Title of Thesis: ARCOLOGY OPTIMIZATION AND SIMULATION FRAMEWORK

Rowin Andruscavage, Systems Engineering Master of Science, 2007

Thesis directed by: Dr. Mark Austin

Institute for Systems Research

Abstract

... Tradeoff studies focus on feasibility of coverage based on differing transportation network topologies. Finally, this document outlines a verification and validation plan for models created using the simulation engine.

Arcology Optimization and Simulation Framework

February 5, 2007

by

Rowin Andruscavage

Thesis submitted to the Faculty of the Graduate School of the

University of Maryland, College Park in partial fulfillment

of the requirements for the degree of

Systems Engineering

Master of Science

2007

Contents

Part I

Introduction

1 Purpose

This project serves to realize an urban multi-modal transit simulation designed during the course of the systems engineering master's program. The program will take a systems approach to modeling human habitats and the transportation networks that keep them running. We would use such a simulation framework to create a baseline model of current day capacity, and then create future models to compare the effects and quantify the benefits of investments in future infrastructure. These kinds of tools would be instrumental in making a case for the development and construction of highly efficient arcologies or other forms of well-integrated compact cities. But nominally, we could apply it towards evaluating and tracking the effectiveness of present-day city growth philosophies.

The distinguishing characteristics of this simulation framework includes:

- A hierarchical level-of-detail organization that allows available data from both top-down parametric models to interact with data generated from clusters of detailed simulation objects. This allows us to seed detailed objects in a sub-system using available aggregate data (e.g. Using data on the total gallons of fuel consumed by an airport per month and distributing that consumption across the aircraft that use that airport) and compare it to data generated by tallying up the individual fuel consumption of those aircraft. This would help calibrate & validate the model by quantifying the effects unknown fuel flows, such as waste or other fuel sources. The hierarchical organization also makes the simulation easier to partition across distributed compute nodes.

- Definition of a data interchange schema between elements of a multi-modal transit infrastructure. The communication provides just enough information about each piece of passenger, cargo, vehicle , and connectivity graphs and defines minimal interfaces to allow them to report to and receive suggestions from a global transit optimization engine.

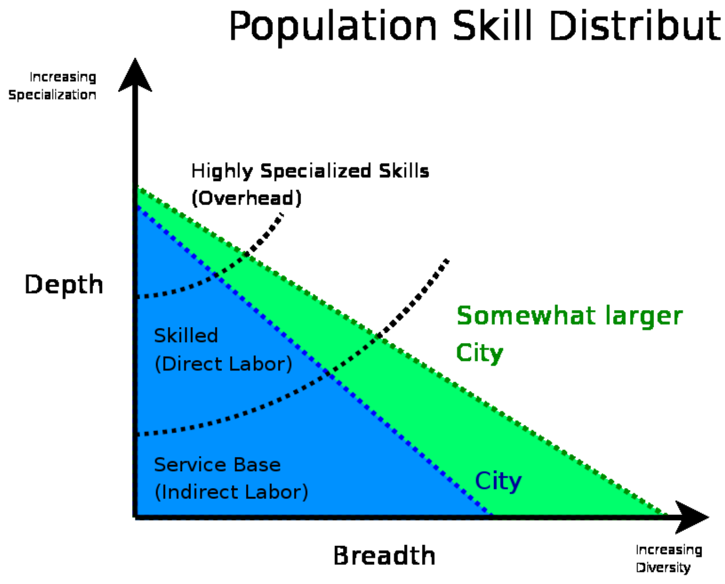
- An inherent focus on meeting the needs and goals of the inhabitants. Many transportation simulations focus on maximizing throughput or minimizing delays or fuel expenditure. However, these metrics may not serve to help evaluate the layout of the urban area itself. This simulation infrastructure would ideally be used to measure the effectiveness of optimizing the layout of an urban area to reduce the need to load the transit infrastructure with commuters, people running petty errands, and other frequent but necessary tasks. An ideal city would have a higher "efficiency" ratio, tracked by an admittedly somewhat elusive "productivity" metric divided by the amount of energy needed to produce and nominally sustain it.

  $\eta = \frac{GDP}{E_{direct}+E_{sustinence}}$

A simple multimodal mass transit optimization solver coupled to the simulation attempts to create a demand-responsive fleet schedule for several types of defined vehicle types that service transit networks within the sim. This tool aims to provide a quasi-optimal means to transport people and goods around within city clusters.

What makes a city special compared to a cluster of businesses and residences? Hans Blumenfeld would argue that a metropolitan area would attract corporations and residents with highly specialized skill sets. Also, as the population grows, a wider variety of niche businesses can sprout up and sustain themselves while catering to a relatively small segment of the market. So by this consideration, a good metropolitan area draws businesses and populations to it by maximizing the diversity and variety

Figure 1: Population Skill Distribution


Population Skill Distribution
Increasing Specialization
Depth
Highly Specialized Skills (Overhead)
Skilled (Direct Labor)
Service Base (Indirect Labor)
Somewhat larger City
City
Breadth
Increasing Diversity

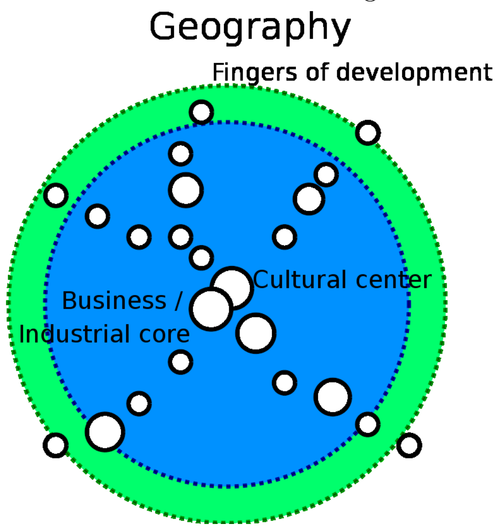
of specialized skills and jobs, summarized in 1. A more developed metropolitan area (represented by the green shaded area compared to the blue shaded area) would have more positions requiring advanced degrees, as well as offer more variety in terms of restaurants, services, etc.

Geographically, as cities grow in population, they often grow "outwards" in area before they grow "upwards" in density. As Blumenfeld notes, this typically follows a pattern of "fingers of development" that grow outwards from the urban core along established transportation corridors such as highways or waterways.

So most metropolitan areas eventually become victims of their own success. Drawing more diverse and skilled population eventually increases their geographical size towards a point where a resident of the city can no longer access all of the resources the urban area has to offer due to congestion.

The goal of the optimization tool embedded within the simulation component of this thesis is to demonstrate a flexible modeling scheme that could investigate the potential effectiveness of various mass transit topologies and strategies, especially with regards to:

Figure 2: Geographical Distribution



- The distribution of and various loads generated by work nodes and residential nodes
- The size and connectivity constraints of various shared vehicle networks shuttling people and goods between nodes
- The ability for the passengers and cargo to make transfers between different vehicles as well as modes of transit

As the urban area grows, we attempt to preserve an ideal population density while also preserving the practical reach of the transit system to prevent fragmenting the city. For the civic planning authorities, this traditionally involves zoning and building out roads and utilities. At some point along the city's growth, they might consider the efficiencies of building infrastructure based on a futuristic arcology hyperstructure in order to meet their urban development goals in a compact physical package.

Together with the simulation, this project seeks to provide a (minimalist) framework necessary to analyze such an urban sytem. We evaulate the effectiveness of an urban complex by creating a demand / sustainment / measurement framework that is used to determine its ability to satisfy its resident and employer needs. Primarily, the effects of the municipal transportation infrastructure's availability and operation on the commute of a worker between their residence node and workplace node. This framework would also allow us to experiment with different urban planning layouts which may ease the optimal solution to the transit problem. From this, a set of urban planning and transportation paradigms should emerge that succeed in making the world smaller by effectively increasing the accessibility of urban nodes by every other node in that metropolitan area.

2 Inspiration

Well, it all goes back to the meaning of life, doesn't it? Here we are, hanging around looking for love or money or happiness, always trying to get the most out of life - in essence optimizing our existence in some fashion. The optimization part is where simulation can be a useful tool, as we often disagree on what infrastructure improvements we could make in order to make us happier or richer or work not so far from our loved ones. For all the aspirations we've had over the decades of reaching for the stars and developing permanent space colonies, I'm surprised by the relatively little success we've had in improving the efficiency of our lifestyles in our dwellings right here on Earth. The ideal american domicile still consists of the single family home, an almost completely isolated pocket of land connected to the rest of the community only by a few wires, pipes, and a stretch of pavement. What goes across these interfaces, and how might they be improved and rearranged by municipal facilities to make the city as a whole more sustainable, flexible, and efficient?

As long as we're dwelling on the meaing of life, another of our tendencies has been to miniaturize complexity, both in space and time. While the sun and stars and universe are beautiful and magnificent only when observed on a grand scale of things, I'd surmise that they are not as interesting when studied on the same scale as, say,

Figure 3: Municipal home interfaces

Category	Current Interfaces	Potential Future Interfaces
Physical	Driveway, Parking	Driveway, Automated Package Transport
Utilities	Electricity, Gas, Water, Sewage	Electricity, HVAC, Fuel (Gas, Hydrogen), Water, Sewage
Wired Communication	Copper / Fiber medium for Telephone, Cable TV, Internet	Junction Box, redundant trunks
Wireless Communication	Broadcast Radio/TV, Cellphone, Satellite, Wifi	distribution points to common aerials, satellite dishes

the inner workings of a paramecium. That's one of the main reasons why we're always searching the rest of the universe for life, looking for something of sufficient complexity that would be interesting and that we could have the hope of interacting with on a similar scale as us. Life as a whole apparently strives to fit more complexity into small spaces. It's a little engine that harnesses energy gradients in order to further decrease the entropy of a localized area. We live on the interesting boundary region of a fractal. And through that, we've begun to expand the boundaries between which objects of vastly different scales can interact. Lately we've been peering into the inner workings of relatively tiny, fast computing devices, which will soon be governed increasingly by subatomic particle interactions, which in turn affect what we do with our lives. That's amazing. Someday soon, we also expect that the tiny electrical processes that occur in our microchips may go on to help us alter the course of celestial bodies, perhaps to allow little us to produce some kind of pronounced impact (or lack thereof) in the cosmic ballet of planets. But for now, one of primary (although not yet fully utilized) uses for our microprocessing technology often is the guidance of the course of our vehicles and information delivery systems.

Most of our interactions with the urban environment that we live in, such as going to work, catching a bite to eat, or (unfortunately) even going out for a hike, involve transportation and delivery systems. These systems take many forms, ranging from various ground, air, and subterranean transit networks to power, water, and even information distribution pipelines that feed directly into each of our homes. Much of this infrastructure is put in place with funding or regulation from government agencies at national, state, and local levels. During these times of rapid modernization, traditional governments can be a bit slow figuring out what infrastructure to invest in. Simulation is one tool that can come in handy to help quantify the benefits of different operational concepts. This analysis can be used to answer questions about design options. However, few governments have validated simulated representations of their jurisdiction that they could use as a baseline model with which to compare several project proposals with a "do-nothing" scenario. Having such a tool would not only give them better access to information about physical arrangement the performance of their existing town, but could grossly cut down on the arguments and political delays incurred when properly used as a "vision communication tool" to the populace.

An advantage to designing cities from the complete-systems perspective of an arcology is that it forces you to take all scale levels into account in the design. This allows the arcology to transition better as new technologies evolve and are put into place. The physical aspect of an arcology is predicated on a municipal "hyperstructure" upon which rests plots for residential, commercial, industrial, and civic construction. The lots would have tightly integrated people and package transportation in addition to the standard complement of water, utilities, and a more minimal road network.

On the national level, arcologies would be constructed to connect well to other cities, with effective transportation and distribution systems and low transit times to most places. Current cities tend to have transportation problems... to put it lightly. Many cities originally sprouted up around ports by major waterways (shipping still accounts for 90% of the mass of goods and materials transported FIXME: citation). However, shipping is not really a means for distributing goods for nonindustrial uses (*i.e.* the vast majority of the city population). Airports are usually built either too far from the city to connect well to mass transit systems, and eventually get enveloped (and subsequently throttled) by suburban growth after which they become a noise nuisance to residents. The United States has invested heavily in the interstate highway system. However, around many cities these get tied up in rush hour congestion, resulting in delays and waste throughout.

On the metropolitan level, rush hour congestion itself is an abomination that most city denizens would readily identify with. We must look terribly silly to outsiders, repeatedly stressing our transit infrastructure past the point where it ceases to be effective. We tend to want to commute simultaneously simply to be in sync with everyone else - most of whom we don't even deal with regularly if at all - for little discernable benefit. And why is affordable housing in such short supply that many have to commute in from suburban or exurban towns 30, 60, 90 miles away? It seems as if our needs to be flexible in the increasingly unstable employment market are not met by the relatively inflexible lease and mortgage agreements.

The urban development paradigm of roughly the last half-century has been characterized by suburbanization. As more massive superhighways are built to relieve the strain on the original interstate connectors, more suburbanites continue to sprawl out along these new corridors. After a certain point, the ratio of space allocated between highways and developable, livable area becomes saturated to the point where we get diminishing returns from building more roadways. Highways take up a lot of space, and when we start to pack those highways close together, we end up spreading out actual useful land into isolated pockets nestled between interchanges. Another interesting tendency which just confuses out-of-town drivers is that complicated intersections between multiple highways call for larger, more complicated, and ultimately more profitable construction project for interchanges. What's more, having multiple highways down busy rush-hour corridors don't really make the world any smaller. A sparse network of good, uncongested highways should make it take just as long to get from point A to point B without having to build and maintain several alternate routes that exist just to relieve rush hour congestion. A good example of this resides in the Baltimore-Washington corridor, with I-95 flanked on each side by 295 and Rt.29 roughly only 2 miles apart.

I don't really have anything against cars. They are likely the most flexible mode of transportation available. All you need is a slab of pavement or even gravel connected to the road network, and voila, you have an interface to the road transportation network. Compared to the equipment you'd need to interface with the municipal power grid or water/sewer lines, this slip of asphalt is one of the simplest yet most capable ways of moving people and goods to and from your home. However, when be build cities almost exclusively around automotive transport, we end up losing a lot of what makes dense cities good for people and sustainable for the environment. Cars act as a multiplier to the amount of space each person takes up. Not only do you need a driveway space to park each person's car at their home, but also a space reserved at their work, as well as some shared spaces at all of the shops and venues at which they'd possibly spend time. Add to this the ganglia of roads connecting those spaces together, spacious service statiosn, and shoulders and extra lanes for safety and additional peak capacity, and we find that our cities have vastly outgrown the human scale. Looking down at our houses from an aircraft, we'd see more land covered by pavement reigned by cars than for buildings and establishments to be enjoyed by people. A well designed city would achieve higher density for people by introducing transit alternatives allowing them to go directly for home to work. All that Park-and-Ride initiatives accomplish in regards to land utilization is just moving the parking lots further away from the workplace. In an urban complex with sufficient transit, people should only need to use their cars to leave the city, but rely on municipal transit to move people and goods within the city.

To address this, a current philosophy for urban planning centers around the Smart Growth concept, which concentrates on building high-density mixed-use developments near existing or planned mass transit stations. Hopefully this will reduce the car land use multiplier.

On the personal level, much of the home infrastructure for living does not have flexibility for change. We are still using much of the same basic physical interfaces developed over a century ago for power and voice communications. Additional systems have sprouted on top of and alongside these networks, such as DSL, cable television, and various wireless and satellite networks. Add to that various combinations of water mains, sewage systems, natural gas pipelines, and perhaps we might begin to appreciate the need for developing more flexible and maintainable living facility interconnect standards that could be easily upgraded and expanded to adapt to support new emerging networks and give us greater flexibility in reusing older homes and living spaces. Such standards help reduce the barriers to market entry, allowing economical deployments of existing upgrades such as fiber-to-the-premises, or even some things for which markets haven't really been created for yet, such as fully-automated package delivery systems or centralized HVAC services.

The purpose of an arcology is to create a compact, highly organized structure for people to live and work. It should be designed to improve and maximize the quality of life of its residents, and not just focus on maximizing personal productivity to maximize economic performance. While the major design challenge would consist of finding a way for getting large groups of people to tolerate living in dense proximity to one another, I would submit that the internal transportation system is one of the keys to making the system perform. This circulatory system for people and packages affects how well most of the rest of the system can perform to meet goals for delivering necessary resources, and meeting safety requirements.

While the simulation and optimization models used in this thesis are certainly generic enough to apply to most ordinary forms of mass transit, I chose to apply it in the context of an arcology for two reasons. First of all, the word "arcology" still remains rather unique in the global namespace of the engineering field, and connotes a flair for futurism (for better or for worse). More importantly, the design focus of arcologies as an autonomous structure encourages us to analyze it in terms of control volumes, defining the flows of input and output products in ways much more conducive to identifying resource consumption and environmental impact. While the concept of analysis via the definition of control volumes may come naturally only to engineers trained in thermodynamics, it is refreshing that emerging efforts to track our carbon footprint as part of a global carbon dioxide emissions budget. Hopefully this is the start of more complete tracking and accounting (and eventually optimization) for more human environmental resouce use and waste reclamation.

3 Background: Arcologies in History, Media, and Current Proposals

The Wikipedia entry for Arcology has a more comprehensive listing of references to works and projects than I could possibly describe here. Yet, the background on the development of arcologies or similar proposals is surprisingly thin, so I'd like to highlight a few major works.

Arcology: The City in the Image of Man[26]

The specific concept of the arcology was first introduced in the 1950s by architect Paolo Soleri as the ultimate urban planning solution to the problems of metropolitan growth. Continuing trends in the expansion of metropolitan areas have contributed to explosive growth of low density suburban sprawl, the decay of inner city urban areas, and finally the indiscriminate destruction of natural environments to make room for a human habitat system which is increasingly less efficient, convenient, and aesthetically-pleasing. The concept of the arcology attempts to reverse those trends by providing a compact city infrastructure that works well and manages to reprocess most of its waste before returning material back to the environment.

What exactly *is* an arcology by definition? Featured in several science fiction works as the cities of the future, an arcology is more than just a structure... a "superbuilding" that contains everything you would expect to see in a current city. The arcology integrates living spaces and working spaces with transportation systems that connect it all. One of the fundamental differences is that it heavily involves the vertical dimension into city planning, whereas current planning is done by zoning commercial / residential / industrial in an ad hoc fashion (well, by government committee through series of votes, demonstrations, bribes, and legal means applied in a reactive manner which might end up fulfilling actual requirements just as well as an ad hoc design process would). Additionally, an arcology's roots in urban agriculture would mean deliberate collection and reprocessing of waste byproducts. The arcology might simply be described as what a city would look like if it was actually designed by competent systems engineers (of course, also a feat easier said than done).

Compact Cities[23]

This work provides the most compelling vision on how this human habitat would work from a technical standpoint. George Dantzig and Thomas Saaty (fathers of Linear Programming and the Analytic Hierarchy Process, respectively) wrote this fascinating book on the feasibility of constructing a livable city of between 250 thousand to 2 million residents within a 2-4 square mile, 4-8 level superstructure. Their proposal addresses many financial and social factors as well as providing major design elements and outlining the major physical characteristics of their ideal layout proposal. I should hope that this simulation framework is flexible enough to simulate some of the main ideas in their design, including their transit network of trams and elevators, the evenly distributed timecycles of its denizens, and even parts of the conveyor-driven automatic package delivery system described in their proposal. The design in this book could certainly be used to establish an upper bound of the types of efficiencies that a city willing to radically re-engineer its design could hope to achieve.

Other Literature

The Modern Metropolis consists of a series of Hans Blumenfeld's essays and articles on urban growth vs. urban planning.*[8]* These treatises generalize how cities have developed and evolved over the decades and centuries, and suggests some design principles for sustaining growth over time. These insights into how to cope with the forces that incrementally shape cities and inevitably stress them beyond their initially planned limits would hopefully reinforce some of the ideas for flexibility provided by the Compact Cities design.

While excitement about radically redesigning urban planning has died down over past few decades, other environmentally-friendly initiatives have taken its place. Several publications focus more on modifying the design goals of current city planners to incorporate more alternative forms of transportation. The book and accompanying website Carfree Cities presents several concepts and examples that make public transit and areas more pedestrian and biker friendly. The author has a particular affinity for Venice, and provides that city as a model for low impact multimodal transit along with some additional improvements.[11, 5] Most contemporary urban revitalization works take this track of advocating increased use of multimodal transportation in current city design to cure the ills caused by an automobile-centric monoculture.

Current Works and Proposals, Arcologies in the Media

The Biosphere 2 is a good experiment in closed-system sustainability.[3] Unfortunately, its primary experiment was widely regarded by the public as a failure.[24, 12] The facility has since come under the management of Columbia University as a research lab.

The closest present-day developments resembling arcologies are smattered around the world in various stages of completion. The truest to spirit arcology project in existence would be Arcosanti and Cosanti, the experimental communities arranged by architect and founding father of the "Arcology" concept Paolo Soleri himself.[2] These reduced scale experiments in the Arizona desert are currently reported to be hovering around 5% complete after 30 years of development. Like the Biosphere 2, this development has shifted in focus into an urban laboratory.[20]

While this paints a somewhat bleak outlook, the influence of these spearheading projects is definitely spreading. Large scale proposals have been cropping up more frequently, especially in population-dense Asia. Predictably, the Chinese have a keen interest in the arcology concept, both for expanding high-density urban areas,[18] and also in the form of constructing sustainable communities that would address their growing problem with semi-rural slums.[28] Several Chinese and Japanese design firms have been promoting various skyscraper appoaches, such as the Ultima Tower,[27] Tokyo's Sky City,[9] and the on-hold Tokyo Millennium Tower[17] (the latter two are covered in Discovery Channel documentaries).[15, 14] Arcology.com has a collection of other notable works and proposals.[1]

Under the strains of urban growth, many former suburban towns are finding themselves faced with less ambitious but more pragmatic “smart growth” proposals for new higher-density mixed-use developments. In the Washington DC metropolitan area alone, plans are underway to construct these types of mixed use centers in Vienna[13, 22] and Tysons Corner[21, 16] off of existing or future mass transit stations. We'll likely see more of this type of development in the near future, especially seeing as how the Supreme Court has recently ruled to allow private homes to be seized for mixed use and other commercial development.[6]

Urban Simulation in the Media

Previous well-known works that tackle the task of urban simulation includes two series of open-ended games from Maxis (now part of Electronic Arts) that approach the problem from different scales: SimCity and The Sims. Certain versions of SimCity (2000 and 3000) even had arcology elements in them, although since they were entirely self-contained, they really did little for the game other than to allow you to boost your population without having to provide additional infrastructure. To some extent, these games could be used to experiment with different urban or residence layouts, but they primarily pattern themselves after common current day paradigms and lack the flexibility needed to really turn its simulated environment upside down. Hopefully they do serve to influence the next generations of urban planners, who might come to expect and demand some of the timely command and control interfaces coupled with instantaneous reporting of the city's condition and resources. Beyond that, there is not much published in the way of complete city and/or lifestyle simulation. This is probably partly because most of this analysis can be done more simply using historical data tracked by government statistical agencies, and because most of the simulation writers are more busy simulating more interesting things such as data[4] and transportation networks.[7]

4 Potential Simulator Applications

Paolo Soleri describes several arcology designs that could be used to replace major cities or serve well in several environmental settings. This project would create a tool that could be used to quantitatively analyze the benefits of enhancing cities with concepts from the arcology paradigm. This report describes the systems engineering of a tool to perform preliminary design & benefits analysis of urban transit sytems.

This simulation would want to be flexible enough to handle most of the suggestions made by Dantzig and Saaty in *Compact Cities.* Indeed, a lot of the design requirements and hooks left for future work were heavily influenced by the desire to tackle some of their recommendations, such as:

- Rotation of work / sleep schedule to prevent what they term "cicadian rhythms" that results in peak infrastructure congestion.
- Multimodal transit architecture of elevators, trams, cars, and automated package transport.
- Star hub & spoke transit topology joined by rings.

Not many people are in a position to design cities. However, almost everyone needs to work within the infrastructure of one, so it would be worthwhile to create a model if only to serve as a dynamic demand generator used to plug input parameters into these data and transportation networks. Surely they can use historical data as inputs, but this breaks down when a system they are designing may have a significant effect on the input data.

One item of study that this type of simulation makes feasible is the relationship between these data networks and transportation networks. For example, if a city decides to spend money upgrading their data infrastructure so more people might be able to telecommute to work, this may have a noticeable impact on the load on their mass transit system. This simulation could aid as a decision-making support tool that could actually tie the network and transportation models together.

4.1 Network Topologies

We could also draw less literal comparisons between data networks and transit networks, especially when it comes to subjects like their topologies. The most efficient topology for providing service to a set of transit nodes linked together by various distances typically involves finding the minimum spanning tree that spans the set. However, we could reap some rewards from building "inefficiently" with additional linkages between nodes to provide alternate pathways. To draw several analogies to computer network topologies, let us consider some of the improvements we could make by investing in additional connectivity.

Fault Tolerance: The easiest way to ensure high availability of service during component failures, accidents, or even routine maintenance or upgrades is to simply build two o feverything. During a failure mode, we simply switch to using the backup resource, be it a highway lane, second runway, port, *etc.* Of course, this approach is terribly expensive, doubling your infrastructure costs simply to go from 99% availability to 99.999999%. But you could get more return on your investment by also allowing load balancing on the additional assets. The backup resources stay active to add capacity to your system. During peak periods, you could run twice as many cars without violating headways, land or take off more aircraft, or unload ships in parallel. Failure modes will reduce system performance, but a single failure will not completely shut down access to a node or connecting segment. Of course, most of the benefits of load balancing only become apparent when your system demand approaches the capacity of a single nonredundant resource.

If we have already committed ourselves to building twice the infrastructure to meet laod demands, we might as well consider placing tha additional infrastructure in such a way as to provide more benefits than we'd have simply by constructing two copies of the minimum spanning tree on top of each other. We can accomplish this in such a way that still preserves some of the redundancy qualities for fault tolerance of the system, while improving capacity and other performance aspects such as latency. The minimum spanning tree is often full o farborized links, which are very, well, tree-like. Network branches reach out and join together into larger common trunks. By creating more reticulated linkages, directly connecting individual branches without necessarily traversing through a common truck, we can form a more densely interconnected network that not only has additional capacity but also has reduced transit times between nodes that would have been further apart in the MST network. This can reduce the overall diameter of the network (the maximum distance between any two nodes in the system).

This type of more distributed topology tends to be more decentralized than the MST, since it spread smaller hubs out throughout the network rather than concentrating them into a few central superhubs in transit trunklines. It can also be more flexible in terms of providing multiple equal cost pathways between pairs of nodes. This can make the distributed topology more resilient to failures or outright attacks on one of its hubs.

A more distributed, heavily reticulated mass transit system would have higher service availability, high capacity, and low latency, making it a more viable alternative to personally owned vehicles that dominate many metropolitan environments today.

4.2 Urban Planning & Design Analysis

Several initiatives are currently underway to rethink the way metropolitan areas are designed. This simulation modeling & analysis framework can provide a design planning and evaluation tool to assess several integrated mass transit network topologies to help identify and accelerate the worthwhile changes.

The use of simulation as a decision-support tool would help avoid or at least temper some of the larger controversies over the past century of rapid technological change. The history of our infrastructure has been peppered with some epic and ultimately costly battles over different modes of transfer, such as the turn of the century Edison - Tesla battle to establish AC or DC as the power delivery standard[19] or the politicized finger pointing over whether GM was duly responsible for taking control of streetcar operations in the 20s in order to dismantle them in favor of GM-manufactured busses.[25, 10] Having detailed records of the simulations used to provide hard data on which broad policy decisions are based could help justify your decision later. With more options pushed by several technology firms, it should be more important than ever to be able to determine the selection of major wired or wireless communications infrastructure or transit modes based on available technical data, and not on which company has the best connections to the civil servants responsible for municipal decision making.

Ultimately, if this were to evolve into a fully-featured urban simulation tool, it could be used as a rapid prototyping environment for proposals to system changes big and small. When this functionality matures, a municipality might require a simulation-based analysis to accompany any new infrastructure proposal as part of a gateway approval process. As standard patterns are built up, the sim framework may morph into a design tool, replete with a library of open-source blueprints, guidelines, and standards (as well as customizable sections) to that can be deployed to achieve a development goal. Furthermore, as the process becomes automated, it might incorporate more direct civil input, turning review and evaluation of problem areas and proposals into something of an experiment with alternative direct digital democracy governance, with which the citizens can interact with as something of a hive mind. Or so goes the vision.

Part II

Generic Arcology System Model

An arcology is a combination of architecture with ecology, essentially forming an environmentally-friendly (or at least sustainable) human living system well-suited to systems engineering analysis. This section defines and describes a network queuing simulation model that might be used to perform trade study analysis on such a system. The model attempts to structurally decompose the human habitat into groups of subsystems on all scale levels that interact through the exchange of several resource types. The resulting resource flows are quantified into performance metrics used to compare different types of arcologies to actual living conditions. Bottom-up scenarios of arcology models will be compared to top-down scenarios constructed based on present day statistical data. Tradeoff studies focus on feasibility of coverage based on differing transportation network topologies. Finally, this section outlines a verification and validation plan for models created using the simulation engine.

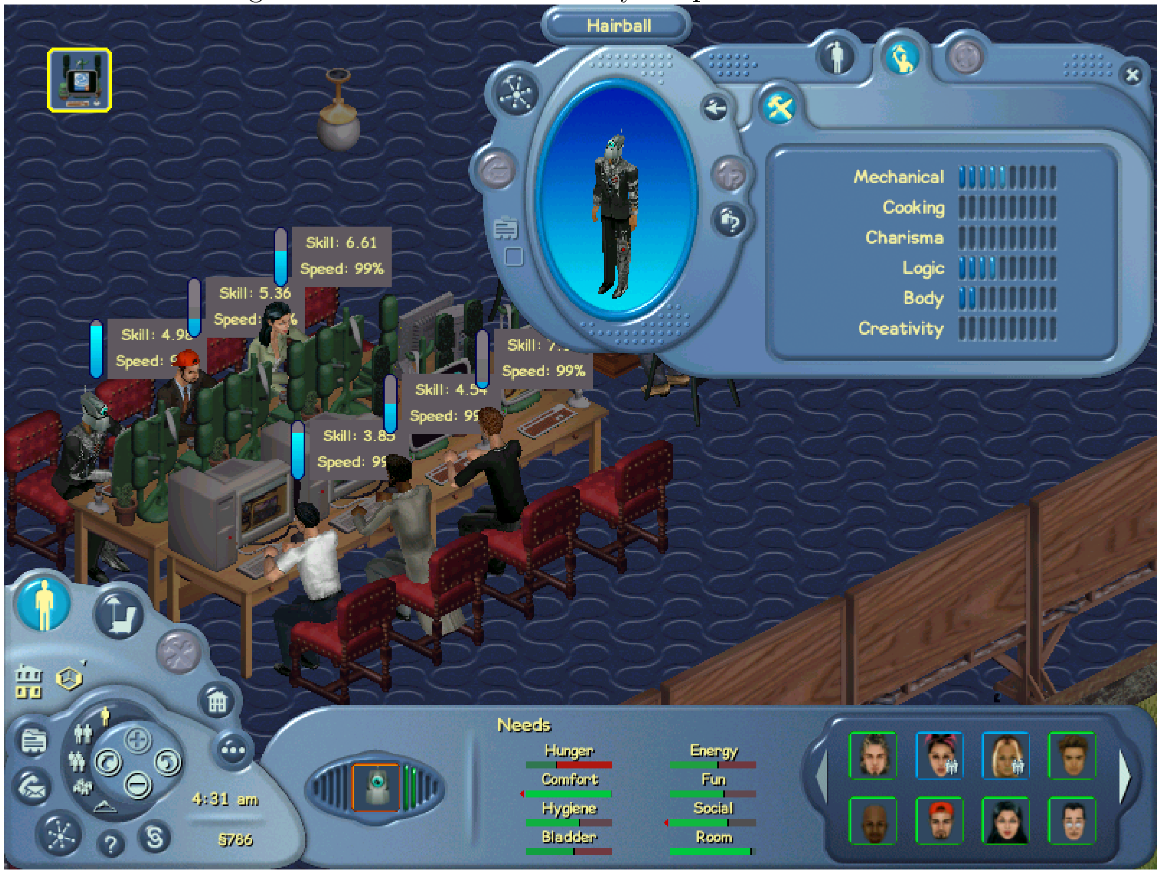
5 Concept Requirements

5.1 Goals

One of the characteristics of systems such as cities that grew by evolution rather than by design is that they lack fundamental policies that drive their design. Components of the city usually come about in a reactionary manner: fire protection services are built after too many buildings burn down, airports are built to serve cities after they have already grown too dense to accommodate one in a central location, tap water distribution systems are gutted out and replaced only after the old ones were too heavily loaded to be sanitary.

Hindsight being 20/20, it is worthwhile to dwell on past mistakes and develop urban planning with a systems engineering process worthy of supporting a megalopolis. The first step is to develop a set of goals and objectives that drive the design of the city. At first glance, the goal of a city (at least as envisioned in Maxis's *SimCity*$^{\text{TM}}$) ought to be to grow and prosper. However, this overlooks the city's primary responsibility to fulfill the needs and look after the well being of its inhabitants. For that, we can look at it from an individual level on par with the scale of Maxis's *The Sims*$^{\text{TM}}$:

Figure 4: *The Sims*$^{\text{TM}}$ Entity Requirements Model[1]



The Sims$^{\text{TM}}$ offers 8 needs for each of their simulated characters: "Hunger", "Energy", "Comfort", "Fun", "Hygiene", "Social", "Bladder", and "Room". This model takes an even simpler approach:

- Shelter : where people live and sleep (accounts for "Energy", "Comfort", and

"Room" from *The Sims*$^{\text{TM}}$ model)

- Food / Air / Water : the raw materials people need to consume to live, or at least not starve to death (accounts for Hunger)
- Health : maintenance factors, such as cleanliness, waste, (accounts for Hygiene and Bladder)
- Work : most people need something productive to do when they aren't attending to their other needs. This could take the form of working for money, or being educated to increase their knowledge bank of information.
- Entertainment : if people aren't doing something productive, they're probably doing something fun to while away their time (accounts for "Fun" and "Social")

In order to fulfill these needs for all of the city's inhabitants efficiently, what they are really looking at is developing infrastructure to move resources around so that each of these needs can be catered to. This simulation model takes on abstract views of these resources and the transportation networks that move them around.

5.2 Objectives

So what should our objectives be, if we are to meet our goals, how can we form an objective function for optimization? Of course, we're talking about multivariate optimization of multiple goals.

- Continually improve the quality of life for inhabitants
- Accelerate development of improvements to the body of knowledge
- Maximize productivity, performance.
- Optimize resource consumption to achieve balance with interchanges with the outside environment.

- Avoid optimizing the whole at the expense of the few by trampling individual freedoms. Add structure to the system by providing opportunities and alternatives, not imposing restrictions on who gets to travel and who doesn't.

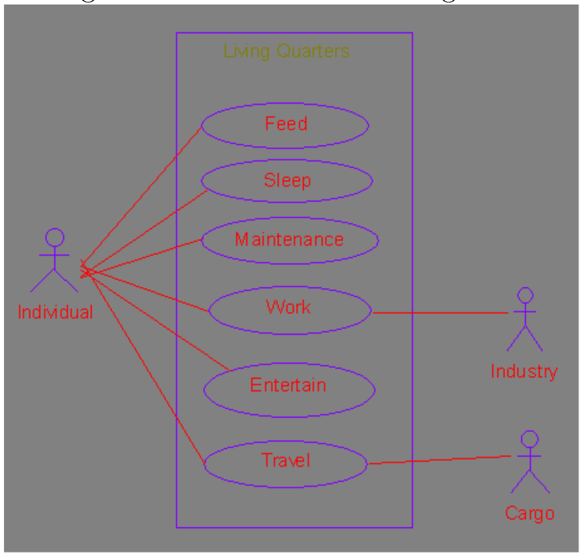
Obviously, we'd need to break down each of these objectives into measurable quantities. In order for the simulation model to be effective, it should be capable of assigning metrics corresponding to these objectives, and computing them based on the simulation inputs. The simulation inputs and execution will have to sufficiently model real life enough to be able to produce a valid estimate of these performance metrics. For example, a "quality of life" metric might be a composite of several measurable outputs, including the length of required commutes, the number of times they are hit with a hunger event that can't immediately be serviced by the resource delivery system, amount of leisure time afforded after all of the "required" work is done, *etc.*

5.3 Use Case Diagrams

As described, the arcology use cases are simple enough, and represent a few different modes of operation. The system boundary is provided by the living quarters, which, contrary to its name, extends beyond the individual's residence and just encompasses all the locations where they go about their business. The arcology simulation model will need to be flexible enough to model these types of activities in order to be used for design.

The one new activity introduced by this diagram is the "Travel" interaction. As mentioned, not all of these use cases occur in one location, so the Travel case takes care of moving the individual from one location to another. This interaction is performed thorugh one of the Cargo Transportation Infrastructure classes, which will be detailed in the System Structure.

Figure 5: “Live” use case diagram.



Actors

Individual : An inhabitant of the system.

Industry : Entity by which the individual is employed.

Cargo : Transportation Infrastructure responsible for moving people around (as well as resources).

Use Cases:

Sleep : Everyone needs a place to rest for a significant portion of the daily cycle.

Feed : Consumption of food and water resources.

Maintenance : Miscellaneous cleaning tasks, such as bathing, brushing teeth, doing laundry, dishes, *etc.* would be represented here.

Work : Work is a transaction between and individual and an industry to exchange money for productivity. In this case, productivity fuels the reactions that the industry performs.

Entertain : Entertainment can take on several forms, from merely socializing with other individuals, engaging in solitary entertainment interactions (TV, games), to mass entertainment (theatre, *etc.*).

Travel : An individual is able to travel through the transportation infrastructure to commute to work or to travel to places to fulfill their other needs, such as for food or social interaction with friends.

6 System Structure

The basic model consists of an overall package named GeneralHabitat, which contains base classes and three more packages to organize resources, reactions, and transportation methods.

6.1 GeneralHabitat Package

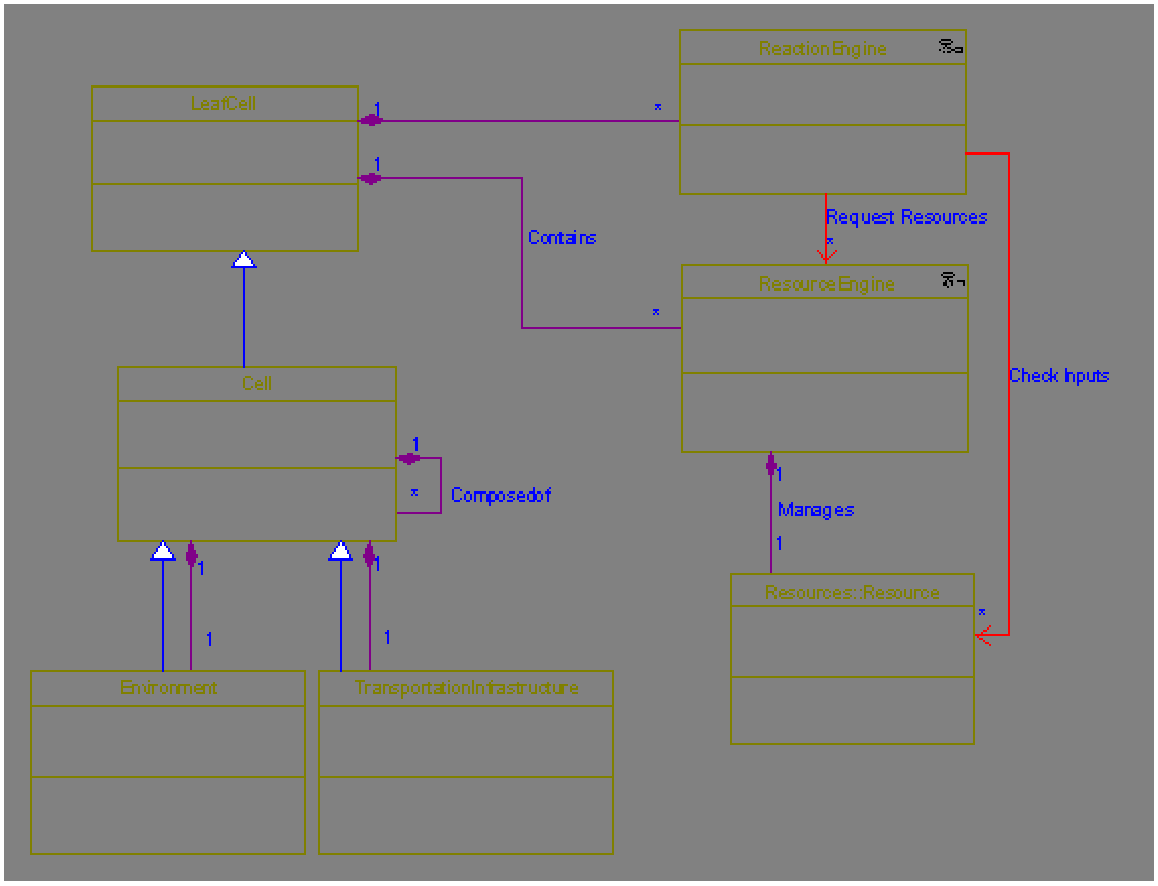
Generalized resource queuing and transportation model of living support systems.

A scenario is required to build up a model of a system by creating a hierarchy of cells that connect to each other via transportation network infrastructures. These cells then begin to perform resource transactions between each other and resource reactions within themselves to simulate the daily operations of the system and observe it from different levels of detail, scaling from the individual to the city to the world. The transaction approach is well suited for implementation in a discrete event simulation.

Much of the model is static, such as monetary costs for resources or the structure of cells. This model is not intended to perform dynamic economic simulations or find ecological balances between the deaths and birth rates of people or towns; those functions have been well studied. (That said, the nature of the event-driven simulation framework makes it easy to patch in such functionality by manipulating variables or cleverly reorganizing the scenario outside of the simulation.)

Instead, this model is merely intended to construct an overglorified spreadsheet used to perform preliminary design and calculate rough benefits analysis on changes to ways of life, quantifying answers to such questions as: "how much energy might a city save if everyone installed more efficient light bulbs?" or "how much time can we save if we staggered a city's work schedule to relieve rush hour congestion?"

Figure 6: GeneralClasses Object Model Diagram



6.2 GeneralClasses

The GeneralClasses object model diagram (Rhapsody's internal name for a UML class diagram) depicts the base simulation classes and generally encompasses the entire design of the simulation. All object model diagrams following this would actually constitute scenario-specific use cases that highlight the use of the base simulation classes.

Cell : The fundamental unit of structure. Each cell represents an identifiable entity, which contains its own collection of resources. These resources can be traded with other cells, or undergo reactions within the cell to transform groups of resources into other types of resources. Generally, there are five basic types of cells that work together: The entity itself, the entity's environment, the entity's transportation infrastructure, and leaf cells to represent individuals and industries. To further complicate matters, entities are arranged into hierarchies of subcells. This allows us to view the system on several levels of detail, from global down to the individual. To do this, we introduce the constraint that a cell's resources always equals the sum of the resources of all of its child subcells.

LeafCell : Leaf cells are a special type of cell reserved for individuals and industries. These cannot be subdivided further into subcells, and thus lack an environment or a transportation infrastructure to support those subcells.

Subclasses:

Cell

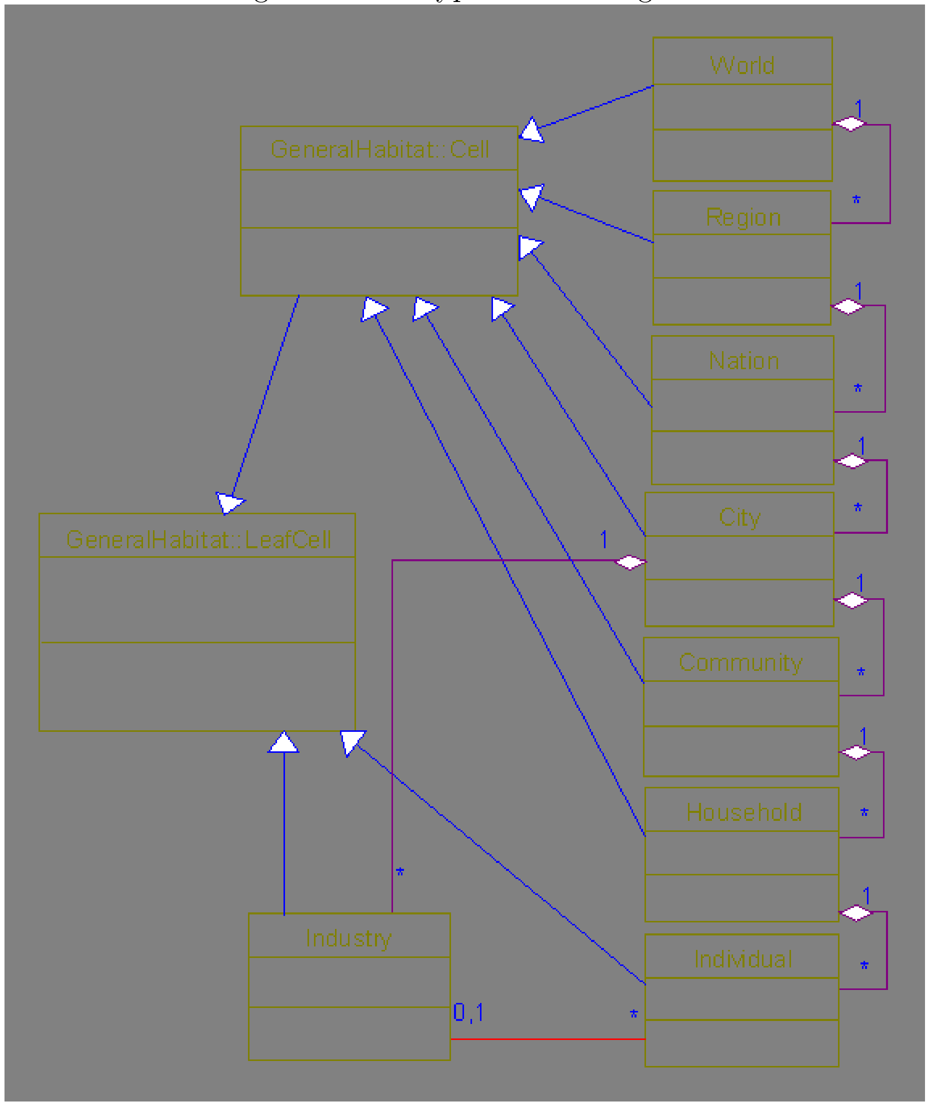
Individual

Industry

CellHierarchy

We model the area of interest by breaking it down into a hierarchy of cells and subcells that work at a different level of detail. There are essentially two types of units, parent nodes and leaf nodes, with the only distinction being that leaf nodes do not have any subcells. One possible scheme for defining this hierarchy is presented in the CellTypes class diagram. It's important that all of the subcells add up exactly to form the parent cell, so in some cases, it would be necessary to define subcells that represent everything that might be left over after allocation into existing subcells. For example, the rural areas not part of a city would be lumped into a special residual "City" subcell to be included as part of a "Nation". Similarly, homeless people and vagrants would be lumped together into a special "Household" or "Community" subcell to be included as part of "City" data. This should be an acceptable practice, since these units may tend have similar characteristics.

Figure 7: CellTypes Class Diagram



Classes:

World The limits of the size of the system. Of course, the architecture of the model is left open to envelop interplanetary commerce between worlds in the distant future.

Region A geographic region would tend to be composed of several nations with a common situation. Of course, large nations may exist over several regions. For our purposes, we'll simplify by assuming all nations are smaller than the regions they are in.

Nation A nation sets the policy for international trade and commerce. Plus, data is often available on the national level for input into the top-down models.

City A city would be the highest level of organization represented by an individual arcology. Several cities would be interconnected to form a nation. One "city" cell unit can be put aside to account for all rural areas not included in other cities.

Community Families tend to cluster into communities, which in turn form cities.

Household A household would consist of a family of several individuals living together in one residence. A family doesn't necessarily include extended family, or preclude the existence of other arrangements such as roommates.

Individual A leaf node in the hierarchy, the Individual cannot be broken down into any more subcomponents (we can only hope). Most individuals will also work for an industry. Individuals are free to move from place to place as part of their daily lives. This allows them to commute to work or to visit friends in another household and transfer their resource consumption to stress the infrastructure at other locations. When individuals travel, it puts a strain on the transportation infrastructure.

Industry Cities have a special type of leaf node called Industry, which essentially employ several Individual units to perform certain specialized reactions on particular resources in bulk. Generally, they consume energy resources to refine material resources.

Environment A special passive cell that will always yield any resources that it has and accept any waste that is ejected into it. Instead of interacting with other cells on the same level, it only interacts with subcells. So, for example, a nation's resources can be split amongst its cities, and city level waste gets deposited in the nation's environment (as opposed to some other nation's environment).

TransportationInfrastructure A special cell that interacts with subcells. It represents the connective tissue that allows resources to transit between subcells, and it takes both money and fuel in the process. Several types of transportation infrastructures can be defined with different characteristics in terms of resource burn rates.

Attributes:

Maintenance Monetary maintenance cost incurred to keep this system up and running per unit cycle.

TransitCost Monetary cost required to move a unit of resource through this transportation infrastructure per unit distance.

Value Infrastructure build value, how much money needs to be invested to put this transportation infrastructure in place so it can be used.

6.3 Transportation

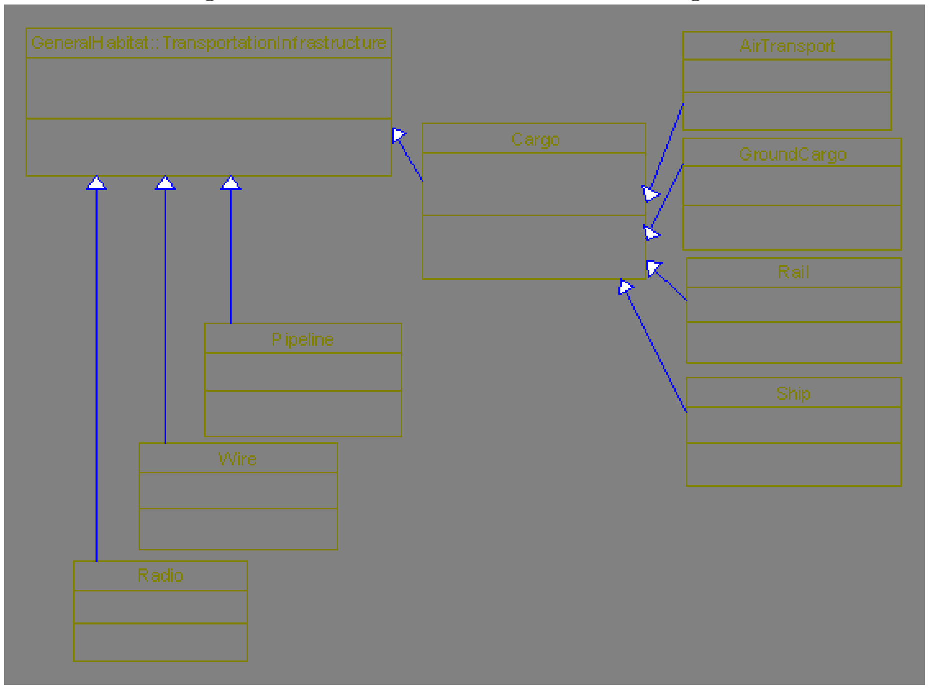
The transportation network serves as connective tissue that joins the nodes of the structure together. It is up to the scenario to define the connectivity graph, but once accomplished this will compute the overhead in terms of resource burn to transfer individuals and cargo through the network.

Classes:

Cargo A generalized form of transportation for passengers and cargo. Only these forms of transportation can handle material goods and individuals.

Attributes:

Figure 8: ConnectiveTissue Class Diagram



NetworkCapacity The number of transport units the transportation infrastructure can handle. As network capacity approaches this number, congestion effects set in.

numUnits Number of transport units actively using the system at any given time. When this number nears the NetworkCapacity, congestion delays set in which begin to cut into the efficiency of the system.

AirTransport Expensive but fast, and often must be used in a multimodal fashion, where households must transfer their wares up to the city level first before making airhops between cities.

GroundCargo Well connected, reaching every location with road coverage.

Rail High initial infrastructure costs and not very well connected, but fairly economical once everything is in place.

Ship Only a boon to certain cities, and requires some contention for port infrastructure.

Pipeline Pipeline infrastructure is good for transporting fluid commodities, such as water, natural gas, sewage, *etc.*

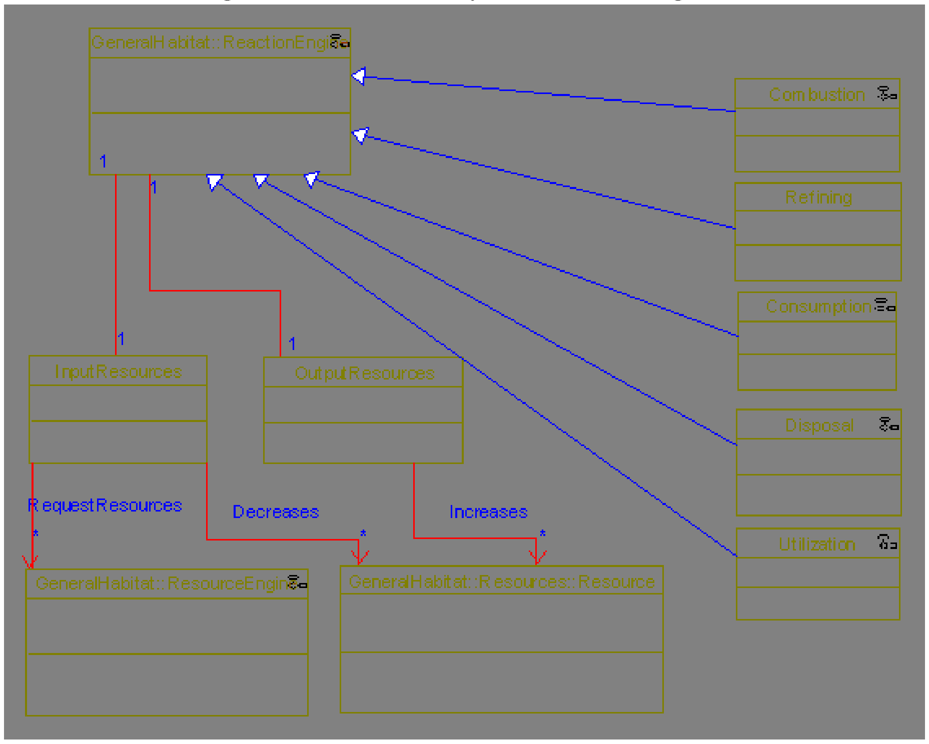
Wire Distribution system for electricity and information

Radio Distribution system for information

6.4 Reactions

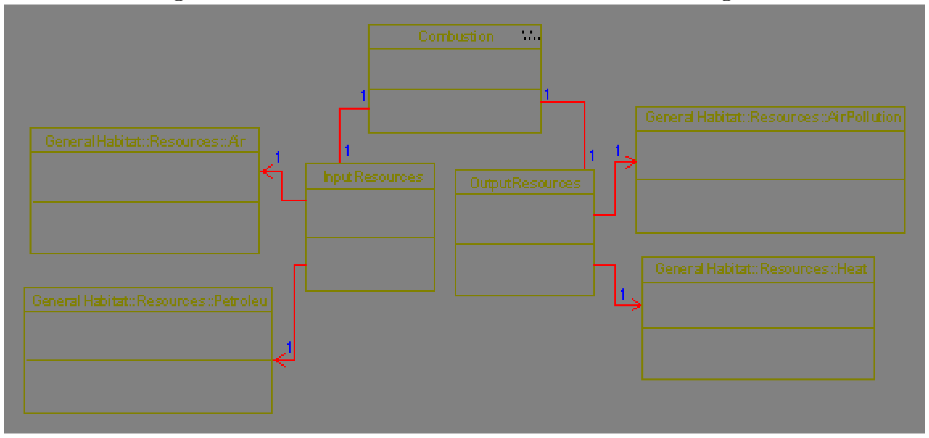
This package defines reactions that can occur within cells to transform one set of resources into another set of resources. The definition of the reaction governs changes to the quantities of inputs and outputs, and balances them the same way a chemical reaction would be balanced.

Figure 9: ReactionTypes Class Diagram



The icons in the top right of some of the classes indicate that those classes have activity diagrams associated with them. These diagrams can be viewed in the corresponding System Behavior section. The specific reactions on the right inherit the activity diagrams from parent classes, where they can be extended. The "Refining" class appears to have "lost" its activity diagram, though, probably due to a bug in the way Rhapsody inherits statecharts; it should be possible to fix by deleting the class and recreating it.

Figure 10: CombustionReaction Class Diagram


Combustion
1
1
General Habitat::Resources::Air
1
1
InputResources
OutputResources
1
1
General Habitat::Resources::AirPollution
General Habitat::Resources::Heat
1
General Habitat::Resources::Petroleu
1

This diagram highlights one of the reactions in detail. Other reaction types would look very similar to this diagram with different combinations of input and output resources.

6.5 Resources

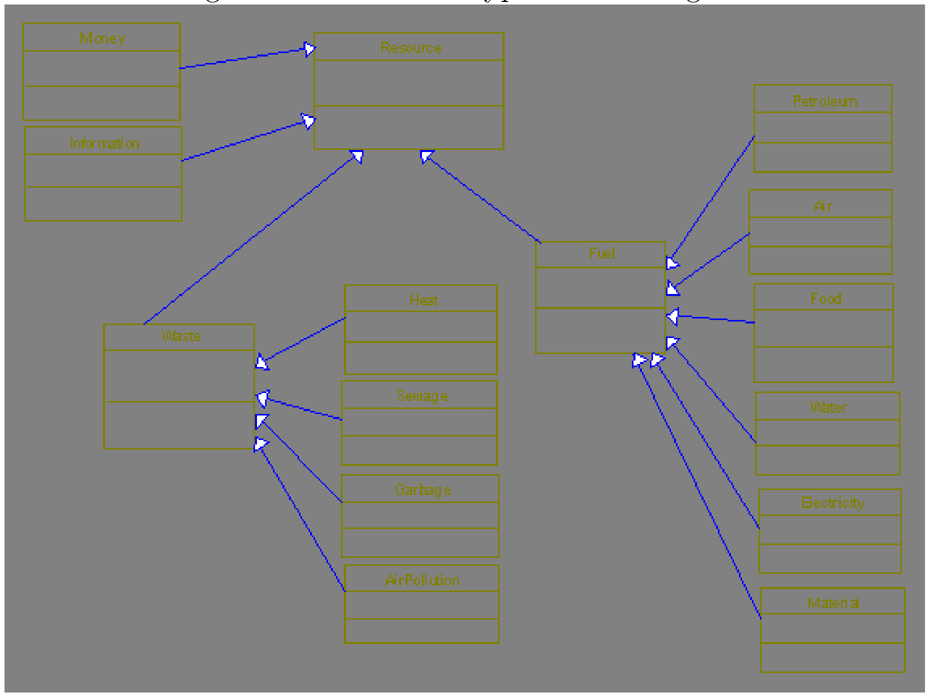
This package details the various generalizations of resources available in the model.

Resource A particular resource of interest that can be contained, traded, or reacted within cells.

Money Financial resources are often exchanged for goods and services, so it's worth tracking how much each cell has on reserve.

Information Information can also be transferred and accounts for education activities or entertainment.

Figure 11: ResourcesTypes Class Diagram



Fuel The Fuel superclass generally refers to any resource that is useful.

Air Rather than get too specific in chemistry terms, this class represents clean useful air for breathing or to provide oxygen for combustion.

Electricity Electrical distribution is one of the oldest networks in the world and serves as a catalyst for many other useful reactions, or merely as a utility to improve the quality of life.

Food Anything people can consume.

Material Any kind of object or artifact that might be transferred. Along with the mass value inherited from resource, material can also have a value density, which can increase with the refinement reaction to represent a lot of what industry does.

Petroleum More traditional fuel products that are not necessarily restricted to oil or derivatives. Anythings that burns to produce energy could be included, such as coal and wood.

Water Clean potable water for drinking or for maintenance.

Waste Superclass that represents byproducts that cells probably don't want to keep around, but that need to be tracked and disposed of appropriately. Waste can still put a load on the transportation infrastructure, and require industrial resources to treat and refine properly.

AirPollution Any kind of gaseous waste.

Garbage Solid waste products, mostly discarded materials.

Heat Otherwise known as entropy, almost every process surely creates waste heat that needs to be dissipated.

Sewage Liquid waste that might be drained through the sewage system.

6.6 Transportation Infrastructure Overlay

Demand Model

As an exercise, let us consider some of the data elements we would want a schema to include that would lend themselves to a good schedule optimizer. Each of these values of interest might need to be expressed and measured in different forms, to indicate whether their values have been projected from previous data, predicted based on current known conditions, or are the actual measured values after the fact. Additionally, projections and predictions would want uncertainties attached to them in order to be of use for contingency planning.

First off, we will list out the information a passenger or piece of cargo wishing to traverse the system would want to convey to us. The simplest schema would consist of a source location, a destination, and a desired time of arrival or departure. But much other information could be collected that would be of use:

- Unique identifier: every database needs to refer to its elements by some unique ID at some point. Many privacy rights activists cringe every time a system forces them to assume one that is traceable back to them. It's beyond the scope of this paper to address the requirements of what can or cannot be gleaned or pieced together by data mining this information. But suffice it to say that privacy and security concerns could be met by currently existing encryption, digital signature, and authentication technology. As an example, suppose that after payment, a unique system identifier was associated with an encrypted, one-time signature generated by the passenger's private key. Only that passenger would be able to decrypt the digital fingerprint that associated their personal identity information with the unique ID stored in the passenger roster. They would be able to prove that it was them who generated that unique signature ID at a later time, say, if they needed an alibi. However, government or private entities that somehow got a hold of the passenger roster wouldn't be able to runs searches, such as "give me a list of all the people who traveled to this shopping mall" or "list all the places John has traveled to lately." For more restrictive governments or law enforcement / monitoring agencies, all or part of this data could be exposed through a key escrow system. The point is all of this framework exists and should be set up from the inception of the system, since the security and authentication model will likely be deeply ingrained into how the rest of the software systems operate. The main problem that most privacy advocates see is that the minimum basic anonymity safeguards are simply not being deployed into the systems of today.

- Schedule constraints / flexibility : optimization thrives on having some slack or flexibility in its constraints. We could achieve more optimal schedules if only passengers could more adequately express things like:

  - What range of times could they be expected to arrive at their destination? *e.g.* Not later than 9:00?
  - How much extra would they be willing to pay to reduce their time in transit, say be giving them preferential treatment in the schedule optimization algorithm? In the same vein, would any of them be interested in paying less to reduce their "pull" on the scheduling algorithm, so their scheduling might flow around "hitchhiking" economically around the empty seats left over in schedules generated to server passengers paying for higher priority routing?
  - What kind of safety factor or time buffer are they comfortable with? Would they be willing to run through an airport to make a tighter connection?

- Accessibility needs : handicapped passengers could make special requests to suit their situation. This could help budget transfer time and resources better. For example, instead of equipping all of the vehicles in a fleet with minimal accessibility features at great expense, a bus system could have 5% of their fleet be fully equipped and serve handicapped passengers as their first priority.

Cargo would have much of the same properties as passengers, perhaps a few more to encode other special handling instructions, hazmat designations, and so forth. As cargo might spend significantly longer stretches of time in the system between warehouses and transfer stations, they might have more stringent tracking and tagging requirements, as well as more flexibility in routing preferences, especially between low priority bulk and high priority overnight shipments.

FIXME: Cargo security via digital signatures, accountability.

Having all this passenger and cargo data pretty much takes care of knowing the transportation system demand inputs.

Route Graph

The next set of standardized data should describe how the transit network itself is set up to handle the demands placed on it. Every transit system could be expressed as a network, so we will liberally apply terms from the networking field to describe some of these concepts. The first assumption we'll have to make is that any transit system could be expressed and modeled as a collection of nodes and connector links. They might vary significantly in complexity and level of detail between transit systems, but they all need to be able to "plug in" to each other for intermodal optimization to work properly.

FIXME: network diameter, node degree

A simple light rail or tram network might consist of a few dozen stations connected by a single track. On the other end of the spectrum, a metropolitan road network modeled in detail would have thousands upon thousands of connective paths, links to probably all of the other nodes of transit, relatively few fixed source and destination nodes, and likely not enough user planning data will ever be made available to predict traffic congestion resulting from construction, weather, accident, or just plain rush hour delays.

In any case, the minimal elements needed to represent this transportation network would include:

- A unique node identifier
- A geographic node location, represented in a standard reference frame such as the WGS-84 latitude, longitude, and altitude used by the GPS system.
- A connectivity matrix, minimally of transit times between node pairs. A special value would indicate that certain node pairs (probably most of them) are not connected at all. This might even be digested from much more complicated routing algorithms, such as street navigation systems. The connectivity matrix

will need adjustments over time, to schedule in planned closures for maintenance, or new routes opening up at particular times.

- Buffer and storage nodes, such as maintenance bays or taxiway queues. These might have special properties with regards to what can and cannot take place.

Vehicle Model

In order to finally traverse this network, though, a transit system ultimately needs some set of vehicles (though many parts of a transit network might be represented as walkways on foot, which we might as well model too in order to help design capacity for escalators, moving walkways, ticketing and security checkpoints? perhaps even to make sure hallways and doorways are wide enough to meet capacity and fire codes). Each vehicle would have associated with it:

- A geographic location within the network, whether it was a geographic location in transit, at or waiting for arrival at a station node, or even occupying a storage or a maintenance bay.
- A passenger or cargo capacity
- A set of rules governing how fast it can navigate its network, how long it takes to load and unload, *etc.*
- Various maintenance details, such as fuel supply, crew refresh schedules, and at least some indicator of the probability that it will reach its destination without breaking down along the way or running late for some other reason.

The system would need a way to introduce its own arbitrarily fixed schedule or other constraints. This could be required merely as a way to allow legacy timetable-based systems to nominally interact with the optimized system. While we could squeeze a more optimal solution by imposing fewer constraints, for various reasons (such as lack of equipage to perform last-minute reroutes), we need some way of communicating and enforcing pre-existing schedule constraints. In the end, this probably isn't any different than the mechanism we'd use for introducing scheduled maintenance stops.

Environmental Factors

The last major category might include "environmental" factors that would affect the performance of the system. These factors could either be predicted in advance with some degree of certainty, or suddenly evolving events such as accidents or breakdowns that require a reformulation of the optimization problem to mitigate.

Weather conditions can have a predicable effect on a system. Updates on rain or snowstorms should be able to make their way into the system so it can plan on having some degree of constrained capacity in advance. Airports can plan to shut down for a few hours while "convective weather cells" (thunderstorms) pass by overhead. As better forecast data has become available, air traffic control centers have actually been able to institute ground delay programs for aircraft all the way at their points of departure, so they don't end up circling in holding patterns near the destination airport, waiting for the inclement weather to abate. Such contingency planning based on externally available data could make their way into streamlining other forms of transportation, albeit less dramatically.

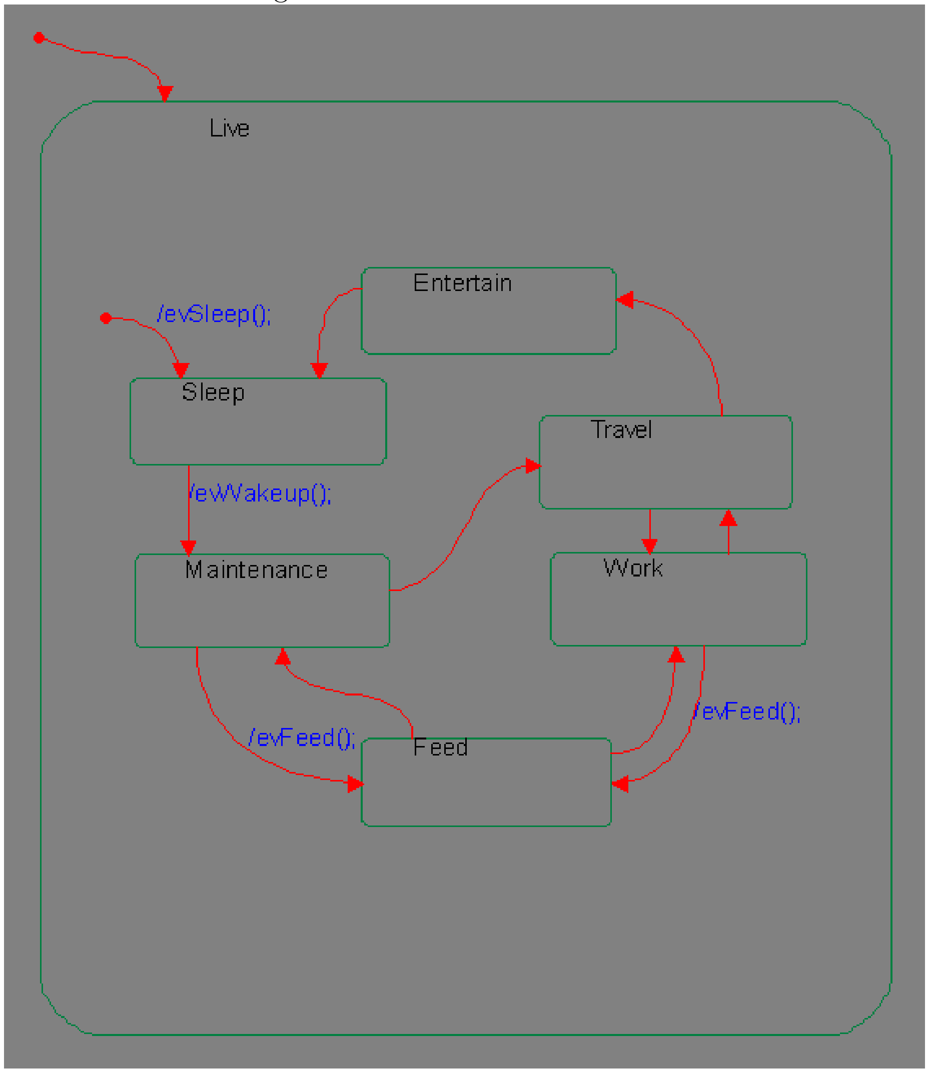
These types of entries will manifest themselves by time-dependent changes to the network connectivity matrices. Each cell would have a probable new value for transit time on that link, accompanied by probable start and end times of the effect.

7 System Behavior

The simulation model is based on a discrete event simulation engine. This means that state changes in the system structure are triggered by the firing of events which occur along the global time line queue. The model executes by populating the global time queue with scheduled events and firing those events in order. Every time an event is activiated, the system global time is advanced to that time. Any state transitions in the model that were blocking on this event are executed so they can perform their activities, which often result in the scheduling of more events in the future. Thus the simulation perpetuates events and continues in time until there are no more events left on the simulation queue.

7.1 Individual

Figure 12: Individual Statechart



The individual transitions from state to state in their daily activities triggered by these events. A fairly simple schedule could be arranged as follows to implement a statechart representing a typical person's day. The statechart depends on having the right combination of events defined and triggered to advance the individual through the full daily cycle.

7.2 ResourceEngine

Each resource engine keeps track of the flow of one resource within a cell. This includes the input of resource from the environment, trade of resources with other cells, internal reactions that transform resources to and from other resources, and waste resource output back to the environment.

The resource engines are initialized to fire push/pull transaction events at regular intervals. Pull transactions would offer to exchange monetary resources for goods and services such as food or electricity. Push transactions relate to the expulsion of waste, and would end up in the immediate environment unless picked up by a transportation system to take to, say, a waste processing plant (represented by an industry) first.

evWakeup Event signalling a person to wake up and begin their day.

evFeed Event signalling that the person should make an attempt at going somewhere to eat.

evSleep Event signalling person to go somewhere (preferably home) so they can sleep.

Relations:

itsResource Each ResourceEngine manages the quantity of one resource for each Cell unit through transactions in/out of the environment, trade with other Cell units via connective transportation cells, or internal reactions within a Cell.

Attributes:

Amount Amount of resource requested per cycle. Type of double, Public

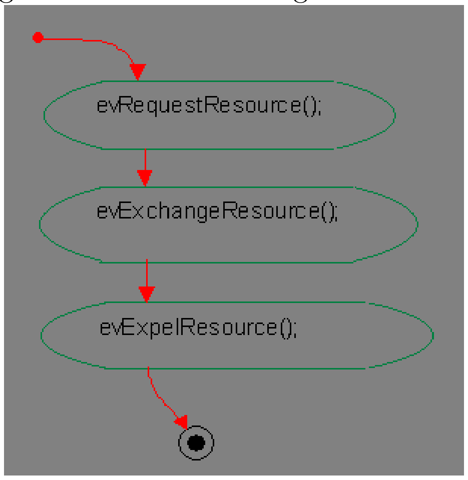
Interval Time interval between requests. Type of double, Public

Activity Diagram

Resources essentially attempt three types of transactions:

1. A request for resources from its parent cell or environment, in client/server pattern.
2. A peer-to-peer trading agreement scheduled through the scenario setup.
3. A dump of resources back to its parent cell or environment.

Figure 13: Resource Engine Statechart



Expel Push transaction to deposit waste into the waste management infrastructure (or the environment).

Action State EntryAction evExpelResource();

Out Transition Target: Terminate

Pull During initial scenario setup, set a starting

> **Action** State EntryAction evRequestResource();
>
> **Out** Transition Target: Trade

Trade Transaction to exchange resource with another cell on the same level.

> **Action** State EntryAction evExchangeResource();
>
> **Out** Transition Target: Expel

Terminate Local Termination State

7.3 ReactionEngine

Resources are required to fuel several reactions that occur within a cell. These reactions are driven by the reactionengines associated with a cell to consume several resources and turn them into other resources and waste.

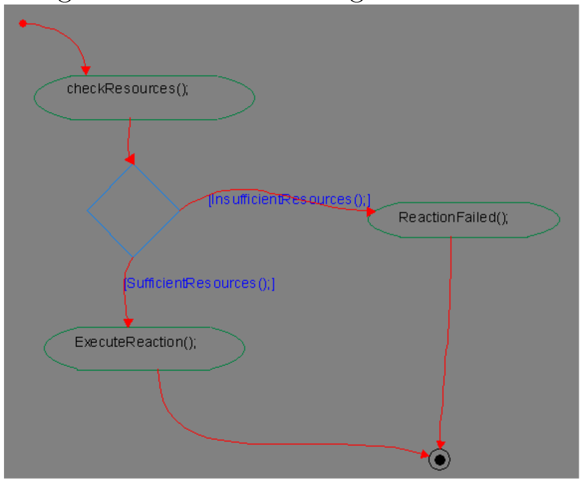
When enough of the input resources become available, the reaction event can commence. Otherwise, the reaction engine asks the resource engine to request the required resources through pull transactions.

Several reaction types are available, but the assigment and scheduling of reaction events is up to the scenario builder.

Subclasses:

> **Combustion**
>
> **Consumption**
>
> **Disposal**
>
> **Refining**

Figure 14: Reaction Engine Statechart



Utilization

WaitForTrigger : Poll input resources to see if there are enough raw materials ready to undergo the reaction.

Action State EntryAction checkResources();

Out Transition Condition Connector Branches:

SufficientResources() Target: ExecuteReaction

InsufficientResources() Target: ReactionFail

ReactionFail Record an appropriate penalty for a failed reaction. If no penalty action is defined, then the failure is merely recorded. These failures could then lead to a detraction in the quality of life output metric.

Action State EntryAction ReactionFailed();

Out Transition Target: Terminate

ExecuteReaction Reduce input quantities and increase output quantities in the ratio defined in this reaction.

Action State EntryAction ExecuteReaction();

Out Transition Target: Terminate

Terminate Local Termination State

InputResources Proxy to ReactionEngine that requests or collects resources available within the cell.

OutputResources Proxy to ResourceEngine that increases associated output resources upon successful reactions.

8 System Requirements Allocation

As with everything else in this design document, a distinction must be made between requirements for the arcology and requirements specific to the arcology simulation model (the actual system of interest). The ability for the simulation to successfully model the fulfillment of fundamental arcology requirements is in itself a requirement.

The simulation tool should be able to quantify estimates for real life occurrences. One of the odd requirements for this system is to provide special failure cases in the event that all of an individual's use cases cannot be met. While the consequences for failure to fulfill a need (such as starvation, homelessness, or sickness do to poor hygiene all very real-world problems) does not necessarily have to be simulated to achieve its purpose in the system, failures do need to be noted and become part of the output of the system. It is of interest to note that failure is an option, and must be properly accounted for as part of the normal operation of the system.

We present separate requirements for the arcology and the simulation model. The simulation requirements are driven by the ability to model interactions between elements that affect the arcology requirements, so they may be seen as further derived.

8.1 Arcology Primitive Requirements

1. Attend to basic occupant needs defined in the Individual use cases described in Live.
    (a) 1.1.Provisions (Feed)
        i. 1.1.1.Food
        ii. 1.1.2.Water
        iii. 1.1.3.Other consumables (vitamins, nutrients, *etc.*)
    (b) 1.2.Indirect assets & qualities
        i. 1.2.1.Shelter, security (Sleep)
        ii. 1.2.2.Health, hygiene maintenance not covered by (Maintenance)
            A. 1.2.2.1.Waste removal
2. Self-sufficiency & sustainability (Work)
    (a) 2.1.Extract required resources from environment
    (b) 2.2.Extract labor from occupants
3. Improve quality of life for occupants (Entertain)
    (a) 3.1.Education
    (b) 3.2.Entertainment
    (c) 3.3.Social interaction

8.2 Arcology Derived Requirements

1. Transformations of resources
   (a) 1.1.Fuel to Waste - byproducts of Arcology Requirements
   (b) 1.2.Construction / deconstruction mechanism - resulting from
2. Accounting & transportation mechanism for resources
   (a) 2.1.Solid - Arcology Requirements
   (b) 2.2.Liquid - Arcology Requirements
   (c) 2.3.Gaseous - Arcology Requirements
   (d) 2.4.Information - Arcology Requirements ,
   (e) 2.5.Monetary credits - intermediary between exchanges and transformations.
3. Transportation mechanism for resources & occupants in order to satisfy all of the above (Travel)

8.3 Arcology Requirements Traceability

FIXME: insert content

8.4 Arcology Specifications

8.4.1 Specs for Arcology Primitive Requirements

1. Attend to basic occupant needs defined in the Individual use cases described in Live.

(a) 1.1.Provisions

i. 1.1.1.Food : $>$ 1.77 kg per diem

ii. 1.1.2.Water : $>$ 2.3 kg per diem

iii. 1.1.3.Other consumables (vitamins, nutrients, *etc.*)

(b) 1.2.Indirect assets & qualities

i. 1.2.1.Shelter, security : distribution of 5 - 10 hours of sleep, personal living quarters with $>$ 37 m2 of personal living space.

ii. 1.2.2.Health, hygiene maintenance not covered by (1.a), *e.g.* timely delivery of emergency supplies & services.

A. 1.2.2.1.Waste removal - roughly equivalent to total of Provisions.

2. Self-sufficiency & sustainability

(a) 2.1.Extract required resources from environment - varies, should balance with environmental production rates, if known.

(b) 2.2.Extract labor from occupants - a distribution of around 1/3 of the daily cycle. Provide $>$ 19 m2 of work space.

3. Improve quality of life for occupants : maintain or increase amount of leftover time dedicated to the following:

(a) 3.1.Education

(b) 3.2.Entertainment

(c) 3.3.Social interaction

8.4.2 Specs for Arcology Derived Requirements

1. Transformations of resources

   (a) 1.1.Fuel to Waste - roughly 1 to 1 conversion factor by mass.

   (b) 1.2.Construction / deconstruction mechanism

2. Accounting & transportation mechanism for resources - Conversion, creation, consumption of each class of resource.

3. Transportation mechanism for resources & occupants

   (a) 3.1.Quantify measures of effectiveness - cost, latency, throughput, efficiency

9 Simulation Design

9.1 Use Cases

Use cases for the simulation model:

- Set up a modeling scenario using input data.
  - Build bottom-up scenario: Since the arcology is designed from the ground-up, starting at the individual level, the structure of our model would allows us to calculate the aggregate performance at higher levels of organization, such as a the city and national level.
    - Define # of simulation units, connectivity between units, schedule of transaction events, schedule of reaction events, initial conditions.
    - Output aggregate performance for groups of units.
  - Build top-down scenario: The present day scenario is built in a top-down fashion from various data sources. Statistics are only tracked from relatively high levels on the organizational hierarchy, so we must extrapolate some data to flow down to fill the detailed subcells of the structure.

    - ∗ Define high-level consumption rates for groups (using publicly tracked & available data), provide distribution histograms for each type of resource & transaction rates for each subunit.
    - ∗ Output unit-level quality of life, performance.

- Execution of simulation model to produce output data.
- Postprocessing & analysis of output data into performance metrics.
- Design-of-Experiments method of parametric analysis for solution space exploration & optimization.

9.2 Operational Concept

The simulation basically boils down to an accounting of conversion and transaction events that move resources between themselves and their environment. Therefore, most of the coding involves making and managing container objects. Building from the ground up, here's the implementation plan:

1. Resource containers are the most elementary class. They merely have to choose an identity, and store a number representing how much of this resource the owning object has pooled together. It needs getter and setter functions, and a master dictionary for looking up other useful properties associated with that type of resource (such as density, , market value, *etc.*) that might be used for various other calculations. Money and information are considered resources as well for tracking purposes, but they basically constitute an "activation energy" for a reaction or transaction to proceed, so are treated somewhat differently.
2. Reactors come in various forms and are intended to provide balanced conversions from one set of resources to others (often waste).

3. Cell objects are the hierarchical units. These will be the most complex but most useful class used in the simulation. They each can contain some combination of:

   (a) resources

   (b) reactors for converting internal resources from one to another

   (c) buffers and constraints on the amount of resources they can hold before having to push them elsewhere

   (d) parent, child, and peer cells with which to interact, such as by scheduling transactions and reporting metrics up and down their chain of command.

   (e) internal agendas used to schedule reaction and transaction events.

4. Connective meshes define which cells can actually interact with each other, representing the function and capacity constraints of various transport networks that move resources between the cells in the system. They can exact a cost (in terms of money transactions and resources consumed.

5. The instantiation framework is what reads the scenario file and begins to create cell objects and set up the simulation. This is where a modeling language would come in.

6. A reporting engine collects data from the simulation at desired intervals and needs to be programmed to extract useful data and analyses from the simulation.

What potential uses could this transportation network simulation have? The types of problems I hope it will be useful for is demand generation. Different types of transportation infrastructures could be evaluated against each other to determine how well they meet that demand. Many existing transportation optimization problems tackle ways to increase throughput or capacity. But the task of urban planning should focus more on minimizing demand in addition to maximizing capacity. For example, instituting staggered work hours or telecommuting programs can relieve peak rush hour traffic congestion without spending a fortune widening highways and building additional infrastructure just to handle a few hours of peak usage a week. It would be nice to know how much incentives to provide to encourage employers to implement flexible work hours, or how much to invest in telecommuting infrastructure (such as municipal broadband) in order to provide productivity benefits similar to simply adding highway lanes or additional thoroughfares.

Also, by simulating demand, we can create a transportation system that is more sensitive to individual needs rather than the aggregate flow of travelers. This would allow us to create schedules around the traveler's itinerary rather than forcing the traveler to always plan around fixed train, bus, ferry, and aircraft timetables. For instance, if everyone starts work exactly at 8:30, but buses only run hourly on the hour to that particular stop, then the extra half hour everyone spends waiting per day essentially counts as extra commuting time in their books, even though the bus operators might only measure the time the passenger spends sitting on the bus and perhaps waiting for known connections.

An advanced bussing system that dynamically generates routes and schedules based on individual source and destination requests from each passenger could achieve efficiencies and meet customer requirements far better than what we have today, and could make public transportation more attractive to people who drive their own vehicles in order to maintain that degree of flexibility. During peak commuting hours, this has the potential to reduce individual commute times, as buses could be scheduled more like express routes and fill up at one location and proceed directly to stops at a common destination with minimal stops or transfers or jaunts down back roads along the way. During off-peak hours, buses would not run nearly empty along the same routes with very low frequency, but would run on demand, cutting down wait times and making them a more convenient option for midday or late night errands. An effective public transportation system should make a metropolitan area "smaller", where each of its districts are easily accessible for connecting places where people live, work, and go for necessary errands and entertainment. Under the current hub and spoke paradigm, unless your source and destinations are near hubs or just down the street, travel on the system through two hubs can take up a significant portion of time. This time would typically consist of at least 5-10 minutes of waiting for each connection and perhaps 10-20 minutes riding each segment; the result being that driving independently in one's own car would take between half or even a quarter of the time that the trip would take on public transit, even with traffic. For commuters, this time savings doubles, so it is of little surprise that most commuters prefer to spend the extra gas, auto maintenance, and toil to gain 1-2 hours of family time at home a day. Public transportation systems could still use a lot of improvement to make mass transit desirable over driving, rather than just an alternative to driving that merely relieves congestion on the roadways so that other drivers end up with a better traffic experience.

9.2.1 Urban Systems comparison framework

FIXME: Sensor, Decision Maker, Effector roles.

9.2.2 Performance Metrics

What defines a good inter-modal transit system? The conflicting goals might be characterized as: speed, response, coverage, and efficiency.

- "Speed" refers to how fast the transit system can get a passenger or cargo item from point A to point B. Unfortunately, this does not depend entirely on the cruise speed of the vehicle alone, but also time spent making transfers and additional preparations (such as passenger check-in and luggage screening at airports)

- "Response" refers to the frequency of service, particularly how well it matches and meets demand. Extra time that people have to wait at their source or destination should be counted against the system... though this is almost always overlooked in transit performance metrics today. The data just isn't available, or people have relegated themselves to adjust their schedules around the system's timetables. This "response" metric will usually be at odds with efficiency due to economies of scale, since making passengers wait longer times between pickups can cluster them into larger groups.

- "Coverage" refers to how well the transit system covers the service area, which should include how far people have to walk from their doorstop to enter the system. Broad coverage is more difficult to achieve for a mass transit system, especially as population density decreases and residences and businesses are more spread apart.

- "Efficiency" might refer to two terms: that in terms of frugal monetary spending on operating costs and fixed infrastructure investments, as well as in terms of conservation of fuel and resource utilization. Efficiency pretty much always counterbalances against each of the three other goals, so we often must express how much extra money or fuel we are willing to expend for whatever modest gains in speed, response, or coverage.

9.2.3 Model Validation against Actual System

9.2.4 Multiple Model comparison

So what can we do once we have a coupled system of transit networks, a simulation of that system, and an optimization framework that can set up schedules for the simulation (or the actual system) to evaluate? We can set up yet another iterative optimization - this time of the actual system configurations and not just one schedule. This will help us evaluate urban design and infrastructure in ways that should help drive progress towards efficient and sustainable societies that serve the people who live in them. We can propose a new construction or infrastructure project, show its benefits in a simulated model, and later validate those benefits using data collected from the real system. Competing models for improvements might even have the chance to provide benchmarks using the same methodology.

FIXME: Insert diagram

The ability to compare several optimization components, several system structures, different modeling methodologies, all using the same data interchange format to facilitate direct comparisons between both real and simulated evolution of the scenarios, allows us to take a systematic, objective approach to tackling urban improvement projects. Adapting such a simulated and real system performance comparison framework will allow us to have more complete impact assessments by making sure every study or proposal is analyzed consistently, using the same inputs, and doesn't sweep away or ignore unwanted side effects and consequences. Urban planners could use these studies to provide ammunition for driving changes toward the way they envision their communities. And the focus on operational efficiency and continuous improvement driven by pervasive measurement and analysis will lead towards a leaner, sustainable society where more resources could be directed towards forward progress instead of consumption.

9.2.5 Multiple Schedule Optimization Algorithm comparison

The main way we'll be able to improve efficiency (aside from simply improving fuel efficiency) would be to use existing resources smarter - through extensive use of optimization. With enough planning and foresight, optimal scheduling is straightforward to perform. However, things never quite go as planned, due to a variety of unpredictable factors such as weather and accidents and just plain last-minute changes in schedules. In order for the optimal plan to be of much use, we ought to continually collect enough data in real-time to monitor and reevaluate schedules as able. This requires that we have a communications system in place that allows us to poll the status of our cargo, passengers, and transportation vehicles. Equipage for this type of system would have been cost prohibitive in the not-too-distant past, but now that geolocation devices, mobile computing, wireless networking, and cellular data network backbones have become nearly ubiquitous, we'd be silly to not put all this capability to good use.

So instead of having fixed timetables locked down and set weeks, months, or even years in advanced, based only on projections from previous observations of seasonal, aggregate flows of the past, and barely ever followed to the minute, we could perform schedule optimization on actual data. This data would factor in individual requests from each customer, including their destination and schedule constraints (or better yet, their schedule flexibility). Vehicles could report their current location and status, meaning they'll always be right on time - especially since they could report their arrival time themselves. Monitoring and reporting of deteriorating road or weather conditions could automatically update the schedules of every vehicle in the network to account for and mitigate the effects of new delays.

We live in an uncertain world. How will the system deal with uncertainty and unexpected events in schedules? Probability should be built in to the optimization problem formulation, and one of the goals of the optimizer might be to minimize the impact of unfavorable (but probable) events. Analysis of historical records can generate performance metric associated with each vehicle, route, weather prediction, *etc.* A useful way of representing on-time performance probabilistically is to reconstruct the data from the cumulative distribution function (CDF) associated with the prediction.

This would work much better than simply providing means and standard deviations, since most transit data is so skewed towards being late than being early. It's much easier to break down and be several hours late, than to speed across a transportation link in record time. The CDF can be quantized to reduce computational complexity, at the cost of adding extra conservative wait time buffers between connections.

FIXME: figure 1

While this type of data will be monitored and collected, only certain parts of the tail will interest the schedule optimizer.

FIXME: figure 2

We're primarily interested in what time the vast majority of the vehicles will arrive, as well as what hopefully small percentage are beset by schedule-impacting delays. There's no fixed "magic percentile" that would determine how much extra buffer time to schedule to make sure everyone makes their connections. This will likely be set arbitrarily at the beginning, as all of these factors contribute to an overall "confidence in planned schedule volatility" metric (maybe more easily expressed as an opposing "schedule stability" metric). With the optimizer system, we can recompute new schedules whenever an unexpected event comes up - such as when a vehicle is delayed enough to fall on the tail end of the CDF and it misses its connection. The optimizer can take that new information into account and simply create a new schedule based on these existing conditions - which will likely result in diverting other vehicles over to take care of the late straggling passengers. So the risk analysis that determines how aggressively to schedule extra buffers into the system would depend on how much impact a schedule recovery plan would have. Planning in large buffers to reduce risk likelihood means extra wait time for passengers and more idle time for vehicles in order to ensure that the schedule stays stable. The ability to drastically reduce these buffers means the whole system could run at a faster pace. If the cost of recovering from missed connections is low - say to catch a subway train that runs every 5 minutes - then the scheduler can comfortably deal with smaller buffers and higher schedule volatility risk. In the case of an airplane network where flights run between cities maybe once or twice a day, a missed connection would mean putting people up in hotels or chartering additional make-up flights. In this case, increased schedule awareness can also help by figuring out the total impact on whether it's even worth holding flights for latecomers to make their connections.

FIXME: figure 3

So in addition to the overall transit system performance optimization goals we discussed in 9.2.2, we also want to introduce some practical optimization goals that will help the scheduler intelligently create and maintain buffers to deal with uncertainty. Now, how to formulate and computer this enhancement is beyond me, since it would likely require the optimizer to do risk-impact assessments on every combination of missed connection. But that's no reason to shirk away from providing the necessary information about on-time performance in the data protocol now, so that future generations of engineers could tackle it.

The final category of optimization constraints would come from the operators of the various transit networks. This would allow them to add crew and maintenance schedules, such that they can pick up and drop off drivers, pilots, and other staff at certain locations, or make sure that a vehicle ends up in a certain maintenance bay every so often for refueling and service.

These constraints are typically easy to add without a lot of heartburn, since they tend to help reduce the number of branch and bound paths that a mixed integer programming optimizer needs to search through to converge on a solution - at least as long as the solution remains feasible. The challenge comes in that expressing these constraints should be the job of the separate transit network organizations, and the abstract protocols needed to express these constraints would likely require extensive knowledge of how the global optimization problem is formulated and solved. It is undesirable to have this information format coupled too closely to the formulation, since it will make it more difficult to change and upgrade the optimization engine in the future. We don't want to force everyone to have to radically change their code at the same time throughout the system every time we want to introduce an incremental upgrade. We also don't want the entire systems upgrade to fail because of one or two late development efforts. We want enough abstraction built in so that they might make changes at their own pace to take advantage of new scheduling and optimization features and capabilities. Their abstract representation of their constraints needs the ability to compile itself so it can be applied to both the old and the new versions of the optimization formulation.

Unfortunately, I'm not able to come up with a language abstract enough that would allow the businesses to express what maintenance needs a generic optimizer must meet, without cheating and taking advantage of intimate knowledge of the formulation and the meaning of its various variables. A sophisticated abstraction language processor would have to take the expression and transform them into equations that relate particular variables to each other or to newly introduced variables. This processor would likely be nontrivial to implement and be prone to unexpected behaviors and errors. So a more practical way to handle crew and vehicle maintenance schedules would have the operators compute maintenance schedules separately from the main globally optimized schedule, and insert them as fixed constraints using the legacy scheduling interface. The end result of performing iterations of this would not be as optimal as if the global optimizer took maintenance into account. But at least it starts close to an optimal solution, and provides our necessary layer of abstraction. The iterations would proceed something like:

1. Transit network operator would provide the number and current locations of available vehicles at the beginning of the day

2. The global optimizer takes the customer demands and those initial conditions,

and furnishes the schedule desired of that transit system.

3. The operators manually (or semi-heuristically) tweak the schedule to ensure that particular vehicles end up in nearby maintenance bays when they're due. These get fed back into the global optimization as constraints.

4. The global optimizer find a new solution taking these new constraints into account, filling in new gaps in the schedule and hopefully not straying too far from the original optimal objective function result.

This would let us converge on a solution set somewhat near the optimal one that takes maintenance factors into account without tying down the programming to a particular implementation of the optimizer.

A global optimizer that did include operator goals and scheduling constraints isn't out of the realm of possibility, however. Additional complexity could be added by allowing these third parties to add their own set of constraint statements, even weighted objective functions. Some discipline would still be needed to keep the system stable. In the original form, the problem is formulated in advance, and the data provided by passengers and schedules add constraints in a consistent manner - the worst thing we should need to worry about are infeasible solutions. However, by allowing third parties deeper control of objective functions and constraint statements, we're exposing the system to a host of potential problems and vulnerabilities:

- Malformed or even malicious statements can make the problem intractable. There may be ways to identify some offending statements and automatically detect and flag them to somehow alert or even filter them out of the calculations - but the latter approach could likely create unpredictable results.

- We'd need ownership and permissions on variables to separate the components provided by different parties. This would ensure that operators don't introduce constraints that could penalize their competitors.

- Many companies pride themselves on their own optimization capabilities. We may need a mechanism to protect proprietary information about their mode of operation revealed in their contributed code statements. We could allow them to submit "black box" modules that manage to interact properly with the rest of the global optimization. An alternative method may be to partition the problem such that they're entirely responsible for optimizing their segment of the global calculation, interacting with the rest of the system through the input and output protocols.

Hopefully these reasons (and probably others) have helped to articulate why I haven't addressed these issues in the current incarnation of this thesis. But this might be the beginning of an outline to tackle these considerations in the future.

9.2.6 Intentional data interchange

FIXME: Publish / Subscribe plan interaction

9.3 Simulation Requirements

1. Insert scenarios as inputs
   (a) 1.1.Numbers of units involved (people, transportation mechanisms, industrial entities, *etc.*)
   (b) 1.2.Available resources from environment, initial conditions
   (c) 1.3.Resource conversion rates, schedules, functions
2. Simulation execution - model resource consumption/production rates, providing estimates on actual performance (pending validation of model)
3. Output metrics defined and calculated
   (a) 3.1.Qualitative measures of performance
   (b) 3.2.Quantitative measures of performance
   (c) 3.3.Allow possibility for formulating optimization problems to aid in benefits analysis & decision-making in arcology design.
4. Significant events (to be defined by modeling use case scenario case studies) should be modelable by scenario architecture - important activities that have impact on performance measures should not be ignored. Should provide at least approximate methods of simulating effects that are difficult to model.
5. Specification of accuracy in estimates & predictions. (Goal of ~20%)

9.4 Specs for Simulation

These mostly deal with measures necessary to create hardware and programming efficiency requirements for successful execution, and have little else to do with the planning or setup of the model. Therefore, we won't dwell too much on these, but provide a placeholder for lower-level specification by software engineers.

1. Ability to model baseline scenario on the order of magnitude of $\sim 10^{10}$ units processing a 24 hour period of events on currently available computer hardware.

2. Attain a reasonable execution time of less than 10 hours to process the scheduled event queue such that the baseline scenario, assuming ~100 events per unit during the 24 hour period.

3. Achieve realtime or faster simulation speed of the baseline scenario.

Part III

Implementation Notes

10 UML Diagram Tools

The approach for this project began using Ilogix Rhapsody® in C++ Development Edition to construct UML diagrams of the Arcology model. Work proceeds under the expectation that the code generation facilities of Rhapsody could be used to embed C++ source code in the framework to compile and run a working executable as part of the Systems Modeling and Analysis course in the future. As an added benefit, Rhapsody also provides documentation generation of the model in rich text format. This project documentation is interspersed into this report with appropriate commentary and then exported to html.

One of the side effects of using Rhapsody include some subtle differences in naming conventions, presumably used to simplify the merging of the standard OMG UML specification with the practical realities of software engineering frameworks. Notably, Rhapsody uses "Object Model Diagrams" in place of both "Class Diagrams" and "Instance Diagrams". Since this project deals with abstract models, we will almost always be referring to class diagrams except when dealing with actual scenarios.

More recent diagrams covering the design of the simulation framework itself were done using the Umbrello UML diagram tool. While its code generation capabilities are nowhere as strong as that of the commercial tools, it can generate stubs for several languages, including the XML Schema that will be used for cross-component data interchange discussed below.

11 Data Interchange Schema

In order to operate in an inter-modal fashion, however, different segments of bus, rail, and even taxi and aircraft platforms must be able to exchange data with each other in order to feed the formulation of the global optimization problem. This also needs to interoperate between multiple jurisdictions and carriers, who will still want control over their own vehicle resources.

What kind of features would such a schedule collaboration system need to make a diverse set of platforms interoperate? First of all, we need to define a common language used to publish and exchange schedule and status data. Next, we would want to define schemas representing the types of data that are actually required, desired, or merely expressed as comments for general informational purposes. Some of the properties desired by this scheme could certainly be handled by an data representation framework like that provided by XML (extensible markup language):

- It should have a standard set of tools for processing and manipulating the data, a la XML's parsers and stylesheet transformations.

- The data representation format should be extensible, allowing newer versions of software to introduce new data types and tags without breaking older software that doesn't expect or understand the additional data. In a similar vein, older software in the system should still preserve these newer data structures in messages that it passes along between other, perhaps newer or more capable software components that understand and can make use of it.

- The schemas should be centrally version controlled and available for verifying data types, *etc.*

This language feature set would allow different organizations to continue to share and integrate their logistics information, even as the set and functionality of the data schemas grow, change, and evolve over time. Incremental additions can be introduced, such as adding field for, say, the error or uncertainty surrounding a predicted arrival time - information that we might not be able to make good use of now, but could give us tangible benefits once we learn to process it better. Major version changes that alter the meaning of data fields in ways that are fundamentally incompatible with earlier versions could be introduced and managed by a central standards body, while a set of standard transformation filters could be provided to convert as much data between major revisions as possible.

12 Discrete Event Simulation Framework

The prototype framework consists of two major parts. The simulation code is written in python making heavy use of the SimPy module, while the formulation of the schedule optimization problem in lp-solve's modeling language is handled by a perl script. The simulation code initializes the optimization problem's variables using a simple text file, while the resulting model formulation file is read and solved by python's lp-solve module at various times throughout the sim.

The schedule optimizer was written first. Being the "brains" of this framework, it imposes a few major constraints to the way our transit network can be modeled.

The transit system must be modeled by a network of "station" nodes representing the entry, exit, and transfer points for passengers and cargo. Passengers and cargo can only move between nodes on vehicles, which can transfer between any two connected stations at regular, synchronized intervals. Several vehicle types can be made available, and can differ in passenger capacity per vehicle, cost per transit event, connectivity graph between nodes, the maximum number of vehicles allowed to visit a station at the same time, and a host of other measures and constraints.

The most crippling part of the model deals with timing. Time is dealt with in terms of synchronized discrete timesteps, during which the state of the entire system can be represented at one point in time by a complete set of variables. At each time step, the state of the system must be such that every vehicle is stopped at a node. By the next time step, all passenger transfers must have been made and all vehicles must have completed their transit to the next station node (or else stayed in place at their current station). When the simulation translates this to events in continuous time, this means that all stations synchronously act in unison, where every vehicle departs simultaneously, travel all at the same time, and offload passengers at their destinations simultaneously, and all wait together for passengers to transfer to make connections. While this obviously constrains the flexibility of the model in a big way, this arrangement allows the schedule optimizer the flexibility it needs to balance hub-and-spoke transfers with more direct paths, depending on the capacity and economics of the vehicles made available.

Therefore, the model used in this analysis is that of transit stations that are an equal distance apart (at least in terms of transit time) and that each and every vehicle waits the same amount of time for passenger transfers to complete before they disembark for their next destination. The result is that, in reality, a fair amount of time is bound to be wasted under this model as all vehicles must stop and wait for transfers at all intermediate stations on their paths, even if they are not transferring passengers.

There are at least three approaches to making the models a little more realistic. One is to introduce longer transit segments between nodes that are two or more times longer than the "unit" segments between adjacent nodes. This can be accomplished by adding "non-station nodes" in between pairs of actual stations, and enforcing conditions that prevent passengers from transferring off of the vehicles they are already riding. By adding more and more of these nodes to all segments, we could achieve greater precision when attempting to match the real-world state with our representation of the system with discrete time steps.

A slightly cleaner solution may be to rewrite the optimization problem formulation to support transit between nodes taking a configurable length of time in discreet timesteps. This would eliminate many of the extra variables that would otherwise be associated with the phantom non-station nodes, at the expense of needing a bit messier initialization and solution parsing logic for the state information that is no longer carried by actual variables at in-between departure and arrival times. For example, suppose a long trip would take more timesteps than a particular schedule optimization run was handling. If that vehicle doesn't end up at a station by the end of the modeled time, then it would not even have any variables created to represent it or its passengers, and logic external to the schedule optimizer would have to be created to make sure its state continuse to evolve such that it gets closer to its destination in the sim.

12.1 Computing Considerations

Large scale global optimization can require a lot of computing power. It falls under the class of NP hard problems that scale exponentially with the number of transit nodes we add to the transportation system. Let's look at some of the ways in which problems of this size can be tackled.

The solver should be set up to run in parallel across several CPUs, scaleable to a massive clustering system. Many linear and mixed integer solvers have the capability to run on this type of platform, so it's not something we have to worry about directly.

We still would need to resort to a host of other tricks to reduce the computational complexity enough to approach any problems of any appreciable size. Most of them involve introducing some sort of constraint to reduce the number of branch and bound paths search in the solution space.

- The easiest way to reduce the computational complexity is to partition the problem into smaller parts. Since these types of "traveling salesman" problems scale exponentially with respect to the number of nodes, the number of branches to search would be drastically reduced.

- Adding link constraints is also another way of reducing the search space. Not every node needs to be linked to every other node. So often we will resort to building a connectivity matrix to define which source nodes can get to which destination nodes. With road and rail, only adjacent nodes are directly connected. Distant nodes would requirer transit through other city or station "nodes"

- With aircraft, of course, most vehicles can travel directly from any node to just about any other node in the network. In this case, it may be helpful to add "max connections" constraints, to keep the system for searching through impractically long schedules. An itinerary that made a passenger jump between more than two or three connecting airports would likely be rejected by that person. Of course, low priority bulk cargo may find some advantage through waiting for these multiple connections, filling in otherwise "empty" space leftover on any flight where the opportunity arose to get it slightly closer to its destination. But at some point all of the extra handling and transfer overhead ought to outweigh whatever small price break.

- Just about any schedule constraint that would help "lock down" otherwise free-floating variables would help reduce the search space. Feeding in initial conditions - like the current location of the fleet, or stops that must be made by a certain time (for example, to ensure buses take all passengers to a stadium well before a game starts) would help speed the optimization along.

- Sometimes it may be necessary to simply add other heuristic or even arbitrary constraints to help the system converge on a solution. Many of these constraints probably won't even affect the solution, but constrain the search space enough to allow a much quicker answer.

All else failing, many mixed-integer programming solvers also allow "good enough" solutions to be given without a complete exhaustive search of the solution space. Modern MIP solvers can be pretty clever about searching the "most promising" paths first, so completing the entire exhaustive search would yield little improvement on the objective function. Of course, this technique only applies if a feasible solution is found at all.

Finally, a sophisticated optimization would involve precomputing most of the possible schedules in advance., and then have the ability to account for the effects of small changes with only minimal recalculation of the final optimal solution. This type of incremental adjustment may be necessary to recover from small, unexpected schedule breakdowns. Suppose a vehicle suddenly announces that it will be arriving 30 minutes late to a hub node. If recomputing the entire optimal solution taking this new information into account would take a few hours of number crunching, we obviously don't want everything to grind to a halt while waiting for the scheduler to tell us what to do next. An "incremental update" to the solution performed with minimal recalculation might be achieved by determining which vast majority of system variables shouldn't be affected, and formulate a highly-constrained optimization problem that only searches through a small set of variables affected by the unexpected change in one or two schedule input values. We'd need to develop a heuristic to determine exactly how far out this limited set of "affected variables" should reach.

Another scheme might involve jumping back into a snapshot of the state of the large optimization and only recalculate internal values that have changed with the modified inputs. Perhaps some solvers have this ability.

1.

Part IV

Multimodal Mass Transit Simulation

13 Purpose

This programming project serves to realize an urban multi-modal transit simulation designed during the course of the systems engineering master's program. The program will take a systems approach to modeling human habitats and the transportation networks that keep them running. We would use such a simulation framework to create a baseline model of current day capacity, and then create future models to compare the effects and quantify the benefits of investments in future infrastructure. These kinds of tools would be instrumental in making a case for the development and construction of highly efficient arcologies or other forms of well-integrated compact cities. But nominally, we could apply it towards evaluating and tracking the effectiveness of present-day city growth philosophies.

Framework Capabilities

The primary features that this optimization framework sought to achieve include:

- Demand-responsive routing rather than operation on a fixed schedule. This is necessary for us to worry less about generating transit designs around peak demand levels that do not function as efficiently with nominal demand levels. We also hope that the system would utilize command and control networks that take advantage of available communications infrastructure to make requests and guide passengers through the system.

- Allow optimal transfer strategies to emerge. At different loading levels, the system vehicles may organize themselves like hub & spoke / feeder & trunk networks for efficiency, or begin to resemble more direct point-to-point routing during lighter loads or when existing hubs become constrained.

- Multi-objective goal functions, including terms for maximizing service quality such as low average latency from sources to destinations, high throughput, and efficiency terms that would minimize general operating costs associated with the number of vehicles operating in the fleet and the number of segments they would have to travel.

The SimPy discrete event simulation framework in the Python scripting language forms the core of the system model. Including the Psyco Python runtime optimizer helps certain routines run closer to native speed and gives the model a 1-2 order of magnitude increase in computation speed. The LP_Solve package performs schedule optimization tasks and feeds the results back to the simulation for execution.

14 Concept Requirements

14.1 Mass Transit Optimization Goals

This simulation constructs a simple transit network with passengers traveling from source nodes to destination nodes. The scheduler attempts to provide an optimal or quasi-optimal schedule of transit fleet vehicles with various capacities, operating costs, and nodes serviced that will transfer the passengers to their final destination. Through parametric analysis of different demand loading and network topologies, we hope to define some characteristics of urban areas that enable the system to meet the opposing passenger demand and vehicle utilization objectives efficiently.

14.2 Fleet Schedule Optimization Objectives

The objective function of the transit vehicle schedule optimization is a weighted composite of the number of passengers served, the time they are delivered, and a flat cost incurred per vehicle leg. The weight on each objective typically puts them on different orders of magnitude, such that a secondary objective will not be considered until the primary objective reaches an optimal point.

1. Maximize the number of passengers delivered to their final destinations. All passengers are currently weighted equally, which means during instances where the system is operating beyond capacity, the optimizer will favor passengers who are close to their destinations. There is currently no zone tracking to ensure that passengers traveling long distances can "pay more" to compensate for the higher transit cost. The objective function also provides no reward for moving passengers partway, so a particular solution will either move a passenger all the way to their destination node or not at all.

2. Minimize the amount of time the passengers spend in the transit system. This is accomplished by adding a linear bonus term to the objective function that rewards the system for delivering passengers to their destinations at earlier times.

3. An optional objective to minimize deviation from a desired fleet size could be used. We could simply minimize the number of vehicles in use, but we'd have

to find some way to balance this with the passenger service objectives. Plus, most service operators have a fixed number of vehicles and drivers to employ, and the optimizer could make use extra vehicles to improve passenger service quality, as well as make recommendations of when to rent additional vehicles and drivers temporarily to meet demand.

4. Minimize the cost of operating the vehicle fleet. This is currently expressed by a simple flat cost incurred by each segment a vehicle travels. Each vehicle type could have a different cost per segment, such that a vehicle with a higher capacity would have a greater cost per time unit. Currently no cost is deducted for vehicles simply idling at stations or for prepping a vehicle entering service, but those terms would be easy to add.

These objectives would be subject to the following model constraints:

- Conservation of passengers and vehicles moving between nodes. Passengers and vehicles should be neither created or destroyed during the course of the schedule.

- Passenger movement between nodes constrained by the capacity provided by vehicle movements between nodes. Passengers can only move in the network when carried by vehicles. The optimization problem currently allows passengers to wait and transfer freely between vehicles at station nodes.

- The transit system constrains vehicle movement by many factors:

  - A connectivity matrix allows vehicles of a certain type to only travel between connected nodes. This allows us to model different modes of transit that are only available from certain nodes. For example, a certain subset of nodes could be served by a rail system, while the rest of the nodes would only be accessible via bus service. The connectivity matrix provides enough flexibility to model a transit system as a collection of directed graphs, so nodes could be connected by one-way or bi-directional links.

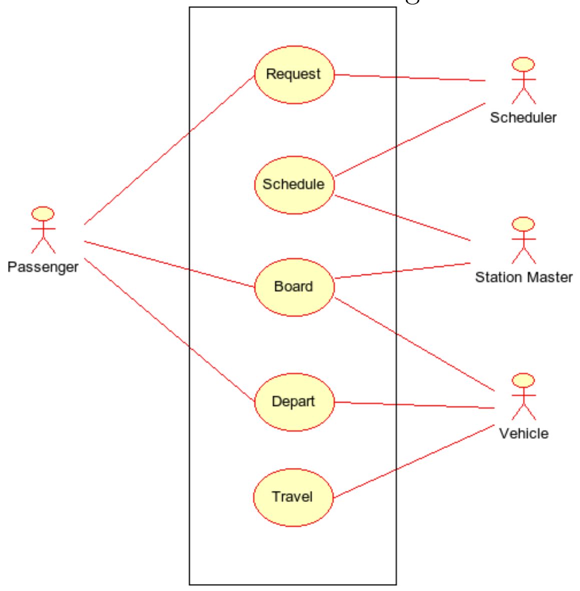
– Station and waypoint capacity constraints could prevent too many vehicles from visiting the same station or route simultaneously.

– A hard maximum fleet size might prevent some unrealistic solutions.

We could add some arbitrary constraints somewhat easily. These could include a maximum number of vehicles on a group of segments or waypoints that have been grouped together to represent a constrained resource, such a bottlenecked intersection or canal.

One notable constraint that this optimization does not attempt to handle is a required time of arrival (RTA) for passengers, it only optimizes based on the time people specify that they are available to depart. Oftentimes people would want to arrive at their destination just before the fixed start of their work day, or at an airport in time to catch a flight. Because this optimizer uses an inventory management approach, adding this information would result in an exponential increase in decision variables. This would add a lot of complexity to the problem and make it take much longer to solve. Combined with the fact that many of the passengers wouldn't have need of this functionality (such as the ones who are leaving work or the airport and just want to get to their destination as soon as possible), the schedule optimizer declines to consider this constraint. An algorithm external to this schedule optimizer would need to provide a rough estimate of the required time of departure (RTD) necessary to meet a passenger's RTA, and submit a transit request with that RTD into the optimization. If the itinerary provided to the passenger falls behind their RTA or even significantly ahead (making them wait too long at their destination), the algorithm could redact the transit request and try again with a slightly different RTA. A few cycles of this incremental optimization on a much simpler schedule optimizer

Figure 15: Use Case Diagram



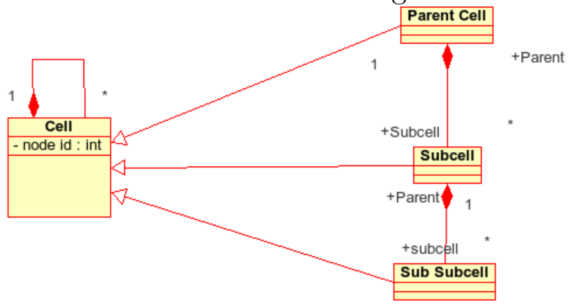
for the subset of passengers who actually need it should provide an acceptable solution more quickly.

14.3 Transit Use Case Diagram

A passenger begins by submitting a transit request for sometime in the future to the global scheduler. The scheduler collects requests and generates an optimized vehicle schedule that separates passengers into several pools based on their current and final destination node. When the time to execute the schedule comes around, a station master at each station loads passengers from each bucket into its pool of available vehicles, and then assigns the vehicles to travel to their next destination node. When the passenger reaches their final destination, they depart the transit system.

For simplicity, all passengers deplane at each station so they can be sorted into their next/final destination pools. Another logistics layer could be implemented to provide the convenience of maximizing the number of passengers that could stay aboard their vehicles during transfers.

Figure 16: Cell Class Diagram



15 Mass Transit System Structure

The model is arranged in a hierarchy allowing the partitioning and relocation of units at different levels of the structure. This allows us to use flexible recursive algorithms to facilitate a lot of searching and reporting tasks, including incremental exports of state snapshots of the system hierarchy in graphML format for viewing in yFiles's yGraph application.

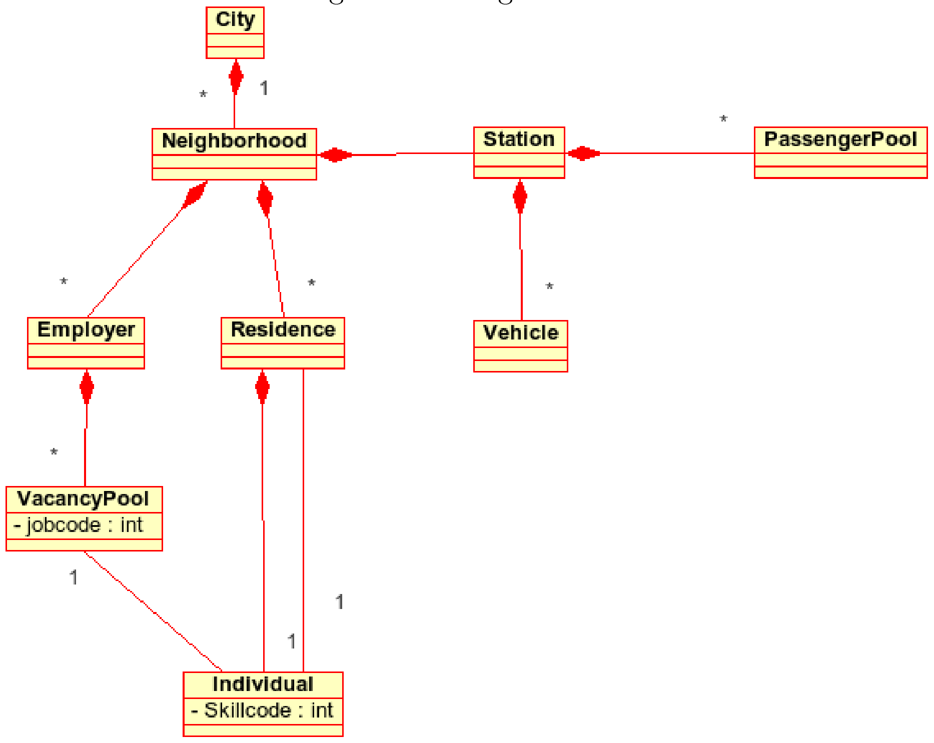
15.1 General Cell Class

All simulation entities inherit from the Cell class, which provides a subcell container for any children classes. The cell class stores a handle to its own parent cell as well, so algorithms may traverse the tree in either direction. Subroutines allow child cells to move about the tree, updating associations so cells never have more than one parent. Each cell also has a className to distinguish between different types of children as well as filtering functions that can search for and return subcells meeting certain criteria.

15.2 Neighborhood Nodes

The rest of the elements in the model are comprised of various incarnations of the general cell class. A master city cell forms the root of the tree hierarchy and contains several neighborhood node cells representing clusters of employers and residences that

Figure 17: Neighborhood Class Structure



share a transit station.

Each neighborhood can contain any number of employers or residences.

An employer would have a number of job vacancies associated with a particular jobcode (indicating the skill required by an employee) and additionally a work schedule that would dictate the employee's commute schedule. Assuming that each vacancy could draw a qualified employee into the metropolitan area, an individual would attach themself to fill that job vacancy, and proceed to look for a residence elsewhere in the city.

Since we're not interested in modeling real estate trends, we simply have the individual create a new residence cell in any neighborhood in the city. Currently we use a simple uniform random distribution to allocate residences, but we could add additional factors to study by using different distributions, *e.g.* perhaps tied to the individual's socioeconomic status relative to their available set of skillcodes.

This skillcode-jobcode accounting allows us to model the distribution of diversity in the urban area relative to zoning policies in relation to their impact on transit demand. The workschedule paradigm allows us to adjust the demand on the network to create or reduce peak congestion.

15.3 Transit Network

The transit network operates within the same cell hierarchy

15.3.1 Stations

Each neighborhood contains one station cell that corresponds to a node in the transit network. All passengers tranferring through a station are sorted into PassengerPool containers, one for each other station node in the network. While every passenger in a PassengerPool has the same final destination, they might take separate vehicles or even entirely different paths to get there.

Additionally stations have a fixed number of vehicle berths that serve to constrain the maximum number of vehicles that can dock simultaneously.

15.3.2 Waypoints

Waypoints are typically one-way nodes in the transit network that allow the system to preserve state of vehicles and passengers in between stations. There are no constraints that prevent passengers from transfering to vehicles at the same waypoint, so to prevent passengers from train-hopping or plane-hopping enroute, we apply and additional constraint that all the passengers and vehicles that enter a waypoint at one timestep must leave it the next timestep. For some models, we might desire this kind of behavior, however, which might allow us to delay vehicles enroute or put them in congestion or holding patterns outside of a station. In the future we may want to ease those constraints somewhat to allow these other types of behaviors.

Waypoints don't really have any meaningful parents, since theirs not much reason to interact with them. They are typically attached to the master city cell since they would typically exist between neighborhoods.

15.3.3 Vehicles

The vehicles in the various transit fleets traverse the network picking up passengers from stations and dropping them off at the next station. Each vehicle type is represented as a completely separate transit layer, each with its own connectivity matrix that details the segments and waypoints that type of vehicle can traverse. Each type of vehicles has only two properties of importance to the schedule optimizer: a maximum passenger capacity and a cost per segment traversed.

15.3.4 TransitTokens

TransitTokens are used to identify passengers and cargo within the transit system, storing information on thier final destination. This is used to sort them at each through station. Additionally, they log the path taken and timestamps for each passenger, so they come in handy for collecting transit times and wait times during postprocessing analysis.

16 Mass Transit System Behavior

The simulation model is based on a discrete event simulation engine. This means that state changes in the system structure are triggered by the firing of events which occur along the global time line queue. The model executes by populating the global time queue with scheduled events and firing those events in order. The system global time advances to the time of the last event, and any state transitions triggered by that event are executed so they can perform their operations, sometimes scheduling additional events in the future event queue. Thus the simulation perpetuates events and continues in time until the program stops or there are no more events left on the simulation queue.

This transit simulation consists of a conglomeration of relatively simple entities working together. We'll introduce them roughly in order of increasing complexity.

16.1 Individual

The simulated people entities exist purely to create demand on the transit system. In the current simple commuting scenario, they simply live in a residence at one node and work at an employer at a possibly different node. They will enter the transit system based on their work schedule. Some configurable time in advance of their travel, they will submit a TransitRequest to the global transit scheduler system. By having advance knowledge of when the passenger needs to travel, the fleet schedule optimizer can ostensibly do a better job reducing passenger waiting time.

They enter the transit system by traveling to their local Station and procuring a TransitToken programmed with their final destination. From there on, they are shuffled around by the other entities of the transit system until they reach their destination station. Once they arrive at their final stop, they are placed into the appropriate employer or residence cell in that neighborhood.

16.2 Vehicle

Vehicles of the same type are basically interchangeable, so the only state information of any importance for them is their capacity and their immediate destination node (either a station or a waypoint). Vehicles simply wait to receive a transitEvent and then they pick up as many people as they can from the station's PassengerPool and all leave for the next destination, which they'll arrive in a predetermined amount of time.

If they arrive at a station, they will dock in an available berth and immediately empty out all of their passengers into the station for sorting into transfers.

16.3 StationMaster

Each station has a StationMaster process that reads the global fleet schedule distributed with each transferEvent and organizes all passengers and vehicles. It first sorts all passengers into PassengerPool queues and all vehicles into rosters each grouped by a common next destination. After a brief period of time allowing passengers to make their connections onto the next vehicle, the firing of the transitEvent signals that all transfers have completed and the vehicles disembark to their next destination.

16.4 GlobalScheduler

The global scheduler receives incoming passenger requests, occasionally triggering the generation of a new optimized schedule. Then it gradually advances the global clock until the time comes to serve the first passengers arriving at the station. Then the global scheduler fires a succession of transferEvents and transitEvents at regular intervals to synchronously push the Vehicles and StationMasters through their state actions.

17 System Requirements Allocation

17.1 Primitive Requirements

People can get to where they are going in a reasonable time

Should not need to use more vehicles than necessary

17.2 Derived Requirements

17.3 Simulation Requirements

17.4 Requirements Traceability

17.5 Specifications

Specs for Simulation

Part V

Analysis of Sample Scenarios

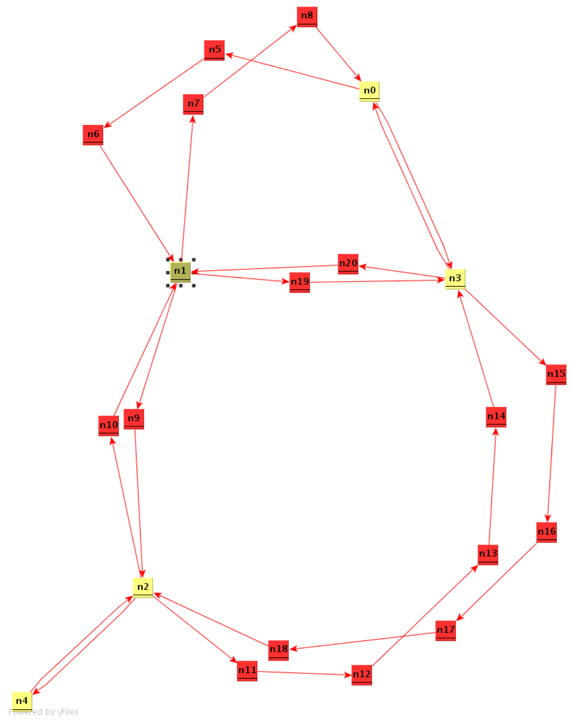
The simulation framework we have gives us the flexibility to model several combinations of loads, vehicle fleet sizes, and network topologies connecting the nodes together.

18 Scenario Descriptions

Verification and Validation was performed on several arbitrary transit networks such as figure #18, which provided several different combinations of connections between stations and waypoints.

The graph demonstrates the functionality of both bidirectional station-station links, and different combinations of unidirectional station links connected via 1 or more waypoints. Additionally it provides multiple equal-cost routes linking several stations to encourage utilization of alternate pathways during congestion.

Figure 18: Arbitrary transit graph used for V&V
Yellow nodes indicate stations, red nodes indicate waypoints.



19 Verification of Simulation Engine

19.1 Simulation Requirements Verification

Checks:

Passengers get sent to their destinations.

Connectivity constraints not violated.

Vehicle capacity constraints not violated.

19.2 Simulation Specification Verification

20 Validation of Analysis Data

Follow paths of individual passengers and vehicles to ensure they make sense.

Check for optimality. Attempt to find improvements to the schedule. It should be hard, even impossible if the solver had found an optimal solution.

21 Sample Scenarios

The program generates histograms plotting the transit system response to an input demand "pulse". The demand pulse is currently a uniform random distribution across all source and destination nodes.

A script produces parametric analysis sets of results for two types of systems: a light-rail system and a PRT type grid.

Blank rows in the chart summaries indicate where the optimzal fleet scheduler was unable to find a feasible solution in under 30 minutes.

21.1 1D Rail Transit Network

The light rail system has two types of operation, strict linear rail (trains stop at every station) and an express rail (where stations are off the main line and trains can save time by bypassing stops). The system is constrained such that a maximum of 4 200-passenger trains can stop at each station at a time.

21.2 2D PRT Transit Network

We can also model a PRT-type system with a bunch of simplifications to allow my fleet scheduler to scale up to a 25-node 2D triangular grid. The main simplification was to make 1-passenger vehicles, which eliminates transfers (they still get counted as station nodes traveresed, but we can assume the passengers just stay in the same vehicle). Otherwise the structure and behavior follows the same rules used by any other simulation based on this framework.

Figure 19: 1D Rail

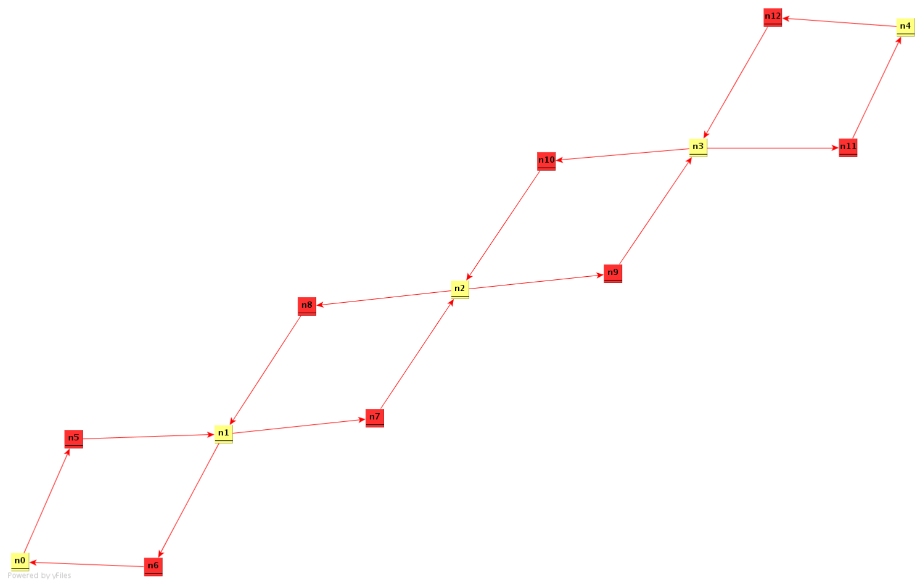


Figure 20: 1D Rail with express service

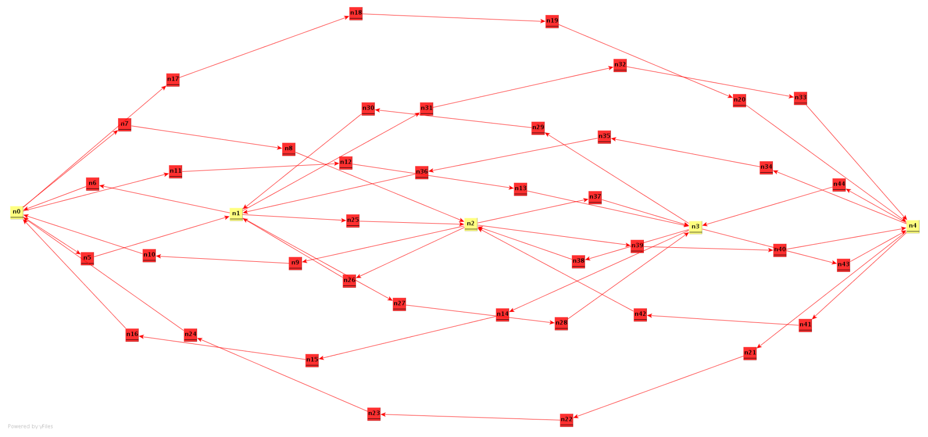
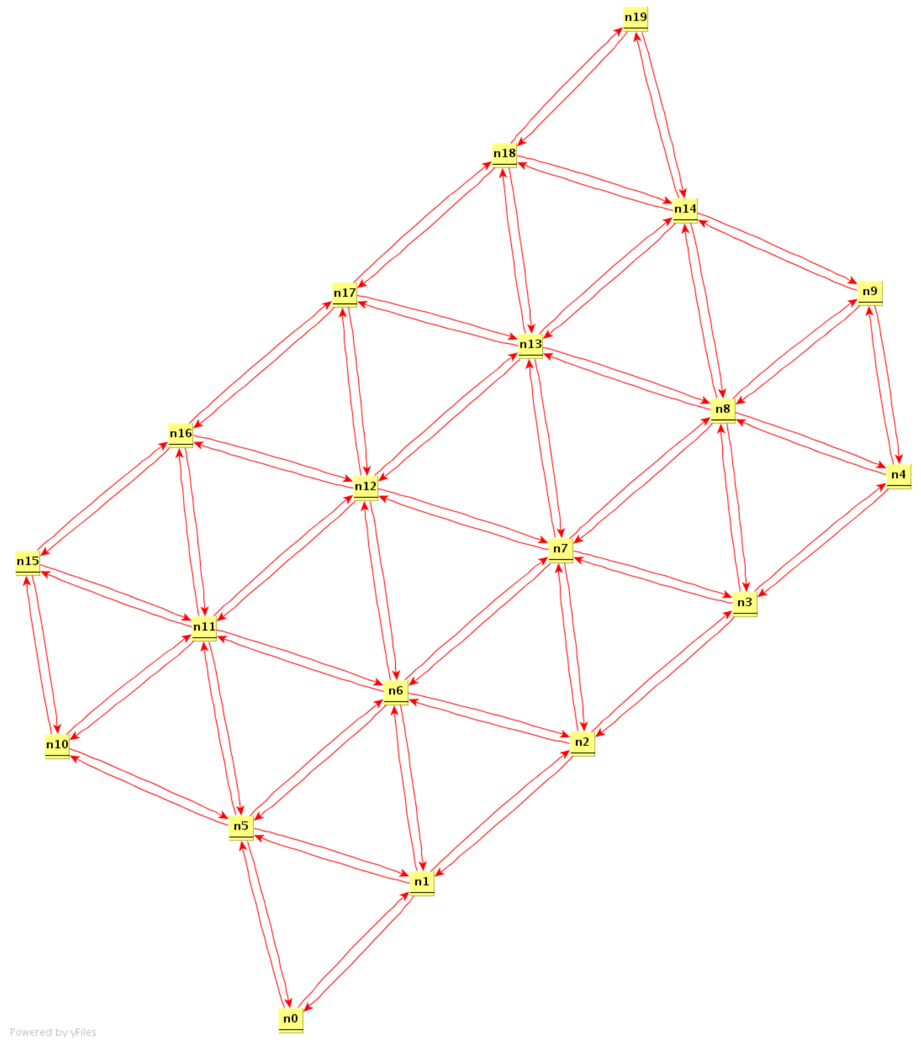


Figure 21: 20-node PRT Triangular mesh network

width ymax = 4 nodes

length xmax = 5 nodes



22 Post Processing

22.1 Performance Metrics Gathered

22.2 Data and Histograms

There are three red histograms that pertain to the transit system performance from the point of view of the passengers (number of transfers taken, their departure time relative to when they requested, and their total transit time). The blue histograms relate to vehicle fleet activity (number of vehicles in motion at a particular timestep, and the passenger load percentage). Each row represents a different network configuration and demand level. These independent variables are summarized in the leftmost column. The second column summarizes how many active vehicles are needed to meet the demand and the total network segments they must traverse.

Part VI

Conclusion & Future Work

The fleet schedule optimizer's work grows exponentially with the number of nodes, so I hit a scalability limit with about 8 stations... beyond that, it takes more than 30 minutes for my 1.87Ghz AMD K7 PC to find any feasible suboptimal solution. Trying CPLEX instead of lp_solve might help here, especially if CPLEX can do some row reduction to eliminate variables.

Todo:

- Simulation: Allow setting vehicle and passenger initial state, to allow continuous evolution of simulated state (currently only allows one schedule optimization run)

- PostProcessing: data reduction for Monte Carlo analyses
- Visualization animation
- ClusterKnoppix liveCD packaging

Future Work

- Hierarchical optimization to increase scalability
- Introduce optimization heuristics to increase scalability
- Constraint linking: allow several waypoints, stations, and other resources to share cumulative constraint terms
- Interactive scenario builder / data editor GUI
- Specification for a continous time model, that can be aliased into a discrete time model with variable length timesteps
- Allow waypoints and perhaps even transit segments to become possible passenger pickup / dropoff nodes (no transfers), representing stops along a bus or rail line.

[Deprecated] Arcology V&V Sample Scenarios

This entire part is deprecated; salvage what we can into the rest of the paper and delete.

23 Verification of Simulation Engine

Verification testing of the arcology simulation engine would consist of running through a series of test cases presented in a test package. Testing will likely occur in phases, each which primarily concentrate first on testing pieces of the core discrete event simulation building blocks defined in the GeneralClasses object model diagram, and then on testing the integrated interactions of scenarios built using those classe, broken down as follows:

1. The ability of the program to set up scenarios using the basic building blocks of units, partitioning them into smaller sub-units, each containing resources.
2. The ability for units to exchange resources based on processing transaction events from the event queue.
3. The ability for units to internally convert resources into other resources based on processing reaction events from the event queue.
4. The placement of the transportation network to apply constraints on the connectivity of the units. The transportation links should also be represented as units, so that they may consume resources themselves in the course of delivering their cargo.
5. The successful execution of a simple example scenario.

23.1 Simulation Requirements Verification

FIXME: Add figures & labels

1.*Insert scenarios as inputs*

Simulation must read input files and populate data structure. Provide printout mechanism to dump state of data structure for verification of the following capabilities:

1.1.*Numbers of units involved (people, transportation mechanisms, industrial entities, etc.)*

Unit test scenarios:

- Create two peer units with transportation connector unit between them
- Create several nested subunits within a superunit. This is illustrated in the Figure 4: CellTypes OMD diagram.

1.2.*Available resources from environment, initial conditions*

Unit test scenarios: (refer to Figure 9: ResourceTest)

- Create two different resources inside a unit
- Create & verify two different resource quantities in a unit and a nested subunit
- Place maximum resource capacity constraints on a unit
- Create an environment unit, which can optionally act as an unlimited source and sink for various resources.

Figure 9: ResourceTest

1.3.*Resource conversion rates, schedules, functions*

Event queue test scenarios:

- Process resource conversion event within a unit to transform a quantity of resource into an equivalent quantity of waste. See Figure 7: CombustionReaction OMD for an example.
- Process a push-based resource conversion event, where a multitude of inputs are converted to a multitude of outputs at the specified conversion ratio until one or more of the input resources are exhausted.

- Process a pull-based resource conversion event, where a multitude of inputs are converted to a multitude of outputs at the specified conversion ratio until either: 1) the requested output quantity is achieved, or 2) one or more of the input quantities is exhausted, resulting in a lower yield than requested.

- Process a unit transaction event, in which a specified quantity of a resource is transferred from one unit to another unit to which there is connectivity through the transportation network. The total quantity of resource in the system must be conserved. See Figure 10: TransactionTest below:

Figure 10: TransactionTest

- Test planning function events, which monitor the internal resource state of a unit and schedule more resource conversion events or transaction events on the queue at regular intervals and based on triggers

2.*Simulation execution - model resource consumption/production rates, providing estimates on actual performance (pending validation of model)*

After individual unit tests & operations have been verified, attempt to load and run a simple but complete simulation scenario. Given inputs of a unit topology, initial resource distribution, and a queue of events, verify that simulation ends in the correct final state. See Figure 11: ScenarioTest

Figure 11: ScenarioTest

3.*Output metrics defined and calculated*

Simulation engine must have a logging and output mechanism. Logging events consist of polling functions that sit on the event queue, passively poll the state of the system when executed, and log the data to a file. The logging functions should not affect the state of the simulation in any way during the course of their execution.

3.1.*Qualitative measures of performance*

As this simulation primarily performs counting, most of the metrics will be quantitative. However, some qualitative metrics gathered could include:

- Error flag. If a function within the simulation throws an exception, the entire simulation run should be marked as tainted. This flag could also be raised when certain preconditions fail, for example, if a unit somehow ends up with a negative quantity of resources. This type of situation could potentially allow the simulation to continue running, without any indication an error had occurred.

3.2.*Quantitative measures of performance*

Most metrics gathered by the logging functions would take snapshots of:

- The state and topology of the unit mesh
- The amount of resources in each unit
- The total amount of resources that has been transferred over a transportation link
- Deficiency reports on amounts of resources that have been "pulled" from a unit, but upstream nodes were not able to provide enough resources.
- Congestion reports on amounts of resources that were unable to be "pushed" to downstream nodes without violating their maximum resource capacity constraints.

3.3.*Allow possibility for formulating optimization problems to aid in benefits analysis & decision-making in arcology design.*

Optimization can take place in the planning function events that schedule and place new events on the simulation event queue. While the algorithm can vary, we can test the ability for the planning function event to affect the scheduling of future events. The planning functions must have the ability to:

- Query the event queue for future scheduled events
- Cancel an event in the event queue, but only if the event occurs in the future. It cannot cancel events that have already been executed.
- Place new events on the event queue at an arbitrary future time.

4.*Significant events (to be defined by modeling use case scenario case studies) should be modelable by scenario architecture - important activities that have impact on performance measures should not be ignored. Should provide at least approximate methods of simulating effects that are difficult to model.*

This requirement affects the ability of the simulation core engine to assemble itself with enough complexity to create a valid model of the system, and is thus in part tied to the validation of the engine. At a minimum, the simple building blocks of units, transportation connectivity, resources, conversion events, and transaction events must be flexible enough to be composed into more sophisticated structures with complex behaviors

For example, let us attempt to model the transportation of a resource from unit A to unit B. In the simplest case, they have a transportation network link between them, and a resource transaction event fires to transfer the specified quantity of resource from A to B. End of story.

In a more complicated scenario, we might want to model different mechanisms to transport those resources from A to B, so we'll use the basic building blocks to model the resources being loaded up, say, into a series of truck units. The truck units would start with a transportation connectivity link to unit A, and a quantity of resources will be loaded up to the truck's capacity. The truck would then disembark, firing a reaction that disconnects its transportation link to A, and schedules a new transportation connectivity link to B at its arrival time. It would also trigger a resource consumption event that would allow it to burn fuel enroute as a function of cargo weight and distance between A and B.

The simulation core building block units would need to demonstrate this sort of composability to pass this verification test.

5.*Specification of accuracy in estimates & predictions. (Goal of ~20%)* This is related to an output metric. The simulation engine would need a self-monitoring performance metric to keep track of the buildup of standard uncertainty errors to estimate a system-wide predictability error. As the simulation proceeds with its projections, the errors will compound to a point where it is not worthwhile for the simulation to proceed much farther.

23.2 Simulation Specification Verification

1. *Ability to model baseline scenario on the order of magnitude of ~1010 units processing a 24 hour period of events on currently available computer hardware.* An example scenario of such proportions needs to be created and loaded into the simulation engine without exceeding the memory or storage limitations of the target computer platform.

2. *Attain a reasonable execution time of less than 10 hours to process the scheduled event queue such that the baseline scenario, assuming ~100 events per unit during the 24 hour period.* The example scenario needs to be created with the appropriate number of events of "average" complexity.

3. Achieve real time or faster simulation speed of the baseline scenario.

24 Validation Evaluation Plans

Validation of the arcology simulation tool occurs when it can be used to build models of arcology systems. These models would represent some function performed in a real life habitat system, and its behavior and results would in turn need to be validated against performance metrics collected from the actual system it represents. Since we cannot directly assess the sim tool's ability to create generic models, the validation evaluations will focus on a range of sample model calibration and validation scenarios. We can measure the performance of the models by comparing them with real world performance. If the models can prove their viability, then the capability of the underlying simulation to build models will have been intrinsically validated.

Since the simulation model essentially deals with discerning large scale effects based on the accounting and execution of small scale actions, we will need to validate models on two levels: macro and micro.

24.1 Macro Level Validation Scenarios

The macro scenarios focus on evaluating the model's predictions of aggregate changes to the system, e.g., what happens to city-wide energy utilization when the population doubles. At first glance, you might expect the energy use to double. However, if the increase is due to the size of the average household increasing, then things like lighting and air conditioning would be shared among individuals, implying that the increase would be less than double. This process turns out to be as much a calibration step as a validation step.

To go about validating this scenario, you would first want to create a model that measures energy use as a function of household size. Then, to double the population, create a new distribution of household sizes (accounting for both the creation of new houses as well as the increases in existing households). Using the earlier function, we should be able to estimate the new total energy usage based upon the distribution of household sizes, and summing up the energy used per household.

Live historical data containing this information is available in the US census. It will be a stretch, since the census is only taken every decade. Energy utilization is also dependent upon a wide variety of other factors, such as local fluctuations in the cost of power, the development & use of technologies such as efficient fluorescent light bulbs, and of course the proliferation of new power-hungry devices such as personal computers and microwaves. Hence the necessity for the calibration portion of the validation. Calibration for this example scenario can occur by using a “control group” of cities whose populations and distributions of household sizes did not change at all. This would allow us to add a systematic increase/decrease factor to each individual’s energy utilization before factoring in the household growth.

After the model is properly calibrated to agree with a group of baseline cities, we can assess its prediction accuracy by giving it energy and census data for one year, having it project the energy usage a decade later based on that census data, and comparing that to the actual. The percentage error over a range of different cities would give us some evaluation of the level of confidence we can place in its projections.

24.2 Micro Level Validation Scenarios

The micro scenarios, on the other hand, focus on how well the model can handle the accounting of changes on the individual level, to determine if different operations make sense. Listed here are some sample scenarios that should be modelable and give predictable, expected results:

- Suppose an individual installs their own power generator (e.g. solar panel) and hooks it into their electrical grid. Their power consumption from the public utility should go down. They may even be able to pump some power back into the public grid.

- Another individual does not have a kitchen, and must drive to the nearest food store to “eat-out” every time they need to contend with a hunger event. Their extra fuel consumption should show up on their usage metric.

- A commuter in the suburbs moves downtown and begins using public transportation. The impact to their finances, commute time, fuel consumption, trips to the grocery store, and remaining leisure time should show up on their monitored metrics.

25 Sample Tradeoff Analysis

This simulation model will essentially be a scoring system in and of itself, as it provides output parameters that can be combined to form a composite score. Analytic Hierarchy Process would probably not be appropriately applied here, since the model produces quantitative results well suited for the scoring process, and is not so much intended to rank cities based on their performance.

The simulation system culminates in the ability to reduce piles of data into simple measures of effectiveness that can be used to compare several city designs or evaluate several potential changes to the operation of one city, and also reflect its impact on both higher levels of organization on down to the low levels of individual life.

Since most of the measures of effectiveness are ratios that range between 0 and 1, there are no concerns about recentering scores or establishing scale factors. Breaking down metrics to be expressed in this type of nondimensional form has been a longtime stalwart of supporting aerospace engineering research, where oftentimes theorists raised on entirely different unit systems could compare performance based on dimensionless parameters.

The only task remaining to form a composite score is choosing the weights to assign to each performance metric. The scoring method works best when all of the weights are normalized.

For this project, we will perform an example preliminary design tradeoff study focusing on the transportation system used to distribute goods to the population at large.

The optimization is performed using a transportation model that optimizes delivery schedules through a series of equidistant nodes. The nodes are specialized in producing a particular resource, which needs to be distributed to the other nodes that don't have that resource. A common distribution paradigm is to have a large hub that collects all of the resources, and then distributes the complete package to the destination nodes. This works well for relatively small nodes, for example, distributing goods from a city (hub) to its surrounding suburbs. However, as the relative sizes of the nodes increase to the point where adjacent nodes are peers, a different approach may be warranted.

The optimization problem used is a mixed integer / linear program that takes several input parameters characterizing the transportation network topology and demand levels, and outputs an optimal schedule of flights. We will model our demand in terms of unique resources generated at a particular node that must be distributed to all of the other nodes in the system proportional to their population. The problem was originally designed for moving passengers through the airliners, thus the passenger/aircraft terminology. However, the same model could apply just as well to moving freight through a network served by a fleet of cargo vehicles.

As a simplification, we will wrap up all of the user requirements declared in the specifications into a unit package. Therefore, every unit that makes it through the transportation network will satisfy part of the requirement for one individual in the system.

25.1 Design / Decision Variables

System Wide Measures of Effectiveness

In general, the complete city system can only improve properly if we choose the right performance metrics to judge it by. An optimization function that optimizes the wrong metric will certainly cut you short of fulfilling your goals. For a city, the metrics we would want to track include:

- Resource production / consumption ratio per cell. An effective system would need to be efficient at doing a lot with the resources it has available to consume. The emphasis should not be merely on stinginess with resources, because that can only cause stagnation.
- Transportation overhead - Establish metrics to track the ratio of resources spent on the connective infrastructure compared to the nodes and activities it actually supports. Of course, this also needs to be balanced with the need for growth and interconnectivity, so it should be considered secondary to productivity.
- Sustainability - The environment is usually the first to give resources or absorb waste when they are not serviceable elsewhere. However, it is often not well known what the capacity of the environment to perform restorative reactions on waste resources to turn them back into useful resources. The burden should be placed on the industries who exercise the environment the most to prove what its capacity is, and to achieve a suitable equilibrium.
- Quality of life - This can be measured by tracking the rate of failed reactions scheduled by the population to maintain their desired standard of living. Of course, this is dependent upon how high that initial standard is set. The only qualification for this model is to attempt to keep the quality from dropping below former levels.

Transportation System Preliminary Design Input Parameters

Define a distribution system topology. All nodes will be fully connected, but have different hub/node size ratio. What characterizes the difference between hubs and ordinary nodes? Hubs will have higher transit demand levels as well as larger throughput limitations.

Number of people per node ratio. For the same total population, is it better to have them distributed across several nodes, or occupy relatively few. This will likely be dependent on throughput limitations.

Population / number of transport vehicle ratio. For the same total population, is it better to have fewer vehicles working complex routes, or many vehicles working in parallel.

Transportation System Measures of Effectiveness for Tradeoff Analysis

The following criteria form the core of the multi-criteria optimization performed for tradeoff studies.

- Maximize profit of running the system, represented by the total freight revenues less the total cost of operating the fleet of vehicles. Revenue is generated by sending a unit of cargo to its destination. A flat cost characterizes the expenses of each segment that the cargo haulers travel. Plus, another flat cost characterizes daily maintenance expenses for having the vehicle in operation, to encourage the schedule to use fewer vehicles if possible.

- Maximize the coverage & service of the system (minimize the number of unserved people). Usually it's unprofitable to transport the last few units of cargo left at a node for the day, since they don't fill up the cargo holds enough to offset the fixed cost of the journey. Minimizing this criteria attempts to serve all of these "leftover" bits at an operating loss if necessary.

- Minimize the deviation from an arbitrary desired fleet size. A somewhat artificial metric, this criteria attempts to find a solution using a specific number of vehicles. Often, the solution would require more vehicles than you may have in operation, and you would want to avoid temporarily leasing if possible, even if it affects the other optimization criteria. On the other hand, the profit optimal solution may require less vehicles than you have in your fleet, but union labor mandates or other reasons may drive you to want to use those vehicles anyway without giving them a throwaway schedule.

Bibliography

References

[1] Arcology.com: Architecture + ecology. http://www.arcology.com/.

[2] Arcosanti: A prototype arcology. http://www.arcosanti.org/.

[3] Biosphere 2 biospherics. http://www.biospheres.com/.

[4] Scalable simulation framework research network. http://www.ssfnet.org/.

[5] Carfree cities. http://www.carfree.com/, 1996-2005.

[6] Homes may be 'taken' for private projects. *The Associated Press*, June 23 2005. http://www.msnbc.msn.com/id/8331097/.

[7] Jarkko Niittymaki; Ari Karppinen; Jaakko Kukkonen; Pekko Ilvessalo; Erkki Bjork. Citysim - validated assessment tool for simulating urban traffic and environmental impacts. In LJ Sucharov, editor, *Urban Transport V: Urban Transport and the Environment for the 21st Century*, pages 394–403. WIT Press, 2000.

[8] Hans Blumenfeld. *The Modern Metropolis: Its Origins, Growth, Characteristics, and Planning.* The M.I.T. Press, 1967.

[9] Takenaka Corporation. Takenaka's engineering: Sky city concept. http://www.takenaka.co.jp/takenaka_e/engi_e/c02/c02_1.html, 2000.

[10] Christine Cosgrove. Roger rabbit unframed: Re-
visiting the gm conspiracy theory. *Institute of Transportation Studies*, 3(1), 2005. http://www.its.berkeley.edu/publications/ITSReviewonline/winter20042005/gm.html.

[11] J.H. Crawford. *Carfree Cities*. International Books, July 2000.

[12] Daniel and David Carasso. Biosphere: Recreating the world. *Aish HaTorah*, September 17 2002. http://www.aish.com/societyWork/sciencenature/Biosphere_Recreating_the_World.asp.

[13] Laura Olsen; Cheryl Cort; Roger Diedrich. Smart growth groups support vienna metro development. Technical report, Coalition for Smarter Growth, October 13 2004. http://www.smartergrowth.net/pressroom/PressReleases/2004.10.13.Viennametro.html.

[14] Discovery Channel. *Millennium Tower*. http://dsc.discovery.com/convergence/eti/projects/towermain.html.

[15] Discovery Channel. *Tokyo Sky City*. http://dsc.discovery.com/convergence/engineering/skycity/interactive/interactive.html.

[16] Richard A. Etlin. The future of tysons corner: A fifteen-point blueprint for the new "downtown" of northern virgiina. University of Maryland School of Architecture, Planning, and Preservation, October 21 2004. http://www.smartgrowth.umd.edu/research/pdf/Tysonspdf.

[17] Foster and Partners. Millennium tower. http://www.fosterandpartners.com/internetsite/html/Project.asp?JobNo=0504, 1989-.

[18] Francis Frick. A seaside arcology for southern china. Master's thesis, University of Hong Kong, 2000. http://www.cityfarmer.org/frick.html.

[19] Jill Jonnes. *Empires of Light : Edison, Tesla, Westinghouse, and the Race to Electrify the World*. Random House, August 19 2003.

[20] Susan Metros. Creativity and leadership: A heart to heart on leadership: How to use your life experiences to become a better leader. *Association of College & Research Libraries News*, 66(6), June 2005. http://www.ala.org/ala/acrl/acrlpubs/crlnews/backissues2005/June05/hrttohrt.htm.

[21] Fairfax County Department of Planning and Zoning. Tysons corner transportation and urban design study. http://www.co.fairfax.va.us/dpz/tysonscorner/, August 2005.

[22] Lisa Rein. Metrowest battle turns partisan. *The Washington Post*, page A10, July 28 2005. http://www.washingtonpost.com/wp-dyn/content/article/2005/07/27/AR2005072702295.html.

[23] George B. Dantzig & Thomas L. Saaty. *Compact City: A Plan for a Liveable Urban Environment.* W.H. Freeman and Company, 1973.

[24] Jeffery A. Schneider. Environmental investigations: Was the original "biosphere 2" project a failure? Technical report, SUNY Oswego, 2003. http://www.oswego.edu/ schneidr/CHE300/envinv/EnvInv01.html.

[25] Cliff Slater. General motors and the demise of streetcars. *Transportation Quarterly*, 51(3):45–66, 1997. http://www.hawaiireporter.com/storyPrint.aspx?7ece2dbe-c0ec-4590-ba1c-3d548bb8a758.

[26] Paulo Soleri. *Arcology: The City in the Image of Man.* Bridgewood Press, 1999.

[27] Eugene Tsui. The "ultima" tower, two-mile high sky city. Technical report, Tsui Design and Research, Inc., 2005. http://www.tdrinc.com/ultima.html.

[28] Jo Twist. Eco-designs on future cities. *BBC News*, July 14 2005. http://news.bbc.co.uk/1/hi/sci/tech/4682011.stm.

Additional Resources