope with interoperability problems. In other words, the modeling, simulation and testing assist in ensuring the right data are available for an individual system at the right time. The modeling, simulation, and test framework provides:

- Automated DEVS Model Generation from various standardized requirement specification formats (e.g., Business Process Modeling Notation, Business Process Execution Language, Statecharts).
- Collaborative model development using DEVS Modeling Language.
- Automated Generation of Test-suite from DEVS simulation model.
- Net-centric execution of model and test-suite over a service-oriented architecture (XML, middleware infrastructure).

The DEVS approach uses the concept of an experimental frame. A frame can be viewed as a system that interacts with the system of interest to obtain and analyze the data of interest under specified conditions. The frame is characterized by its implementation as a measurement system or observer; it typically has three types of components: a generator that generates input segments to the system; an acceptor that monitors an experiment to see the desired experimental conditions are met; and a transducer that observes and analyzes the system output. For example, a frame responsible for job processing performance metrics such as round trip time and throughput may have the following components:

- A generator that produces service request messages at a given rate. The time that has elapsed between sending of a request and its return from a server is the round trip time.

- A transducer that acquires the departures and arrivals of requests allowing it to compute the average round trip time and other related statistics, as well as the throughput and unsatisfied (or lost) requests.
- An acceptor that determines whether the performance achieves the developer's objectives, for example, whether the throughput meets the desired level or a specified percentage of the round trip times are below a threshold.

The DEVS approach adopts methods for transforming objectives of a system (goals) into experimental frames from the literature. Experimental frames translate the objectives into more precise experimentation conditions for the source system or its models; a model is expected to be valid for the system in each such frame.

To the best of our knowledge, however, the DEVS approach does not address predicting the emergent behavior of multiple candidate "to-be" solutions and their ranking, in particular to the context (environment) of the SoS.

CONCLUSIONS

Our preliminary work has focused on establishing a prototype SoS for the simulation game to explore the research questions with a concrete, example system. Our example system is a distributed, web-based simulation game for an intelligent transportation system. It appears to have the potential to provide a flexible, well-structured solution, however, much work remains to demonstrate this. Our next step in the research is to explore alternative techniques for the SoS Adapter component. There is a strong body of literature from the component-based software engineering community that can be drawn upon, where rigorously defined and verified approaches to identify, select, compose, and retire smaller components (i.e., not independent systems) have been proposed.

The students in the graduate level Advanced Software Engineering Project course who selected this project to work on reported that it provided an excellent opportunity to enhance their technical skills on system of systems engineering, middleware communication standards, web technologies and protocols, thin client development, and the MVC architectural style.

ACKNOWLEDGEMENTS

We would like to acknowledge the UT-Dallas, Fall 2013, CS 6387 Advanced Software Engineering Project team Shirin Dharmadhikari, Nirav Nagda, Pratik Raichura, and Jayant Sonrexa, for their contributions in creating the simulation game prototype. A technical report capturing their results in detail is in progress.

REFERENCES

Dharmadhikari, S., Nagda, N., Raichura, P. and Sonrexa, J., CS 6387 Fall 2013 Final Project Report, "Intelligent Transportation Simulation Game System of Systems Engineering", The University of Texas at Dallas.

Federal Transportation Advisory Group (FTAG), "Vision 2050 – An Integrated National Transportation System", February 2001, available at: http://scitech.dot.gov.

Hu,X., Zeigler,B.P., Mittal, S. "Dynamic Configuration in DEVS Component-based Modeling and Simulation", SIMULATION: Transactions of the Society of Modeling and Simulation International, November 2003.

IBM Rational Method Composer, available at http://www-03.ibm.com/software/products/en/rmc, accessed Feb. 28, 2014.

Krutchten, P., The Rational Unified Process An Introduction, Prentice Hall, U.S.A, 2003.

Maier, M., "Architecting Principles for Systems-of-Systems", Proceeding of the 6th Annual INCOSE Symposium, p. 567-574, 1996.

Mittal, S. "Extending DoDAF to Allow DEVS-Based Modeling and Simulation", Special issue on DoDAF, Journal of Defense Modeling and Simulation JDMS, April 2006, Vol 3, No. 2, pp.95-123.

Mittal, S. and Douglass, S., "Net-centric act-R-based cognitive architecture with DEVS unified process", SpringSim (TMS-DEVS) 2011: 34-44.

Mittal,S. and Douglass, S., "DEVSML 2.0: the language and the stack", SpringSim (TMS-DEVS) 2012: 17.

Mittal, S., Zeigler, B. Martin, J., Sahin, F., Jamshidi, M. System of Systems - Innovation for the 21st Century, Chapter 5, Wiley,Inc. 2008.

Object Management Group (OMG), Version 1.3 of SysML specification standard, 2012, available at: http://www.omg.org/spec/SysML/1.3.