Architecture for the Modeling and Analysis of Rapid Transit

A-MART

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

SYST 798/OR 680

Table of Contents for Main Section

Executive Summary

Mass Transit is a topic that impacts millions of people across the world and in the US every single day. From buses and aircraft to light rail and metro transit systems, commuters, tourists and everyday people depend on it. While these systems may seem quite different at first glance, they all in fact share the same goal, to provide a safe, reliable, efficient and economic means to transport passengers en mass. Individual Rapid Transit Systems are often developed in a stove-piped manner, which often limits future options for development, upgrade and expansion. It also impairs the ability to share best-practices and innovations which could potentially benefit the entire Rapid Transit domain. It is with this theme in mind that the Architecture for the Modeling and Analysis of Rapid Transit Team (AMART) was implemented.

Mass transit systems share many operational attributes and therefore a common architecture for their automation is not only feasible but also very useful. There are over 160 urban mass transit systems in existence, and more under construction [6]. Additionally, such systems are continually being expanded and overhauled to enhance rider capacity, performance, or efficiency of the network. This represents a much broader and larger market for such an architectural paradigm as opposed to developing a single architecture for a specific rapid transit system. Developing this paradigm would allow AMART to model and analyze any rapid transit system and focus in on numerous problems plaguing current systems such as crowding, reliability, safety, control systems, energy consumption and optimization.

This project involved the development of an architectural paradigm which can be applied to a variety of existing or planned railway-based rapid transit systems, such as the DC Metro, to model and analyze system performance. The resulting analysis can be used to illustrate how the existing or proposed underlying systems would likely perform and provide recommendations on how to improve the operational efficiency of the mass transit system.

This project encompasses numerous systems engineering and operations research concepts and is a system of systems approach in several ways. The architectural paradigm represents an overlaid system building from existing systems abstracted to the common rapid transit model at various levels. As opposed to developing an entire working solution for a specific rapid transit system, this project strove to explore an extensible solution with a more generic design space of the problem common to all rapid transit systems and determine a flexible, marketable solution which can work with as many rapid transit systems as possible.

Our company has developed a flexible architecture paradigm with an executable architecture that we can apply to any mass transit system throughout the world to analyze causes of non-desirable behavior and recommend changes to infrastructure and operations to optimize operational efficiency across systems, or system components. Our experts can easily tailor our paradigm and model to any customer's needs and conduct thorough and meaningful analysis resulting in actionable recommendations for improving performance.

1. Introduction

The Architecture for Modeling and Analysis of Rapid Transit (AMART) Team took the approach of an SE/OR firm that focused on architectural analysis of multiple aspects of various transit systems and encompassed both systems engineering and operations research focused operational problems. These problems ranged from automated control and control centers to scheduling, safety, and rider satisfaction. For this project our goal was to develop an architecture paradigm that could be applied and marketed to various metro transit systems, regulatory agencies or developers/designers or further developed for use in other transportation modes. Ultimately the team defined the following problem statement:

Problem Statement: Crowding in rapid transit systems worldwide is a major source of crime, dissatisfaction of ridership, stress on the system, and danger to the public. Overcrowding is indicative of insufficiencies and/or inefficiencies related to:

- Station Size/Configuration
- Train Schedules
- Human Factors (bikes/strollers or hurrying to catch train)
- Schedule/configuration flexibility (reversing escalators etc)

Worldwide, rapid transit control systems and their components have been created, modified, analyzed, and optimized on an ad hoc basis. Considering each system as an independent design as is currently done leads to inefficiency and waste. Failing to consider the commonality of such systems prevents finding the best control solutions for the whole and also prevents future design evolution to improve the whole.

Our reason for choosing the crowding problem is two-fold. First, this is a common problem across all the studied transit systems. Second, there was sufficient availability of information to thoroughly explore this problem.

Based on the definition of the Problem Statement, Team AMART devised a Mission Statement to research and analyze aspects of metro systems and gain insight into the crowding problem that would enable us to develop an architecture paradigm to promote future study and application to other transit systems and modes. The following report details the development and refinement of these statements, the architecture selections (both documentation and executable), the architecture development and the analysis performed. It further describes how our architecture can be applied to other problems within the metro transit system as well as other transportation systems to include multi-modal analysis.

Mission Statement: Develop an architecture paradigm that can be expanded or applied to multiple levels of transit system/modes to gain insight and information about crowding by:[use consistent grammar for bullets]

- Comparing a minimum of three different systems to develop a common [the grammar would be more interesting since the language instantiations are generated from the grammar] language (grammar and vocabulary) to identify network architecture (nodes, edges, flows etc) for each zoom level in addition to the relationships between the zoom levels
- Modeling the common network flow pattern at multiple chosen related layers of abstraction, concentrating on the level of crowding observed within the station (starting from the station level and zooming out iteratively)
- Driving neighboring models with modified input from modeling results to perform analysis on "flow down" of delays and other propagating behaviors
- Abstracting an architectural [level], that could make it possible to model any rapid transit system and determine common throughput and efficiency concerns across any level, while also providing a framework for analyzing potential solutions for both specific instances and generalized system constructs

Our company has developed a flexible architecture paradigm with an executable architecture that we can apply to any mass transit system throughout the world to analyze causes of non-desirable behavior and recommend changes to infrastructure and operations to optimize operational efficiency across systems, or system components.

2. Scope Definition

The team went through several iterations of scope definition before ultimately settling on our final Problem and Mission Statements. The following sections describe how the team went through this process to determine these statements and ultimately choose our documentation and executable architectures.

2.1 Origins of Idea

The origination of the idea for modeling and analyzing a common rapid transit system paradigm arose from years of fascination with rapid transit as a public good, the familiarity of the DC Metro System, and a world-wide notion that most, if not all, rapid transit systems share common operational and system concepts and practices. The customer on the project was to be a specific rapid transit system's owner or operator, but the team was to attempt to address concerns which would simultaneously address their concerns and others worldwide. The team would see a broader opportunity: this call for proposal as an opportunity to develop a flexible architecture that could be applied not only to this particular mass transit system, but to others throughout the world as well.

A general architecture would be described which can be a framework for developing such automation systems in other systems throughout the world. This represents a much broader and larger market for such an architecture as opposed to developing a single architecture for a specific rapid transit system. Additionally, this project was seen as being in line with the current economic conditions and recent initiatives: infrastructure projects are often seen as a way for providing job growth, and such an improvement through the follow-through of this architecture is such an example.

The original vision was very broad in scope, and included the comprehensive modeling of systems, and the integration of such a model into real-world control and monitoring of systems. The visualization and control centers were to be considered in detail. As the project began, the team quickly determined that this scope was infeasible for the duration of the semester and the available expertise of the team. Nonetheless, these concepts shaped the final scope of the project in the semester. The critical aspect of the original scope that was maintained was the creation and modeling of the common architectural paradigm of all rapid transit systems worldwide, as opposed to modeling a single system.

As systems engineering is in interdisciplinary field, it was important to consider a project outside the originators' normal field of work. This project would utilize numerous fields of engineering and concepts, to include potentially distributed algorithms, modeling, simulation, and visualization, graph theory, as well as civil engineering, queuing theory and human behavior modeling. In relation to the team's skill set, Dave, a network engineer working for a defense contractor, had never been involved with rapid transit control systems design, and has a primary focus of wireless networking and cyber security. By choosing a domain so far from his typical domain, the team maintains a sense of novelty, prevents the blurring of work and school, and promotes learning over practicing already learned behaviors and skills. As it would turn out however, Mahesh has experience with rapid transit architectures, and was able to bring domain specific expertise to the team.

2.2 Team forming

Our team was lucky enough to incorporate members with diverse backgrounds and talents which were all put to use for this complex analysis of a rather large problem.

Introduction to Team Members. Team AMART is comprised of four members: David Claypool, Kim Baumgartner, Mahesh Balakrishna, and Yimin Zhang. Both David and Mahesh are Systems Engineering Graduate Students. Dave received his BS from Virginia Tech and is currently a Software Engineer. Mahesh also received his BS from Virginia Tech and has a background in Air Traffic Control and formal training in rapid transit systems. Mahesh is currently a Systems Engineer. Kim and Yimin are Operations Research students. Kim received her BS from the Air Force Academy and is former Air Force Logistics Officer now working C2 Program Management. Yimin is a full time graduate student in the GMU Engineering department specializing in Air Transportation Systems. With both SE and OR backgrounds in the group we were able to devise a well rounded approach that would yield an insightful product. The following table describes how our team put to use our various backgrounds for this project.

Team Member	Dave	Mahesh	Kim	Yimin
Background	SE Student Systems Engineer Software Developer	SE Student Air Transportation Analyst	OR Student Program Manager	OR Student Concentration in Air Transportation
Contributions	Strategist Developer SysML Consulting	SysML Developer Background Research Domain Specialist	Output Analysis Schedule Website	Output Analysis Background Info Researcher

2.3 Problem Statement Development and Selection

In defining our ultimate problem statement, Team AMART considered a wide range of different issues and problems centering around rapid transportation systems like metros. The first step was to conduct background research on various train systems to discern commonalities in operational, infrastructure and other issues.

2.3.1 Train System Research and analysis and selection

At the beginning of the project we did a lot of background research on rapid transit systems [1]-[14]. This study gives us a broad view of different rapid transit systems and a deep understanding of how a rapid transit system works. In order to cover systems that were as diverse as possible, we chose 30 rapid transit systems from all over the world including Asia, South America, North America and Europe. Some systems are extremely new while others are very old. We tried to fully compare the different systems by defining several mission areas as follows:

- **General Information**. This included history, cultural influences, along with how famous the system was and why and funding conditions and planned renovations for both short and long term.
- **Control System**. The Operations Control Center (OCC) functions as the nerve center of the system. The integrity of OCC varies which affects level of automation. For example, the Bay Area Rapid Transit (BART) is a fully automatic system. By studying the current methodology and previous improvement efforts of systems, we can see that some brand new systems incorporate new technology to build high level automatic systems while some old systems were trying to update current technology from manual control to automatic control.
- **People.** Daily ridership and capacity are important parameters to consider whether the system is heavily used and whether expansion will be needed to provide better service.
- **Cases and Reliability.** We researched previous disasters and accidents in addition to overall reliability.
- **Track and Trains**. This incorporated the station and system designs: layout of tracks, number of lines, and overall size and complexity of the system. We also considered utilization of trains as compared to demand of the customers.
- **Management and Funding**. Laws, Policies and oversight agencies are totally different from government owned systems and profitable corporations.
- **Control Detail and Communication Technology**. Since most systems are trying to update their facilities to automation, it is necessary to compare the detailed control technology to find differences between them. Automatic Traffic Operation (ATO), Automatic Traffic supervision (ATS) and Automatic Traffic Protection (ATP) are three main aspects to look at. A fully automatic system like BART has all of these automation scenarios. Others have a mixture of manual control with computer control.

By comparing scenarios above, we decided to choose London Underground, Metro Atlanta (MARTA), Massachusetts Bay Transportation Authority (MBTA), New York Subway and DC Metro. Each system has its own story and systems are quite different from each other such that we could have a broad application if we build a general architecture based on these systems. A breakdown of our reasoning for choosing these systems is located in Appendices D, E and F.

After determination of these 5 systems, we conducted further research to determine the best system to focus on for this project. We settled on the DC Metro because it was not only complex, but also because we felt we had to best chance of getting real data and information due primarily to our close proximity. AMART then had to focus on individual stations to model. We chose stations of varying location, complexity, and layout to ensure that our model was extensible enough to be applied to the full range of station scenarios. The stations we chose to focus on are:

- **Union Station.** Union Station is one of the busiest stations in the Metrorail system, averaging 33,000 passengers per weekday. Union Station is a located on the Red line. One reason why we chose this station is

that it is a simple station which only serves one line. This station has two tracks, both of which are serviced by a single platform.

- **Farragut West Station**. Farragut West Station is one of the most heavily-used stations in the entire system. And is serviced by both the Orange line and Blue line. With the exception of Metro Center and Smithsonian, more Blue and Orange line riders use this station than any other. There are two platforms in the station. Each platform is used for one direction. Orange line passengers and Blue line passengers heading to the same direction share the same platform. There are two entrances, two platforms, three elevators and 12 escalators.
- **Rosslyn Station.** The station is located at 19th and N. Moore Streets in Arlington and is part of a large office, shopping and apartment complex. South of the station, the Orange and Blue lines split, with the Blue line continuing to Arlington Cemetery. Since both the Orange line and Blue line reach Rosslyn Station, this route diverges and the need to keep the lines from separating at grade led to the decision to create the bi-level station. One level is for trains going east and the other is for trains going west. Like Farragut West station, both Orange line passengers and Blue line passengers heading to the same direction share a platform. There are two entrances to the station.
- **Metro Center Station**. This station is located at the intersection of 12th and "G" Streets NW and serves the Red line on the upper level and the Blue and Orange lines on the lower level. The upper level has two bay platforms and the lower has a single island platform. The station gets very busy during rush hour and severe crowding is a major problem, especially on the lower level's island platform. This station's central location provides easy access to many of the district's tourist attractions (White House, Ford Theater, FBI building, etc.) and there are major stores and hotels nearby. Eighteen different bus routes connect to the Metro here.
- **Piccadilly Circus Station**. Piccadilly Circus is one of the most famous areas of London and a major road junction in the City of Westminster. The Circus is adjacent to major shopping areas and entertainment centers in the central part of the West End, which makes the station heavily used and subject to crowding. The station has seven entrances in different corners of the Circus which makes the mezzanine very complex.

2.3.2 Problem Statement Consideration

After completing our system research and analysis we compared our selected systems with our stakeholder mapping and came up with the following set of Problem Statements:

- Overcrowding of stations and trains leads to reduced safety and customer satisfaction
- Varying existing levels of Automatic Train Control create difficulties in common management strategy
- The stove piped development and analysis of systems prevents a common ability to plan for and react to major a disaster
- Despite a common mission and operational structure, the actual systems are quite different from a systems view. Most systems were developed in an independent stove piped manner limiting options for further innovations, expansion or upgrades.
- Control center methodology is not common through various systems. If a common policy was formed with similar interfaces, it would allow operator from DC to pick up and go to NY with little training etc (interoperability)
- Independent systems have developed individual processes and procedures to pass information (C2) to and from train operators rather than determining best practices to enhance safety and performance.
- Passengers are dissatisfied with wait times, cost, crowding, crime and this directly impacts ridership and profitability of system
- Static train schedules fail to account for unanticipated changes in system availability and utilization.
- Rapid Transit Systems have been designed, deployed, and studied independently in the past without considering the commonalities, resulting in inefficiency and stove piping of solutions and evolutions of the systems.
- There is a broad concern of station crowding in rapid transit systems worldwide, in particular the DC Metro.
- Despite common language and common mathematical concepts, little has been studied on the iterative abstraction of network-types to generate generic solutions to common network-related problems. (rejected outright)
- Current rapid transit system lack a standardized approach to handle unanticipated events in the system

- Previous system modeling efforts in rapid transit systems have not considered the interaction among the adjacent range of scopes (zoom levels) in studying such a system, resulting in a lack of concordance between the models and potential inaccuracies in conclusions.

2.3.3 Selection

Many of these problems are interrelated while others seem to be completely disparate. Once our initial list was complete, Team AMART had to scope down our problems into one statement that described the problem we would intend to analyze and hopefully solve. To do this we considered a variety of factors, from lack of available information, to complexity of the problem and the ability to complete a meaningful piece within 14 weeks. Ultimately, we settled on the crowding problem specifically in respect to individual statement platforms with the ability to zoom in and out to various levels to see the effect of an overcrowded station platform on the rest of a system. This problem was complex enough to encompass meaningful SE and OR attributes, yet simple enough to be accomplished within the 14 week time period and be able to complete specific abstractions to validate our model and its relationship across multiple zoom levels (see section 2.5). Furthermore, with respect to crowding and passenger loading, there is ample data available to conduct meaningful analysis of the architecture.

2.4 Stakeholder Analysis and Requirements Elicitation

Formal requirements engineering methodologies [15] were used in eliciting requirements. Figure 1 illustrates the inputs, controls and noise that impacted the process.

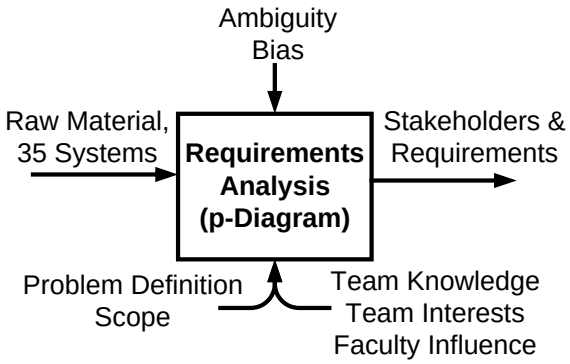


Figure 1. Requirements Engineering for A-MART

For identifying stakeholders, their needs and wants, five orthogonal systems were chosen and studied in detail. These included Washington's WMATA, Boston's MBTA, New York's MTA, Atlanta's MARTA and London's Tube. From these systems we conducted research and analysis of possible stakeholders through determination of management and oversight structures, funding streams, political implications and operational issues.

Listed below are the resulting stakeholders that were identified. Also since A-MART is a university activity, faculty and the A-MART team members were also deemed to be key stakeholders.

Top-Down	Bottom-Up
Policy Makers Politicians Administrators Transit Management Transit Monitoring Office Schedulers and Planners Industry Train Manufacturers Metro Construction Contractors GMU Faculty A-MART Team Members	The Riders Commuters Tourists Special Events Operating Staff Ticket Counters Station Managers Train Drivers Transit Security Maintenance Staff

After the identification of the stakeholders, a value mapping matrix was used to rate each stakeholder's needs and wants and to derive a composite score for each. Figure 2 below shows the resulting scores. It can be seen that reduced

station crowding, increased safety and security and reduced train crowding were found to be the top needs for potential A-MART users.

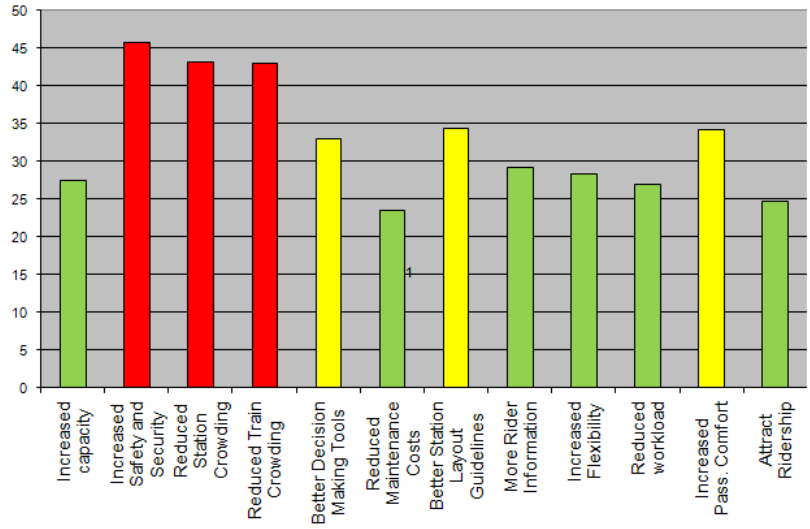


Figure 2. Stakeholder Value Mapping

The stakeholder value mapping allowed for the team to further refine the scope of the project and to revisit the existing systems to identify associated customer requirements. This activity led to a total of 37 customer requirements to be collected. With several members of the team having requirements engineering experience, we decided that each team member would independently develop functional requirements for A-MART that would satisfy all 37 customer requirements. Through group discussions and analysis, a final set of 10 functional requirements were agreed upon for A-MART development.

In order to formalize the relationship between the customer requirements and functional requirements, a Quality Function Deployment (QFD) in the form of a House of Quality was developed [16], which can be found in Appendix J. This analysis helped the team understand the interrelationship between functional requirements and to also to identify technical priorities and benchmarks. In addition to the QFD, all stakeholders, needs, wants, customer requirements and functional requirements have been added to the A-MART Enterprise Architecture, which is described in detail in Section 3.1. This allows for stakeholders to be traced to specific functional requirements in the architecture, thus providing a means to assure through the product development process that stakeholder needs and wants are met per the value mapping. The Enterprise Architect package containing this information can be found in Appendix H.

2.5 The Zoom Concept

In understanding the levels at which models could be developed and simulated, the team incorporated the notion of zoom levels for the model of a rapid transit network. This section describes those levels, and their users and interconnections. The levels of zoom as described here are not a discrete concept as in a hierarchy, but rather continuous, with notional integer and non-integer values assigned. The integer values are used for logical zoom levels whereas non-integer values are used to signify intermediate levels between the logical zoom levels. In the modeling and simulation efforts however, only those levels assigned to logical stopping positions are considered. Provided below are further descriptions of various zoom levels.

Zoom level -1 is a level above the rapid transit paradigm, including the demographic, external transportation networks, and cultural models that influence ridership of the rapid transit system. At this level, the rapid transit system is a single component within a larger model. While not modeled, the thought process of defining this zoom level illustrated some of the concepts explored in the models we did describe, such as the arrival of a large load of passengers due to a conference in the center of a city, which would be modeled at this level if performed in detail. At this zoom level, passengers have desires and schedule requirements of which the rapid transit system is a small part, making modeling this layer a massive effort.

Zoom level 0 has the scope of all interconnected transit systems, to include the rapid transit system of focus. More specifically focused on the transportation of riders than the motivation and high-level movements of the riders,

this model is more realistic to current analysis methods, such as those seen in TranSim. This is a massive area of study and already has a lot of effort put into synchronizing traditionally disjoint transit system operations.

Zoom level 1 is the rapid transit system as a whole. If viewed in a picture form, the model would closely resemble the system maps commonly seen in trains of rapid transit systems worldwide. Riders enter the system at one station, leave at another. The obvious applicability of this zoom level is an operator's eye view of system status, if the model were to be implemented as a monitoring service with the proper visualization. Numerous rapid transit systems already have this in place.

Zoom level 2 is a single line of a rapid transit system. This zoom level was defined as an integer because more often than not, individual lines, especially in older systems such as the London Underground, are operated independently of one another. In the DC Metro, such a model at this zoom level would be the entireties of the Orange and Blue lines, as they, for example, share mostly the same physical track in the center.

An intermediate level between zoom levels 2 and 3, which could be assigned a notional value of 2.3, drills down to a single neighborhood of stations. A neighborhood of a given station is defined as that station, along with all stations that are one-station away from that station. Therefore, in the case of Metro Center, the neighborhood would be Metro Center itself, Federal Triangle, McPherson Square, Farragut North, and Gallery Place-Chinatown along with the four sections of track that separate them from one another. A study at this level allows for managed analysis of the effects of different stations on one another.

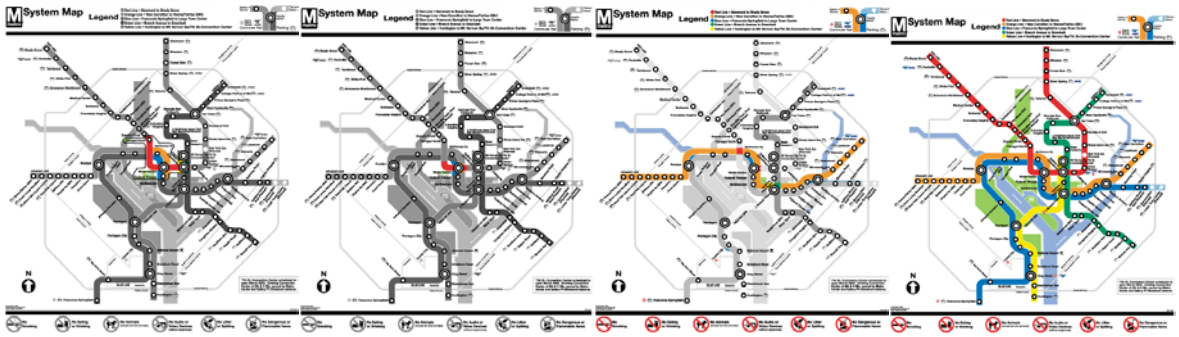
Another intermediate level between zoom levels 2 and 3, which could be assigned a notional value such as 2.7, deals with the modeling of a large transfer station such as Metro Center. Because this station involves multiple lines, and is more complex and larger in scope and potential from a single station on a single line, it is bumped out from level four, which is a single station with a single line.

The next logical zoom level 3 is a single station with a single line, such as Dupont Circle, representing a single node of the entire system with no transfers or complex sprawling layouts. This is the focus of our modeling efforts. Riders are modeled in detail at this zoom level, which permits a fairly comprehensive analysis of movement and crowding.

An intermediate level between zoom levels 3 and 4, assigned the notional value of 3.3, focuses on a single region of a station, or a specific pathway through a station, such as the south side of Dupont Station, or the lower level of Metro Center. This detailed level permits careful study of chokepoints within regions inside a station.

Another intermediate zoom level between levels 3 and 4, assigned a value of 3.7, deals with a logical section of a station, such as a given mezzanine and associated turnstiles, and the crowding effects therein. One such modeling aspect of a platform involves the consideration of doors as an interface into the train, supported by the capabilities of the platform being models. Such a more detailed study permits more accurate understanding of the underlying crowding behaviors of the higher zoom levels.

The next logical level, zoom level 4, takes the model to a logical station element such as a single platform, escalator or turnstile and allows for studying crowding at that one element. This zoom level could for example be used to study how bottlenecks could be reduced by using different types of turnstiles. The last zoom level considered is zoom level 5, which takes the model to a subcomponent within a single station element. Examples of this zoom include electrical system within a turnstile, a ticket reader of a turnstile or a gate of a turnstile. This definition level ends the discussion of zoom levels as considered for A-MART.



Zoom Level 2.3 Zoom Level 3 Zoom Level 4 Zoom Level 5

Figure 3. Zoom Levels

3. Architecture Development

The A-MART project includes both the definition of an abstract rapid transit architecture paradigm as well as the development of an associated execution model for modeling and simulation to enable analysis of a wide range of scenarios and excursions to model system performance across multiple levels. The following sections detail our efforts to develop both the abstract rapid transit architecture as well as the executable architecture.

3.1 Enterprise Architecture Documentation

Model-based Systems Engineering (MBSE) is a fairly new methodology that integrates requirements, architecture development and design into one repository. Such a methodology allows for better understanding of the system, earlier identification of design issues and reduction in overall risk to the project. As a result, Team A-MART decided to follow the MBSE approach using the Sparx EA® Enterprise Architecture (EA) as the tool. This EA tool also allows for the use of formal engineering languages such as Universal Modeling Language (UML) and Systems Modeling Language (SysML) for all modeling [17] – [22].

Over the semester, the A-MART team has also noticed that the tool has been beneficial in better defining the A-MART architecture itself by forcing further thought and discussions as each of the model components (described further below) were added. A weekly-log of the EA development process is provided in Appendix I.

Sparks EA® allows for automatic generation of very detailed systems engineering documentation, all of which is provided in Appendix H. The EA model is also available on the A-MART team website:

http://mason.gmu.edu/~kbaumga1/AMART.htm

The remainder of this section describes the organization of the EA, but does not include details beyond that as is fully provided in other locations just mentioned.

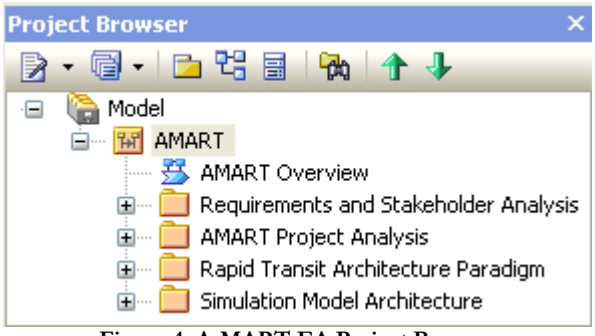


Figure 4. A-MART EA Project Browser

Figure 4 is a screenshot of the top-level of the project browser of the A-MART EA. As can be seen, separate packages have been created to document the components, each of which is further described below.

A-MART Project Analysis

This package fully describes the PDP, team members and project scope. The project scope is shown using UML analysis diagrams and also includes the various scopes that were considered prior to picking the final one. The PDP is also documented using UML activity diagrams and each activity can be navigated into for further detail. The PDP process and the sub-processes follow a spiral type development model. The spiral model is a means to reduce risk by iteratively developing elements of a design. Given the nature of A-MART, the team decided that the spiral model could be ideal for defining the rapid transit architecture paradigm at different zoom levels while validating each element through an executable model. Also, through the duration of the project, documentation was generated and maintained in a manner that could be efficiently consolidated into a final deliverable at the end of the semester.

Requirements and Stakeholder Analysis

This package fully documents the requirements analysis that is described in detail in a different section. It is divided into sub-packages for stakeholders, requirements and mapping. In addition to having all customer and functional

requirements included as UML requirements elements, the House of Quality and stakeholder value mapping are included as embedded documents. Furthermore, the numbering of the requirements elements is consistent with the numbering in the House of Quality. This along with the stakeholder to functional requirements trace diagrams provides the traceability needed to assure that stakeholder needs and wants are met by the end product [23].

Rapid Transit Architecture Paradigm

This key package fully describes the common rapid transit architecture paradigm developed by the project team. Figure 5 below shows the layout of the sub-packages, which include a behavioral model, structural model, network neighborhood, and a spatial taxonomy.

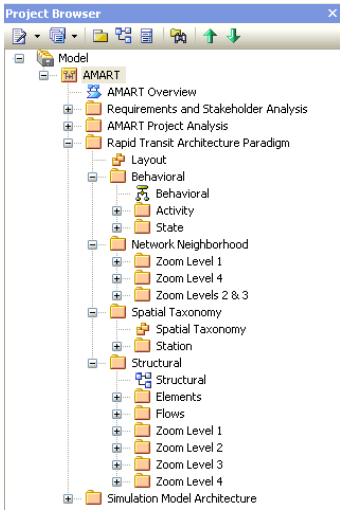


Figure 5. A-MART Integrated Artifact Hierarchy

Behavioral Model

The behavior of the various elements of the common rapid transit architecture paradigm is shown using UML and SysMLstate diagrams. This includes components at different zoom levels including the rapid transit system, an individual line, an individual station or individual elements within a station. In addition, riders and trains each have an associated activity diagram which describes all the activities they perform within the rapid transit paradigm.

Structural Model

The structural model of the common rapid transit architecture paradigm in this package shows elements using SysML Block Definition Diagrams (BDD). Each diagram shows not only the attributes and operations of each element, but also the relationships between different elements. In addition, package diagrams are used interactively to show the concept of zooming between different levels. Therefore, it is important to note that the layout of the packages themselves is part of the definition of the architecture.

Network Neighborhood

The network neighborhood sub-package describes the rapid transit architecture paradigm using concepts from graph theory which applies particularly well to transportation networks. In graph theory, structures of vertices and edges are used to model relationships between objects [25]. Using neighborhoods, which are a subset of vertices and edges immediately surrounding a given node, large, complex flows can quickly by analyzed by looking at key areas within a network. A detailed description of the concept of neighborhoods is provided in Appendix E.

The traditional sense of the neighborhood in a rapid transit system deals with the influences a station experiences during operation. By modeling the neighborhood of a station, and only the neighborhood in isolation, one can fairly accurately see the impacts the neighboring stations have on the central station without modeling the entire system. We make use of this in our modeling efforts. This neighborhood concept corresponds to zoom level 3.3. The network neighborhood EA sub-package describes vertices, edges and neighborhoods for the different zoom levels of the rapid transit architecture paradigm.

Spatial Taxonomy:

The spatial taxonomy sub-package contains abstract groupings of the station components. This taxonomy was developed as the team realized that station components, though separate, may have common characteristics. For example, both a turnstile and an entrance are access ways, though turnstiles have an additional characteristic in that they require payment for access. The abstract relationships between station components are shown using object diagrams and object relationships.

Simulation Model Architecture

As described in section 4.1, a simulation model called StationSim has been prototyped in Microsoft C#® to model the common rapid transit architecture paradigm. The EA contains a package which documents the class diagrams of StationSim, including the attributes, operations and relationships between classes [24]. In addition, notes are provided to describe the function of each class and to document any key algorithms.

3.2 Executable Architecture Selection.

In choosing the simulation tool for the modeling of the rapid transit architecture paradigm, the team used the following criteria:

- Learning curve for the team
- Data collection capability
- Licensing
- Integrity
- Extensibility
- Visualization

Various tools were considered, some of which are specifically designed for transportation systems (such as Transim), for flow analysis (such as Pajek), discrete-event simulation (such as Arena), Colored Petri Net (CPT) tools and generic modeling tools such as Microsoft C#, MATLAB, Excel, etc. A full comparative analysis of the various simulation tools is provided in Appendix K. We began by attempting to model stations in traditional discrete event simulation tools such as Arena; however modeling the train as the resource was impractical as described in detail in Appendix K. We then looked at existing transportation models that may have been able to be adapted for our purposes, but quickly found that they were extremely complex and designed primarily for multi-modal systems. They did not have the ability to focus in on individual elements of a station such as the platform or to compare how different elements could potentially affect that platform, nor did any of our group members have a high enough understanding of these systems to effectively tailor them to our needs. Ultimately Team AMART determined that the best fit for our purpose would be to develop our own model using a generic modeling tool. In the end, C # was chosen for modeling A-MART because the language is ideal for rapid prototyping and aligns with the object oriented methodology used for describing the architecture paradigm.

4. The Project

In planning our model development for AMART, we selected a two-fold approach: The first effort was the description of an architectural paradigm capable of describing a broad range (ideally all) of rapid transit systems worldwide. This also includes object diagrams illustrating instantiations of this paradigm in the various chosen models, as described in the slides. The second effort included the development of an executable version of this model capable of simulating riders within a stations and later on trains within a rail network.

4.1 The Model

Our executable model is a flexible framework based on discrete event simulation. In the station model, riders (users of the system) are the tokens, and station locations/facilities are places, and the riders are exchanged via ports in the station elements to accomplish their desired modeled goal in the station, such as leaving on a certain train or exiting the system in a certain direction. The framework is designed to allow for station elements to be instantiated through "drag and drop" capabilities and is highly extensible for future work through additional coding enhancements. All instantiations, configurations and input files (e.g. rider "agent" patterns, train schedules, etc) can be saved/loaded to XML.

The model has a broad capability to generate raw data on the ongoing simulation execution: detailed reports of each rider's "experience" in the system – with the ability to track a full walkthrough, step by step, of where a rider was. Secondly, it tracks the status, in terms of how many, and how dense, riders were at any given station location, or groups of locations, at any given interval of time. From this, numerous variations of statistical analysis can be performed. The

data outputs will allow us to perform several different types of analysis, to include parametric analysis (regression) to link different parameters to station crowding levels, and then perform sensitivity analysis on potential solutions to determine optimum courses of action. A detailed tutorial on the grammars used in the model is in Appendix E.

4.2 Use of StationSim

Usability of the tool is three-fold and can involve a combination of analysis of existing pre-built models of stations and track systems, creating entirely new models from scratch, and enhancing the software itself through extensible coding. The StationSim Framework is presented with built-in stations for analysis.

Creating new models involves learning the user interface for such interaction, including such activities of placing station and track components, manipulating their properties, connecting their ports, and setting up the data files. During any time in the process, the current progress can be saved to an XML file for later work, much as a word document or other software package would be expected. A guide for this interaction is provided, both in high level and detailed form, in the appendixes.

The most advanced use of StationSim is in the enhancement of the software itself. By using C#, the team was able to bring the most accessible, least expensive programming environment for the purpose of allowing future users to enhance the software. Thanks to the freely available Microsoft Visual Studio C#2008 Express Edition Integrated Development Environment, users may make changes to the code and debug such changes in an accessible manner. The code is in appendix O.

The primary use of StationSim for AMART was for analysis. The first step was to develop detailed models of each of our stations. This included physical features, entrances, exits, escalators, turnstiles and platforms in addition to the determination of train schedules and passenger flows as inputs into each station model.

4.3 Data Inputs

While the rapid transit architecture paradigm can be developed with information widely available through research, the simulation runs using the executable model are theoretical, though accurate until the model is configured with real-world data. For example, it is possible to simulate the behavior of a particular station while using approximate dimensions and rider flows and generate useful results. However if real rider statistics and accurate station element dimensions are used, results that apply to real-world situations could be achieved.

However since the September 11, 2001 terrorist attacks, the availability of detailed information such as exact station layout, dimensions of walkways, escalators, platforms, etc is very tightly controlled. Additionally while train schedules are available, actual time of arrival at stations is generally not published. Where possible, other means and sources of information have been used by the team to make the simulations as accurate as possible given the constraints. It is anticipated and hoped that AMART will gain the interest of WMATA and potentially allow for simulations using configuration, ridership and train operations data provided by the operator. The following sources of information have been used thus far for configuring the simulation models:

Ridership Data

AMART was able to determine realistic data by studying yearly reports, briefings to the Board of Directors, station capacity analyses and other documents developed by rapid transit management on ridership on their systems. Information on transfer of riders at stations where multiple lines cross is still difficult to find. Since the purpose of this study was to focus on the causes and effects of platform crowding, we used varying loading factors for riders to test how the system handled different situations. These types of loading factors included constant rates, surges, and simulated real day loads taking the system from opening through rush hours to closing while simulating increases in passenger flows and changing train schedules.

Physical Dimensions of Stations, Lines and Systems

While track length is easily estimated, dimensions of station elements are difficult to find. As a result, the team has gone on field trips to estimate and record station layouts and dimensions.

Guidelines on Station and System Design and Level of Service [26]:

The Transportation Research Board, sponsored by the Department of Transportation, has published a Transit Capacity and Quality of Service Manual, which it describes as:

"TRB's Transit Cooperative Research Program (TCRP) Report 100: Transit Capacity and Quality of Service Manual, 2nd Edition contains background, statistics, and graphics on the various types of public transportation, and provides a framework for measuring transit

availability and quality of service from the passenger point of view. The report contains quantitative techniques for calculating the capacity of bus, rail, and ferry transit services, and transit stops, stations, and terminals."

The TCRP manual is used extensively by the rapid transit systems across the country, both for adding new stations and lines, but for also analysis of existing infrastructure. The manual provides guidelines for estimating parameters such as train head way, train dwell time, station layout, station capacity, line capacity, system capacity, etc. It even provides information such as amount of time a typical rider takes to traverse through a turnstile, how walls impact the flow of riders on walkways, behavior of queues, etc.

Part 7 of the manual provides general guidelines on passenger densities in station design. Additionally, it various parameters related to densities and Level of Service (LOS) measures, which can be found in exhibits 7-3, 7-4 and 7-9 of the manual. Given that station crowding is the primary problem being addressed by AMART, rider density guidelines provided in TCRP manual have been used in the executable model and for analysis.

4.4 Types of Analysis

StationSim allows for the station element parameters (as described in the TCRP manual exhibits 7-3 and 7-8) to be set/changed/measured in station simulation runs for different time durations (peak hours of a day/whole day/over weeks, months, years etc). In addition to these, other parameters that impact crowding in stations such as train inter-arrival times, escalator speeds, escalator directions, turnstile directions, rider characteristic (i.e. high numbers of unfamiliar riders during special events) etc can also be changed .

StationSim thus provides the ability to identify and analyze real-world operational issues. Such an example might be a specific platform or region of a station (such as an escalator plus an associated platform) that is identified as a "Hot-Spot" (i.e., has frequent over-crowding). Studies can be performed to study specific station elements, groups of elements, as well as station-wide. Figure 4 illustrates the inputs, outputs, uncontrollable and controllable [27].

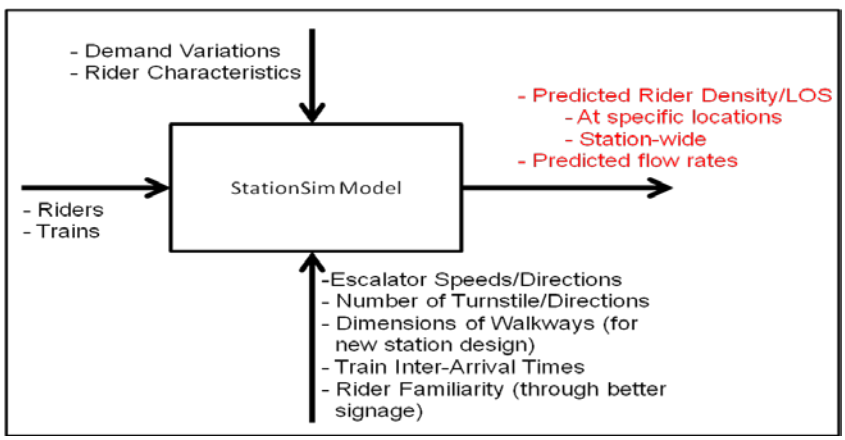


Figure 6. p-Diagram of StationSim for Density/LOS Analysis

Team A-MART conducted several different types of analysis using StationSim and was able to make several interesting conclusions.

Parametric Analysis

The first part of our analysis included determining what parameters contributed to platform crowding. Our basic tool for determining this was a regression analysis based on running the models under various loading conditions with varying train schedules and passenger flows. Through the simulation we collected various data elements to determine their correlation to platform crowding. Data elements collected included, queue lengths through various station elements (turnstiles, escalators etc), Pedestrian flow rates into the station, interval times of trains, time since last train, system time, train utilization and capacity, and wait time on the platform. We were able to determine that the elements with the greatest correlation were Pedestrian flow rates into the station, train interval times and train capacity (number of open spaces on trains entering the station). In order to convert the data into a form that can be compared, we had to take averages based on specific points in time. While the regression analysis showed an extremely linear correlation under constant loading factors, it was obvious that the averages affected the analysis under simulated real-world

conditions. Figure 7 below shows a plot that compares the actual average passengers on the platform in each loading scenario against the predicted values determined from the regression analysis.

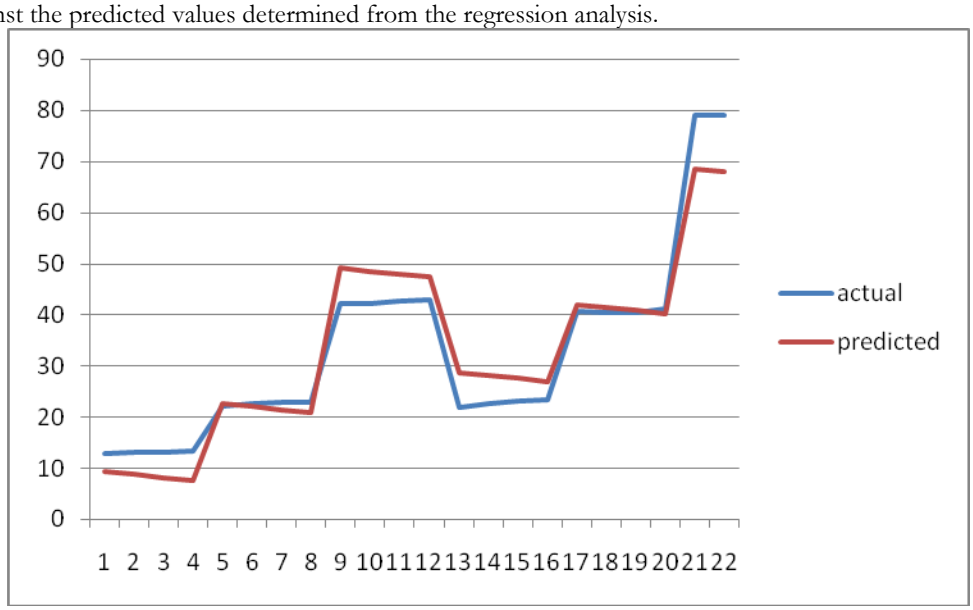


Figure 7. Plot of Predicted Passengers on Platform vs Actual

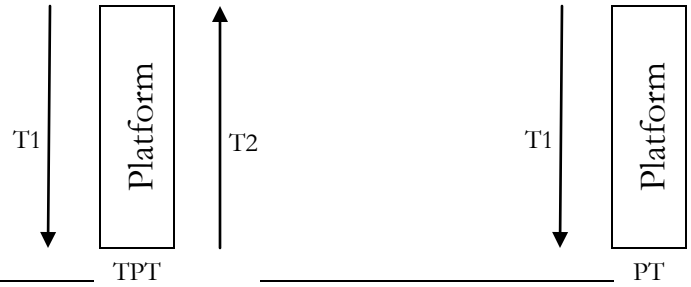
We were able to achieve an R Square value of .92, which means that those three factors account for 92% of the variance in riders on the platform. As you can see from the plot, the predicted values follow the actual values pretty closely through a wide variance of system performance. It is important to note one limitation of this analysis. Sustained periods where more riders are entering the station than the trains can accommodate are not covered by this prediciton.

4.5 Sensitivity Analysis

Once we determined that three parameters (Passenger Flow into the station, Interarrival times of trains, and open seats on arriving trains) were directly correlated, we decided to hone in on how to improve performance on these parameters to reduce platform crowding and positively validate cause and effect. While many different aspects of the passenger flow rates could be explored, the only way for anyone to control those rates to result in less passengers entering the platform would be to restrict the flow into the station by closing entrances and turnstiles. This would result in less passenger throughput and most likely increased rider dissatisfaction if their access to the station was limited or denied. However, the inter-arrival times of the trains is easily controlled through scheduling. Our group focused in on the train arrival times to determine optimal performance of the schedule, not so much as how many trains and how often they arrive, as that is a problem that has been studied numerous times, but more what is the best frequency of those arrival times in relation to other arriving trains or specific events affecting passenger flow. In addition, we explored the effects throughout the system and the neighborhood concept of overcrowded trains on a specific station platform by isolating system responses to varying degrees of train crowding.

4.6 Interarrival Times of Trains

Before we continue we must explore the physical layout of the tracks and platforms. There are two archetypes of platform layouts, and both of these types were modeled and analyzed in our project. The first is a platform that services 2 tracks, one on either side of it as depicted below. We call this a Track, Platform, Track (TPT) layout. The second is a platform that services a single track (possibly with multiple lines). We call this a Platform Track (PT) layout.



We first explored the TPT layout. In this situation we studied a variety of different scenarios to determine the optimal phasing of train inter-arrival times.

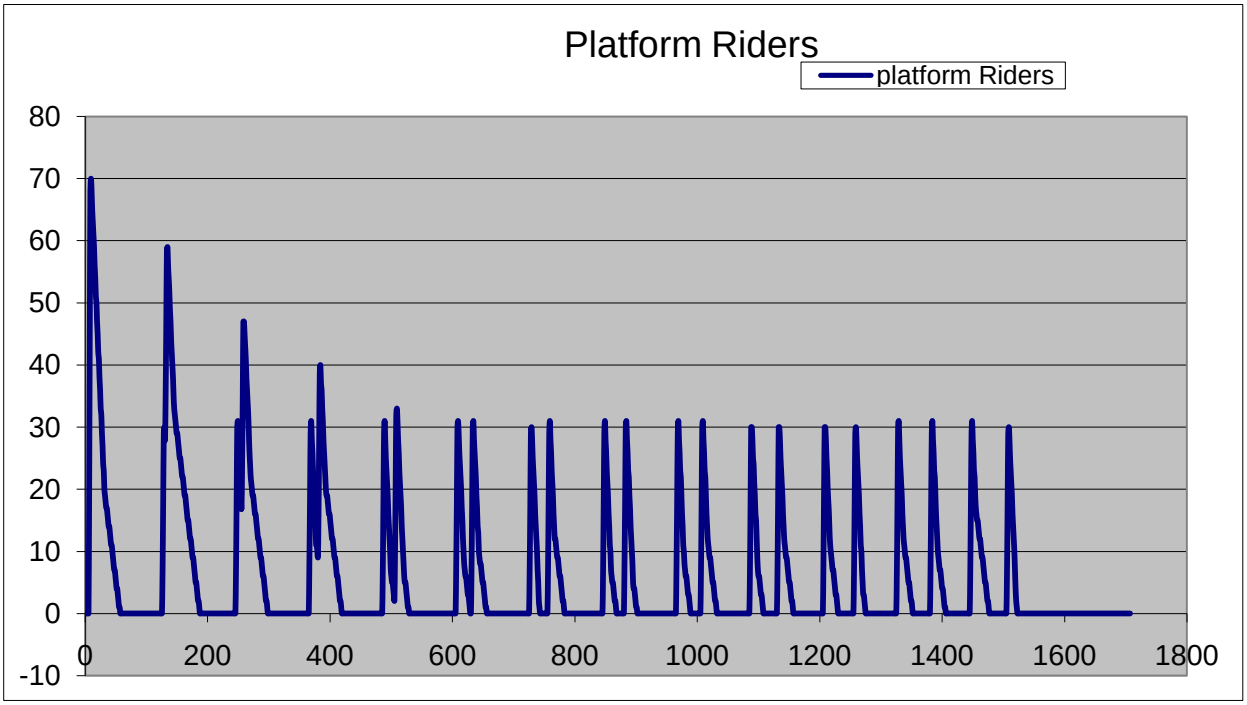


Figure 8. Phase Variance for TPT layout

Figure 8 above shows a train on one track arriving every two minutes, and another train on the other track arriving every two minutes, 5 seconds, sweeping from 0 to 180 degrees phase variance. This analysis initializes with both trains arriving and delivering passengers onto the platform at the same time. As can be seen, as the interarrival of these trains spreads out, the maximum passengers on the platform (due to riders exiting the train) is 2.5 times less than when two trains arrive at the same time. This shows that the optimal schedule for interarrival times is evenly spaced arrival times.

Similarly we see the same result from Destructive type variance. Figure 9 below shows a single train on one line that arrives every two minutes with a load of passengers, while another, empty train on the other line arrives every two minutes, 5 seconds to pick up riders waiting for it, sweeping from 0 to 360 degrees phase variance.

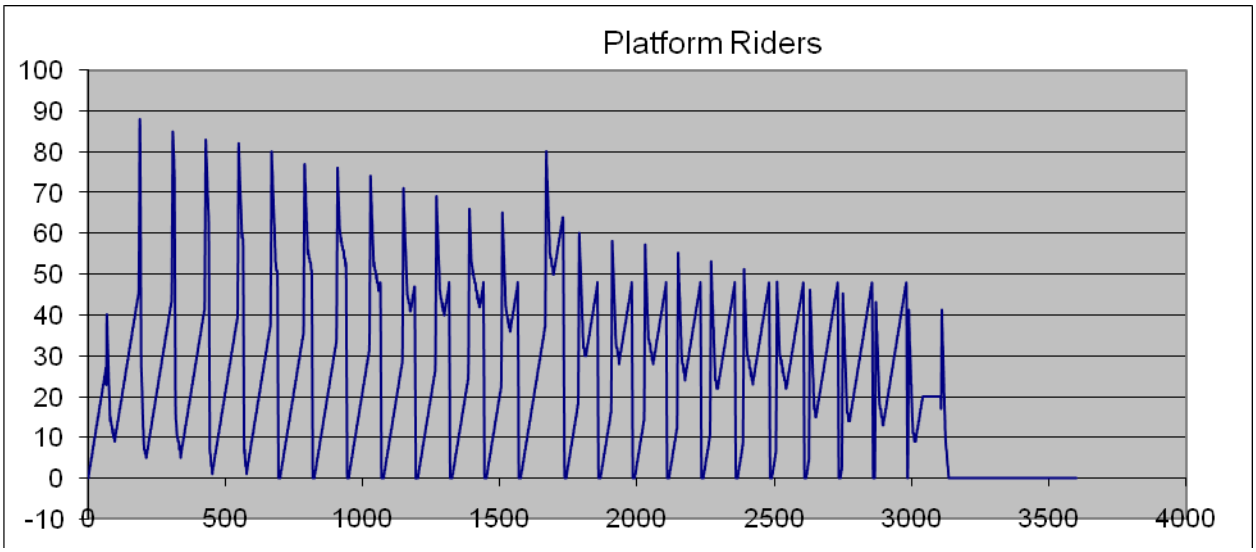


Figure 9. Destructive Interference

Both of these analyses were specific to the TPT archetype. The difference between the TPT and PT layout is that, the PT layout prohibits 2 trains depositing riders on a platform at the same time since only one train can service the platform at a time. When two trains are scheduled to arrive at the station at the same time on a PT layout, the model automatically holds one of the trains until the first train departs the station. The blue line in Figure X below depicts the PT layout using the exact same inputs as the TPT layout Phase Variance Plot. Notice that the number of riders on the platform does not appear to have the same phase effect in that each peak is relatively evenly spaced out and occurs at the same height – approximately 30-40 riders on the platform at once. In other words, with the PT layout, by design, even when you have one train right behind the other, the passengers exiting the first time have enough time to get off the platform (if exiting) prior to the next train arriving so there is no compounding effect like you see with the TPT layout. So we decided to add a new dimension to see if we could gain further insight. The original analysis (blue line for PT) and the TPT analysis isolated the effect of riders arriving from a train and exiting the system on the platform density. For the PT analysis we took it a step further and added passengers entering the station from the street to wait for a train on the platform in addition to riders exiting a train and remaining on the platform to wait for the next train (such as might happen at Rosslyn where a single platform services both Blue and Orange line trains). The red line on Figure 10 shows this effect.

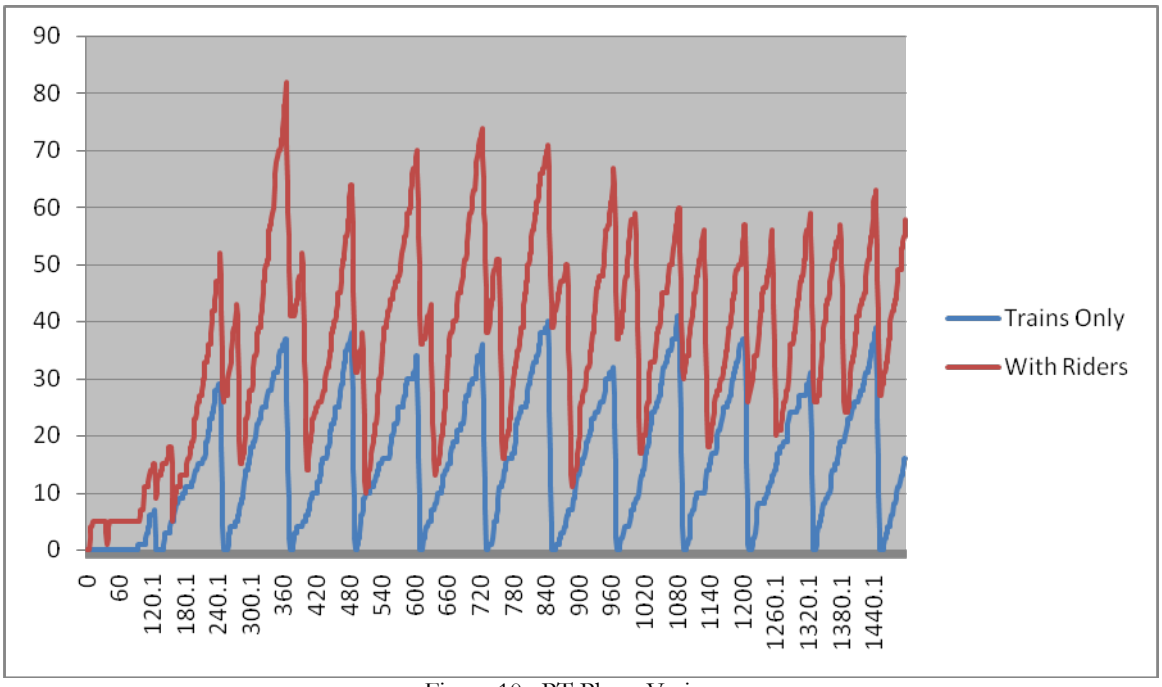


Figure 10. PT Phase Variance

The interesting result from this is that including the additional riders in the analysis appears to follow the same pattern as the original TPT layout. At the 180 degree phase variance when the inter-arrival times of the trains are opposite, you see that the peaks become more evenly spaced and in general are approximately .25 times lower than when the trains are coming one right after the other.

From our initial analysis we can deduce that both station layouts (TPT and PT) respond better to evenly spaced inter-arrival times of trains in regards to platform crowding.

4.7 Train Utilization Analysis

The other parameter that we explored was that of available train capacity or more simply put, the amount of available room on trains arriving at a given platform. It is common sense to state that if you have more riders entering a station than the trains can carry, that you will see increased crowding. We see this happen generally in two situations. The first is during rush hour when trains are over crowded and tend to arrive at a station already full, therefore without the capacity to pick up all of the riders on the platform waiting for that particular train. The below plot shows a sustained period of more passengers arriving at the station then the trains can carry away.

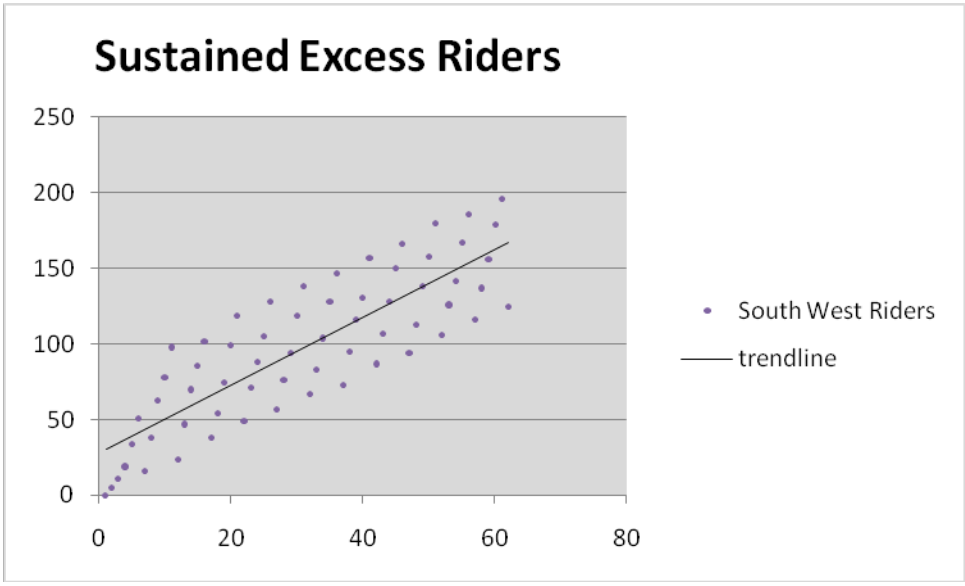


Figure 11. Excess Riders

The other scenario we experience in the system is a surge such as at the end of a sporting event or concert, when a large pulse of passengers descends on a station in a short period of time. The below plot depicts approximately 4000 passengers descending on a station in a short period. The blue line corresponds to the passenger crowding under normal scheduling conditions while the red line injects additional empty trains every 10 minutes to help dissipate the crowd in a more timely fashion.

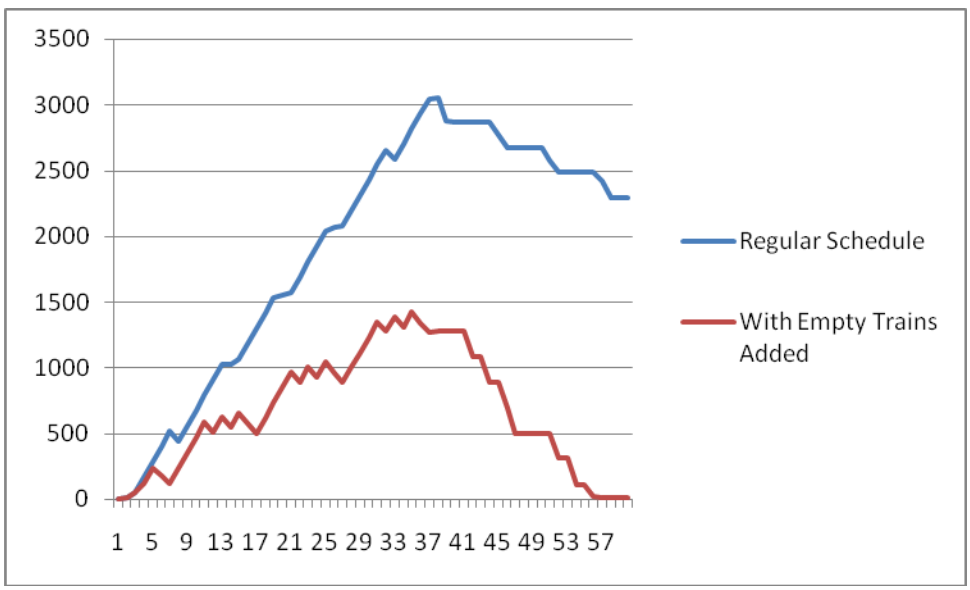


Figure 12. Surge

4.8 Model Grammar

A main portion of our project involved developing and defining concepts such as grammar, and neighborhoods, both within the paradigm of the rapid transit systems and the executable StationSim model. In this process a formal grammar was developed and was used to formalize relationships and building blocks of the systems. This formal grammar is actually comprised of several other grammars including several string type grammars and one shape grammar. A complete appendix detailing this work is located ad appandix E.

5. Future related work

As we have previously discussed, StationSim was developed in a flexible manner to allow us to model multiple stations and aspects of metro transit systems. This model and architecture paradigm can be easily adapted to study and focus on other problems not only of a metro transit system, but other transportation systems and concepts as well.

5.1 Air Transportation

Intuitively, air transportation systems are pretty similar to the rapid transit system studied here [30]. For example from a network flow point of view, aircraft in the air transportation system in some ways behave like passengers in rapid transit system. Similarly, airports can be viewed as stations and airways, arrival and departure routes can be viewed as tracks. Both have schedule, delay and congestion problems, etc. Therefore, A-MART can very easily be adapted to model air transportation systems to study issues.

Layout of tracks at a station and layout of runways at an airport

Different stations contain different track layouts. Take DC Metro for example, the Dupont Circle station has only one track for each direction while two tracks in Farragut West station share the same platform. In the Metro Center station, there are two levels of platforms serving three different tracks for both directions. Similarly, for a single airport, there are various layouts of runways. Most big airports have independent runways which are parallel runways far enough from each other allowing aircraft on different runways to land and depart independently. Other airports have dependent runways which are laid out in parallel. Crossing runways is another type of layout. The bottleneck for dependent and crossing runways is that the aircraft can not land and depart from the runways simultaneously. This situation is same as two trains on different tracks can not arrive at the station at the same time.

Station and airport

In the station, passengers wait for trains on the platform. Similarly, holding points can be used for re-sequencing aircraft for a take-off sequence and that is different from the sequence of their arrivals at the holding point in order to maximize runway throughput via minimizing total safety separation distances in the air.

High density at the platform can cause safety problems and over-crowding can lead to the station being closed. The space for aircraft to hold is even comparatively smaller. The goal is to prevent the aircraft from queuing up. Trains arrive at the station at a certain rate because of schedule and dwell time while air traffic control has minimum separation time, which is also a bottleneck for the overall throughput. In addition, the number of escalators and the direction of escalators in our project can be changed to number of taxiways and direction of taxiways, which can affect queue length of the aircraft holding point.

Platform Crowding and Tarmac Congestion

Reducing platform crowding (i.e., to minimize the total amount of time passengers spend in the station, including the escalator, mezzanine and platform) can be compared to minimizing tarmac congestion (i.e., to minimize the total amount of time aircraft spend on the tarmac, including the taxiways, parking aprons and runway entrances and holding points but excluding the gates). It can be recognized that minimum fuel consumption, minimum environmental impact, and minimum passenger discomfort can be achieved simultaneously by minimum tarmac congestion.

Robustness of schedule

The uncertainty considerations lead to the objective of maximizing the robustness of traffic planning and control in the presence of uncertainty. Note that minimization of tarmac congestion will help reduce or even virtually eliminate congestion at runway entrances and on taxiways, which is one of the two major recurrent sources of uncertainty.

Gate assignment problem

Gate assignment is a frequent problem faced by airlines and airports. One problem is that during peak times, aircraft are often waiting to park at already occupied gates. This leads to increased costs and chances of passengers missing connections. The issue is further complicated gate assignment restrictions due to size of aircraft (regional jet

versus a Boeing 747), ownership/control of certain gates by airlines, etc. Another problem is the loss in revenue due to fuel burned and time wasted during long taxis to and from the gate. Could aircraft be parked at remote terminals and have passengers bused to the main terminals? How could airlines be convinced to share their gates with others? They would need to see a benefit in overall drop in surface delays. To study these problems, A-MART could easily be adapted to model airport layouts, restrictions and aircraft flows. The inputs such a modified A-MART simulation could be aircraft of various sizes and different goals, different arrival rates, different turn-around times, etc. The architecture of A-MART very easily allows for airport surface operations to be modeled and studied.

Baggage system problem

The baggage system is a significant part of the air transportation system. The baggage system also faces quite similar problems to the single station in a rapid transit system. Take Dulles airport for example, the original baggage process is that baggage is checked in from the check in counter and put on the conveyor belt. They are delivered to a lower level of the building where securities are to be done by machines. If there are no problems with the cases, personnel will put a red sticker to the baggage to signify clearance. Otherwise, cases will be opened by personnel. After examining, baggage are delivered to the aircraft by shuttle.

The problem is how to increase efficiency of the baggage handling system. Our model is very suitable for testing and simulating the performance of the whole baggage system because unlike the traffic flow in the air, it has a fixed track. All components in our model can be used and easily transferred to create a real baggage system. Based on the layout of the building, an optimization study could be done by testing different operational architectures of baggage systems and a frequency analysis could be done by testing real data.

5.3 Parcel Service

When it comes to parcel service, naturally, people tend to think about FedEx® service. It is reported that FedEx® Corp's home airport in Memphis, Tenn, remained the world's busiest cargo aerial port in 2008.[31] The efficient integration of warehousing and transportation services is very important to the company. Nowadays, FedEx® relied on Surface Management System (SMS) developed by Metron Aviation to optimize its surface operations in the face of reduced arrival and departure capacity.[32] Our model could accommodate the complexity of taxiway operations through simulation similar to the track portion of the model. We can zoom out from a single warehouse in Memphis to a higher level which considers warehouse as vertices and routes between warehouse as edges by using the neighborhood and network concept.

5.2 Multi-Modal Extension of A-MART

While developing the zooming concept for the A-MART architecture paradigm, the scope has thus far been limited to looking at flows of riders and trains within a single rapid transit system (i.e., zoom level 2 as shown in Figure 6 below). While new riders entering the station are an input into the analysis, how the riders entered the station is not considered at this time. The riders may have walked to the station, been dropped off by car, arrived by a bus, etc. In order to take into account these other modes, A-MART could be extended (i.e., zoomed out further) to be multi-modal. *Multi-modal* planning refers to decision making that considers various modes (walking, cycling, automobile, public transit, etc.) and connections among modes so each can fill its optimal role in the overall transport system [28], [29].

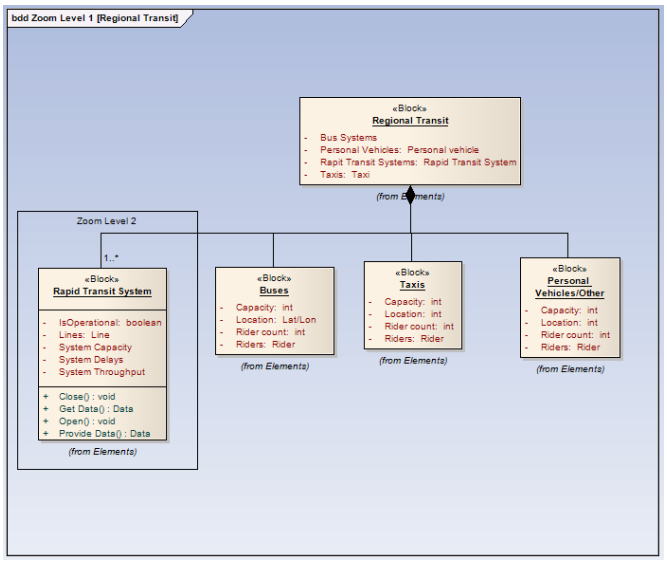


Figure 13. Multi-Modal A-MART

While looking at station crowding within rapid transit systems, including multi-modal aspects in the analysis could help further improve the ability to reduce station crowding. An example of this can be seen frequently at the West Falls Church station on the east-bound Orange line of WMATA in the morning rush hour. West Falls Church station is a primary entry point for suburban commuters who reach the station either via Fairfax Connector or Dulles Connector. Both are scheduled bus services that bring large numbers of riders to the station who are heading into Washington, DC. As the riders exit the buses, they rush to enter the station, pass through the turnstile and reach the platform through escalators and stairs. When there are delays in the Orange line, the platform can become crowded very quickly as new buses arrive. If there is a way to delay the buses or synchronize the buses with train arrivals, the situation of crowding could be mitigated. Therefore, one possible future extension to the A-MART model would be to zoom out to level 1, which would then include new elements such as buses, automobiles, taxis, etc. in the model. Beyond this, the work could be further expanded to look beyond stations at major road arteries which serve stations (e.g., I-66 and the Dulles access road for Orange line stations). This extension may show flow patterns which could be used to optimize the overall system.

6. References

1. Yurikamome, Tokyo, (2006 March) Kobelco AGT System, Kobe Steel Ltd, Tokyo J

2. Precision Station Stopping Progress Update, Presented to the Board of Directors, WMATA, March 2005

3. United States Cong. House. General Accounting Office. Testimony to the Subcommittee on the District of Columbia. WMATA is Addressing Many Challenges, but Capital Planning Could be Improved. Washington, September 2001.

4. Rosslyn Station New Entrance Study, WMATA Department of Planning and Joint Development, 2007

5. Joint Development Policy and Guidelines, WMATA Department of Planning and Joint Development

6. Wikipedia used extensively for general information on various rapid transit systems, (http://www.wikipedia.org/)

7. Urban Rail extensively used for general information on various rapid transit systems (http://www.urbanrail.net/index.html)

8. The official websites of WMATA (http://www.wmata.com/), London Underground (http://www.tfl.gov.uk/modalpages/2625.aspx), Stockholm Metro (http://www.sl.se/Templates/SubStart.aspx?id=1906), New Delhi Metro (http://www.delhimetrorail.com/index.htm) and Buenos Aires metro (http://www.subte.com.ar/contenido/home.asp) were used extensively for analysis of existing systems.

9. The concept of AATC

http://ntl.bts.gov/lib/6000/6600/6646/attc.html

10. The official website of MBTA(http://www.mbta.com/),

11. Station reporter (http://www.stationreporter.net/dcorange.htm)

12. Automatic Train Control in Rail Rapid Transit, online book, United States Congress Office of Technology Assessment, 1976

13. United States. Cong. Senate. Office of Technology Assessment. Automatic Train Control in Rapid Transit, May 1978

14. Banerjee, M.K., Hoda E. Nour, "Review of Automatic Train Control System for Cairo Metro Line 2", Power Engineering Journal, 1998 : 217-228.

15. Mehta, Sandeep, "Requirements Management and Stakeholders", The PROJECT Perfect White Paper Collection, 2005

16. House of Quality Templates, from the QFD Online website http://www.qfdonline.com/templates/qfd-and-house-of-quality-templates/

17. Model Based Systems Engineering – Opportunity to Improve Efficiencies with SysML, artisan® Software Tools, Inc, Beaverton, OR

18. Introduction to Robust Design – Robustness Strategy, from the isixsigma website http://www.isixsigma.com/library/content/c020311b.asp

19. OMG Systems Modeling Language (OMG SysML™), Version 1.1, OMG Document Number: formal/2008-11-02, Object Management Group, 2008

20. Fridenthal, Sandord, Alan Moore and Rick Steiner, (2006, July). OMG Systems Modeling Language (OMG SysML™) Tutorial INCOSE 2006, Orlando, FL.

21. Weilkiens, Tim, Systems Engineering with SysML/UML – Modeling, Analysis, Design, The MK/OMG Press

22. Hause, Matthew, et al "Systems modeling language (SysML) addresses UML shortcomings", EE Times – India,

23. Requirements Management with Enterprise Architect – An Introduction to Enterprise Architect, Version 1.2, Sparx Systems, 2008

24. Mappings between C# and UML elements, from the IBM website http://publib.boulder.ibm.com/infocenter/rsmhelp/v7r0m0/index.jsp?topic=/com.ibm.xtools.viz.dotnet.doc/topics/r_cs_mapping.html

25. Neighborhoods and Graph Theory, from the Wikipedia website http://en.wikipedia.org/wiki/Neighbourhood_(graph_theory)

26. Transit Capacity and Quality of Service Manual, 2nd ed., Transportation Research Board, Washington DC, 2003

27. Robustness P-Diagram: An Overview, from the Quality Portal website http://thequalityportal.com/p_diagram.htm

28. Litman Todd, Introduction to Multi-Modal Transportation Planning – Principles and Practices, Victoria Transport Policy Institute, November 2008

29. Proposed system requirements for the design of a dual mode transportation system, from the University of Washington website http://faculty.washington.edu/jbs/itrans/DMsystemrequirements-CEETI.htm

30. Tsao and Wei, AN OPTIMIZATION ARCHITECTURE FOR AIRPORT SURFACE OPERATIONS, TRB Annual Meeting, 2009

31. JCTrans.Net(http://info.jctrans.net/jcnet/news/asn/2009210729266.shtml).

32. Metron Aviation, FedEx Relying on Surface Management System Developed by Metron Aviation to Optimize Operations During Runway Closure, GISUser.com - The complete GIS, LBS, and Geospatial Technology Resource, Wednesday, 18 March 2009

7. List of Appendices

The following appendices are included in this volume:

- A. Grammar, Neighborhoods, and Future Parallelization (included within this document)
- B. Generic Modeling Environment Example (included within this document)
- C. CPN Tools Comparison Example(included within this document)
- D. System Background Information (included within this document)
- E. Original Concept Material (included within this document)
- F. Glossary of Terms (included within this document)

The following appendices are separate volumes:

- G. Master Spreadsheet System Comparison Spreadsheet
- H. SysML Model of Rapid Transit Paradigm
- I. Enterprise Architect Journal
- J. House of Quality (HOQ) and Requirements Analysis
- K. Model Comparisons
- L. Introductory Model Tutorial
- M. StationSim Input Data Samples
- N. StationSim Excel Output
- O. StationSim Development Environment, Source code, and Logs
- P. StationSim XML Example
- Q. Regression Analysis Detail
- R. Arena Example
- S. Project Schedule and Metrics
- T. Team Member Biographies

Architecture for the Modeling and Analysis of Rapid Transit

A-MART

Major Appendix A

Grammar, Neighborhoods, and Future Parallelization

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

SYST 798/OR 680

Introduction

This appendix describes the grammar and neighborhood aspects of our work, both within the paradigm of rapid transit systems, as described, and also in the executable form of that paradigm in StationSim.

By formally expressing the architectural paradigm of rapid transit systems, we were able to transcend a generally structured description of the rapid transit paradigm and instead elicit an underlying formal grammar which describes the system and its components, including the rider behavior and associated interactions with the system. From the grammar, we further formalized the relationships of the building blocks of these systems. With this formalization, we open up additional avenues of description, analysis and simulation, as well as providing another means to describe formally a rapid transit network.

Within this formal paradigm definition and concurrent executable architecture development effort, there are several different grammars to be considered for describing rapid transit structure and behavior: several string grammars and one shape grammar. Our discussion in this section begins with the most basic: the string grammar of riders flowing through a station, specifically, the path they took through a station. This particular grammar is highly suited towards StationSim itself, as it includes the current behavioral models and constraints in place for the given simulated elements..

Basic Formal Grammars

The there are several lexicon grammars defined as a result of the logic in StationSim. As an introduction to this description, the definition of grammar, as defined by Wolfram Mathworld, is shown below:

Definition From Wolfram Mathworld

A grammar defining formal language L is a quadruple (N, T, R, S), where N is a finite set of nonterminals, T is a finite set of terminal symbols, R is a finite set of productions, and S is an element of N. [G1]

In the case of grammars involving rapid transit, terminal symbols represent the actual items comprising the network of riders, locations, and equipment which makes up the system, while non-terminal symbols represent intermediate abstract constructs that do not directly correspond to physical entities. The productions, also called production rules, describe how the organization of symbols can grow to define a new such organization. The S represents the starting nonterminal symbol This discussion assumes the reader will have had some knowledge of grammars already, but numerous introductions to the formal grammar concept are readily available for a detailed review.

Rider Path History Grammar

One major concept of the grammar is in the definition of a rider path as a result of using the system. We begin our discussion of grammar with this example because it is one of the easier to understand, not because it necessarily is of the broadest scope. This rather simple string grammar is a way for us to see what riders experienced, in order, between when they entered a station, and when they left in a concise manner.

StationSim outputs this string each time a rider leaves a station, whether that departure is out an exit, off a walkway, by some error in the model instance, or via a train. By exporting these strings, we can then also verify through observation or string matching, that the rider path history calculated within station sim is syntactically correct according to the grammar defined here. Over the course of execution, within our models, we have found this correct behavior to be the case.

The grammar is formally defined in the following table:

Nonterminal Symbols	Production Rules
N = {A,B} Where A represents the abstraction of an element that can always accepts riders And Where B represents the abstraction of an element that can only conditionally accepts riders	A → qB A → pl B → A A → wA B → tA B → xA B → kA A → e
Terminal Symbols	**Start:**
T = {q,w,x,t,p,l,e,k} q = Queue w = Walkway x = Escalator t = Turnstile p = Platform l = Line e = Exit k = Ticket Counter/Machine	S = A

Table 1. Rider Experience Grammar

The Example instances of this grammar are illustrated below and are comprised of strings of the terminal symbols, in order of their encounter by the particular rider. In the first example, a rider queued at an escalator, then walked up to another queue, before entering a turnstile, onto another walkway to another queue, and then down an escalator to a platform, and onto a train (l). As one can see, this string grammar, while not specifying which queue or walkway was experienced, lists the general ordering of events in a riders particular journey through a station, and proved valuable in recording concisely the experience of thousands of riders in a simulation run. The other example show other journeys within a similar run.

A Rider entering Dupont boards	qxwqkwqtwqxpl
2nd Rider transfers at Dupont	qxwqxpl
3rd Rider leaves Dupont	qxwqtwqxe

Table 2. Example Rider Experience Strings

One aspect of this grammar that is particular useful is in stochastic behavior of riders: within a given station, such as metro center, there are more than one way to reach a given goal, and as such, the rider sharing a common goal of “redwest” may in fact have had widely differing experiences of how they eventually got there. By recording this station element trace empirically within the simulation, not only can the execution be validated against the grammatical rules, but also the distribution of riders taking a given path can be discerned, thanks to this string being printed as part of the rider report for each rider.

Rider Goal Modeling Grammar

Another grammar is the rider goal oriented string grammar: each rider in the transit system has a destination goal. Part of the human behavioral model, StationSim includes a rudimentary, but effective goal-based behavior model to control how riders move and make decisions within stations and while on trains. Simply stated, this simple grammar is a list of one or more (or even none in simple models) goal strings delimited by a vertical slash (“|”). This grammar, along with the restriction such as all letters and numbers for the goal statement, is formally defined below. Every rider begins the journey with the first of the goal statements active. Each and every time a rider boards a train (if the rider began on a train, then that does not count as a boarding), the goal statement moves to the next goal statement. If no more goal statements are available, the goal statement is pulled from the beginning again and the cycle repeats.

Nonterminal Symbols N = {A} Terminal Symbols Σ = {a-z,0-9,|}	Grammar Rules Start → A A → A|A A → [a-z,0-9]*

Table 3. Goal Grammar

Examples instances of this grammar are illustrated below. The grammar itself is fairly intuitive, and permits very simple goal statements (which are simply the singular goal statement of the rider during the entire run), and complex goal statements involving transfers.

A Rider is placed into the system at the entrance to Dupont Station and simply gets onboard a redline westbound train there.	redlinewest
In a larger model, 2nd Rider takes Dupont to Metro Center	redlineeast|metrocenterexit
3rd Rider takes Dupont, transfers Metro Center to get to Vienna	redlineeast|orangewest|viennaout

Table 4. Example Goals

This language enables routing of any path through the system while maintaining the simplicity and compatibility of the original station-only model. This was important as a the time of its adoption, individual station models had already been developed, and backwards compatibility was desired to limit conversion of previous work.

Shape Grammars

While generally less well known, but potentially more intuitive and more applicable to our domain of systems engineering, is the shape grammar of the paradigm itself. The grammar of the architecture defines the rules for instances of the station elements and stations themselves. These two distinct zoom levels of grammar are defined here for future use and analysis, but grammars for other zoom levels could also be determined in a similar manner. This section describes both the shape grammar at the station zoom level, and also the grammar at a larger context: the entire rapid transit network.

Station Grammar

The grammar illustrated below shares commonalities with the formal lexicon grammars described in the previous section. In this grammar, all of the shapes are terminal variables, representing a slightly different approach than the concepts represented in the paradigm definition in SysML, in StationSim, and that which what was formally declared in the rider path grammar.

The starting shapes are the station entrances and the tracks. From here, the station can be generated according to the shape grammar rule set to form the layout of the station. This layout is in accordance with our paradigm definitions and our StationSim model. One can think of the graphical user interface of StationSim as mimicking the application of these rules to the current shapes in an interactive way.

While not comprehensive, the following table illustrates a working set of rules, which describe the shapes of the elements comprising the station models. It is designed such that other station elements could be added in to account for unique station elements yet to be discovered through existing or yet-to-be-constructed systems. This extension concept is also in line with how StationSim was architected through generalized classes and interfaces to support expansion without duplication of effort.

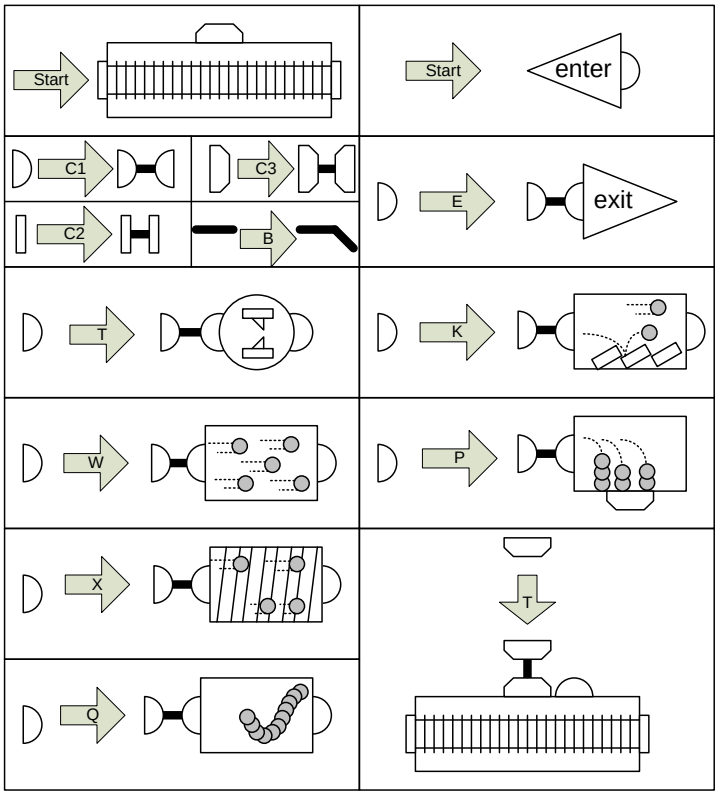


Figure 1. Station Grammar

Shown below is the application of this grammar to build an arbitrary, rather simplistic station. This is representative of the same grammar that StationSim ends up using in its model. As shown, it represents a station with a single street entrance (left side) with a single track on the bottom right. Normally stations would have two or more tracks (one for each direction), but this is a simplistic example for illustration. By viewing models within StationSim, one can see much more complex stations.

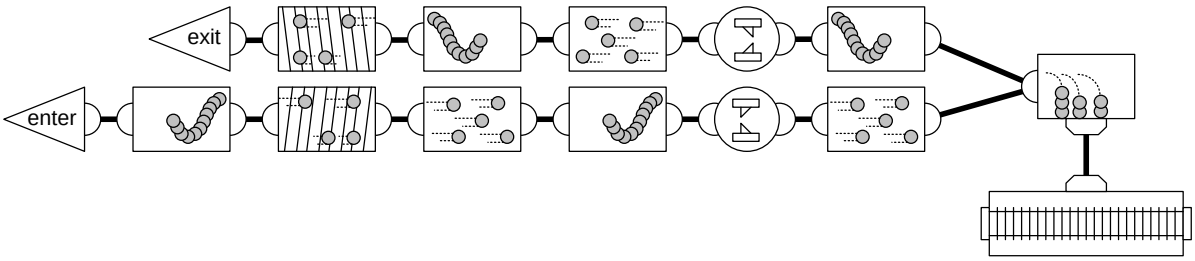


Figure 2. Grammar Instance (Station)

Shown in Figure 3, StationSim, of Metro Center is a more complex example of this grammar, yet still uses a similar rule set. Noteworthy is that while the shape grammar defines the overall layout of the station, there is data embedded as properties of the shapes themselves. For example, the escalators have varying lengths, just as in real systems. Their logical drawing layout need not be consistent with their underlying properties. A Long-drawn escalator is not necessarily a long escalator within the simulator, but it helps to maintain some degree of consistency between those.

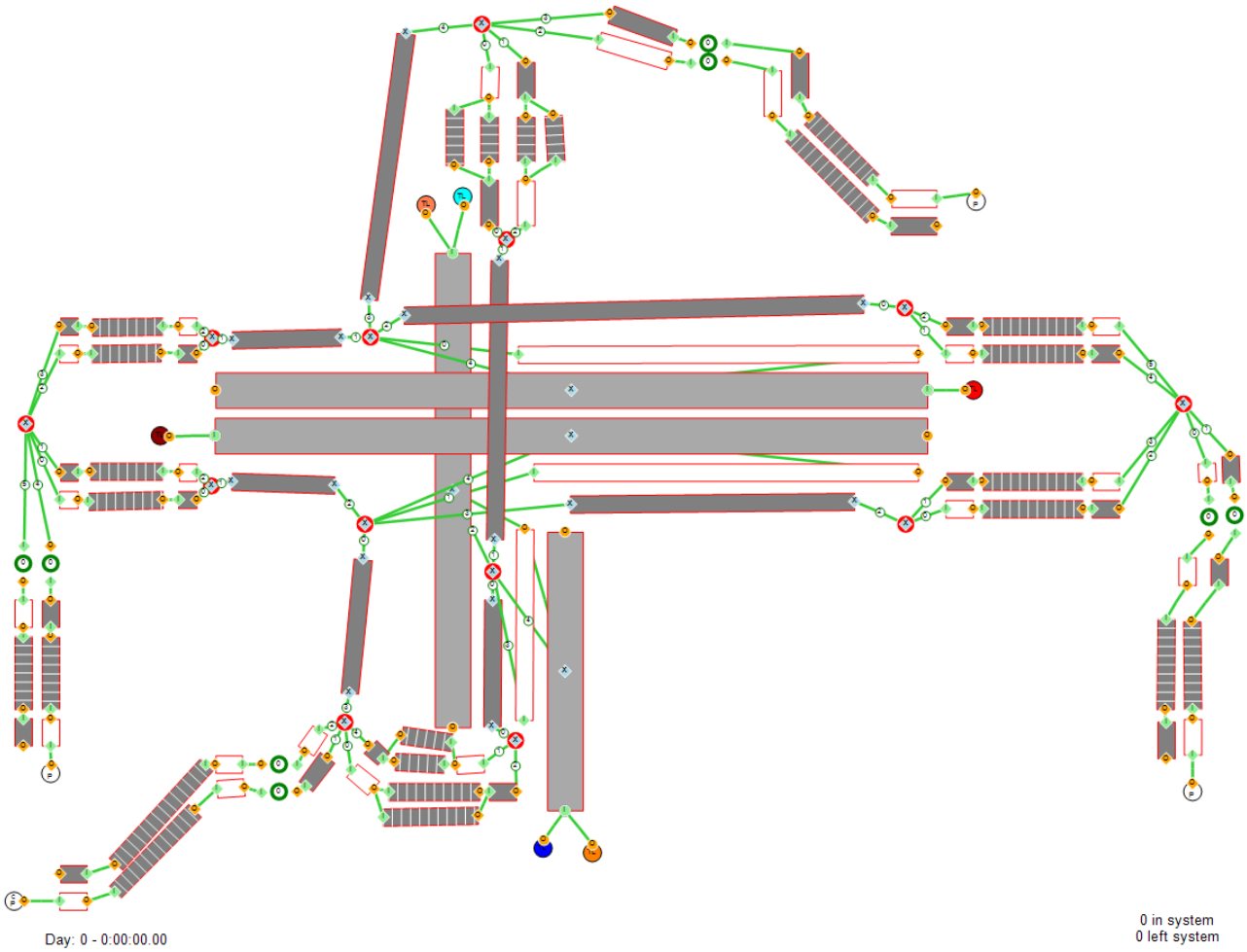


Figure 3. Metro Center Example (StationSim)

System Grammar

On the higher level of zoom, one can define a grammar for the entire system, where the stations are abstracted into a small collection of symbols, represented by the rail lines flowing through it. The grammar for this level is defined in the following diagram, and contains similar concepts as the grammar above. It is important to note that this grammar can effectively be considered part of the overall grammar along with the grammar above, and in effect, there is a singular grammar defining all of the system layout, regardless of zoom level. Just as phonetic grammars within individual words are technically part of the English grammar, it is easier to consider them separately from sentence grammar, and as such, we have split them out for description here as well.

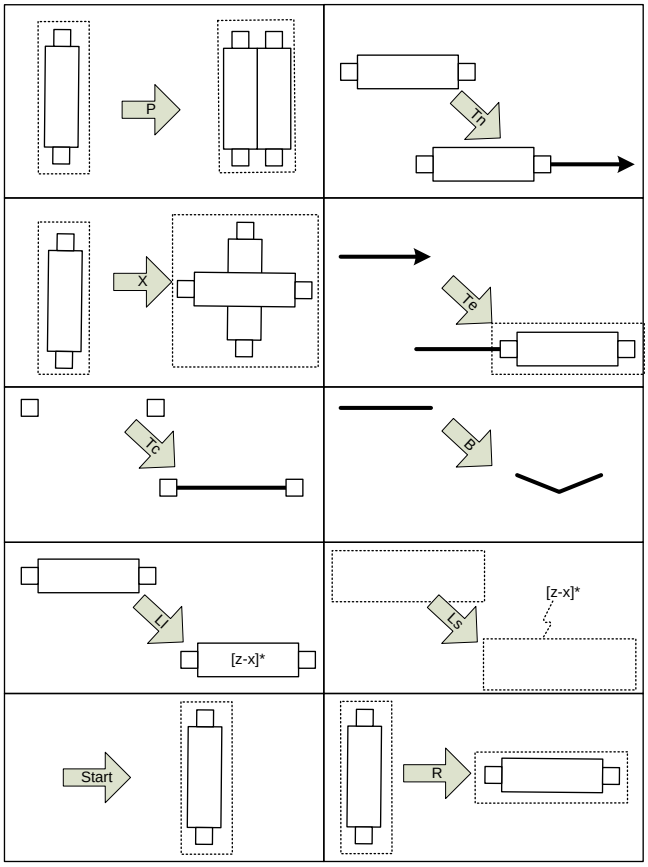


Figure 4. System Grammar

When this grammar is instantiated, the entire system can be defined clearly. Stating with a single track of a station, a system can grow outward to define the entire system. Shown below is one section of the DC Metro built using this grammar. In the directions of the arrows, the grammar can expand through application of Rule Te, and the stations can expand via rule P or X, for example. In this case, the stations are already defined by their logical bounds (their actual physical characteristics), which brings up a critical insight in this generation grammar. The grammar can be used to describe generally any rapid transit system, but must be applied logically to constitute a particular system according to reality. For example, one would not continue to add parallel tracks to Federal Center station in the instance below, even though the grammar supports it.

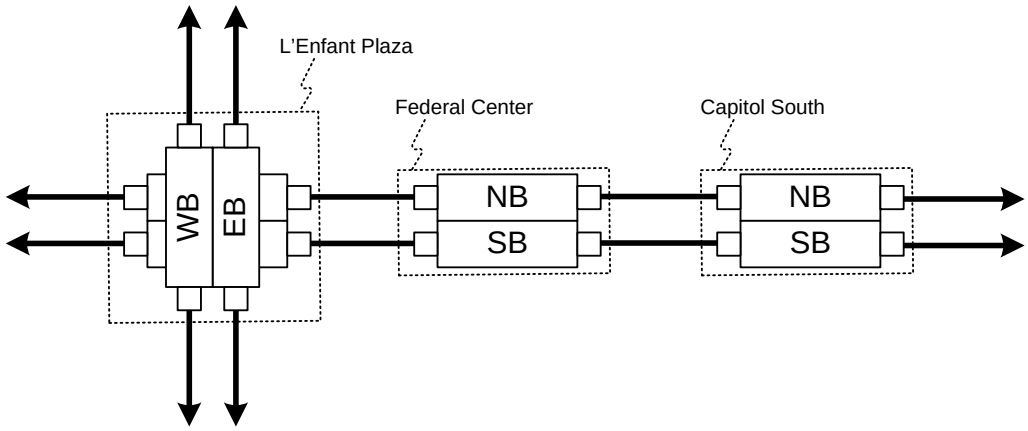


Figure 5. System Instance

Formal Neighborhoods

With the architectural paradigm of rapid transit systems defined, it became evident that there were logical subdivisions of the networks that could be defined, which had several purposes described later. The concept of a neighborhood can be defined in different ways for differing domains depending on the need for such groupings, such as operations control or distributed simulation. The concept of neighborhood has over the course of system evolution been operationally defined as part of logical procedure and administration. The London Underground, for example, maintains a separation of management among various lines: different lines are managed by different authorities.

Graph Theory

In graph theory, the neighborhood is defined in relation to vertices and edges: there exists a given neighborhood or set of neighborhoods for a particular vertex. Depending on the level of paradigm zoom, a vertex in our paradigm can represent various elements of a rapid transit network. Most of the time, however, the vertices either represent entire stations, or atomic station elements.

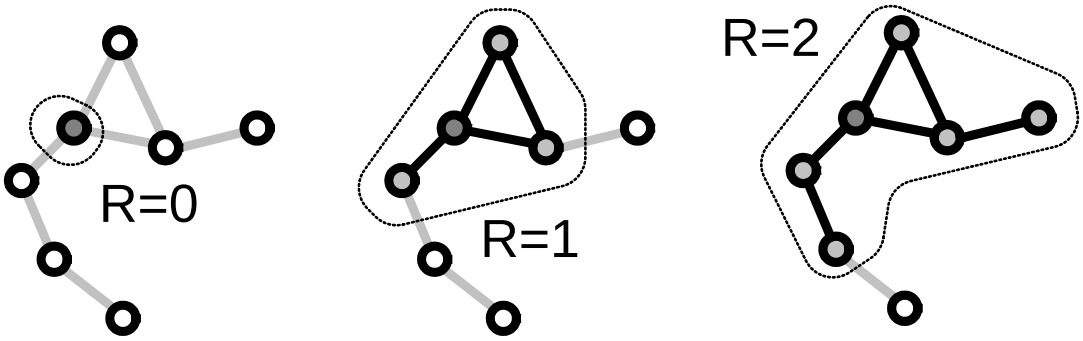


Figure 6. Graph Theory Neighborhoods

Neighborhoods within the rapid transit system defined in this manner can be made easily and in line with the existing network-like feel of the system maps currently accepted as typical representation of the systems. In this example, the DC Metro will be used and in particular, Metro Center is presented as the station of interest for which the neighborhoods will be defined. Shown below is the rather simplistic neighborhood of radius 1, which includes Metro Center itself, the four next-stop stations, and the tracks which connect them.

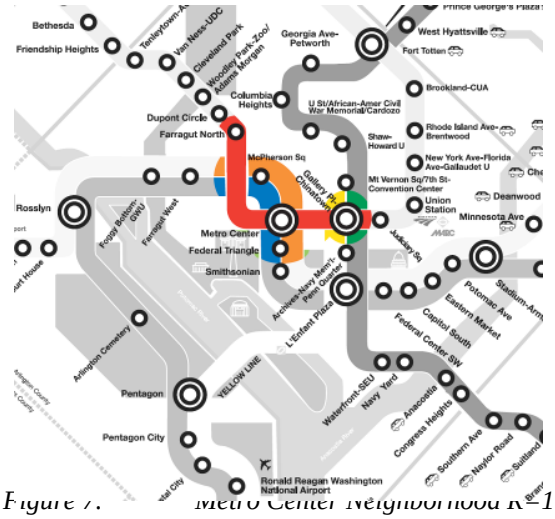


Figure 7. Metro Center Neighborhood R=1

The case of R=1 is rather simplistic, and to really push the definition of neighborhoods in this domain, greater R values must be considered. The following diagrams explore two considered options for the behavior of this expansion. The first, shown on the left, illustrates only expansion on existing tracks that are part of the neighborhood, while the second, on the right, illustrates allowing expansion along other lines as well. In the left, the green line and yellow lines are not part of the original neighborhood, and are therefore not included in the extension, while on the left, they are considered in a more general sense and are included within the range.

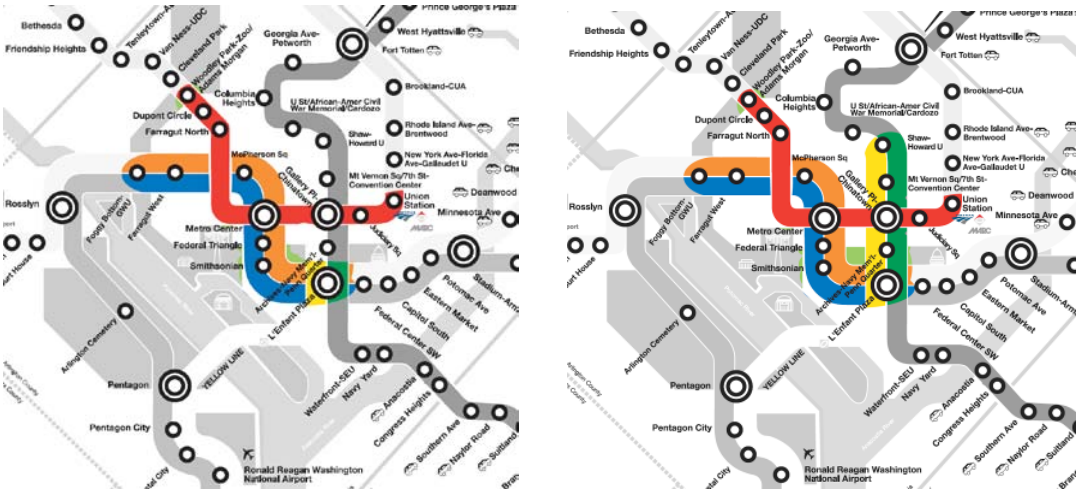


Figure 8. Metro Center Neighborhoods R=2

The first diagram illustrates the reason why the first neighborhood definition is limiting: in this instance, the Archives Station was orphaned and formed into an island neighborhood completely surrounded by the much larger neighborhood colored in the diagram. On the right, the definition restricts such orphans.

Following this concept through to an entire system, any number of partitions can be created. Shown in the following diagram are two such partitions. In the first, smaller neighborhoods are created, each notionally of radius 1, which results in approximately 30 neighborhoods. In the second, larger neighborhoods, of radius 2, are defined and results in a smaller set of neighborhoods, each comprising a larger area.

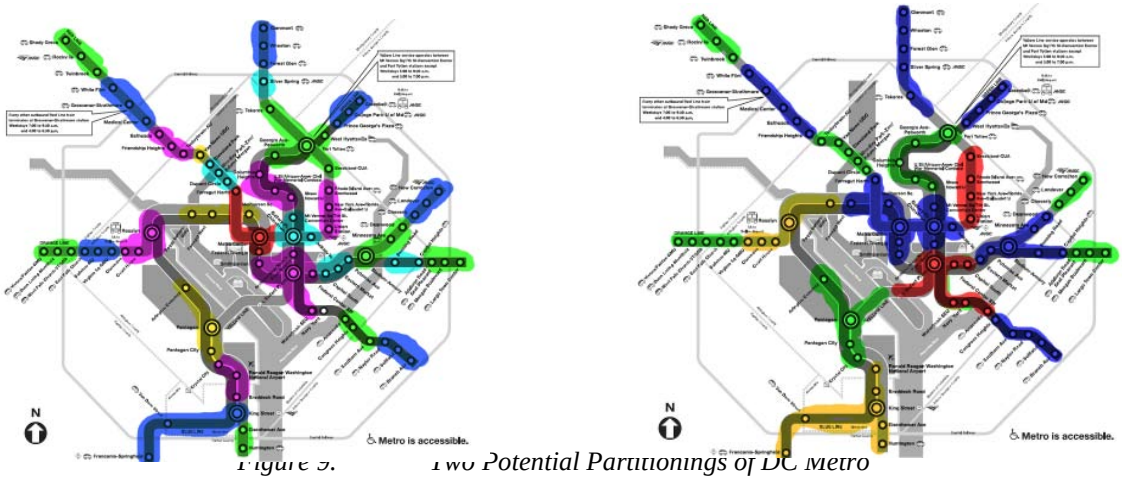


Figure 9. Two Potential Partitionings of DC Metro

Neighbor Interfaces

The interface between two neighborhoods, when defined in the preferred manner (logical splits along connecting tracks), becomes clearly defined and an excellent point of interaction among neighbors to support data sharing and parallel processing, as discussed later.

The interface of such neighbors is along a single strech of track, and therefore, under normal circumstances, riders are encapsulated within trains, and therefore only trains themselves are passed among neighborhoods. As such, along a single point of interaction (a stretch of track), a neighborhood would pass a train, at a given speed or characteristics, along with its contents (various riders with varying goals) to the associated neighbor for that track. Likewise, neighborhoods can accept such trains only at these shared track positions. This well defined interface is much more effective than, for example, splitting the neighborhood down the middle of a station where there is then numerous points of interaction including multiple station elements.

Cellular Automata

In cellular automata, or more generally in grid-space, the Moore neighborhood[G3] is defined as the cells adjacent in cardinal directions and diagonally, out to a given radius R. Shown below are the Moore neighborhoods of a grid cell at a given value of R.

From Wolfram Mathworld
Moore Neighborhood: A square-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0,y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid. [G3]

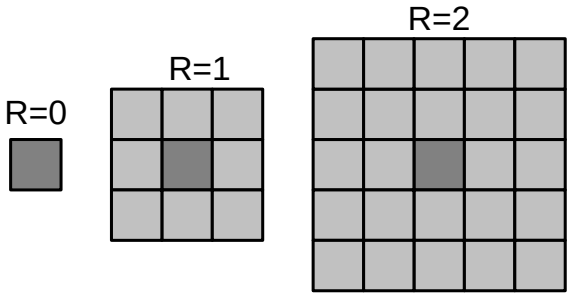


Figure 10. Moore Neighborhood

Another type of neighborhood is common: The following diagram illustrates the concept of the von Neumann Neighborhood with various values for radius. In this model, diagonals are no longer considered adjacent [G2]. This model is perhaps more applicable, as it will become apparent, to rapid transit systems as the interfaces are defined by tracks, which can be modeled to not pass over diagonals.

From Wolfram Mathworld

von Neumann Neighborhood: A diamond-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0,y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid. [G2]

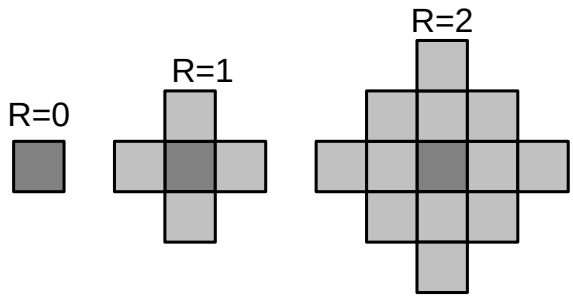


Figure 11. von Neumann Neighborhood

When considering such cells in relations to the previous discussion on graph theory, it is helpful to observe the associated graphs of the Moore and von Neumann interaction spaces as illustrated by the graphs below. For using the graph theoretic approaches to neighborhood definitions, such a neighborhood is only directly useful in the traditional graph theoretic approach when the rapid transit network happens to follow one of the graph layouts shown below. The first graph is known as the Knights Graph [G4].

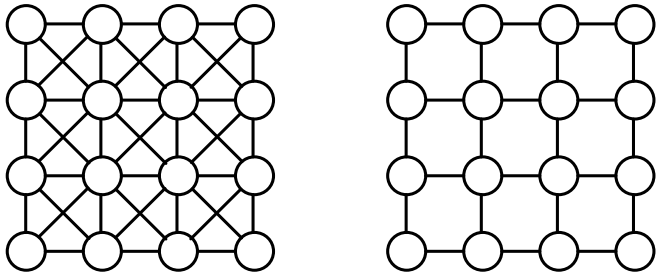


Figure 12. Kings Graph (Left) and Square Graph (Right)

In practice, no rapid transit system follows such a grid-like pattern inherently to its track layout, and a different strategy can apply. In the application of cellular automata to our rapid transit paradigm, we use the London Underground as an example. Shown below is the map from which this partitioning will be performed. One could use the actual geographical track layout, or a symbolic layout as shown here, using the same scheme described here.

Figure 13. London Underground Map

Taking this map, one can apply a grid as an overlay to define the cells of the system. This cell topology is shown below for the map above. A simple square grid is overlaid on the map to create the partitions.

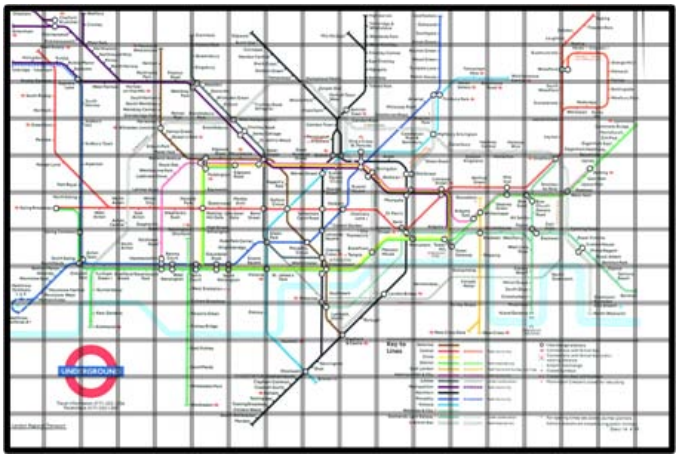


Figure 14. Cellular-Partitioned London Underground Map

These clearly demarked grid lines form the basis of the neighborhoods in the cellular automata domain. Each cell can be classified by its level of involvement. In the sample partitioning below, grey represents cells with no track or station information to represent, and are therefore passive and need not be simulated as they contain no elements. Red cells have particular high levels of components and might, in a given deployment, be simulated on a dedicated machine whereas several green cells could be aggregated into one simulation instance or area of study.

Figure 15. Cellular Zones Flagged for Complexity

It is very important to note that such partitioning can be made via the symbolic map representation, such as in the above example, or by the actual physical topology of the track. The concept can extend to other adjusted representations as well, but in the cellular automata domain, the resulting cells are always arranged in a grid with no more than 4 active neighbors.

As shown below, the points at which the tracks or lines go between cells represents the interaction among the cells in the grid. In some cases, there are multiple tracks, in other cases, there are none. Obviously, when a green (normal cell) is adjacent to a grey cell, there are no interactions at all. This neighborhood definition interaction logically corresponds to a subset of the von Neumann Neighborhood of radius 1. That is to say that a given cell has interaction with zero to four of the cells adjacent to it in the cardinal directions. The interfaces remains exactly the same as in the graph theory model, so long as, just as in that model, splitting cells along the middle of stations is not permitted (a station would be logically adjusted to be entirely within a cell, a trivial and non-affecting exercise, given the map is already logical in the spatial context).

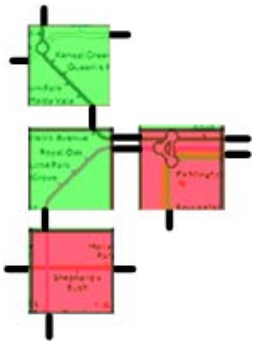


Figure 16. Example Interfaces between cells

In real applications, however, a rapid transit system partitioning into neighborhoods may be more apply tied to operational considerations as well, beyond those expressed in the physical track/station topology. In the very example used here, the London Underground, the system is actually operationally split by various managing units, and so such considerations could dictate how such partitions, whether graph theory based or cellular automata, play out.

Neighborhood Applications

There are important implications in the architecture paradigm's support of neighborhoods: primarily in the computational separation of the neighborhood behavior. There are two methods of interaction

among neighborhoods: passive data sharing and active execution. In passive data sharing, a given neighborhood would generate the list of trains and times to pass to the neighboring node, and then the other neighbor would use that information to run its own simulation.

In active execution, multiple instances of StationSim would pass data, during execution, to the neighboring regions according to a common synchronized clock. This concept is illustrated below.

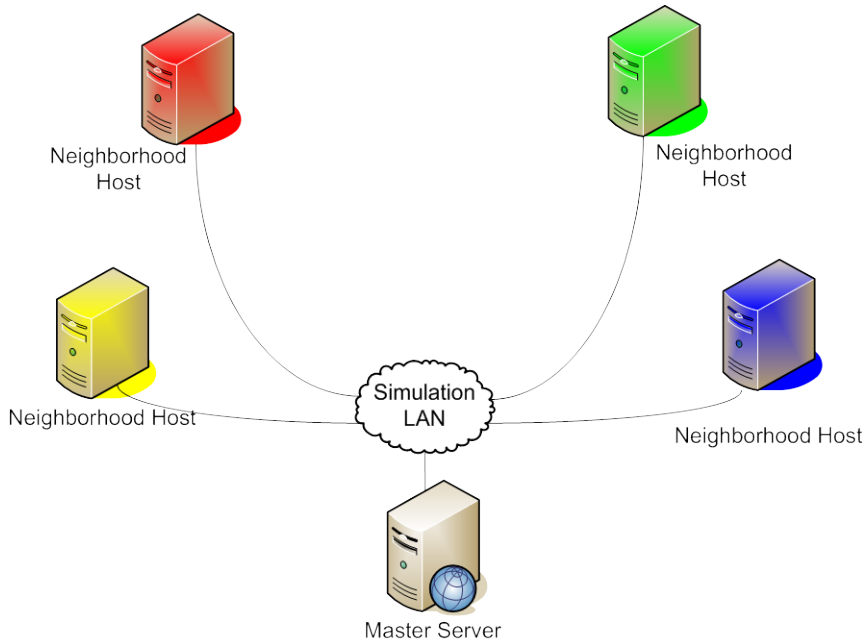


Figure 17. Potential Parallel Processing of System Simulation

Additionally, all of the neighborhoods have similar interfaces to the overarching environment. There is an aspect of time synchronization to this entire process. These include actual coordination such as closing the system at the end of the night or during an emergency from the control center, as well as simulation controlled actions, such as starting and stopping the simulation.

Fortunately, the size of the neighborhoods, which becomes inversely proportional the number of neighborhoods for a given system, can be optimized to match up the degree of parallelization supported by the multi-threaded environment.

At other zoom levels, similar constructs can apply, but this concept of partitioning was seen as most applicable to broad range testing simulating flows of riders throughout the entire rapid transit system efficiently over a distributed simulation platform.

Grammar and Neighborhoods

Finally, the formal shape grammar can be modified for the generation of formal neighborhoods at the station-system zoom level. Shown below are the rules for this shape grammar, noticeably similar to the shape grammar for the system as a whole without limitations on neighborhoods.

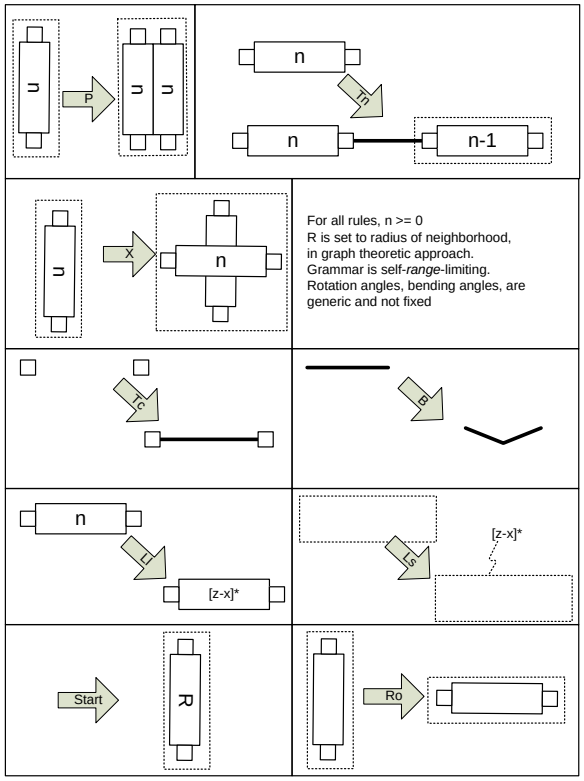


Figure 18. Neighborhood Grammar for System Zoom Level

This grammar can then be applied for the purpose of defining an arbitrary size neighborhood of a particular station. The following diagram illustrates how this grammar does so for an R=2 neighborhood for a basic station which is connected to a large transfer station. In this grammar, the stations adjacent to the transfer stations, even though they are on another line, still are part of this neighborhood.

This grammar is similar to the grammar for the entire system, but includes extensions to limit the definition of the grammar to a given range of interest. Once the “r” value of a given station reaches 0, it can not be defined outwardly anymore, and the grammar terminates in terms of adding new stations. Adding additional tracks to stations is not syntactically bounded, but is logically bounded by the description of the systems themselves. This is analogous to how, technically, English grammar allows for the sentence “I ran very very very… very fast” with infinite instances of the adverb “very”, but the overarching goals restrict the logic to only the desired components.

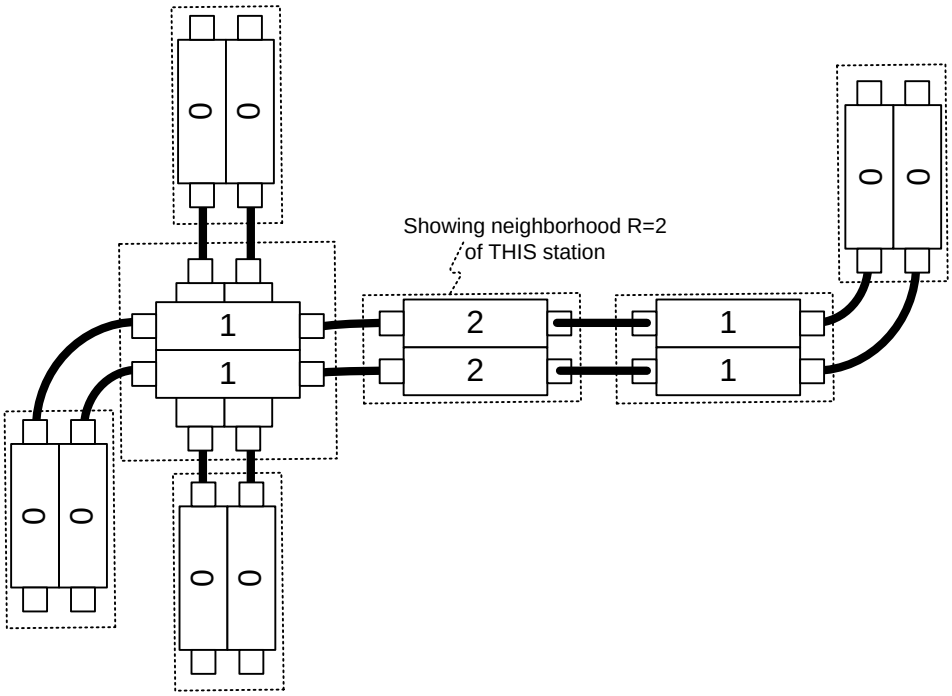


Figure 19. Neighborhood Grammar Instance (R=2)

References for Appendix

[G1] Sakharov, Alex. "Grammar." From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/Grammar.html

[G2] Weisstein, Eric W. "von Neumann Neighborhood." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/vonNeumannNeighborhood.html

[G3] Weisstein, Eric W. "Moore Neighborhood." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/MooreNeighborhood.html

[G4] Sloane, N. J. A. Sequences A002943, A140521, and A158651 in "The On-Line Encyclopedia of Integer Sequences."

Architecture for the Modeling and Analysis of Rapid Transit

A-MART

Minor Appendix B
Domain Specific Modeling Languages Using the Generic Modeling Environment (GME)

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

GEORGE MASON UNIVERSITY

SYST 798/OR 680

Introduction

In formalizing the grammar of our paradigm, an excursion into the study of Domain Specific Modeling Languages (DSML) was seen as a way of presenting this grammar for future work or for additional explanation. To create a DSML for the rapid transit paradigm, the Generic Modeling Environment (GME) was chosen. GME is produced by the Institute for Software Integrated Systems (ISIS) at Vanderbilt University. GME is driven by UML-class diagram notation along with constraints for the language/grammar in OCL. The resulting language is a carefully tailored toolkit for expressing a given instance of a particular domain. Domain Specific Modeling Languages are a relatively new field, and one that has great potential in the software and systems engineering domains. The following is a definition provided by the DSM Forum.

"Domain-Specific Modeling raises the level of abstraction beyond programming by specifying the solution directly using domain concepts. The final products are generated from these high-level specifications. This automation is possible because both the language and generators need fit the requirements of only one company and domain. Your expert defines them, your developers use them. Industrial experiences of DSM consistently show it to be 5-10 times faster than current practices, including current UML-based implementations of MDA."

- From the DSM Forum [1]

Very uniquely, GME allows the users to create languages using a meta-language itself - a language created within GME that is in fact self-defining. This language allows users, such as team A-MART, to construct, in the meta-modeling grammar, the language they are attempting to represent. The following illustration presents this idea.

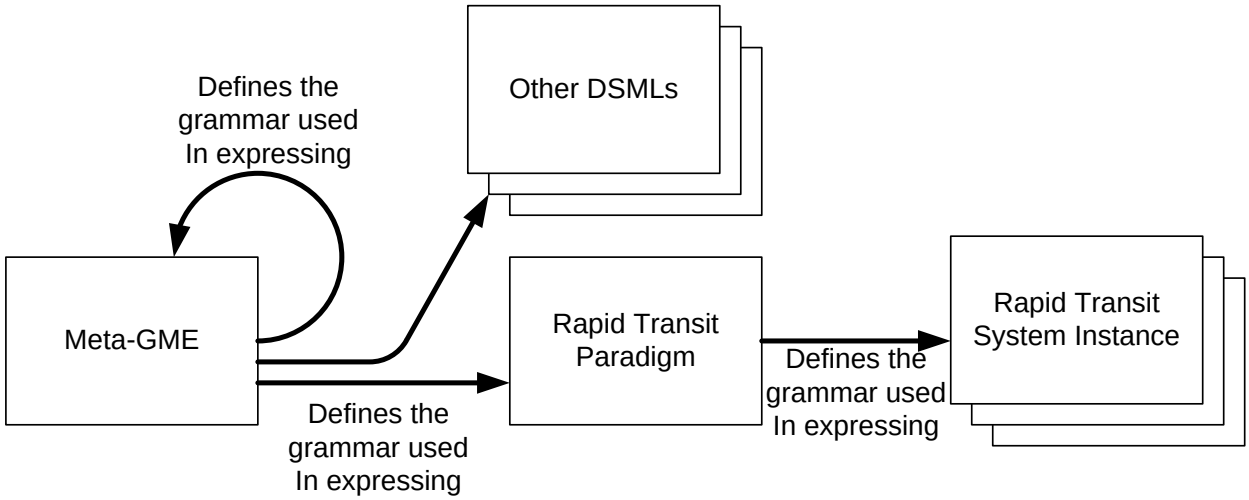


Figure 20. Self-defining META-GME with Paradigm

A full introduction to GME, and especially the learning curve for creating your own instances or languages, is well beyond the scope of this appendix, but Vanderbilt University has gone to great lengths to supply supporting documentation for its use. This appendix walks through the process by which we represented our paradigm in an abbreviated manner to serve as an introduction for further work.

Information on GME, and the installation files for installing it, can be found here [2]:

http://www.isis.vanderbilt.edu/projects/gme/

To begin the process, a DSML of the rapid transit paradigm, based on the SysML relationships and diagrams, was defined within GME using the META-GME paradigm as our language. Shown below is the user interface, and part of the DSML under construction.

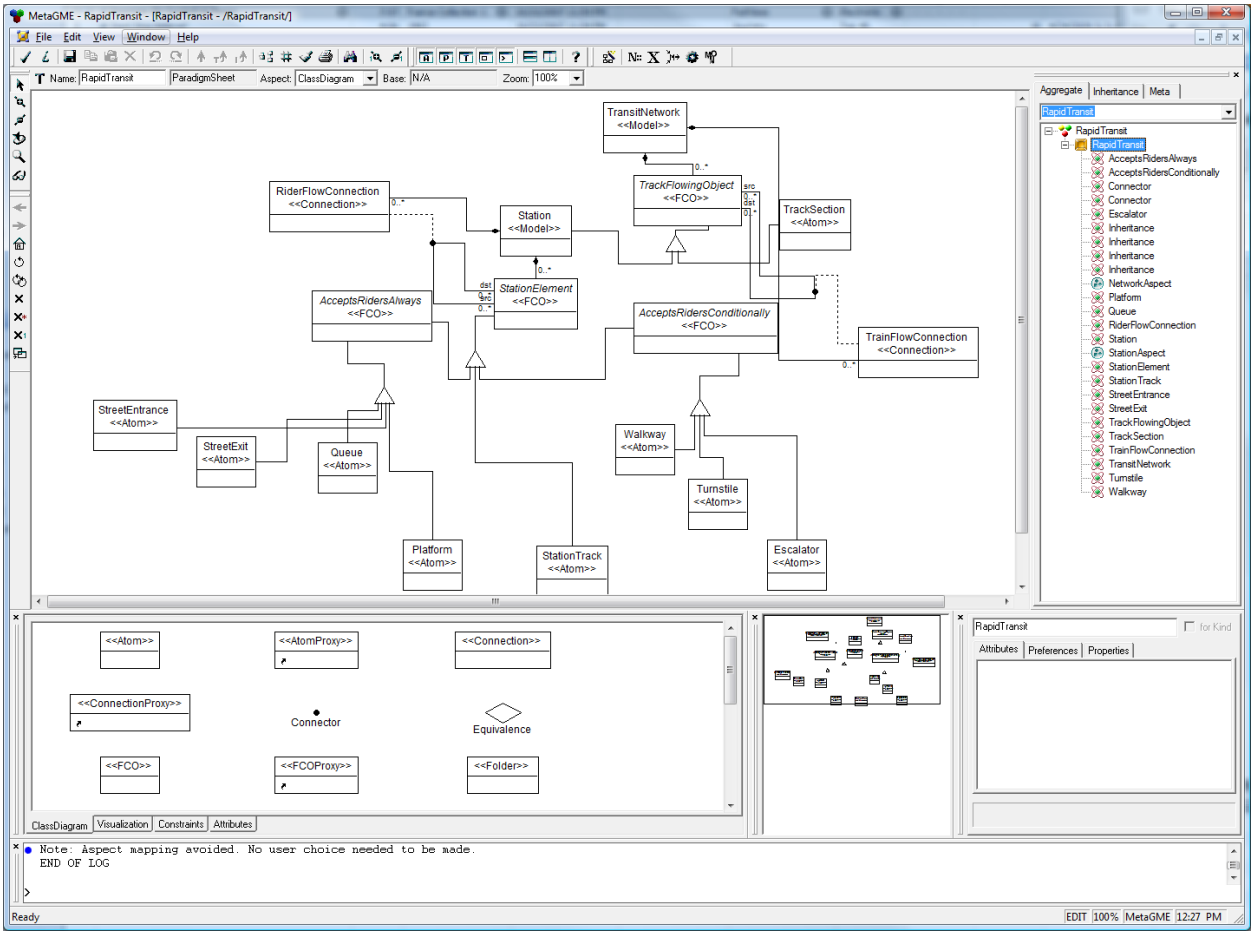


Figure 21. The interface for creating a new DSML

Finally, the resulting language definition is shown on the following pages. First the class diagram description is shown, followed by the aspect mapping, which is a concept described within the GME documentation. Additional aspects of the language can be modeled, but it was beyond the scope of this semester project to dive into a more detailed model.

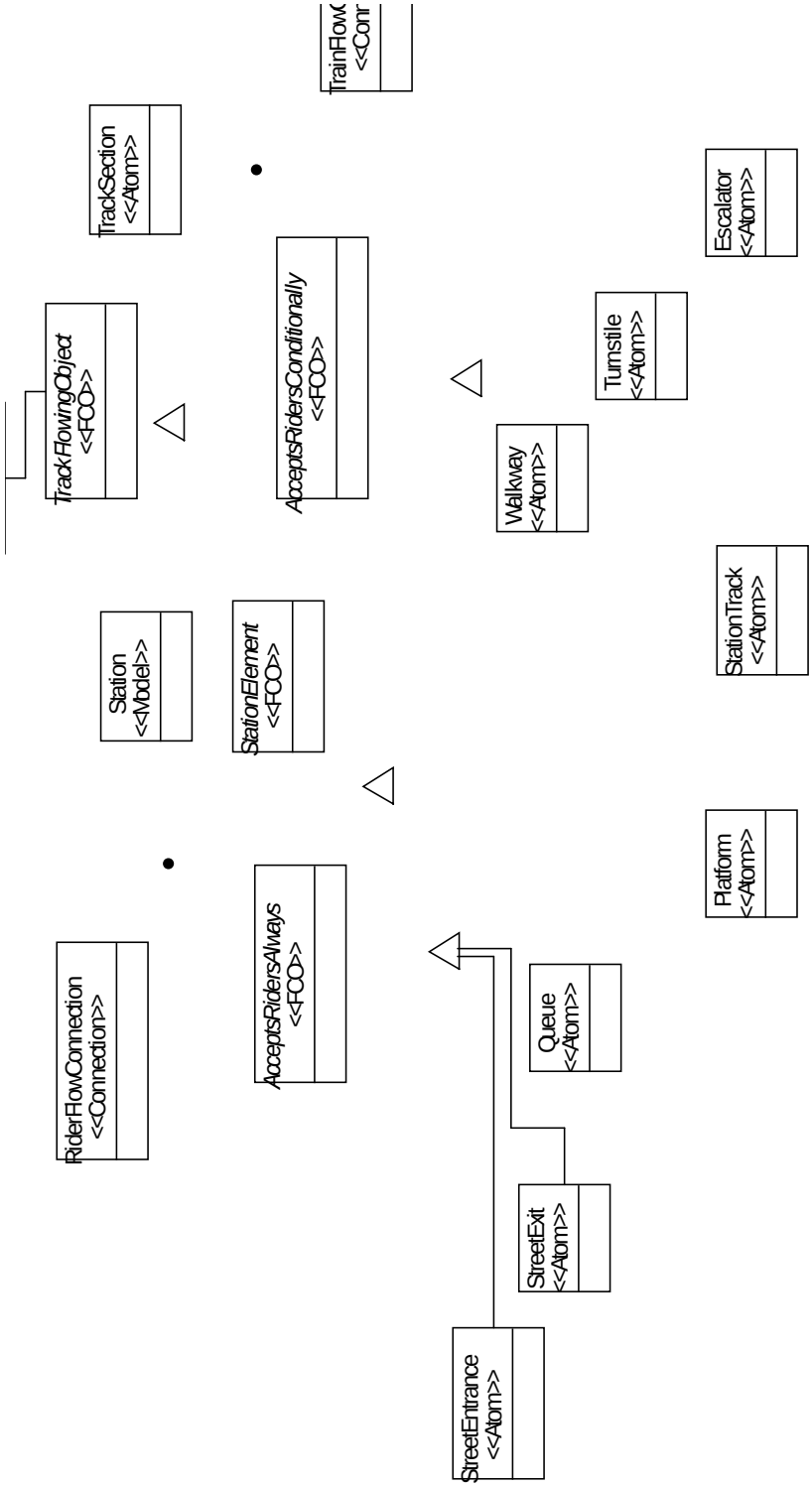


Figure 22. One of the supporting aspects of this DSML, represented in META-GME

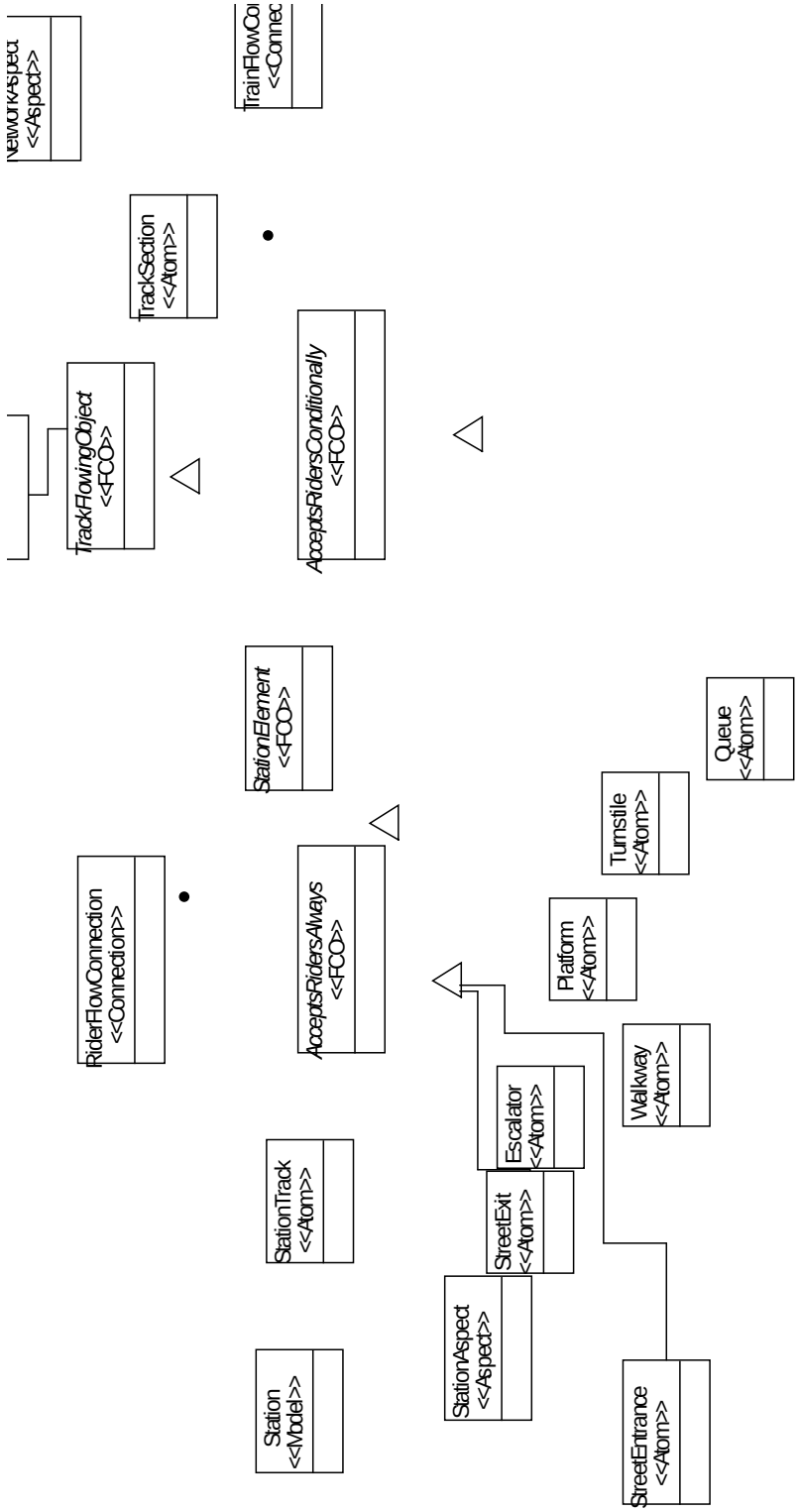


Figure 23. A second supporting aspect of the DSML for systems and stations in META-GME

With the paradigm defined, it was compiled into a usable language. Now within GME, the compiled domain specific modeling language we created is selectable as a language from which models can be created. We created such a model, and an instance of the model is shown illustrating the system-wide model scope. Notice that the tool-palette on the bottom left contains station elements that can be dragged and dropped into the main diagram window. A model hierarchy is displayed on the right.

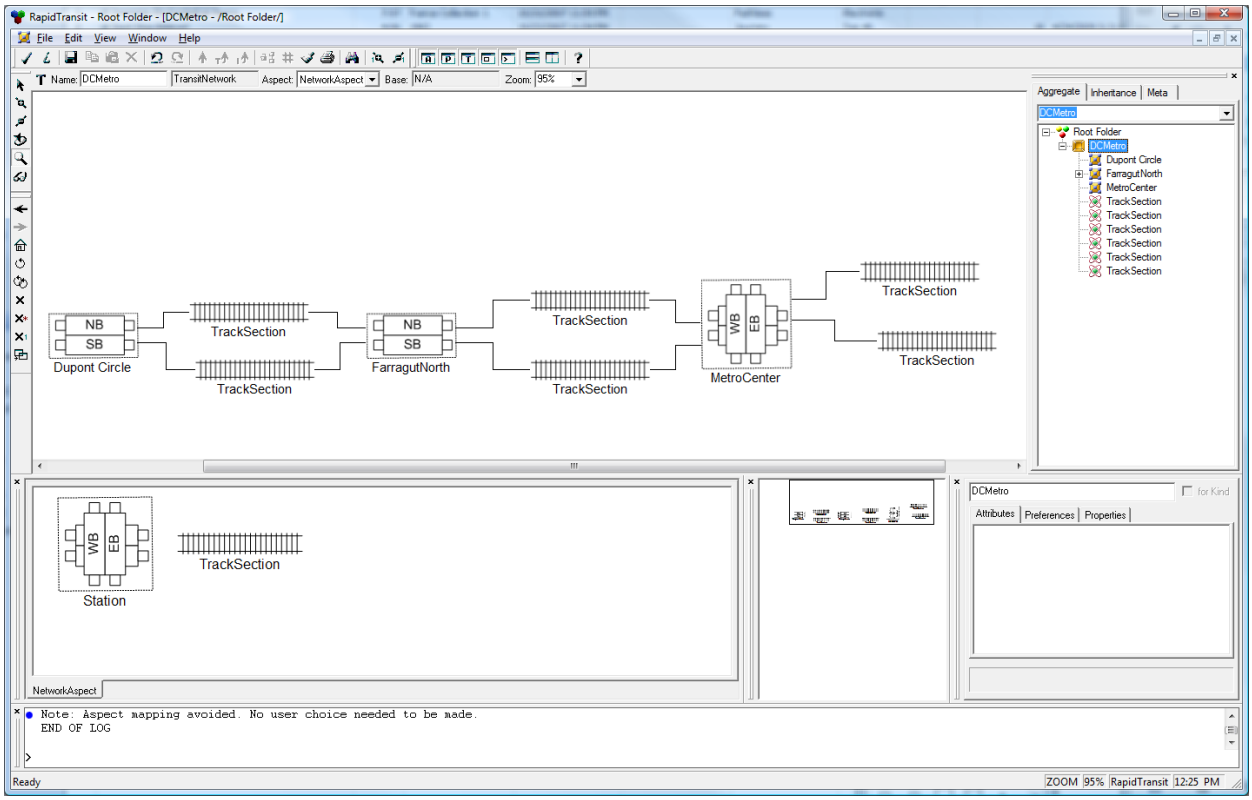


Figure 24. User Interface for using the DSML for System-wide Models

Within the same model, when decomposing the system down to one of the stations in the previous diagram, one can create a station model using the palette of station elements, as formally defined in the modeling paradigm. The connections between the station elements are created as part of the model, and the restrictions on what can connect to various objects are part of the paradigm itself.

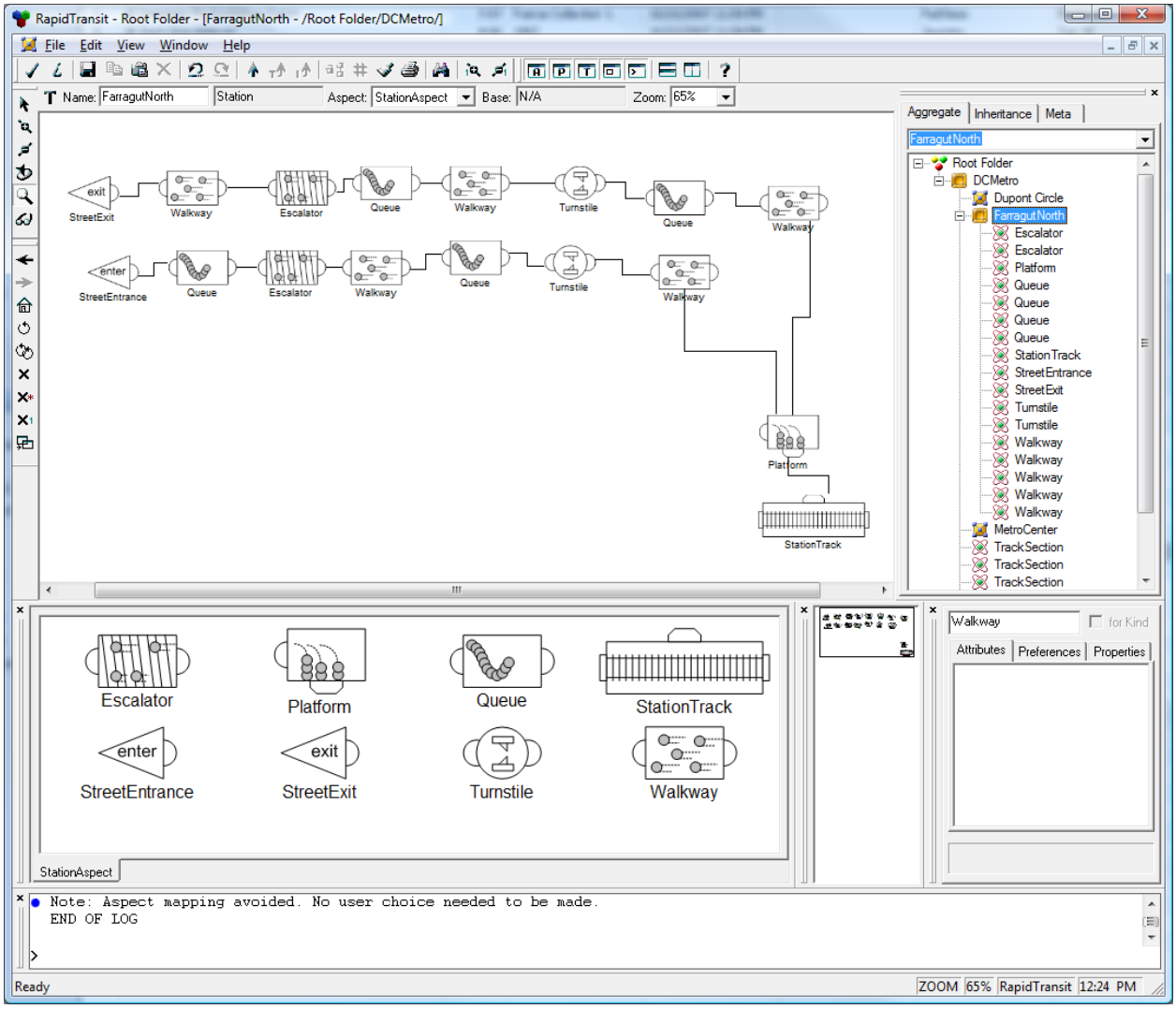


Figure 25. User Interface for using the DSML for Station Models

The following pages illustrate expanded views of the instances of both the system and the station of the paradigm instance.

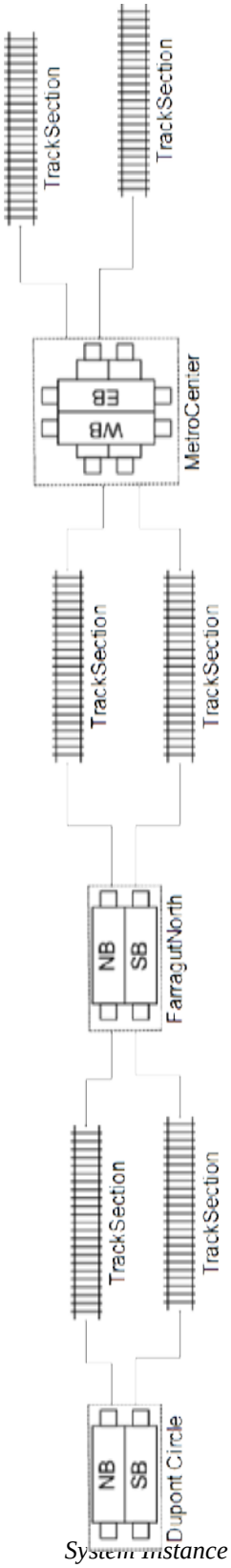


Figure 26. System Instance from GME

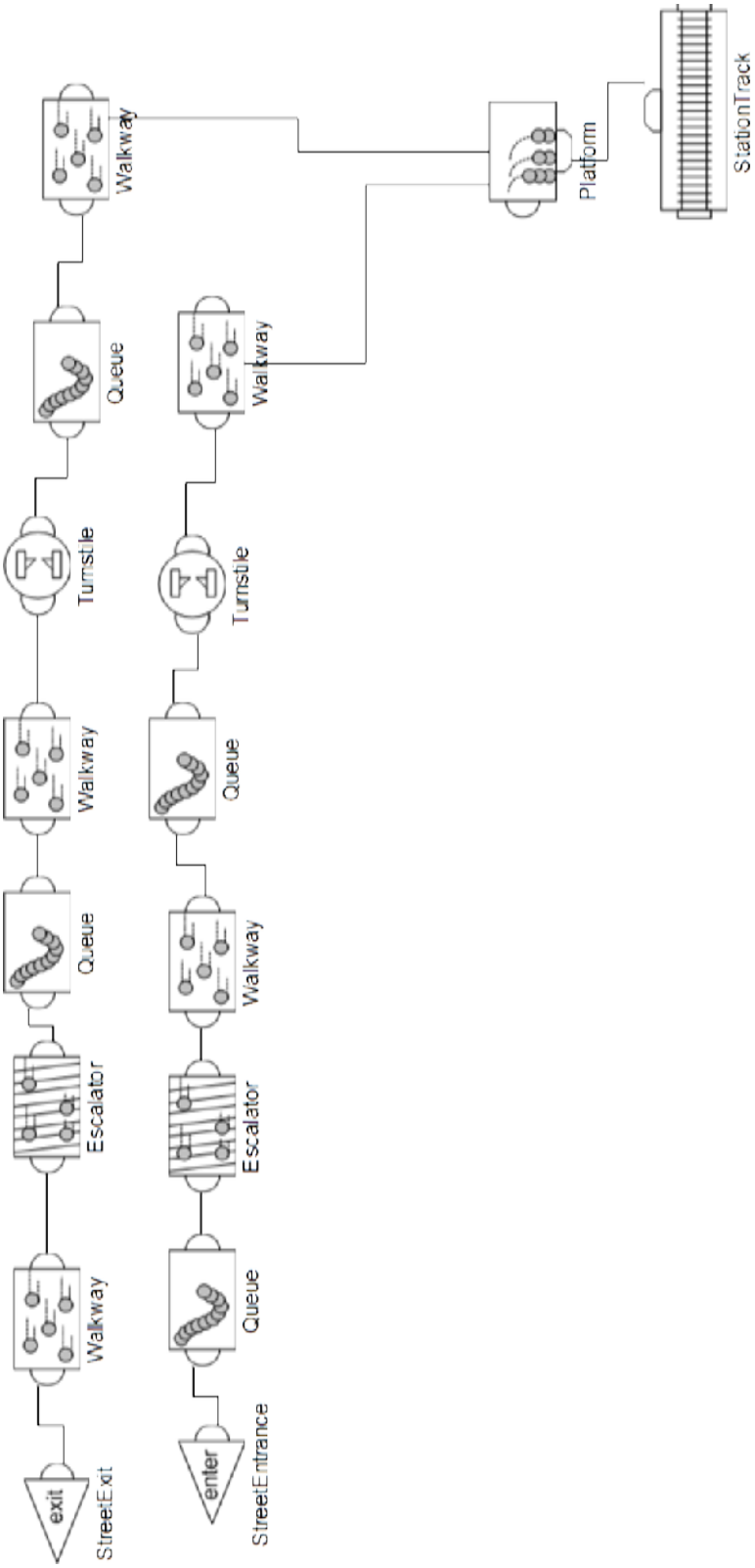


Figure 27. Station Instance from GME

Conclusions and Way-Forward

As demonstrated our paradigm can be formally expressed as a DSML within GME, but the GME tool was not chosen as the focus of our project because David Claypool was the only one on the group with experience with DSML or GME. We have provided the foundations from which our entire model could be represented within GME, and used for formal analysis and other machine-readable extensions (Such as automatically exporting to StationSim from this formal definition).

It is important to note that in the process of formalizing our architecture via StationSim, we in effect replicated the same functionality seen here, but in a domain-specific application. Now, we do not face the learning curve of GME, which in of itself can be expected from such a capable tool, but is not feasible in the time constraints of a semester project. By using our own tool, we were able to get immediate usability by not only our group, but anyone picking up StationSim.

References for this Appendix

[1] DSM Forum. http://www.dsmforum.org/

[2] Vanderbilt ISIS. http://www.isis.vanderbilt.edu/projects/gme/

Architecture for the Modeling and Analysis of Rapid Transit

A-MART

Minor Appendix C
Supporting CPN Preliminary Work

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

GEORGE MASON UNIVERSITY
SYST 798/OR 680

Introduction

As part of our evaluation, we developed the start of CPN models for the station paradigm. One team member developed preliminary models in CPN Tools for a simple station section: A queue leading up an escalator and the escalator itself. This model was used to judge the usability of CPN for this project by both the team members and future users. Since only one team member had experience with CPN tools and the principles and capabilities of CPNs were not as well suited to the complex nature of the rider flows, CPN was not preferred. Nevertheless, the CPN investigation provided valuable insights into both the paradigm definition and the StationSim programming. See the concluding section of this appendix for this rationale. Shown below is the high level view, comprised of the queue and escalator compound transitions forming the top layer of a hierarchical net.

The Resource token tells the queue when it is acceptable to send in another rider into the escalator. This artificial construction is necessary in modeling this as a CPN, but is not at all present to the user in StationSim.

Riders in this model are rather simple, and simply include the goal statement and identifier. As demonstrated in our paradigm and instantiated in the class definitions within StationSim, the actual rider is a highly complex data structure that would have been very obtuse to model and display as a token in CPN.

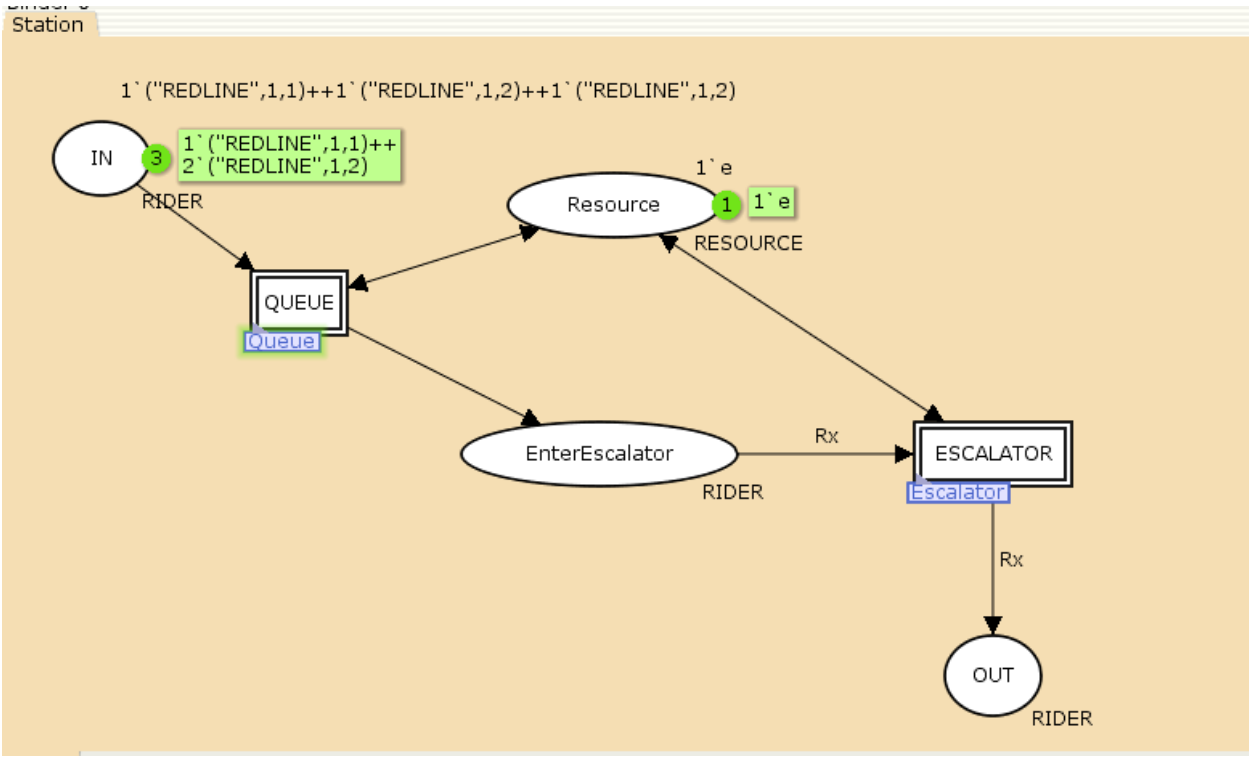


Figure 28. High Level View

The following diagram is a drill down into the escalator structure. Despite the ambiguous nature of its structure, it is in fact far less complete than even the early StationSim representations. It, for example, does not include the complex code that would have been required to model the fact that Riders can only pass when they are not on the same side of the escalator.

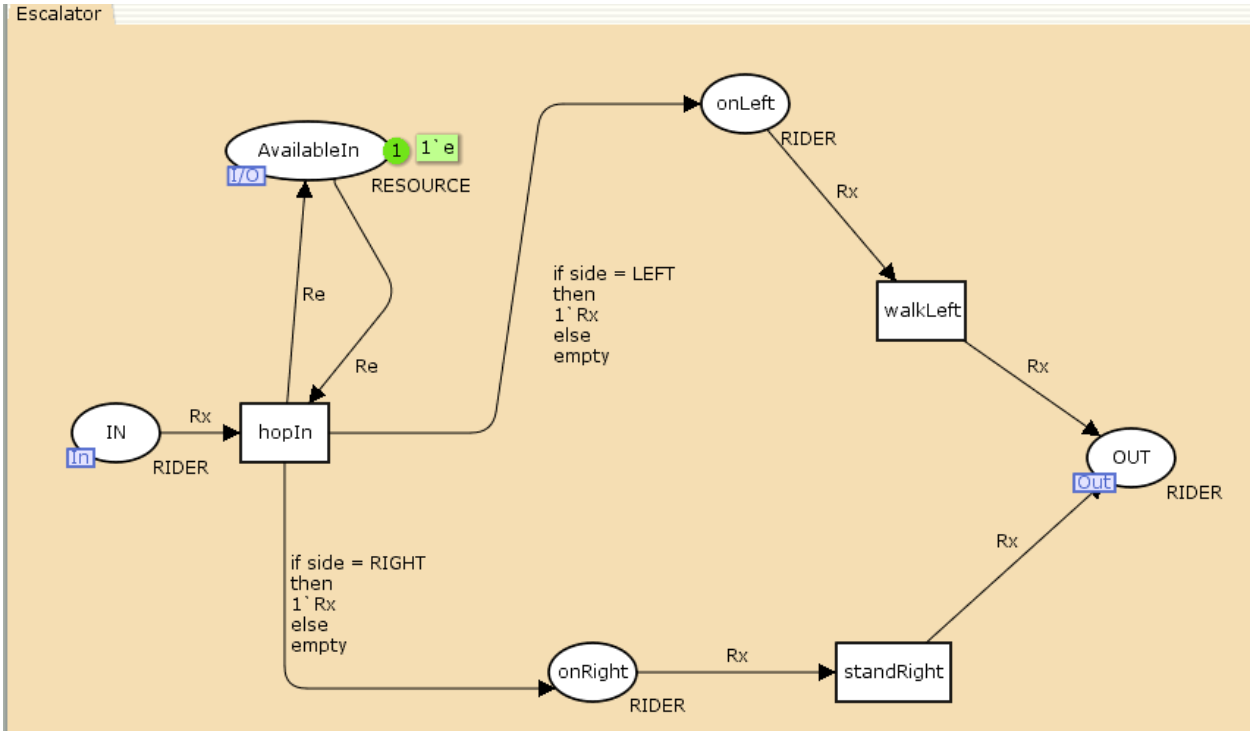


Figure 29. Escalator

The final diagram shown below illustrates the Queue. It is a standard structure in CPN, yet still needs to be modeled for a particular token type, and was enhanced with an I/O place for the resource triggering. The queue can only send out riders when the resource is present in the I/O Resource place.

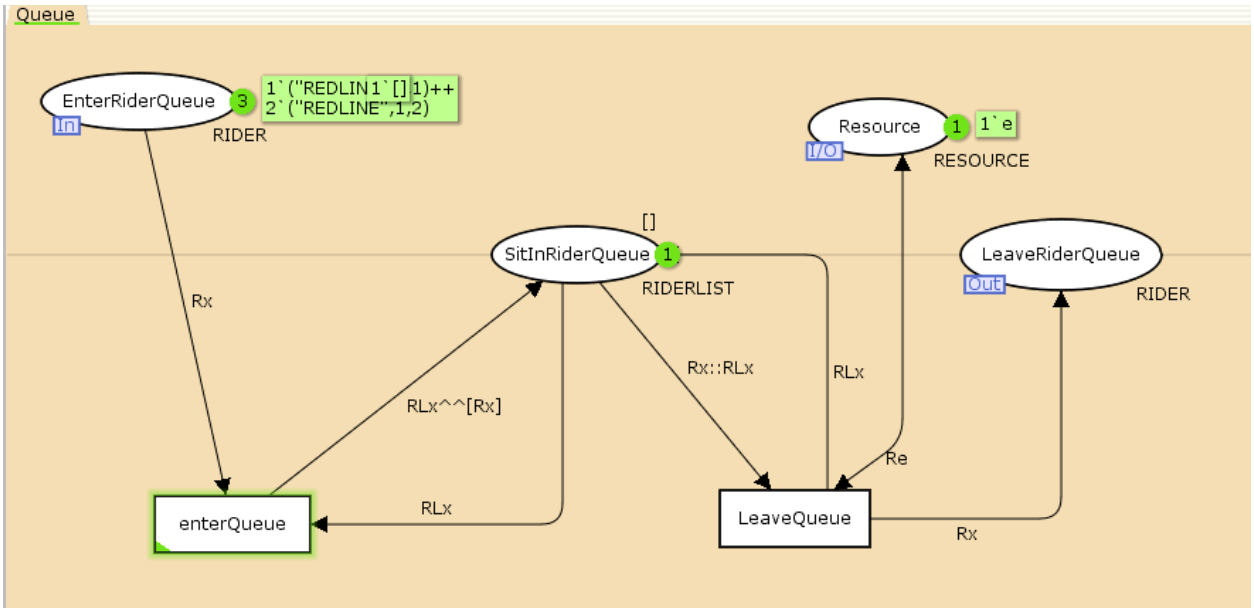


Figure 30. Queue

This effort led to a first cut formalization of our ideas, but ultimately, it was viewed as not the best solution for our purposes for the semester for the following reasons:

- Only one team member had experience creating CPNs and using them, where as StationSim could be immediately used for analysis by all team members
- The amount of ML required to create the more complex behaviors we wanted to model would have taken too much time.
- The CPN would get overly large for a station such as Metro Center
- Since the paradigm and the instances are in effect held in the same file, every time the paradigm was updated, the instances would all have to be updated as well.
- With StationSim, when updates occurred, all instances were also upgraded automatically.

Architecture for the Modeling and Analysis of Rapid Transit

Minor Appendix D
System and Station Background Information

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

SYST 798/OR 680

Introduction

This document talks about the background information about how and why we chose certain systems and stations. In systems part, we will talk about what systems we concentrate on, how we process the information for each system and finally how systems are chosen for selection. In stations part, each of station is described in detail.

Systems

A rapid transit system is an electric passenger railway in an urban area with high capacity and frequency, and which is grade separated from other traffic. Rapid transit systems are typically either in underground tunnels or elevated above street level. Outside urban centers rapid transit lines sometimes run grade separated at ground level. The first rapid transit system was the London Underground, which opened in 1863. The technology quickly spread to other cities in Europe and then to the United States, where a number of elevated systems were built. Since then the largest growth has been in Asia and with driverless systems. Nowadays, there is a various type of rapid transit systems.

Process

In order to make our selection of systems as diverse as possible, we consider the existing rapid transit systems all over the world. Then we define a couple of criteria such as history, track layout, control system, management and reliability etc. to look at different systems. In this case, we make a spreadsheet including all the rapid transit systems we studied which is in appendix E. Finally, we narrowed down our selection down to five based on the spreadsheet,

Systems Chosen for Selection:

Team AMART originally looked at over 30 systems throughout the world and collected information on these systems to include historical significance, operational oversight, infrastructure, ridership, automation, political implications and availability of information. Systems researched included Mexico City, Munich, Naples, Brazil, Chicago, Tokyo and more. From these original 30 systems we narrowed the field down to 5 and conducted even further research to determine which systems were best for modeling and analysis.

This appendix provides a very brief overview description of these systems to aid in understanding the models and the context for their selection.

DC Metro

The DC metro system is utilized by many white collar workers and carries passengers to many special events including concerts, sporting events, the National 4th of July Celebration on the Mall and even the inauguration contributing to unique crowding situations.

Obviously, the DC Metro was highly used by the A-MART team due to its proximity to the university.

Figure 31. DC Metro Track Layouts

Atlanta

Atlanta's transit system is a comparably simple track layout to the other systems studied, however there are several unique attributes. All tracks meet at one center station. The system is extremely overcrowded and there is a wide spread feeling of unreliability. The funding for Atlanta is largely locally funded by surrounding counties and little state funds reach the system. A majority of the riders of Atlanta's transit system are African American and there are accusations that the state is showing discriminatory treatment towards African Americans and lower class workers by providing more money for state highways in the Atlanta region which are largely used by upper-class citizens.

We foresee that platform and station crowding will continue to be an ongoing problem, particular in a concentrated topology such as this with one station in the middle.

Figure 32. Metro Atlanta Track Layouts

Boston

MBTA is Boston's transit system. A distinguishing feature of the system in Boston is the age and diversity of the rolling stock as there are several different makes and models of various ages of trains and cars in use. Another distinguishing feature of MBTA is that the system has constantly been in the process of modernization and transition. There is are serious overcrowding problems in certain stations.

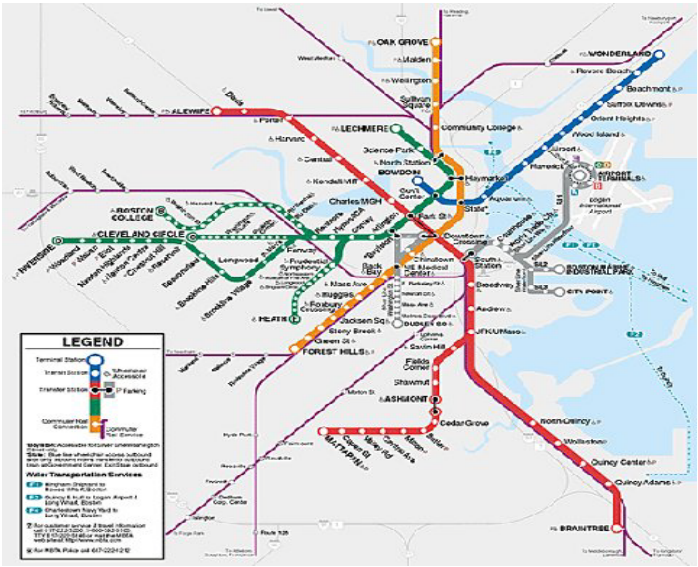


Figure 33. Boston T track Layouts

New York Subway

New York Subway's System has the world's largest single station, Grand Central Station. The whole city is very dependent on the Subway and other forms of public transportation as driving is extremely costly due to parking shortages and the streets are infamously congested leading to unacceptable transit times. Some lines have become saturated and crowding at certain station exceeds design limits.

Figure 34. New York Subway Track Layouts

London Underground

The London Underground was the first underground railway system in the world and is extremely complex. Despite the vastness and abundance of trains (average train inter-arrival time is 2 minutes!) the system is still overcrowded to the point where some stations are made exit-only during certain periods of the day.

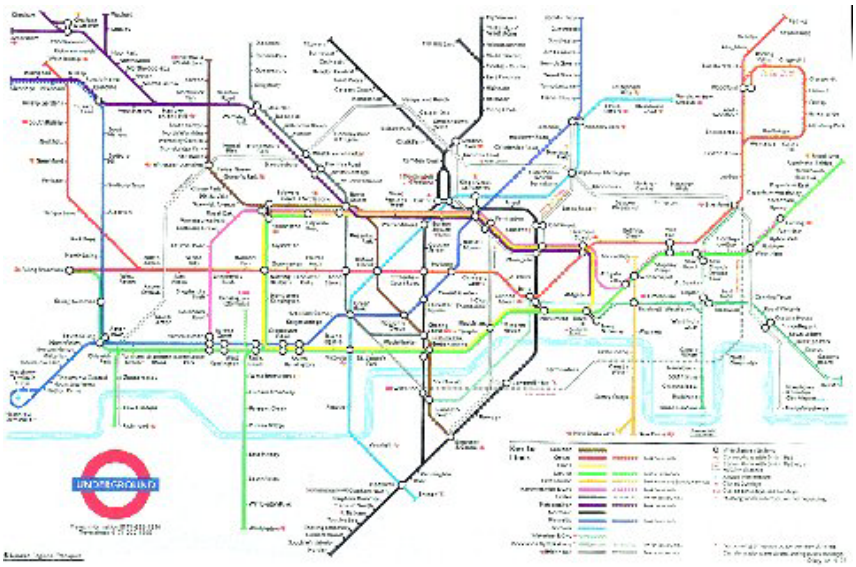


Figure 35. London Underground Track Layouts

Deeper Analysis

These 5 stations were eventually narrowed down to focus on the Washington Metro due primarily to the availability of information and our close proximity which enabled additional research opportunities to gather data and information to properly model individual stations. In addition to the Washington Metro, we used the model to develop an instantiation of Piccadilly Circus. This shows the model's extensibility between different systems and validates our notion of a common architectural paradigm. In addition to this appendix, Appendix E contains the master spreadsheet which shows our complete background research on all 30 systems.

Stations

The following stations were chosen based on complexity, feasibility of information and variety of platform layout. Each station has its own particular platform layout and crowding problem. By modeling different types of stations and doing regression analysis, the relationship of different parameters which will affect crowding could be found.

Union Station

Union Station is one of the busiest stations in the Metrorail system, averaging 33,000 passengers per weekday. Union Station is a located on the Red line. One reason why we chose this station is that it has an island platform which layout is track-platform-track. Union Station only serves red line in DC metro. Serving the Amtrak VRE and MARC lines, as well as a gaggle of bus routes, this station never seems to be empty.

Figure 36. Union Station

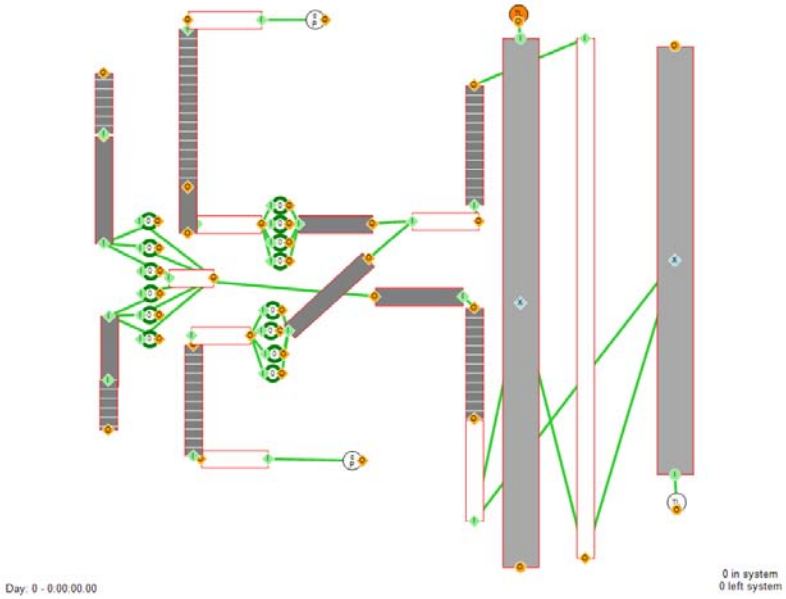


Figure 37. StationSim Representation

Farragut West

Farragut West Station is one of the most heavily-used stations in the entire system and is serviced by both the Orange line and Blue line. With the exception of Metro Center and Smithsonian, more Blue and Orange line riders use this station than any other. Farragut West Station has an alternative platform layout as Union Station since two platforms separate two tracks apart which is platform-track-track-platform.

Figure 38. Farragut West Station

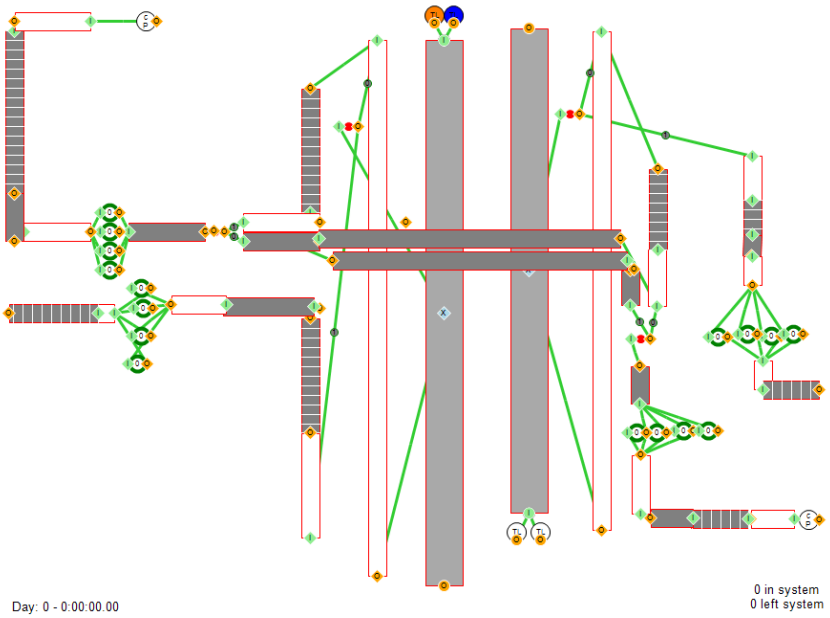


Figure 39. StationSim Representation

Rosslyn

The station is located at 19th and N. Moore Streets in Arlington and is part of a large office, shopping and apartment complex. South of the station, the Orange and Blue lines split, with the Blue line continuing to Arlington Cemetery. Since both the Orange line and Blue line reach Rosslyn Station, this route diverges and the need to keep the lines from separating at grade led to the decision to create the bi-level station. One level is for trains going east and the other is for trains going west. The main reason is that Rosslyn has a unique platform layout which is a bi-level platforms spitted vertically.

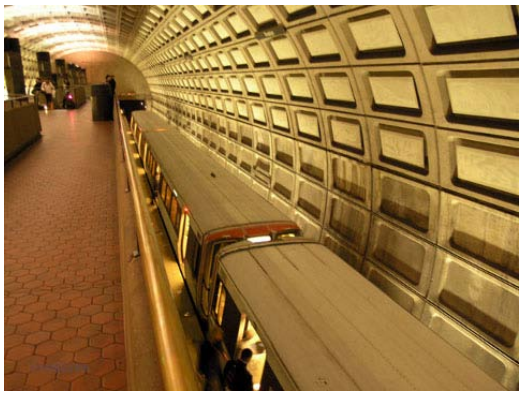


Figure 40. Rosslyn Station

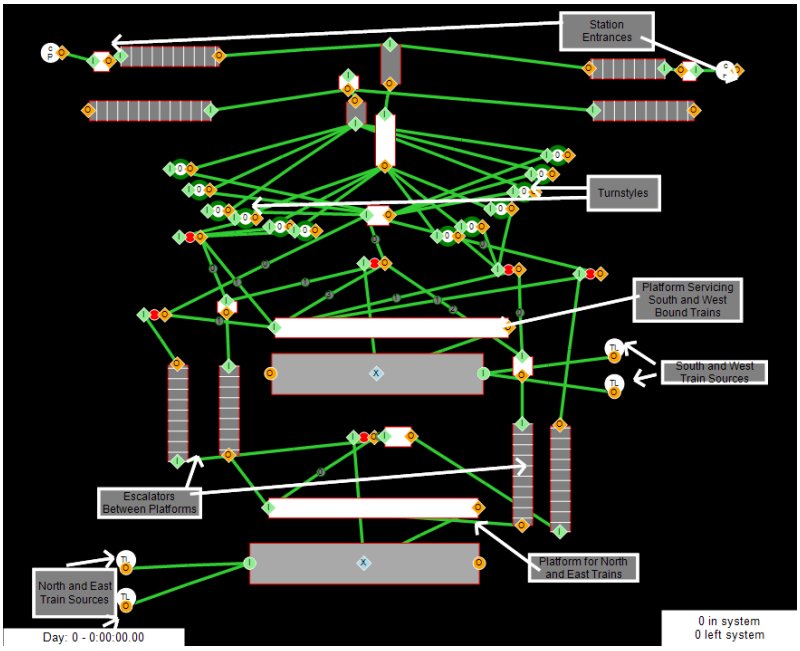


Figure 41. StationSim Representation

Metro Center

This station is located at the intersection of 12th and "G" Streets NW and serves the Red line on the upper level and the Blue and Orange lines on the lower level. The upper level has two bay platforms and the lower has a single island platform. The two vaults intersect right in the station center - 22 coffers on the upper level and 26 coffers on the lower. There are four station exits - along "G" Street NW at 11th, 12th and 13th Streets, and another further south at 12th and "F" Streets NW.

The station gets very busy during the rush hours and severe crowding is a major problem, especially on the lower level's island platform. This station's central location provides easy access to many of the district's tourist attractions (White House, Ford Theater, FBI building, etc.) and there are major stores and hotels nearby. Eighteen different bus routes connect to the Metro here.

The reason why we chose Metro Center is that it is heavily used in the system for transfer lines. It would be a good challenge to model it because of its high complexity.

Figure 42. Metro Center Station

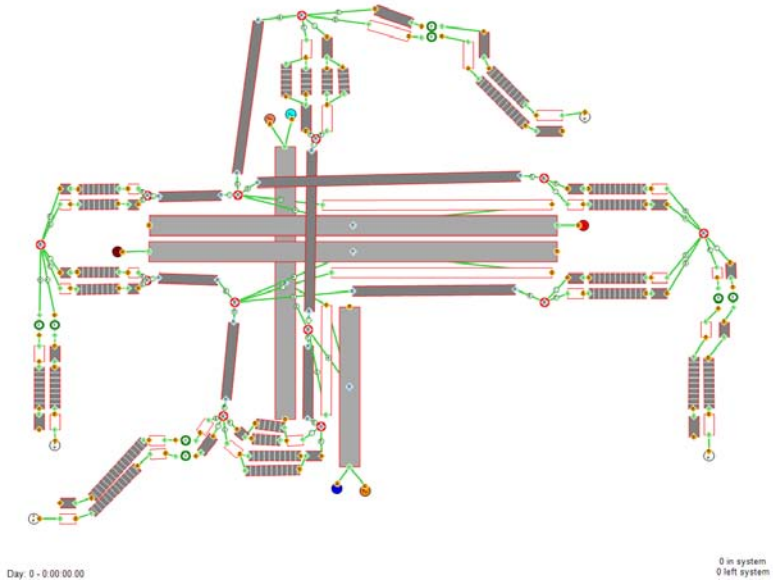


Figure 43. StationSim Representation

Piccadilly Circus

Piccadilly Circus is one of the most famous areas of London and a major road junction in the City of Westminster. The Circus is adjacent to major shopping areas and entertainment centers in the central part of the West End, which makes the station heavily used and subject to crowding. The station has seven entrances in different corners of the Circus which makes the mezzanine very complex. We chose Piccadilly Circus because it is one of the heavily used stations in London underground with crowding problem and complexity of the layout.

Figure 44. Piccadilly Circus Station

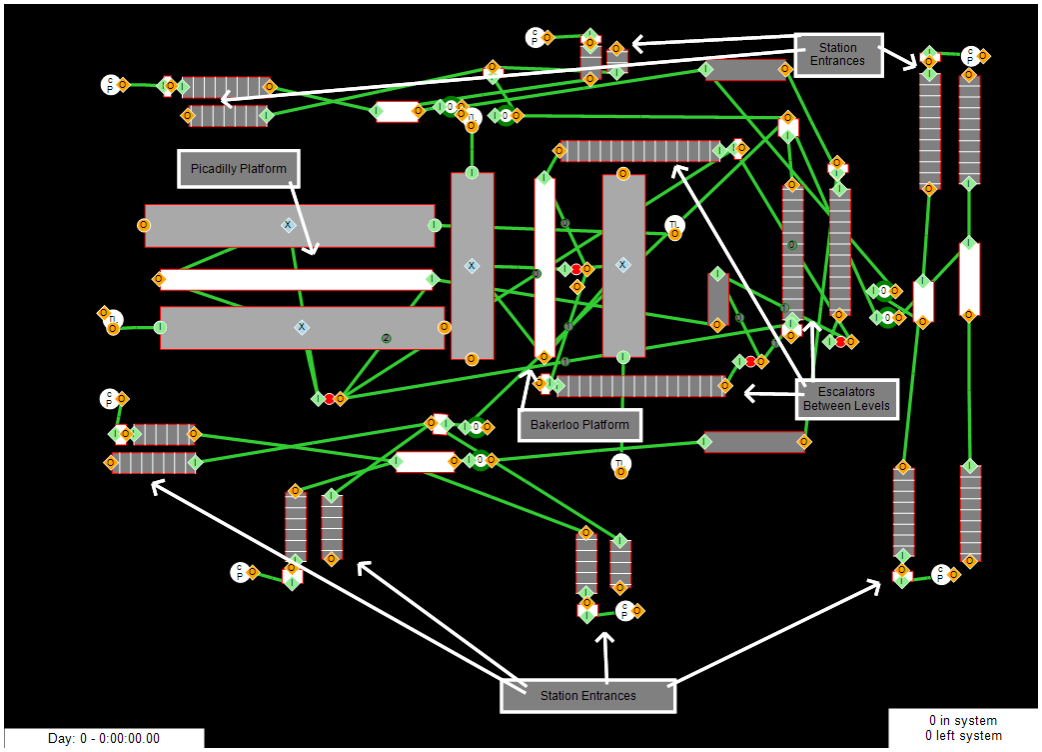


Figure 45. StationSim Representation

Architecture for the Modeling and Analysis of Rapid Transit

Minor Appendix E
Original Concept Material

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

SYST 798/OR 680

Appendix Introduction

This appendix is a reference for our original scope of the project, before team formation. It may be of use for forming new projects, or for understanding the scope and direction from where we started. All material below this point is meant to be the original scope, and may not speak towards the end result of the project.

Concept and Business Case

This project involves the development of an architecture which can be applied to a variety of existing or planned railway-based rapid transit systems, such as the DC Metro. The resulting architecture would illustrate how the existing or proposed underlying systems and novel or existing controlling algorithms can be combined effectively to improve the operational efficiency of mass transit systems through the automated control and allocation of the trains involved.

The hypothetical customer has presented a proposal solicitation for the revamping of a specific regional mass transit system. The existing system is manually run by workers in the train with various automated safety warning systems in place to aid them, and a centralized monitoring system. The goal is to have an automated planning tool and execution engine to allocate, manage, and control the trains throughout the system to reduce customer wait and travel times, system power consumption, and respond to dynamic traffic patterns.

Our company sees a broader opportunity: this call for proposal as an opportunity to develop a flexible architecture that could be applied not only to this particular mass transit system, but others throughout the world. A general architecture can be described which can be a framework for developing such automation systems in other systems throughout the world. The architecture should be specific enough to enable similar performance and behavior improvement on different systems, while remaining abstract enough to work with varying degrees of sensor information coming into the control system. A minimal shim-layer could be created to mitigate the differences between a specific rail system and the common architecture.

Rationale

Besides the obvious improvements to performance and safety potential from an automation perspective, existing mass transit systems are generally stove-piped systems, with no purposeful level of design congruence in the managing systems which monitor or control them. It can be seen that these mass transit systems share many operational attributes, and thus a common architecture for their automation is seen as feasible. There are over 160 urban mass transit systems in existence, and more under construction according to Wikipedia (rapid transit). Additionally, such systems are continually being expanded and overhauled to enhance rider capacity, performance, or efficiency of the network. This represents a much broader and larger market for such an architecture as opposed to developing a single architecture for a specific rapid transit system.

Additionally, this project as seen in line with the current economic conditions and recent initiatives: infrastructure projects are often seen as a way for providing job growth, and such an improvement through the follow-through of this architecture is such an example. Additionally, the potential energy savings and increasing ridership taking cars off the road makes this a green initiative.

Project Eligibility and Feasibility

This project encompasses numerous systems engineering concepts and is a system of systems approach in several ways. The architecture represents an overlaid system building from existing systems abstracted to the common rapid transit model. The system is also not purely a computer or civil engineering domain only, and even potentially includes human behavior dynamics and multiple engineering specialties, making it an ideal systems engineering role to develop. As opposed to developing an entire working solution for a specific rapid transit system, this project strives to explore the

more generic design space of the problem common to all rapid transit systems and determine a flexible, marketable architecture which can work with as many rapid transit systems as possible. As such, many of the multiple-year long integration complexities can be avoided by belaying that work to the actual implementation towards a specific platform. The generic architecture has the validated base technology and supporting algorithms, and descriptions of the integration potential to handle these various platforms, but need not be fully implemented for a specific platform by the end of the semester.

Technology Topics

This project would utilize numerous fields of engineering and concepts. These potentially include distributed algorithms, integration and adaptation of legacy systems, modeling, simulation, and visualization, graph theory, as well as civil engineering, queuing theory and potentially human behavior modeling. There are numerous aspects that can be chosen to be modeled to illustrate the architecture and validate various concepts.

Deliverables

In providing some concreteness to this project, this section outlines the suggested end-products of this project. Most importantly, a thorough description of the generic architecture and rational shall be presented. As part of this architecture, a model of the architecture would be develop to illustrate that given a layout of train lines and tracks, and a model for passenger loading and train performance. Such a model would be demonstrable to verify performance of the proposed solution over traditional simplistic patterns of train movement, and show the versatility of the model to multiple track layouts and capabilities. Secondly, two instantiations of the architecture should be presented, to illustrate the applicability of the architecture multiple mass transit platforms. Tentatively, it is suggested that the Washington D.C. Metro System and the Bay Area Rapid Transit System be used for the architectures.

Goal:

Provide a versatile architecture for automating a variety of rail-based rapid transit systems to provide quicker ride times, greater capacity, higher safety, and lower energy costs

The Business Case:

A stakeholder desires a design for a new automation architecture for an existing rapid transit train system. Our company sees a business opportunity to take advantage of an ever-growing market in similar systems. Where they exist currently, train automation systems are stove piped, independently developed, even though they have common goals and behaviors.

- Worldwide market: 160 cities, 7000 stations, and 25 systems currently under construction, according to "Rapid transit." Wikipedia, The Free Encyclopedia. 26 Jan 2009, 10:39 UTC. 28 Jan 2009 <http://en.wikipedia.org/w/index.php?title=Rapid_transit&oldid=266499794>.
- Green: A better automation system yields less consumption, greater satisfaction of ridership
- Economy: Infrastructure Projects, particular those which help people get to jobs, are always important during times of economic recovery
- Global: Worldwide applicability and complexity

Project Validity

- Integration Concepts: Dealing with existing infrastructure, control systems, sensors, and policies already in place
- Dealing with a new overlaid system on multiple existing independent systems – thus creating a system of systems

- Managing complexity: Creating an versatile architecture – not a specific implementation, that can be applied to multiple varying deployments
- Learning: Numerous modeling and visualization potentials for verifying and validating the architecture

Starting Questions:

How on Earth can this be done in a semester? Since this is an architecture for a versatile system which can be applied to numerous mass transit systems, it can be abstracted and completed separately from the potentially multiple-year specific implementation effort.

What would the deliverables be like? A full description of the architecture and developed algorithms, including integration points with the underlying systems, and guidelines for implementation on specific platforms. Working visual, interactive model of the system shown to be operating on multiple railway topologies with varying traffic loads, differing underlying sensor availability, and train configurations.

What is the design space? Railway-based urban-centric rapid transit: not regional rail, busses, or anything else. Systems such as the smaller Baltimore metro in San Francisco to the massive Moscow or Tokyo Subways, and of course our local DC Metro.

Architecture for the Modeling and Analysis of Rapid Transit

A-MART

Minor Appendix F
Glossary of terms

David Claypool
Mahesh Balakrishna
Kimberly Baumgartner
Yimin Zhang

SYST 798/OR 680

AATC
Advanced Automatic Train Control

AMART
The Architecture for Modeling and Analysis of Rapid Transit

ATC
Automatic Train Control

ATO
Automatic Train Operation

ATP
Automatic Train Protection

ATS
Automatic Train Supervision

BART
Bay Area of Rapid Transit (San Francisco's Rapid Transit System)

BDD
Block Definition Diagrams are charts that contain blocks connected by arrows to depict system interconnections

Cab
The compartment of a rail car where the operator works and where the rail car's controls are located

Colored Petri Nets (CPN)
A concurrency and distributed systems centric modeling language comprised of directed bipartite graphs comprised of a set of places, a set of transitions, and directed arcs. Colored Petri Nets are an extension to this which assigns attributes, "color", to the tokens within the places. For an introduction, see the useful information at: http://en.wikipedia.org/wiki/Petri_net

CPN Tools
A simulation tool for modeling Colored Petri Nets used for rigorously modeling and analyzing system, with a specific focus on concurrent systems

Domain Specific Modeling Language
Shortened to DSML. A modeling language tailored to the needs of a specific domain. Whereas UML can describe a broad range of systems, our paradigm can be used to very accurately describe rapid transit networks only. DSMLs allow a greater level of formality and constraints, facilitating automated processing of models, at the expense of versatility.

Edge
Specifically an undirected connection between two vertices in a graph. Generally any such connection, including directed arcs.

Formal Grammar
A set of terminal and non terminal variables, along with formation rules, that define the syntax of a given language. Typically refers to string grammars, but can also be referring to shape grammars or other syntaxes. See Shape Grammar. Excellent introduction available at

GME
The Generic Modeling Environment: A domain specific modeling language framework created by Vanderbilt University's Institute for Software Integrated Computing (ISIS). Information available at http://www.isis.vanderbilt.edu/Projects/gme/

HOQ
House of Quality is a graphic tool for defining the relationship between stakeholders and requirements

ITS
Intelligent Transportation Systems

LOS
Level of Service, referring to the quality of station crowding. Defined formally in the TCRP

MARTA
Metropolitan Atlanta Rapid Transit Authority

MBSE
Model-Based Systems Engineering

MBTA
Massachusetts Bay Transportation Authority

OCC
Operations Control Center, often referred to as the nerve center of a transit system

p-Diagram
Is used to succinctly describe the external characteristics of a system and to classify the functionality associated with the intended product in the boundary into Noise factors, input signals, output functions.

PDP
Project Development Plan, used to define and establish the management strategy for completing specific goals on a project

PT
Platform-Track Layout is where a single platform services only one track

QDF
Quality Function Deployment is a methodology to transform user demands into design characteristics

Regression Analysis
In statistics, regression analysis refers to techniques for the modeling and analysis of numerical data consisting of values of a dependent variable (also called a response variable) and of one or more independent variables (also known as explanatory variables or predictors).

Sensitivity Analysis
Is the study of how the variation (uncertainty) in the output of a mathematical model can be apportioned, qualitatively or quantitatively, to different sources of variation in the input of a model.

Shape Grammar
A specialized type of formal grammar which uses shapes instead of traditional letters or symbols as the terminal variables.

Sparx EA
Sparx Enterprise Architect is a software tool used to document architectures

Spiral Development
A development model that combines design and proto-typing in iterative, ever-improving stages

StationSim
A graphical and analytical application written in C# created by Team A-MART which is an executable, interactive, and usable instance of our rapid transit paradigm.

SysML
Systems Modeling Language, a modeling language formed for use by systems engineers to perform systems engineering analysis and description. An extension of a subset of UML

TCRP

Transit Cooperative Research Program (A publicly funded program to conduct research projects that promote efficiency of transit systems. www.tcrponline.com)

TPT

Track–Platform–Track is a type of platform layout where a single platform services two tracks (one on each side)

Transfer Station

A train station for more than one railway route in a public transport system, and allows passengers to change from one line to another.

Turnstile

Also known as a fare gate, this is an access-controlled point at which a passenger must pass to enter and/or leave a station.

UML

Unified Modeling Language

Vertices

The nodes, or points, to or from which connections are made in a graph

WMATA

Washington Metropolitan Area Transit Authority

Zoom level

The bounds or scope of the model size, with scaled complexity of components to be studied in rapid transit system describing system views at different levels by zooming in to smaller components or out to a larger system.