A Scalable Methodology for Modeling Cities as Systems of Systems

by

Amanda M. Wachtel

Bachelor of Science in Mathematics
The University of Alabama, 2010

Submitted to the Program in Computation for Design and Optimization
in partial fulfillment of the requirements to the degree of

Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Signature of Author...
Computation for Design and Optimization
June 7, 2013

Certified by................ ....
Olivier L. de Weck
Associate Professor of Aeronautics and Astronautics
Associate Professor of Engineering Systems
Thesis Supervisor

Accepted by...
Nicolas Hadjiconstantinou
Professor of Mechanical Engineering
Co-Director, Program in Computation for Design and Optimization

A Scalable Methodology for Modeling Cities as Systems of Systems

By

Amanda M. Wachtel

Submitted to the Program of Computation for Design and Optimization
On May 10, 2013 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Computation for Design and Optimization

Abstract

As cities evolve in size and complexity, their component systems become more interconnected. Comprehensive modeling and simulation is needed to capture interactions and correctly assess the impact of changes. This thesis presents a methodology for modeling cities from a systems of systems perspective. The framework supplies general modeling guidelines and key steps. Also addressed are the importance of stakeholder interactions, creating the model structure, using smart city sensor data, and applying the methodology to larger, traditional cities.

As an initial step, four city modeling including CityNet, CityOne, Sim City 4, and SoSAT software programs were evaluated from both a user and mathematical perspective. From the assessments, a list was developed of features critical to successful city modeling software including visualization, a streamlined user interface, accurate mathematics, the ability to specify systems and attributes, and the ability to model interconnections between systems.

SoSAT was selected as the modeling tool for the case study, which involved modeling the Army's Base Camp Integration Laboratory. A model of the camp's baseline configuration was built and the camp was simulated for 30 days with results recorded at one hour intervals. 100 trials were run with averaged results presented by time intervals and for the total simulation time. Results were presented at all levels of structural aggregation.

Two sensitivity analyses were conducted to analyze the impact of maintenance personnel and the frequency of potable water deliveries. Adding or subtracting a maintenance person impacted the availability of the generator systems that were being serviced, in turn impacting the performance of the micro grid. Extending the time between deliveries by 24 and 48 hours revealed two systems experienced resource depletions.

Lastly, two technology insertions cases were conducted to assess the impact of adding a laundry water reuse system (LWRS) and a solar powered hot water heater (SHWH). The LWRS provided 70% of the laundry system's water needs, significantly reducing dependency upon deliveries. The SHWH was expected to decrease electricity consumption and increase fuel consumption. However, the reduction in energy demand meant fewer generators were needed to power the micro grid and both electricity and fuel consumption decreased.

Thesis Supervisor: Olivier L. de Weck
Title: Associate Professor of Aeronautics and Astronautics and Engineering Systems

(This Page Intentionally Blank)

Acknowledgements

It's said that it takes a village to raise a child and I now know that it takes a similarly large support network to write a thesis. So many groups and individuals have contributed to the completion of this project, and I am deeply thankful to each and every one. First and foremost I'd like to thank my advisor Professor Olivier de Weck for his guidance and mentoring during my two years at MIT. His compassion, direction, and extensive knowledge had an enormous impact on my academic endeavors and my research.

I am extremely grateful to Sandia National Laboratories for giving me the opportunity to complete my degree through the Critical Skills Masters Program fellowship. A special thanks to my manager Bruce Thompson, who introduced me to the program and also to Camille Valdez in student programs who works tirelessly to keep things running smoothly.

To my department at Sandia, thank you for being the best group of people to work with, and for giving me something to look forward to when I return. I'd especially like to thank Craig Lawton for his help and guidance with the case study, Rob Vander Meer for sharing his expertise on systems of systems modeling, and all those who helped me with questions throughout the process. I'd also like to thank research librarian Laura Calderone for her help with my literature review. The case study for this project involved working with the Army to model the Base Camp Integration Laboratory in Fort Devens, MA. I would like to thank John Viggato for being my contact for the project and for his constant help and input throughout the modeling process.

Above all I must thank my parents. Their constant support, love, and encouragement have carried me through long nights, and even longer projects. Their own work ethic has always served as an example of what can be accomplished with hard work and dedication. They have always offered unconditional support, yet they have done so in a way that allowed me to become independent and successful. They truly are the foundation behind all of my achievements and I am forever grateful for all they have done for me.

Lastly, I'd like to thank many of the students I've met while at MIT. To my core group of CDO friends, who I spent countless hours with at the big table in the office. Whether it was coding until we started typing "end;" at the conclusion of emails, sharing frustrations and funny stories, or learning about each other's cultures, these are the moments that I will always remember. To the students in SERG who were always willing to lend an ear, give feedback, and generally help in any way possible. And finally, to all my other friends who have filled my time at MIT with enjoyable memories and meaningful conversations. I'll remember those fancy dinners and fun outings fondly long after my time in Cambridge has ended.

(This Page Intentionally Blank)

Contents

List of Figures

List of Tables

1 Introduction

In the past, many aspects of a city consisted of simple, stand-alone systems. These systems neither interacted with each other nor relied upon each other to perform their desired functions. As new systems were added, planners were not required to assess how their addition would impact, interfere, or enhance the performance of other systems within the city. In the developed world, and much of the developing world, this is no longer true.

Take for example the process of providing an individual with water for daily use. Potable water was once drawn from wells, which aside from the water table, didn't interact or connect to any other systems. In modern urban areas, we do not go outside to our well when we need a glass of water. Many living within congested cities wouldn't have a backyard in which to have a well. Our water comes from the faucet in our kitchen. Yet the availability of water depends on a complicated infrastructure of pipes traversing the city, which in turn are fed by a water processing plant. The plant itself requires energy to run, creating a nontrivial dependency on the electric grid. Electricity for the grid can be generated from multiple sources, including power plants, solar farms, wind turbines, etc. Overall, the task of getting a glass of water has evolved from a straightforward process utilizing one system, into a complex problem involving multiple interconnected systems with complex interdependencies.

While this is only one example, this tendency towards interconnected systems within cities is increasingly commonplace in today's cities. When systems which are themselves capable of performing a function are integrated into a grouping of systems, we consider the resulting structure to be a system of systems or SoS. For a more precise definition, we consider an SoS to be comprised of components that are also systems and which operate and exist independently of the SoS structure [1]. Even though these systems can and do function on their own, it is important to recognize that the amalgamation of the systems often provides additional functions beyond the scope of those provided by the composite systems themselves. Cities are comprised of many such structures, making them best described as systems of systems. This evolving complexity suggests that traditional city planning methods and methods must be advanced to deal with these highly complex cities.

2 Background

2.1 The Transition of Cities to SoS

As mentioned in the introduction, the systems within cities are becoming more complex and interconnected. Traditional methods of city planning include the use of technical, verbal, and pictorial analysis [2] with minimal use of SoS modeling and simulation or understanding of how systems are interconnected. The lack of SoS models in city planning is mainly due to the fact that the application of such models to city structures is still relatively new, and as of yet, there are few examples of modeling cities in ways that capture the interconnections between city systems from a holistic perspective [2]. Yet as city systems become more intertwined and more complex in their own right, it grows increasingly difficult to predict the dynamics of the city without the use of modeling tools. Policy makers, planners, communities, and individuals are becoming more concerned with the technicalities of their cities and the systems within them, including such concerns as reliability, robustness, resilience, and perhaps most importantly, liveability, which focuses on wellbeing within the community [3]. If we are to maintain and/or create cities that meet our requirements for these attributes, we must shift our perception of city planning and maintenance.

These advanced, complex, interconnected city systems are enabling the evolution of cities by providing key services to citizens such as transportation, energy, and healthcare [4], yet these technologies are much easier to create and deploy then they are to understand, especially as they interact with other complex systems [5]. One of the main problems with failing to understand how systems within cities are related to and interact with each other is that these oversights can lead to catastrophic and cascading failures across multiple critical systems. When systems are increasingly integrated, and when the SoS is under duress, a few things can happen—minor systems can become critical to maintaining operability [6], the performance of the overall structure can be impacted by the decisions or performance of one of the constituent systems [7], and in the worst case scenario, the failure of infrastructure can result in death and costly disruptions [4]. We can no longer afford to ignore the impact that one system might have on another.

Though the behavior of the systems within the SoS is becoming nearly impossible to understand and predict without the use of modeling and simulation, we must also recognize that most SoS involve both social and technical aspects [7]; certainly this is the case for cities. Accounting for a socio-technical point of view means acknowledging that complex systems and SoS within cities are designed, installed, used, and managed by a diverse group of stakeholders [1], [4]. These stakeholders strive to have their interests represented, yet their interests are not always aligned with those of other stakeholders. A key requirement for any models and simulation used for city SoS will be their ability to incorporate these multiple stakeholder views and examine scenarios from multiple angles. Without this flexibility, city models will not gain widespread acceptance, and the adoption of their use will be limited and vehemently debated.

2.2 Current Types of Models Used for City SoS

Recognizing the need for reliable modeling and simulation of cities to capture ever-increasing complexity, researchers have begun creating models using a wide array of modeling techniques. Modeling methods including (but not limited to) cellular automata [8], agent based modeling [8], [9], neural networks, fractals [10], system dynamics [11], computable games [7], and object oriented modeling have been applied to model cities from an SoS perspective [12]. Perhaps the most widely used, agent based modeling represents entities within the city as "agents." Agents are not limited to being defined as people, but can effectively represent buildings, moving vehicles, power plants, etc. Each agent is assigned attributes, but over time, interacts with other agents in the model and "learns" in an adaptive way from previous behavior. This modeling strategy can mimic complex interactions among diverse agents, examine interactions over time for emergent behavior, and is flexible enough to allow for various time scales and long-distance interactions [9], [12]. Each of the other modeling methods listed has its own set of strengths with regard to modeling city SoS. Additionally, these methods are sometimes combined to achieve even greater benefits.

While each modeling method possesses key strengths for modeling cities, they also come with drawbacks. For example, while agent based models are useful for what-if scenario analysis [8], the evolving nature of the agents coupled with the complex and detailed nature of the models often make these models untestable, imparting a high degree of arbitrariness [9]. These models are also extremely sensitive to initial conditions and the behavior of the agents must be well understood if it is to realistically represent complex systems and human interactions [12]. Additionally, agent based modeling cannot easily be combined with Geographic Information Systems (GIS) which are a valuable source of geospatial data, but are not currently suited for integration with dynamic modeling [9]. Lastly, the distinct strengths of the different methods of modeling do not always overlap, meaning that choosing an appropriate modeling method then becomes highly dependent upon the issue being addressed.

2.3 Emerging Concerns

A parallel topic to understanding cities from an SoS perspective, is the creation of smart cities, which combine both physical and digital assets [4]. In the future, sensors attached to systems within cities will help to monitor interactions, guide responses, and prevent failures [6]. Sensors present a new set of problems, since they are constantly generating data, which must be processed at an acceptable level of aggregation, stored securely, and disseminated in an effective and timely manner to be relevant to decision makers. Specifically this will include the creation of databases, tracking the sensors (possibly through GPS), and designing a way (perhaps wireless technology) for the data to be accessed [13]. Engineers and planners also face the nontrivial challenge of figuring out how to integrate technology into legacy systems, which are aging and no longer perform at the level required [14]. When these smart city sensors and technologies become widespread, they will undoubtedly provide valuable data for interpreting the functionality of component systems, but remember that we must still first gain a fundamental understanding of how these systems are interconnected, leading us back to the original SoS framework. Once we can understand and interpret the SoS relationship between the complex systems within cities, we will be able to capitalize upon smart city data to calibrate models and validate simulation results.

As previously stated, legacy cities present a challenge when integrating new digital technologies, yet they are a source of valuable historic data for modelers. More problematic are new areas or cities, where modeling and simulation is key, but no data currently exists. With the absence of reliable data, any models produced cannot be validated. Even when data exists, the quality of data is often inadequate to capture the complexity of the models being used [9], [12]. Such models can be used to examine what-if scenarios or to maintain spatio-temporal data , but should not be used for predictive purposes [12]. Model validation will continue to be a key challenge for

SoS city modeling, and one that must be fully addressed for modeling and simulation to be accepted by planners and key decision makers.

Critical to any modeling effort is the choice of model boundary and level of detail captured in the model. With a SoS city structure, the number of systems within the city, and their corresponding attributes becomes prohibitively large. Modeling everything would be impossible, and at least equally as confusing/non-intuitive as the original SoS [15]. A useful model is one which simplifies the original complex system or SoS into something understandable [15]. Modelers must capture the critical details of the city SoS structure while eliminating those details which fail to impact the analysis. As a general rule of thumb, the structure—in this case our city SoS—should first be modeled at a level detailed enough to ensure key interactions are captured to aid in exposing emergent behavior in the model. Component systems and interconnections that are modeled with high fidelity can be abstracted to the extent that their abstraction does not significantly alter the output of the model. Above all, the model must be broad enough to allow for varied analysis. Early city modeling efforts focused only on a small area of a city such as key subway stations [9], or on a single layer such as transportation [7] or emergency response [6], yielding very specific results capable of answering only limited questions. Additionally in these examples, crucial systems exist beyond the narrow model boundaries and still interact with the examined system/systems in ways not captured by these models. Future city modeling efforts need to focus on how to model city systems and interconnections between systems within the framework of an SoS model while creating models robust enough for varied types of analyses, yet straightforward enough to remain useful.

A final key challenge lies in providing visualization for these SoS city models. For those unfamiliar with programming and mathematical models, understanding outputs is difficult without some type of realistic visual representation to reference. Ideally, the user interface would include a view of the city and its systems similar to the graphics seen in video and computer games such as SimCity. The challenge is that most mathematical city models have only simple realization, while games have realistic graphics but little underlying mathematical accuracy or data content [16]. Software exists to build accurate visual representations of physical infrastructure such as buildings, but usually does not include tools to add attributes such as water consumption, population details, energy use, etc. Visualization within city models needs to capture the physical attributes of the systems, but also should contain their functionality [17]. Similar to the modeling effort, visualization of a city SoS will depend critically on available data, the proper management of real-time data, and creating graphics at a level of detail appropriate to analysis [17].

3 Project Overview

Researchers argue for the need to understand interconnected city systems in their totality, yet current models are usually built to examine a specific area of interest, and are not robust or adaptable enough for varied analysis. Even where broader boundary models have been created, inadequate data means the model output cannot be quantitatively validated and should not be used for predictive purposes. Research needs to focus on how to model city systems and interconnections between systems within the framework of an SoS model, creating models robust enough for varied types of analyses, the level of fidelity needed to create these robust models, how to aggregate historical and sensor driven data into usable data sets, and how to correctly assess the impact of new technologies being implemented within cities.

A large part of creating robust, comprehensive, yet simplistic models is understanding the current state of city modeling software. By comparing and contrasting the various software packages available, the required features to successfully model cities as an SoS structure become clearer. A few different types of available modeling software for cities were evaluated as part of this thesis. Each offers unique functionality, appeals to different types of users, and operates on different platforms. An analysis of each will be presented, as well as suggestions for what needs to be included in an ideal city modeling software. While none of these software packages includes every desirable feature, the models for this thesis will be built and analyzed using the System of Systems Analysis Toolset (SoSAT) developed by Sandia National Laboratories.

Instead of choosing a subset of a city or a single layer to model, this thesis will examine military forward operating bases as miniature, temporary cities in order to create a scalable methodology for modeling cities as systems of systems. These isolated bases, with well-defined external connections, allow us to expand the boundary to capture the entire city and all interconnected systems, while keeping the scope small enough to produce a model which can be verified and validated. As a case study, the Base Camp Integration Laboratory (BCIL) at Ft. Devens will be modeled using SoSAT. BCIL consists of two identical, functioning base camps located side-by-side. One is used as the baseline while the other is used to assess the effects of technology insertion, changes in user behavior, etc. BCIL offers a unique modeling opportunity, since data is routinely collected from sensors placed on the integral systems, against which the virtual model can be calibrated and validated. Modeling methodology, results, and analyses will be presented for the SoSAT baseline model and multiple technology insertion cases that evaluate the impact of new technologies on the base camp. Lastly, how knowledge gained from the modeling effort of temporary cities can be applied to the modeling of permanent cities will be explored. Lessons learned will pave the way for future work in SoS city modeling, particularly with regard to how to successfully capture complexity within a broad boundary while maintaining a manageable and adaptable model.

4 Choosing and Evaluating Existing City Modeling Software

Aside from the different modeling methods discussed in the background section, there are also software packages which already exist and are available for purchase, online, or for academic research. As with the other models previously discussed, the scope is often limited or the software is not concerned with an accurate representation of reality. Some programs focus on precise 3-D renderings of buildings, but not any other aspects of the city. Conversely, other software packages model systems such as water and transportation, but at the cost of generalized infrastructure. Though the need is increasing for an interconnected approach to modeling systems within a city, few programs take this systems of systems viewpoint. We have already established that to accurately capture the intricacies of current and future cities, modeling software must be able to realistically model multiple layers, such as infrastructure, transportation, water, waste, and energy, and also the interactions between these layers.

Though not all available city modeling software captures the SoS structure we require, looking at their strengths and weaknesses is still valuable to understanding what will be required of an optimal city SoS modeling software. To assess the current city modeling software market, a two-step analysis of four programs was conducted. These included IBM's online CityOne game, MIT's Strategic Engineering Research Group's CityNet modeling tool, EA Game's popular game SimCity 4 Deluxe Edition, and Sandia National Laboratories' System of Systems Analysis Toolset (SoSAT). The first step of the analysis was to evaluate each program from a user perspective to assess ease of use and general appeal. The second step was to evaluate the programs from a mathematical perspective to determine the accuracy of models. The perceived strengths and weaknesses, as well as suggestions for future software package features are also included.

4.1 Evaluating Software from a User Perspective

The three programs were initially evaluated from a user perspective. The ability of a program to be clear and accessible to multiple types of individuals will be of key importance to city modeling software. An engineer using the program may have the skills to sift through the code and make changes as necessary, but someone such as a city planner may not have those programming skills. The user interface needs to encompass all desirable aspects of city modeling, yet it also needs to remain simplistic enough to be user-friendly. The program must also be easy to access, whether being downloaded onto a work computer or accessed through the internet.

4.1.1 CityNet

CityNet is a downloadable program that runs on both Macs and PCs. The user interacts with a Java interface where everything is displayed graphically. A .jpg image of an aerial view of the desired area as a "backdrop" for the city is selected by the user. A grid is then generated and displayed over the background image. The user specifies the size of each cell in the grid, and all cells must be the same size unless multiple grids are drawn. An example of this interface for downtown Boston can be seen in Figure 1. From there, the user can use five different windows to specify information on buildings, energy, transportation, waste, and water. These windows are displayed as tabs in the user interface.

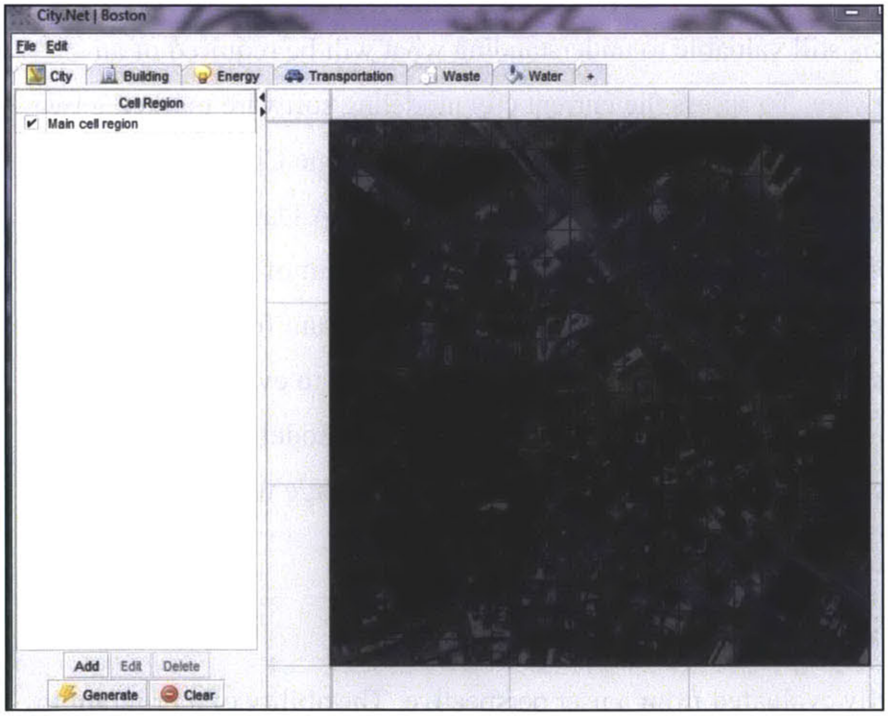


Figure 1. CityNet User Interface

Each window in the Java interface consists of four different items: layers, node types, edge types, and regions. The user specifies multiple layers for something such as transportation in order to create private transportation routes and public transportation routes. Nodes are used for static structures such as buildings, while edges are used for items that can accommodate a flow such as roads and subway lines. Regions can either be collections of edges (ex. a highway), or collections of nodes (ex. a residential area). When a region is a collection of nodes, it appears visually as a polygon. The user draws these polygons directly on top of the image. An example of the building window with defined layers, nodes, and regions for downtown Boston is given in Figure 2.

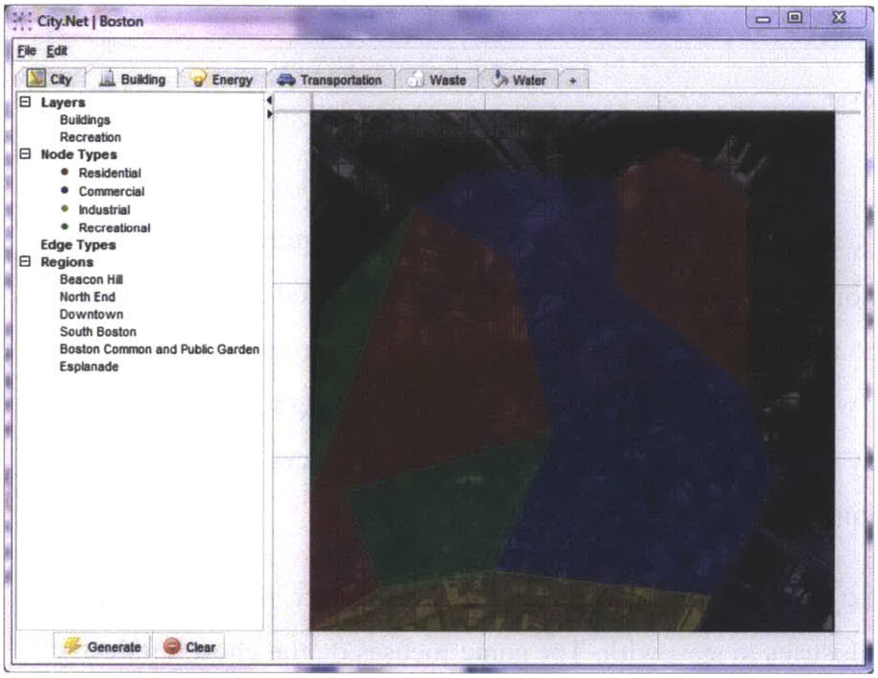
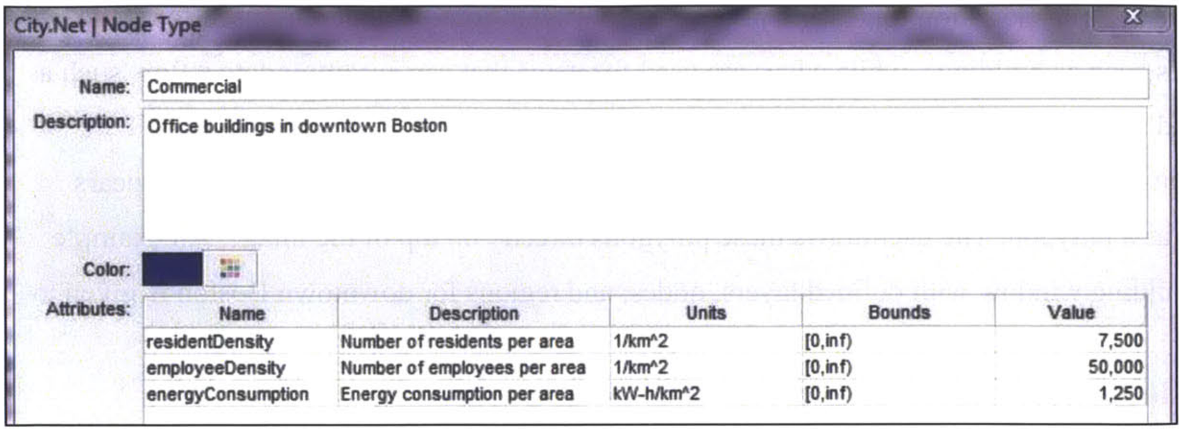


Figure 2. Layers, Nodes, & Regions in CityNet

Attributes can be added to nodes and edges. These can include things such as resident density and energy consumption for residential areas, passenger capacity and average speed for public transportation routes, etc. The user adds these attributes to the nodes and edges and must specify the name, description, units, bounds, and value. These attributes are later exported into an Excel spreadsheet and are then used by Matlab for calculations. The attribute input grid is seen in Figure 3 for commercial buildings in downtown Boston.



Name: Commercial

Description: Office buildings in downtown Boston

Color: ■

Attributes:

Name	Description	Units	Bounds	Value
residentDensity	Number of residents per area	1/km^2	[0,inf)	7,500
employeeDensity	Number of employees per area	1/km^2	[0,inf)	50,000
energyConsumption	Energy consumption per area	kW-h/km^2	[0,inf)	1,250

Figure 3. Adding Attributes in CityNet

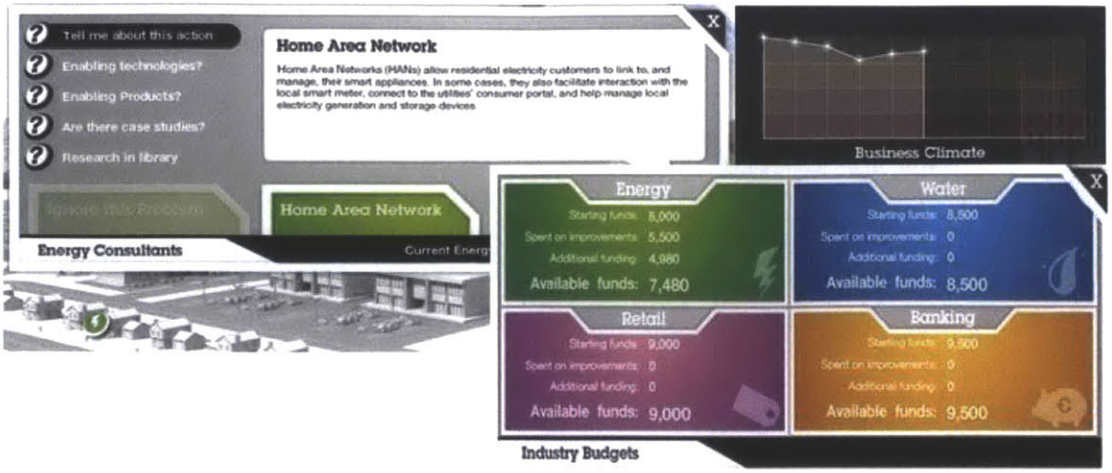
4.1.2 CityOne

CityOne can also run on both PCs and Macs, but is an online game that requires the user to complete a free online registration form. The program cannot be downloaded onto a personal computer and may not run without disabling the firewall (if in use). This program is considered one of IBM's "Serious Games" which according to IBM's website are designed to "prepare professionals to work smarter by enabling them to visualize the consequences of their actions and explore different permutations of events" [18].

CityOne revolves around making decisions in four industries: energy, water, retail, and banking. The user selects one of these industries to start with. The game focuses on the chosen industry but occasionally includes the other industries. The game consists of 20 turns. During each turn, problems arise in the city and are indicated on a picture of the city with icons corresponding to the industry. The user clicks the icons to read the problem and see a list of recommended actions or a list of all actions. Each action has an associated cost that is deducted from the budget for that industry. Figure 4 shows the user interface of CityOne, along with problem icons and a set of recommended actions.

Figure 4. CityOne User Interface

The game contains extensive information to help the player decide which course of action is appropriate. For each recommended action, there is a "discuss" feature that gives five additional resources. "Tell me about this action" highlights what effect the action will have and may include a short video. "Enabling technologies" indicates which IBM technologies are needed for the action, and "Enabling products" lists which IBM products are needed. Meanwhile, "Are there case studies?" provides links to any relevant case studies. Lastly, "Research in library" links to IBM white papers and other IBM resources for more information. In the general interface there are some additional resources. Charts track citizen happiness, business climate, and population after every decision. The user can also view the budget summary for all four industries. Each industry provides a consultant, who makes recommendations for actions, and gives general information about the industry, city, and relays their personal expertise. These user resources can be seen in Figure 5.



Home Area Network

Home Area Networks (HANs) allow residential electricity customers to link to, and manage, their smart appliances. In some cases, they also facilitate interaction with the local smart meter, connect to the utilities' consumer portal, and help manage local electricity generation and storage devices

Energy
Starting funds: 8,000
Spent on improvements: 5,500
Additional funding: 4,980
Available funds: 7,480

Water
Starting funds: 8,500
Spent on improvements: 0
Additional funding: 0
Available funds: 8,500

Retail
Starting funds: 9,000
Spent on improvements: 0
Additional funding: 0
Available funds: 9,000

Banking
Starting funds: 9,500
Spent on improvements: 0
Additional funding: 0
Available funds: 9,500

Figure 5. CityOne User Resources

Bonuses are awarded for each action based on its success. The player can also visually see how they are performing. The picture of the city starts off in black and white but slowly gains color in select areas after a problem is solved in that area. Based on their actions as the game progresses, the player can also earn trophies such as commerce star, energy star, water star, money star, community organizer, business tycoon, water purifier, magnet city, and fresh air. Each turn has a set number of problems to solve, though the player does not have to resolve all of them to move onto the next turn.

4.1.3 SimCity4 Deluxe Edition

SimCity4 is a video game that runs on the PC platform. Unlike the other two programs, there is no free version of this game and it must be purchased and downloaded. A Mac compatible version is also available, though it cannot be downloaded and costs significantly more than the PC version. One of many games within the SimCity suite, SimCity4 boasts 3-D modeling capabilities and extensive user inputs. Additionally, modeling in SimCity 4 is not limited to a single city. Multiple cities can be created and joined by roadways, waterways, or airports. Residents can then move between the cities, and the cities can sell and buy goods from each other.

There are three main modes of play within SimCity 4: God mode, Mayor mode, and My Sim Mode. Each mode has vastly different characteristics and capabilities. Generally, the player spends most of the time in Mayor mode. The game viewed in Mayor mode can be seen in Figure

6. The game's progression is based on a 24 hour clock, which continues at the same pace regardless of whether the player has paused the game or is playing at turtle, rhino, or cheetah speed (slow, medium, fast). The user also has the option of having it always be day, always be night, or for day and night to cycle normally.

ABCs Will Put Sims on Solid Ground

Figure 6. Mayor Mode in SimCity 4

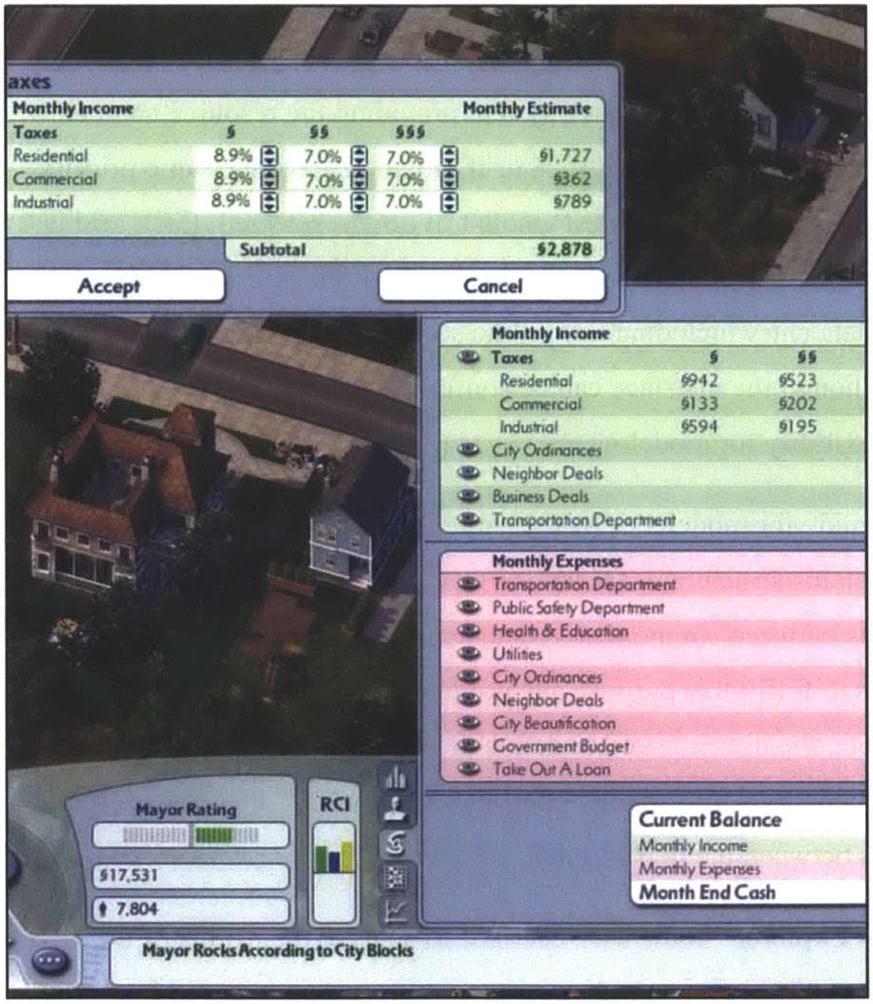
God mode allows the user to create geographical features, cause natural disasters, generate flora and fauna, and specify the day/night cycle. The full functionality of this mode is only available before Mayor mode is entered for the first time. After Mayor mode is entered, then only natural disasters and day/night settings are still accessible. The built in geographical features are extensive and allow the user to create things like mountains, hills, lakes, valleys, etc. An example of user-created terrain can be seen in Figure 7. The user selects the feature from a menu, and then clicks and holds down the mouse button until the feature has reached its desired magnitude. Creative users have also figured out a way to import terrain into a secondary program and then into SimCity, though it is unclear if any of these secondary programs are still functional.

Figure 7. User-Created Terrain in SimCity 4 [19]

Mayor mode is where all of the city modeling takes place. A grid divides up the area of land, and the user uses the cells of the grids to specify model components. The cells are first zoned as residential, commercial, or industrial. The user can then add infrastructure such as roadways, public transportation, and water pipes; as well as add buildings and plants such as police stations, schools, medical facilities, water pumps, and energy plants. Note that the user does not actually build the houses or specify the population. Once the required infrastructure (such as power) is in place, the houses build themselves as Sims begin to move into the city. New infrastructure options become available as the game progresses. All addable items have an associated cost and a set size that they occupy. For buildings that provide services, the user can set the funding level to stay within the budget and prevent excess services from being produced. The user also specifies the tax rate.

My Sim Mode is a special mode that allows the user to interview Sims in the city and find out their opinion of the city and the mayor. The user can also drive, fly, or boat around the city in this mode as part of a "U-Drive-It" mission either for fun, or to earn money or acclaim. Additionally, this mode also allows the user to import Sims that they have created in other Sims games into the city and follow them in their day-to-day lives.

Similar to CityOne, SimCity 4 provides resources to the user. A running list of articles at the bottom of the screen reflects what is happening in the city. Articles appearing in red are about things that require an action from the mayor to fix/improve. These articles also give tips on what to do. Additionally, the user has access to extensive amounts of data and statistics, such as air pollution levels, real time water and energy supply to the city, current population, education levels, etc. There is also a question mark tool that can be used to click on any building or feature within the city to obtain more information. One of the mayor's key concerns is maintaining a balanced budget for the city. Information on the current budget is also readily available within the interface. Sample available budget information can be seen in Figure 8 below.



axes

Monthly Income				Monthly Estimate
Taxes	§	§§	§§§	
Residential	8.9%	7.0%	7.0%	§1,727
Commercial	8.9%	7.0%	7.0%	§362
Industrial	8.9%	7.0%	7.0%	§789
Subtotal				§2,878

Monthly Income		
Taxes	§	§§
Residential	§942	§523
Commercial	§133	§202
Industrial	§594	§195
City Ordinances		
Neighbor Deals		
Business Deals		
Transportation Department		

Monthly Expenses
Transportation Department
Public Safety Department
Health & Education
Utilities
City Ordinances
Neighbor Deals
City Beautification
Government Budget
Take Out A Loan

Current Balance
Monthly Income
Monthly Expenses
Month End Cash

Mayor Rocks According to City Blocks

Figure 8. Budget Information in SimCity 4

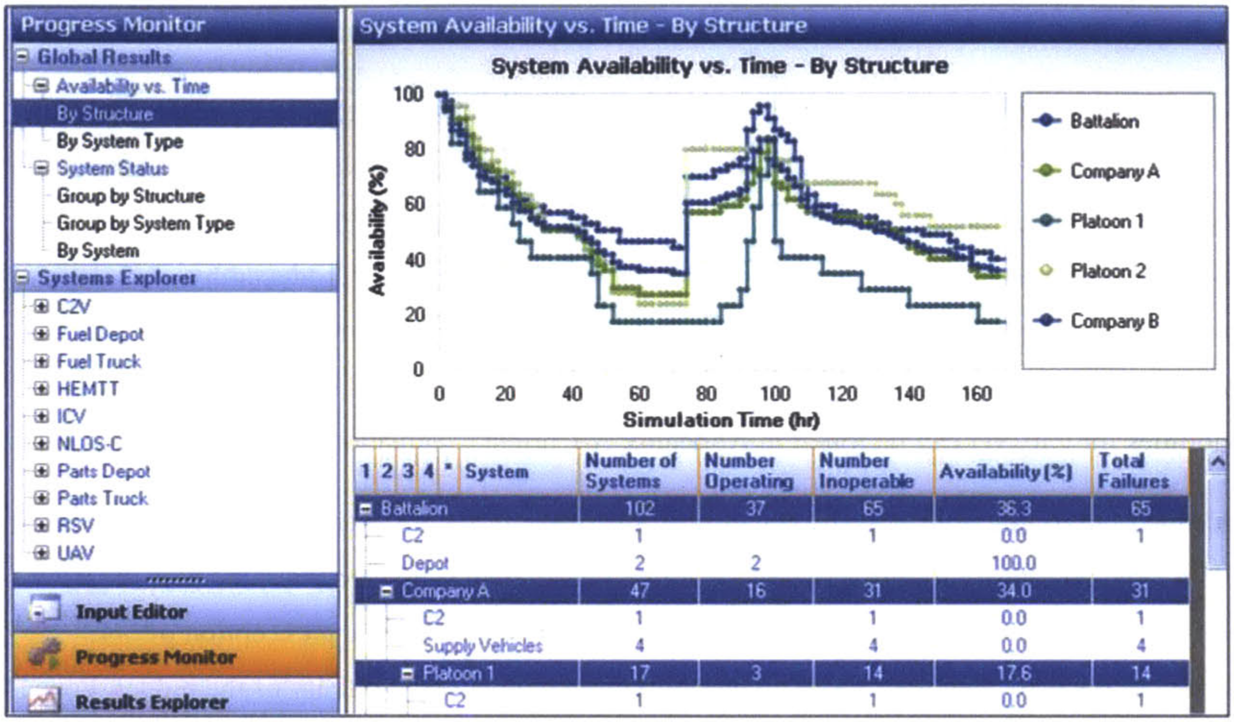
4.1.4 SoSAT

SoSAT is a PC based modeling package owned by Sandia National Laboratories. The software is available only to employees and some customers. Use of the software requires extensive training to learn how to use the user interface, making it a non-intuitive program for those outside of the industry.

SoSAT consists of a series of grids and tables for data entry. Systems are listed in one grid by name and the quantity of each is defined. These systems are organized into a larger structure by the user. This structure is completely user-defined and is extremely helpful in visualizing which parts of the system of systems are grouped together. The consumables for each system and any resources they may supply are also defined in additional grids. For each supply connection between individual systems or groups of systems, each system providing the resource can be assigned a priority. Higher priority systems are used first, or if providers have the same priority, the system that can provide the resource in the shortest amount of time is selected. Each grid is clearly labeled and accessible from a side menu on the left hand side of the interface. Many other grids exist for additional data entry including things like scenario definitions, system functions, and failure modes. Though the interface can be confusing to learn, its flexibility and organization offer maximum flexibility for modeling.

The various grids also act as a guide for modelers. As the name implies, SoSAT is specifically designed to model system of systems structures. To capture this type of structure, all interconnections and interactions between component systems must be fully captured. The user interface dedicates various grids to fulfill this purpose and provides brief tips as to what should be included in each. For example, the user is asked to define the supply connections between systems. If one system supplies energy while another consumes energy from that source, then there exists a supply connection between the two systems.

SoSAT also provides a "Results Explorer" in the user interface to analyze the results of the simulation. Unlike the input grids, the results interface is straightforward and easy to interpret. Graphs can easily be altered or exported and are fully color coded and labeled for clarity. Additionally, for many graphs, summary level data also appears and can be used to help interpret the graph, or for further analysis. A sample graph and data for SoSAT can be seen below in Figure 9.



System	Number of Systems	Number Operating	Number Inoperable	Availability (%)	Total Failures
Battalion	102	37	65	36.3	65
C2	1		1	0.0	1
Depot	2	2		100.0	
Company A	47	16	31	34.0	31
C2	1		1	0.0	1
Supply Vehicles	4		4	0.0	4
Platoon 1	17	3	14	17.6	14
C2	1		1	0.0	1

Figure 9. Sample SoSAT Results

4.2 The Math Behind the Models

4.2.1 More than Meets the Eye

While the user perspective provides insight into whether a certain city modeling software possess the capability to produce a model, that perspective does not provide a complete picture of the software's capabilities. Perhaps a program creates a remarkable model where the user can track energy use, spending within the city, and citizen happiness…but *how* is the program doing this? The mathematical formulations that make up the program are absolutely crucial when determining if the software is capable of realistically modeling a city. For example, does residential energy use vary throughout the day to realistically reflect the periods when residents are away at work or sleeping? And if it does vary, does it do so according to data/statistics obtained from actual cities? The level of fidelity of models produced by the programs directly depends on the accuracy of the underlying mathematics.

4.2.2 CityNet Attributes and Matlab

After the user visually defines the layers of the city and specifies the attribute characteristics, the data is transferred to an Excel spreadsheet. All of the analysis for CityNet is written in Matlab using object-oriented programming. The Matlab functions use the data from the Excel spreadsheet to calculate different city characteristics such as emissions produced by transportation, number of residents in the city, commercial water demand in the city, land area used by the energy system, and many other characteristics. A full list of current calculation capabilities for CityNet can be found in Appendix A-Current Calculation Capability in CityNet.

While the Matlab functions have been carefully specified for the different calculations, there still exists a large possibility for error in the user-input phase. As discussed earlier, the user must specify the attribute name, description, units, bounds, and value. With the exception of description, mistakes in any of these fields could directly impact the fidelity of the model. For example, if the name is input incorrectly, then Matlab will not be able to find the necessary attribute in the Excel spreadsheet. If the units are input incorrectly, the Matlab functions will still calculate the characteristic, but the result will be completely misleading. The bounds and value specified also have an associated risk. It is reliant upon the user to make sure the values are realistic and to have gone through the necessary steps to validate these values. If the values are made up, then the model and calculations are clearly not reliable.

4.2.3 CityOne Scenario Modeling

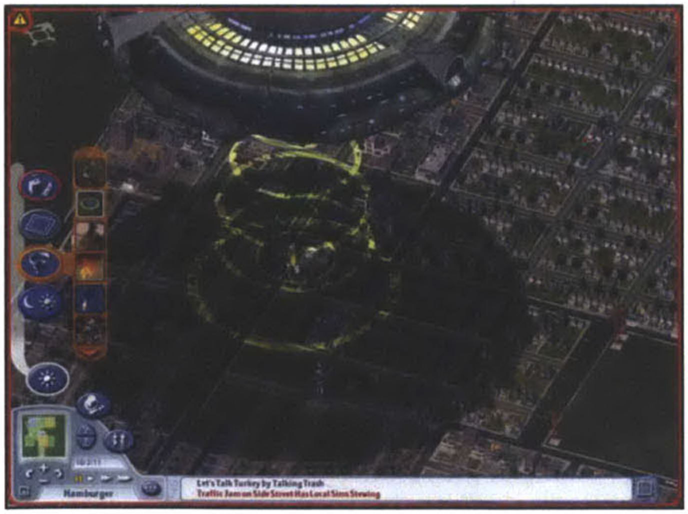
Unlike the other two programs, CityOne does not have physical modeling capabilities. Instead it models potential scenarios in each of the four industries. However, there is a decided lack of mathematics needed for these scenarios since they always appear in the same order and always contain the same possible actions. There are also no user inputs to the program. The only real modeling associated with this program is its calculations of increases/decreases to population, business climate, and citizen happiness. It is unclear how these characteristics are calculated, though they seem directly correlated to whether or not the chosen action requires the use of IBM technology. This bias will be discussed in the next section.

4.2.4 The SimCity 4 Mystery Code

The source code for SimCity4 is notoriously kept secret. The only code that has ever been released is for the original SimCity. The game was renamed Micropolis and released under the GNU General Public Library. However, that first version of the game possesses few similarities to SimCity 4.

A few clues as to the inner workings of the game can be gleaned through reading the user manual. As mentioned earlier, the game runs based on a 24 hour clock. The manual states that "morning and evening rush hours really *do* happen at rush hour" [20]. So at the very least, traffic flow increases based on what hour of the day it is. However, this increase may or may not be realistic for a city of a given size.

Other clues are obtained through the game itself. Potential natural disasters include giant robot attacks, UFO attacks (see Figure 10), and an attack by "Autosaurus Wrecks"—a Transformer-type mechanical monster. These disasters show that at least some aspects of the game are not realistic. Other anomalies can be seen in how infrastructure is built. Buildings are automatically demolished if an elevated rail line or a park is placed in the same cell of the grid. Additionally, homes and other buildings can function without a water source when the city is still small.



Let's Talk Turkey by Talking Trash
Traffic Jam on Side Street Has Local Sims Stewing

Figure 10. UFO Attack in SimCity 4

Much more detailed information would be needed to determine whether the various characteristics of cities within SimCity 4 are being modeled realistically. Based on the aforementioned observations and remembering that the program is designed for entertainment purposes, it seems unlikely that SimCity 4 is capable of creating models with high fidelity. Other complications with using this program as a modeling tool will be discussed in the next section.

4.2.5 SoSAT Mathematics

SoSAT was purposefully designed as a mathematical model and thus its underlying formulations and assumptions have been rigorously reviewed and validated for accuracy. SoSAT is a state model tool with stochastic simulation, and advanced data visualization capability for results analysis. Data entry also allows for maximum accuracy since all data entered can be associated with a statistical distribution of choice with the relevant parameters specified by the user. Including this slight variability enables the modeled systems to more closely imitate reality. Even if technical specifications say a system will run for 20 hours without failing, the reality will always be a value whose distribution is around (but not necessarily equal to) 20 hours.

Additionally, SoSAT employs a discrete event simulation. The model evaluates each system, attributes, and connections to calculate the future time in which the system or associated characteristics are expected to change. At that time, the model reevaluates those parameters and updates all states, statistics, etc. in order to calculate the next change event. This continues until the end of the simulation. Though the delta time between evaluations is not constant since it depends on system changes, the user has the ability to specify the desired detail interval at which the model will update the component systems in order to output results at that time interval. Different detail intervals may be specified on an individual component system basis if desired. Available distribution types and select SoSAT calculations will be examined in detail as part of the case study.

As discussed in the user interface section, SoSAT also is able to fully capture the SoS structure within the model. All connections between systems are rigorously defined by the user and this data is used in the underlying calculations. When looking at the results, you can view multiple interconnected systems in the same chart and clearly see how the behavior of one impacts the performance of the others. The results analysis also allows the user to track cause and effect within the SoS by examining when key systems fail or at what points during the simulation systems are operating.

Similar to CityNet, SoSAT relies heavily on the accuracy of data entered by the user. However, data types are automatically labeled in SoSAT to prevent the potential for naming mismatches seen in CityNet. Reliable data will depend on data collection methods, consultation with subject matter experts, and careful review of entered data to catch any typos.

4.3 Successes, Shortcomings, and Future Characteristics

4.3.1 Positive Aspects

Each of the four programs excelled in specific areas. These positive aspects provide a glimpse of which characteristics an ideal city-modeling program should contain.

CityNet excels in its simplistic user interface. The user can clearly see the aerial view of the city and easily define characteristics by simply drawing them on the picture. Everything is color coded, making it easy to see nodes, edges, and different layers. Additionally, the user specifies the grid size, making it possible to define the city at a high detail level or a more coarse level, depending on the desired accuracy. Furthermore, the user can easily connect layers using interlayers to model realistic connections such as stairways from the street level down to the subway platform. In the user interface, layers can be filtered so information is easier to see.

CityOne is remarkable for its linked resources. Each action is linked to information about IBM technologies and products, as well as white papers and case studies. All of these are real, industry documents that allow the user to quickly learn about various topics and make more informed decisions. This information also mirrors the type of information an individual would need to obtain if making such a decision in real life. In that regard, it trains the user on what type of resources to look for and how to properly utilize them.

The ability to customize almost every aspect is the selling point of SimCity 4. From molding the land, to specifying where buildings and water pipes are placed, to changing tax rates and funding levels, this game allows for maximum user inputs. The graphics and 3-D representations are also incredible. SimCity 4 also surpasses other programs in its ability to model over time. Days pass, buildings are built, people go to work, crime rates fluctuate, etc. Once the model gets going, it's fascinating to watch how it changes with the passage of time.

SoSAT is remarkable for its system of systems structure. It fully enables modeling and simulation from an SoS perspective while allowing maximum flexibility in the types of systems included. Instead of having to "trick" the model into allowing for multiple, interconnected, complex systems, the ability to model interactions are built into the software. Additionally, the results viewer offers a comprehensive yet simplistic way to track changes as they propagate throughout the SoS. The user is able to analyze any system or combination of systems and measure performance by many different metrics of interest.

4.3.2 Drawbacks

Just as every program excels in certain areas, they also have serious shortcomings. These too reveal what aspects are important when modeling cities, since the lack of certain capabilities adds a discernible level of difficulty to the modeling process.

While CityNet excels in the simplicity of its user inputs, getting outputs from the program is a complicated process. Calculations cannot be run directly from the Java interface. The user has to go into Matlab and run various functions. Although there are commands to run all of the functions associated with a specific window (transportation, energy, etc.) there does not appear to be a way to run all of the calculations at once. Users unfamiliar with Matlab or object-oriented programming may find this method of generating analysis overly confusing. As mentioned earlier, the reliance upon user input being accurate is also a limitation of CityNet.

A shocking aspect of CityOne is that the entire game almost seems to be an IBM sales pitch. After playing a few rounds, it becomes clear that actions that involve IBM products and/or technologies are more highly rewarded than other actions. These actions almost always result in higher scores and increases in population, business climate, and citizen happiness. Moreover, choosing other actions tends to only partially solve the problem or even introduce new problems. This theory is supported by the fact that the "library" only contains IBM documents and only IBM products and technologies are discussed as potential solutions. Aside from this, there is no other user input besides selecting an action, the picture of the city is always the same, and the sequence of problems to solve is always the same.

Most of Sim City 4's drawbacks are because of it being designed as a game. As the game progresses, problems within the city start piling up. The mayor then spends all the time trying to remedy these problems and there is no time to think about modeling since the game terminates if too many issues are left unaddressed. Also, the city constantly evolves without direction from the user. Houses spring up, people move into the city, new businesses are built, etc. The user has no control over this progression. The game has so many features that it quickly gets overwhelming, yet the user cannot define important characteristics such as how much various infrastructure items cost, the current population, size of buildings, etc. Some modeling aspects of the city are not realistic. Additionally, users are encouraged (via tutorials in the game) to only deal with issues as they arise, such as not building a fire station until a fire breaks out in the city. Aside from these disadvantages, support for the game seems to be dwindling. The game was released in 2003 and many of the support resources are no longer accessible.

The primary drawback associated with SoSAT is its lack of visualization. While the results explorer is highly visual, there is no way to visualize what the actual SoS looks like. The defined structure helps with understanding the organization, but there is currently no graphical component in the user interface. Another drawback is that structure within the model is highly difficult to change once it has been defined within the software. New nodes can be added at each level, but they cannot be reordered and will always show up at the bottom. While the structure is primarily a tool for the modeler, it helps to have similar structure elements grouped together to make the SoS as a whole easier to view. Entering the structure correctly requires carefully predefining the SoS before building the model. While this is a hindrance, we could alternatively argue that the extra upfront attention to defining the model pays off regardless of the software choice.

4.3.3 Possible Improvements

While testing each program from the user perspective and also for mathematical accuracy, it was easy to see areas where each program might be improved. Some of these improvements are small changes that would increase clarity, while other suggestions would require major additions to the programs.

Since CityNet is set to undergo another round of upgrades, there exist many exciting possibilities for the program. Analyses have to be run from Matlab, but if these could be accessed through the Java interface, it would make the program much more integrated. Perhaps an analysis window could be added where the user would select which calculations to run. The Java interface could then call Matlab to run the calculations and return the outputs to the user. Another potential change would be to use GIS and census data to create the underlying map in the Java interface. Instead of merely being a picture, this would allow it to have inherent properties such as population density. CityNet could then use that data directly instead of having the user specify all of the values. This could dramatically increase the accuracy of the models.

IBM's CityOne has an inherent bias that they should work to eliminate in future versions. By introducing technical solutions from other companies and expanding the library to contain non-IBM documents, they could make the game appear less like a sales pitch and more like a training tool. Additionally, the scenarios need to change. Simply introducing the problems in a random order would improve the user experience. The problems should really be adaptive, meaning that the next problem would be based on the previous answer. Some type of user input would also be helpful—inputs such as population size and what percentage of the city each region (residential, agriculture, commercial, or industrial) comprises would be plausible additions.

For SimCity 4 to be used as a modeling tool, there would have to be an option to separate the modeling from the simulation. There would also need to be an option to model the entire city first, and then run the simulation to see what happens. The user would also need the ability to build residential areas instead of having them automatically generated by the program. In fact, this auto-generation of buildings in general would need to have an on/off option. Additionally, the user would need to be able to change characteristics of the infrastructure such as the size, cost, etc. as discussed earlier. As with CityNet, if there were an easy way to import a GIS map of an area into the program, that would also increase modeling capabilities. Lastly, SimCity 4 already incorporates natural disasters, but it would be ideal to include weather/climate options and vary both natural disasters and climate based on global coordinates and altitude specified by the user.

As discussed in the drawbacks section, SoSAT could benefit greatly from a way to visualize the SoS structure with the aid of a graphical representation. One way to achieve this could be to show the structure as a network, where each node represents a component system within the model and an edge between two nodes represents two systems that are interconnected in some manner. This idea will be explored in greater detail during the BCIL case study.

4.4 Summary and Conclusion of Software Evaluation

City modeling software still has a long way to go before being able to accurately model and assess cities. Each program evaluated brought its own unique view of city modeling, but left out other desirable features. A comparison of the four programs analyzed can be found in Table 1.

Table 1. Summary of Software Characteristics

	CityNet	CityOne	SimCity4	SoSAT
Platform	PC, Mac	PC, Mac	PC (download)	PC
Purpose	Modeling Tool	Serious Game	Game	Modeling Tool
File Type	Download	Online	Download	Download
File Size	54.3 MB	N/A	1.2 GB	18.5 MB
Consider Interconnectivity	✓	✓	✓	✓
Modeling Capabilities	✓		✓	✓
Time Progression			✓	✓
Verified Formulations	✓			✓
Simplistic Interface	✓	✓		
3D Rendering			✓	
Define Attributes	✓		(some)	✓
User Resources		✓	✓	✓
Budget Calculations		✓	✓	✓
Visual Interface	✓	✓	✓	
Human Behavior		✓	✓	(some)
View City's Status during Simulation		✓	✓	(some)
Political Concerns		✓	✓	
Training Required	✓			✓
Accessible Code	✓			✓
Control over Simulation	✓			✓
Generate Results from Interface		(some)	(some)	✓

In comparing the four programs and experiencing their pros and cons from a user perspective, the following features are seen to be key to successful city modeling:

- Visualization
- Simplistic user interface
- Time progression

- Accurate user data input
- Ability to input population characteristics
- Realistic underlying geographical model
- Ability to specify attributes of component systems
- One integrated interface
- Accurate underlying mathematical formulations

4.5 Software Choice for Case Study

After evaluating the four programs and weighing the pros and cons, SoSAT was selected as the software of choice for the BCIL case study models. This decision was highly influenced by the software's previous use for systems of systems modeling, its validated underlying mathematical formulations, and its extreme modeling flexibility. The inclusion of time is also crucial to simulating the evolution and changes of city systems. The key drawback of SoSAT is that it doesn't include a visualization of the city. The results viewer includes many graphics, but there isn't a way to physically see the layout of the city (or in the case study, the layout of the camp). Visualization issues related to SoSAT will be further addressed during the case study.

5 Base Camp Integration Laboratory Case Study

As discussed in the problem statement, this thesis will focus primarily on modeling the Base Camp Integration Laboratory (BCIL) located in Fort Devens, MA. BCIL is comprised of two identical base camps, each designed to support 150 personnel. The layout of the camp can be seen in Figure 11. Aerial View of BCIL. These mimic the 150-person Force Provider camps currently in deployment. A Force Provider 150-person camp is packed into containers that can be transported by a single aircraft and set up in less than four hours [21]. The two camps in the BCIL function as a baseline and a test bed. The baseline is configured to either the standard winter or summer baseline configuration (depending on the season), while the test bed camp is modified from the baseline configuration to test new technologies such as solar power or different energy grid configurations. Sensors have been placed on all of the component systems in the camps to provide real-time monitoring of data, including energy use, water consumption, temperature, etc. Along with historical data, the sensors and the physical test-bed offer a unique opportunity to verify any virtual models of the camp.

Figure 11. Aerial View of BCIL [22]

The force provider camps act as an ideal miniaturized representation of cities, since they fulfill the same functional needs (shelter, hygiene, nourishment, etc.) as a city, and are comprised of similar complex systems. Some included systems include: housing, latrines, showers, kitchen, dining facilities, an energy grid, potable water supply, and waste containment and disposal systems. Each of these subsystems includes many component systems, some extremely complex in their own right. For example, the baseline camp includes a shower water reuse system (SWRS) which collects and processes used water from the showers in an effort to lower the required supply of potable water. Used shower water is sent through a series of filters and a large percentage of this water is purified to a level that can be sent back into the shower system and reused. The remaining water must be sent into a gray water holding tank for disposal. This SWRS is not only a complex system on its own (see Figure 12), but also has many interconnections with the water, infrastructure, and energy levels of the camp. We can see that the complexity of these base camp systems can easily rival those of a city.

Figure 12. Shower Water Reuse System [23]

This thesis will focus on modeling the summer baseline configuration, including the methodology for producing the model and the steps needed to validate the output from the simulation. Additional models will focus on analyzing the impacts of small changes to the baseline model through sensitivity analyses and assessing the impact of technology insertion on the overall base camp system by introducing new/changing technologies into the baseline model. The use of smart city sensor data will also be addressed. Additionally, visualization of the base camp as an SoS structure will also be examined. All modeling efforts will include a time-based simulation of the camp with results and analysis.

5.1 Defining the Project with the Stakeholders

The first step of any modeling effort is meeting with stakeholders and clearly outlining the scope of the project. Stakeholders are considered for this purpose to be any group or individuals with invested interest in the project who possess means to influence the outcome of the project. SoS modeling is by necessity an iterative process, and the modeler must ensure that the stakeholders are involved in each iteration. Individuals are naturally distrusting of complicated models producing unexpected output, which is certainly the case in SoS city models. Involving stakeholders in the process of building the model displaces uncertainty by creating stakeholder buy-in. They are able to see why component systems are modeled as they are, how interconnections are defined, and how unexpected results can be explained by the interaction and availability of sub-groupings of component systems. Since these models are highly complex and time-consuming to create, involving the stakeholder at each iteration also prevents extensive rework once the model is completed.

Large scale SoS modeling projects can be completed using the following approach:

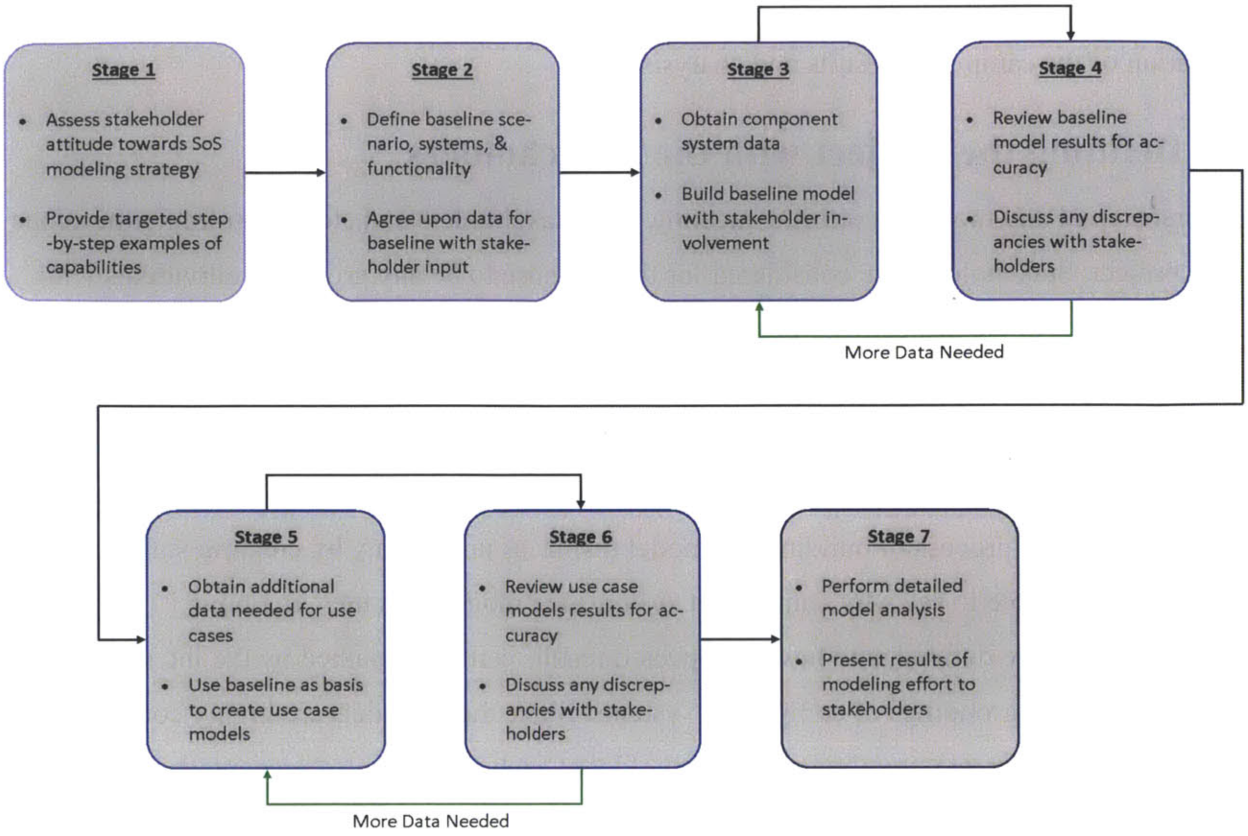


Figure 13. SoS Modeling Strategy

The first stage includes assessing stakeholders' attitude towards SoS models. Until SoS models gain wider acceptance, this step is crucial to the success of the project. Through detailed step-by-step examples, the modeler can convey why SoS modeling is important and specific ways it will benefit the current project. At this stage it is imperative to address the ways SoS modeling differs from other modeling strategies, and also what modeling capabilities and analysis results are possible as a result of using SoS modeling as compared to other types of models.

Once the value of SoS modeling has been established, all parties involved must agree upon how the baseline model is to be defined. This includes defining which physical entity is to be modeled (ex. A specific city/base camp at a particular point in time), what component systems will be included, and how the component systems function individually and with respect to other component systems. These definitions include considerations such as the operating profiles of component systems (when they are operating, operable, or off), which systems are dependent upon other systems, and how failing independent systems impact the availability of dependent systems. Defining a baseline often spurs strong debates among stakeholders who each want specific systems of interest represented. At this point, it becomes crucial to separate out what is normally part of the city, and what should be considered auxiliary. A simple way of doing this is by creating a list of case studies, which are additional simulation models to be built and analyzed after the baseline model has been completed. These are useful for examining new technologies, the impact of removing systems, the effects of natural disasters, etc.

After the baseline model has been defined, all necessary data will need to be gathered. Stakeholders or subject matter experts normally supply such data, again emphasizing the need for stakeholder involvement and buy-in throughout the modeling process. Once data is obtained, the SoS model can be developed, again with involvement from stakeholders. The next stage includes reviewing the simulation results produced by the model for accuracy. These analyses usually reveal small changes that need to be made to the model. Occasionally, the analyses reveal the presence of incorrect assumptions and additional data will need to be gathered before proceeding. This feedback is represented in the diagram by the green arrow leading from Stage 4 back to Stage 3. After the new data is gathered and incorporated, the model will need to be rerun and reanalyzed. This process repeats until the modeler and stakeholders can be assured of the accuracy of the data and assumptions used in the model.

Once the baseline model is complete, the additional use cases of interest will be built as separate variations of the baseline model. Again, these are then analyzed for accuracy and reiterated upon as necessary as depicted in the diagram. When all models have been completed and validated, a detailed analysis is conducted which should include all output of interest; pertinent changes in individual systems, groups of systems, and the entire SoS over time; and the relational impacts between component systems. At the conclusion of the modeling and analysis effort, results are presented to the stakeholders.

5.2 Specifying the Model Structure

The structure of an SoS model is perhaps the most important consideration to the modeling effort. Systems must be modeled and represented in such a way that results generated from the simulation are meaningful and can be aggregated at multiple levels for analysis. An SoS structure is necessarily hierarchical, yet even this specification can yield great variation in modeling methodology. From a city modeling perspective, there are two main ways of structuring the model—from a layered perspective, or from a geographical perspective. The primary difference in these two approaches occurs at the level II system designation.

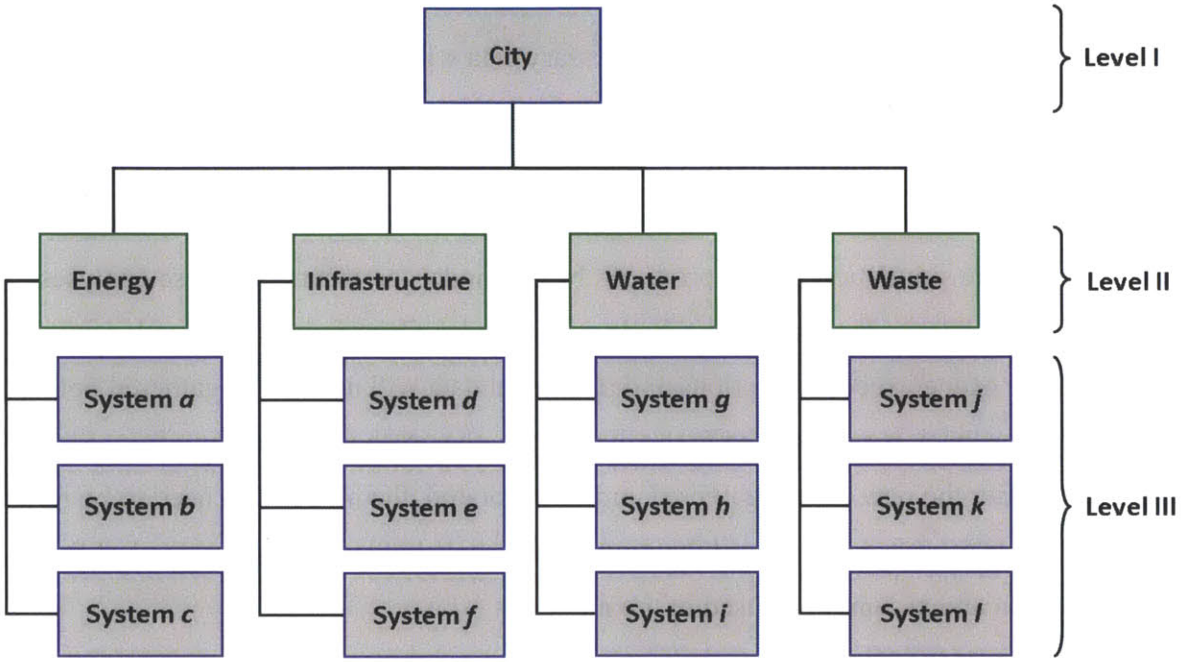


Figure 14. Layered Perspective Structure

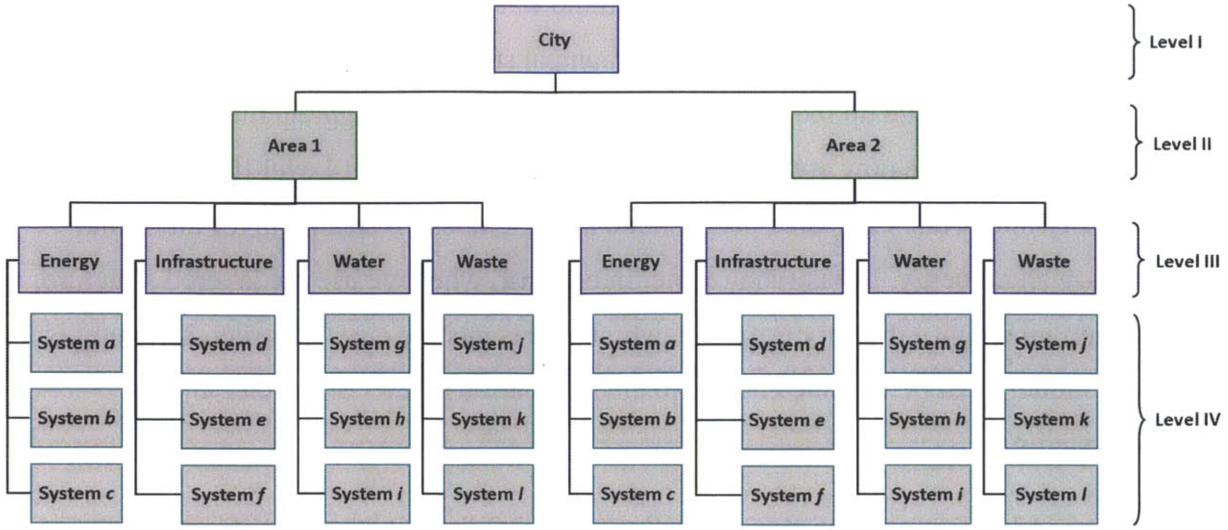


Figure 15. Geographical Perspective Structure

Note that in both structures, the number of structural components at each level can be expanded beyond the structure shown in the figures.

The structure is usually straightforward to determine with most SoS models, but city SoS models present a unique challenge because of the way we are conditioned to think about cities. The approach shown in Figure 15 is most intuitive, since it breaks down the city or base camp into different geographical areas before looking at component systems. This mimics the way we approach our daily use of cities, where we may work in the financial district, dine downtown, and live in one of the up-and-coming neighborhoods. We automatically divide our usage of cities into sub-areas, which each provide a different set of usable attributes to us as individuals.

This geographical perspective, while highly intuitive, fails to create an SoS city model with functional results. We may be interested in looking at different areas of the city, but we are also highly concerned with the city as a whole. Only when the functional layers are included in the Level II structure are we able to obtain city-wide results to analyze how overall water-consumption is changing, how the availability of the energy grid impacts the overall performance of the city, etc. This is because each lower hierarchical level must be a more detailed version of the proceeding level in order to be able to aggregate results. In the layered structure presented in Figure 14, each level is simply a more refined representation of the parent level. If we are considering potable water-using systems, we have the individual component systems at the lowest level, sub-groupings of them above that, and the water "layer" above that which is part of the representative city. But when we use the geographical structure, we are stuck at Level II. The units no longer aggregate up the structure. Whereas we had been aggregating a quantity of gallons of potable water, we now are left with an ambiguous quantity at Level II where we have separated the city into areas. Each area parent node now includes diverse units for all of the layers below—water, energy, infrastructure, etc. In contrast, when we use the layered structure, the units remain intact until the very top node, which represents the city. This representation is key since our model must be flexible enough to analyze various views of the city and its component systems.

For small cities, or base camps, the area is small enough that the model can safely ignore considerations of sub-areas, yet this does not hold true for larger cities. The characteristics of large cities often vary dramatically based on the area in question. Densely populated downtown areas have more residents and therefore higher water consumption per area than spread-out houses in suburban areas. However, one could argue that suburban areas have more outdoor space and appropriate large quantities of water for landscaping and recreational use. These types of comparisons and tradeoffs are something we want to be able to analyze from the results of the simulation. Since including area nodes at Level II of the hierarchy didn't work, we need to move them to a different level. Shifting them downward to Level III as shown in Figure 16 provides a workable solution.

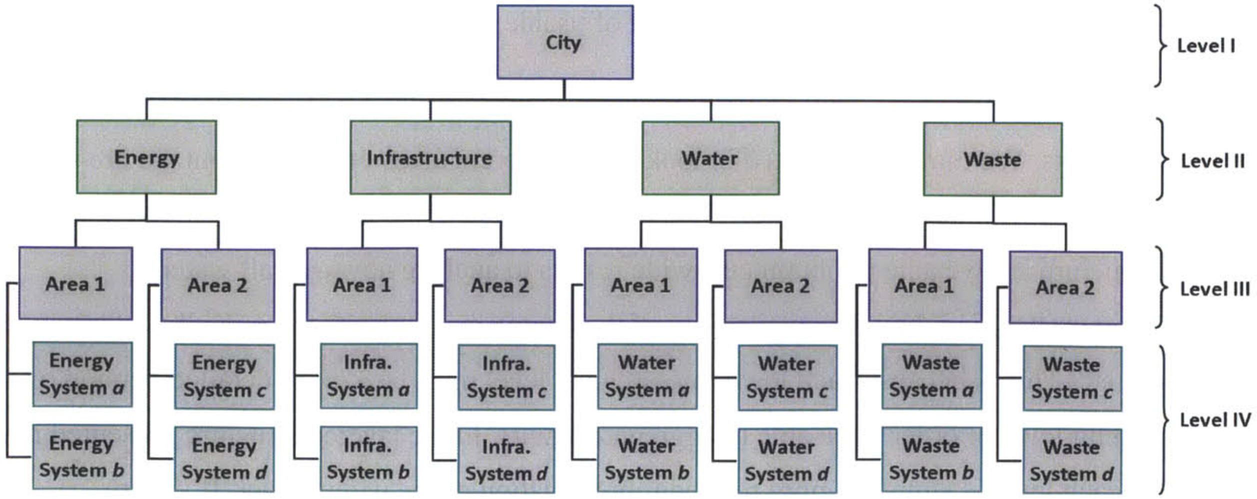


Figure 16. Hybrid Model Structure

From the hybrid structure, we can see that all units are now retained as we aggregate the levels of the hierarchical structure. The value of the various layers (energy, infrastructure, etc.), can be assessed for each area of interest and also for the city as a whole. By slightly rearranging the structure from the original intuitive structure, we have now incorporated the flexibility to analyze all quantities of interest within the SoS city structure. In general, this hybrid structure should be used as a guideline to create the structure for any given SoS city model. Actual implementation will vary depending on the type of software used and how the structure is used within the internal calculations of the model to produce results. Since the BCIL case study does not involve different geographical areas, the modeling effort for this thesis will use the layered perspective structure presented in Figure 14. Note that the difference between these two structures is achieved by removing the Level III area nodes in the hybrid model and shifting the remaining nodes up one level.

The hybrid model is used as a general rule of thumb since it will work regardless of the model being built and software being used, but the modeler should ultimately rely on their knowledge of the software being used to determine which structure is most appropriate. Some software may be capable of tracking overall resource consumption and production even when geographical areas are defined at Level II of the structure rather than Level III. The geographical structure may also be appropriate when modeling separate but similarly located areas (such as a city and neighboring suburbs), or large cities with geographical spread (such as L.A.), where aggregated totals are less of a concern than area-based results. Additionally, aggregated totals for the entire structure may still be obtained through post processing of the resource totals from Level II. The chosen structure should take into account software considerations, stakeholder interests, and the need to create a logically structured model.

5.3 Visualizing the Baseline Camp Layout and Systems

Some type of visualization will be necessary when modeling permanent or temporary cities as SoS in order to facilitate deeper understanding of how the models function. Yet the structure and complexity of these city models make traditional visualizations challenging. Displaying a 3-D visualization with real-time updates within the context of a modeling program requires an exorbitant amount of data processing and graphics rendering. Additionally, it must be recognized that a 3-D representation will not provide a complete view of the city from an SoS perspective.

At best, the user would be able to view the infrastructure layer, the city's spatial features, and select street-level components of the transportation layer. As part of SoS modeling, visualization must be used as an educational aide, rather than a display of graphical prowess.

Other types of visualization are needed to better understand how the component systems are functioning within the city and the many interconnections between the component systems that are driving the output of the simulation model. The ideal visualization methods will be easily generated with few computational demands on the system. They will also convey information about how the city systems are organized and interconnected, in a way easily understood by users with diverse backgrounds and training levels. Three types of visualization are used to represent and understand the BCIL baseline model, including an aerial pictorial description, a network representation, and an undirected adjacency matrix. The combination of these three visualizations creates a powerful method for understanding the SoS model.

The aerial view of the 150 person base camp shown in Figure 11, and repeated below in Figure 17 for convenience, emphasizes the spatial relationships between component systems. Aerial representations of SoS provide the most intuitive visualization, with systems represented as they would realistically appear in real life. The aerial view provides the modeler or user with a simplistic understanding of how the component systems may be related to each other and acts as an important first step in determining the structure of the SoS model, and the feasibility of the representation. For example, the model may be built with the generator micro grid on one side of the camp, with connections to component energy-using systems on the other side of the camp. As the distance grows, the modeler may need to reexamine assumptions that were made about how energy is distributed, since long power lines can affect the efficiency of the power grid. Additionally, quick sanity checks can be made on the basic set up of systems. If two component systems are supposed to share a water source, but are located on opposite ends of the camp from each other, the modeler automatically knows this assumption is incorrect and needs to be corrected. Understanding and integrating this geographical knowledge is essential for building a model that accurately captures the functionality of the city or base camp.

Figure 17. Aerial Visualization

While the aerial view provides important spatial information about the base camp, we must also understand which component systems are interconnected. One way of representing interactions between the component systems is through the use of an adjacency matrix. The size of the matrix is n x n, where n is the number of component systems in the SoS model structure. For simplicity, we will use an undirected adjacency matrix, which means that we only care whether a connection between two systems exists, but do not distinguish any type of order or direction in the implementation of that connection. The adjacency matrix for the component systems of the BCIL baseline camp in the summer configuration is given below in Figure 18.

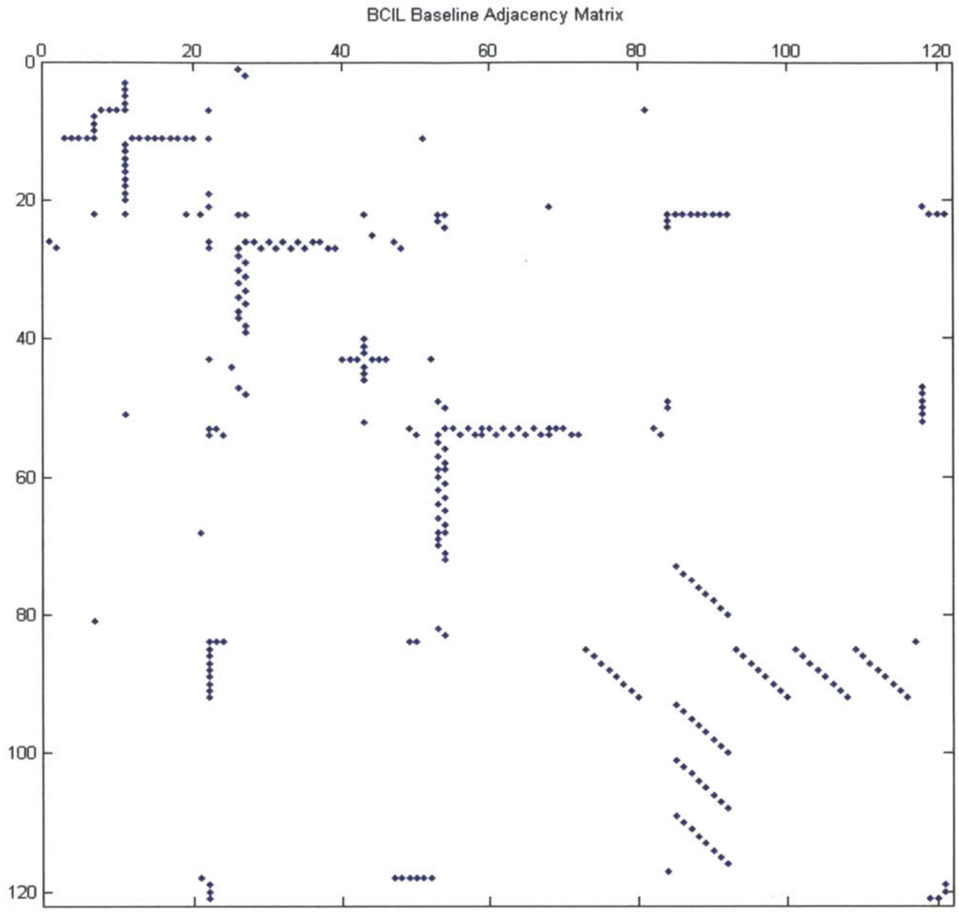


Figure 18. BCIL Baseline Adjacency Matrix

Component systems in the adjacency matrix were organized according to the containerized system or grouping of systems to which they belong. A high level mapping of the numbered component system to their respective groupings of systems is given in Table 2 below.

Table 2. Mapping of Component System Numbers

High Level Group	Component Systems
Water (Black, Gray, and Potable) Storage	1-3, 23-25, 47-52,117
Kitchen and Dining Facilities	4-20
Energy (Fuel and Electricity)	21-22
Latrine Containerized Systems	26-39
Laundry Containerized System	40-46
Shower Containerized Systems	53-72
Billeting Tents	73-116
Camp Operability (Maintenance, Supplies, Parts)	118-121

As can be seen from the adjacency matrix, there are 121 component systems in the BCIL baseline camp. The stars on the graph represent a connection between the two systems. For example if there is a star in row 60, column 53, then there is some kind of functional interaction between component system 60 and component system 53. This could be the transfer of electricity, water, or waste; or a physical connection such as electrical outlets in the wall of the containerized kitchen. Note that since we are using an undirected adjacency matrix, the resulting plot is symmetrical. Going back to the example, this means there is also a star in row 53, column 60. In the actual matrix, stars are represented by the number one, while row/column combinations without interactions are assigned a value of zero. The more nonzero entries in the matrix, the more interconnected the SoS structure. We would expect the number of nonzero entries to increase for larger cities, the inclusion of more component systems, and/or the presence of more complex systems.

Using an adjacency matrix representation fulfills two functions. When developed before creating the SoS model, it acts as a valuable tool for understanding how systems need to be represented in the model and for ensuring that all connections are accounted for and implemented. Modeling assumptions can be cross-referenced with the matrix for increased modeling accuracy. The matrix is also a valuable way to visually convey the importance of SoS modeling to stakeholders. Understanding the various ways in which systems can be connected to each other, or the extent of these interconnections can often be difficult to comprehend without a visual aid. Adjacency matrices provide a concise way of visualizing interconnectivity without the stakeholders or users having to deal with the complex software representation of the SoS model. Along these lines, the matrix also offers a simplistic way to review basic assumptions regarding connections between component systems with stakeholders and/or customers. A large group of people can easily look at the matrix and point out any discrepancies, helping to address errors in a timely and efficient manner before they become deeply integrated through the modeling process.

The final visualization type, the network representation, combines elements of both the aerial view and the adjacency matrix representation. The network representation of the BCIL baseline camp is given in Figure 19 for the summer configuration. This network was generated using the complex network visualization tool Cytoscape [24], an open-source software platform available online.

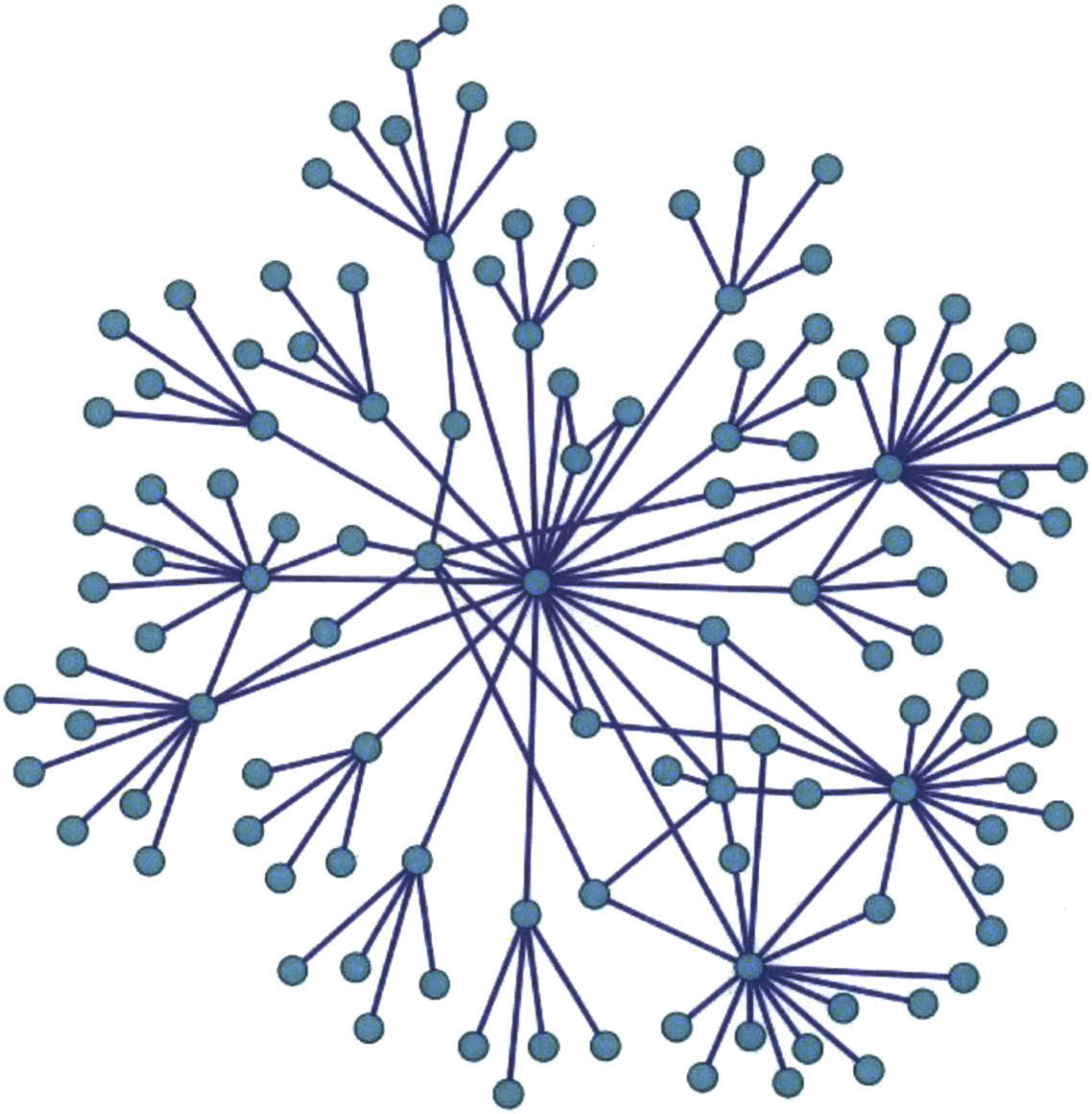


Figure 19. Network Representation of BCIL Baseline

The network representation uses the same concept as the adjacency matrix to designate which component systems are connected. A "star" on the plot of the adjacency matrix becomes an edge between two nodes in the network representation, where the two nodes are the two component systems that are connected. Indeed, the adjacency matrix is simply a compact way of recording the data from the network representation. Using a network however, adds an additional spatial element to the visualization. Networks enable the user to see how systems are grouped together functionally, which often correspond to the physical groupings seen in the aerial depiction. Yet the network considers all of the component systems, not just those which are immediately visible to the human eye. Instead of seeing the kitchen connected to the dining tent as in the aerial view, we now not only see that the kitchen and dining tent are connected, but that they are also hubs to all of the component systems within those two areas, such as the oven, ice maker, and lights. The network representation adds granularity to the visualization, though it loses the geographical accuracy of the aerial view. Networks also provide a powerful way of showing stakeholders how systems are grouped and connected to each other.

Notice that none of the three tools discussed are individually capable of providing a comprehensive visualization of a city SoS model. Each tool adds specific insight to understanding the SoS structure and the component systems, while partially overlapping conceptually with the other visualization types. Only by combining these tools do we arrive at a comprehensive visual representation of the SoS model and all its inherent intricacies. The modeler/user may choose one or two visualization techniques to fulfill a specific purpose (such as a presentation for a client), but all three are ultimately needed to understand the SoS model structure. These types of visualization are all relatively straightforward and simple to create external to the chosen city modeling software. However, any commercially developed SoS modeling software should sincerely consider integrating these features into the user interface for ease of use, particularly for users unfamiliar with modeling.

5.4 Creating the Baseline Models in SoSAT

The first step of creating the baseline model for the BCIL was defining which component systems would be included. This was relatively straightforward since one of the 150 person camps remains unchanged and is designated as the baseline. A listing of the systems was provided for the baseline camp and technical data was gathered. Talking with the customer revealed that the baseline camp is configured differently in the summer than in the winter. After discussing which systems change (added or removed), it was decided that the configurations were fundamentally different enough in their composition and functioning that two baselines were warranted. Both baselines were created as part of the case study for this thesis, though additional use cases were only applied to the summer baseline.

The next step in creating the model was determining how the component systems would be arranged and assigned within the SoS structure. The hybrid hierarchical structure described in section 5.2 was followed. The resulting structure is shown in Figure 20 below. Note that the final level, which would include the individual component systems, is omitted due to space limitations.

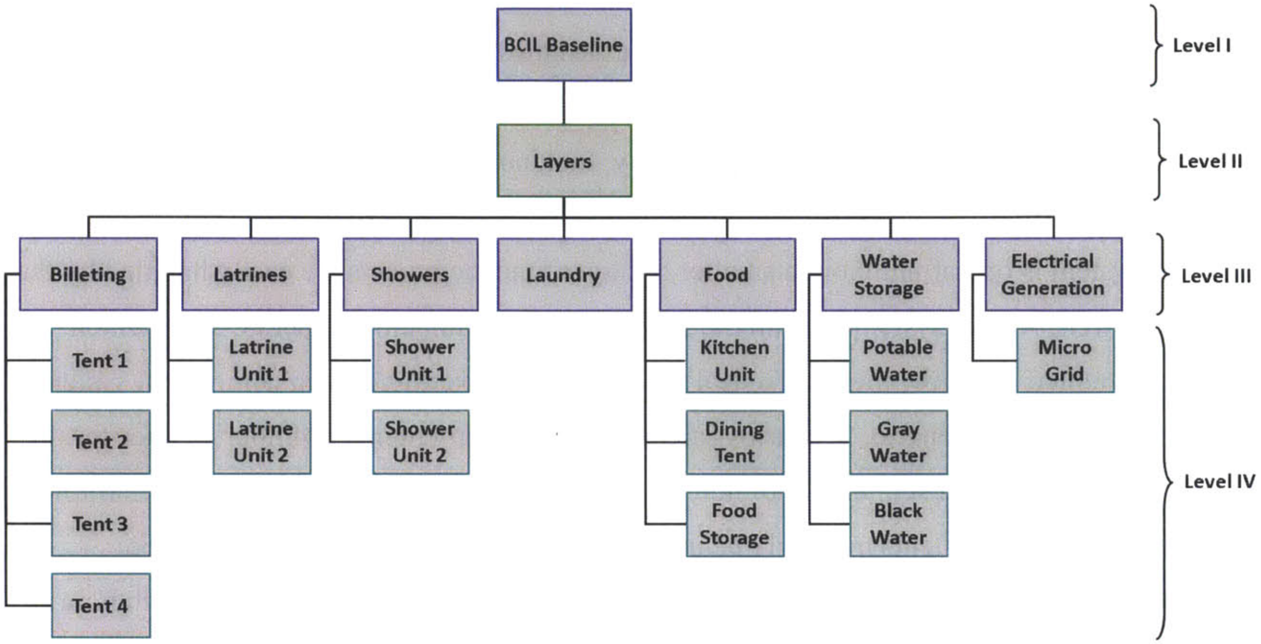


Figure 20. BCIL Model Structure

Notice that instead of the geographical areas seen in the standard hybrid hierarchical structure diagram, component systems are grouped by containerized systems, which are a key concept of the force provider model. Necessary complex systems such as laundry facilities, cooking facilities, showers, etc. are designed within a container to simplify transportation concerns. Each containerized system then contains all of the relevant component systems. The structure was discussed with the customer before implementation. At the level presented here, the structure is the same for the summer and winter baselines. The two baselines differ at the level of component systems, where air conditioners in the summer baseline are replaced with space heaters in the winter baseline. Additionally, systems are added to the winter baseline to protect the camp against cold weather conditions, while other systems used to conserve energy during the summer months are removed.

Data for the component systems was obtained from the customer and leveraged technical specifications and test data of the individual systems. Much of this data has previously been used and validated for other base camps with identical systems. Performance data will be gathered from upcoming BCIL trials and compared to the existing data; however, these trials are slated to occur after the conclusion of this case study.

Once the structure was finalized and the data gathered, the structure was defined within the SoSAT software and the component systems were assigned to the structure. Each component system was also associated with attributes including any resources it consumes or produces. The main resources tracked in the model are electricity, fuel, potable water, gray water, and black water. For this case study, all consumption and production rates were defined as hourly rates while the system is operating, though smaller or larger time increments are easily handled by the software as well. Not all systems within the base camp (or within cities) operate 24/7, which must be accounted for in the model. To deal with variations over time, each component system is assigned an operating scenario. Scenarios can be defined for any number of hours, either for the entire simulation, or for a set number of hours which are then repeated as a cycle. For example, the lights in the billeting tents may only be used at night, and are therefore defined to be operational only at certain hours. After defining whether the system is on or off for each hour in a 24 hour segment, the scenario can then be set to repeat—thereby appropriately representing the use of tent lighting on a day-to-day basis.

Interconnections between component systems are represented by supply connections or distribution networks. Supply connections are a connection between two individual systems, such as a shower containerized system that draws potable water from a water supply blivet (a soft-sided water storage container). Distribution networks include a supplier and/or consumer comprised of multiple component systems. The energy grid provides a good example of this setup. The micro grid is supplied by a group of generators and provides electricity to multiple user systems, including the billeting tents, kitchen, showers, etc.

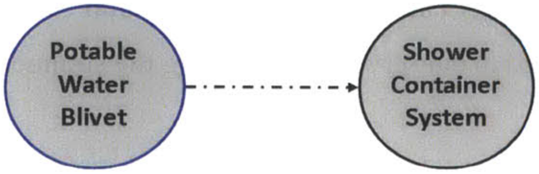


Figure 21. Supply Connection Example

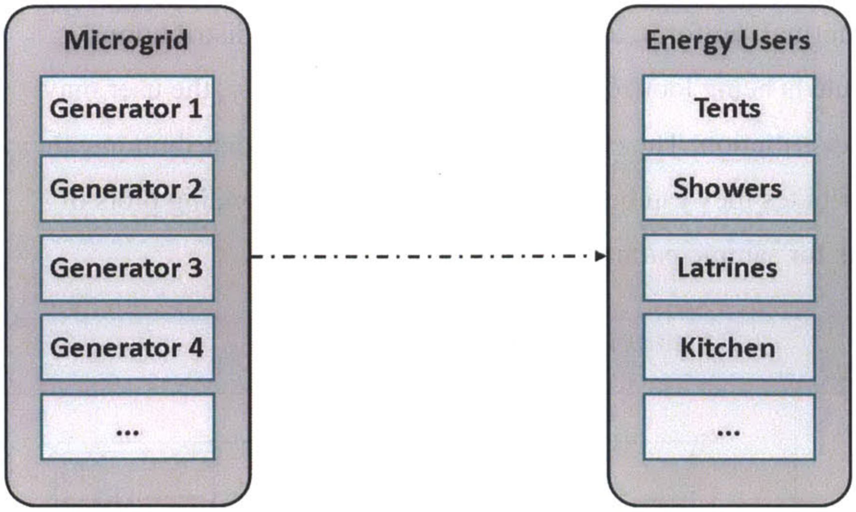


Figure 22. Distribution Network Example

The rules that define how resources and supplies are delivered are defined for each supply connection or distribution network individually. For each supply connection and distribution network, the component systems which can provide the resource or supply are defined. They are also assigned a priority. When a component system or group of component systems requires a resource, the provider system that is chosen by the model as the supplier is the component system with the highest priority. In the case where two or more provider systems are assigned the same priority, the system that can supply the resource in the shortest amount of time is chosen as the provider. Each provider system has a defined time-to-supply which can also be represented by a probability distribution if desired. The model also takes into account whether a potential provider system is currently processing another resource request when calculating which system can supply the resource in the shortest time. If the amount of resource demanded exceeds the resources available, the request will be fulfilled as long as the chosen provider system is able to supply at least 50% of the amount requested.

Every component system can also be assigned a failure rate and upon failure, may require maintenance by a maintenance team before becoming operable again. Rates in the model can be assigned a probability with the user's choice of probability distribution. Random seeds are used in the model to add realistic variability. If desired, the random seeds can be retained from one set of trials to the next to replicate results for debugging purposes. The infrastructure component systems are tracked through their assigned scenarios and also through their location within the defined SoS base camp structure. The types of probability distributions available in SoSAT include exponential, normal, lognormal, uniform, triangular, and Weibull. The distributions available depend on which attribute is being looked at in the model. Additionally, the user may specify a fixed value in lieu of a distribution. The equations for the probability distributions, their inputs, a list of which SoSAT attributes they can be applied to, and the corresponding plots of their probability density functions for sample parameters are summarized below.

Exponential

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, x \geq 0, \\ 0, \quad\quad x < 0. \end{cases} \quad (1)$$

$$\lambda = rate\ parameter$$

SoSAT Applications:

- Time-to-fail
- Personnel

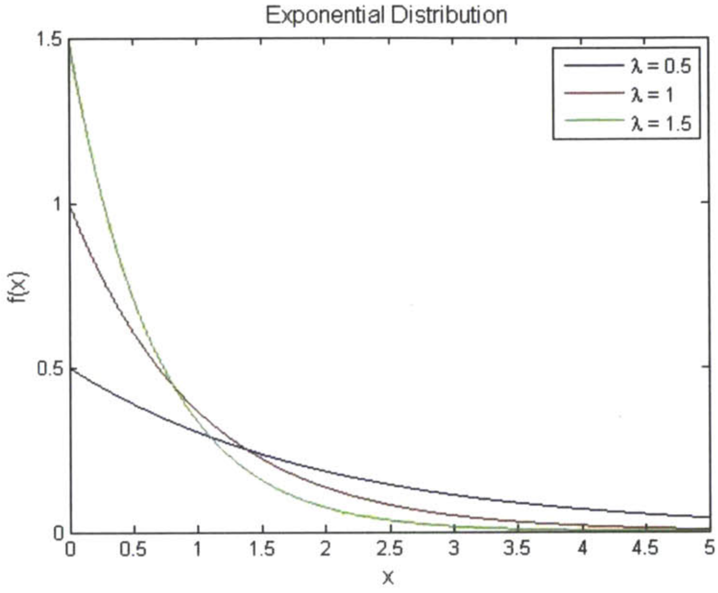


Figure 23. Exponential Distributions in SoSAT

Normal

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2)$$

$$\mu = mean$$

$$\sigma = standard\ deivation$$

SoSAT Applications:

- Time-to-fail
- Delay time
- Personnel
- Multiplier

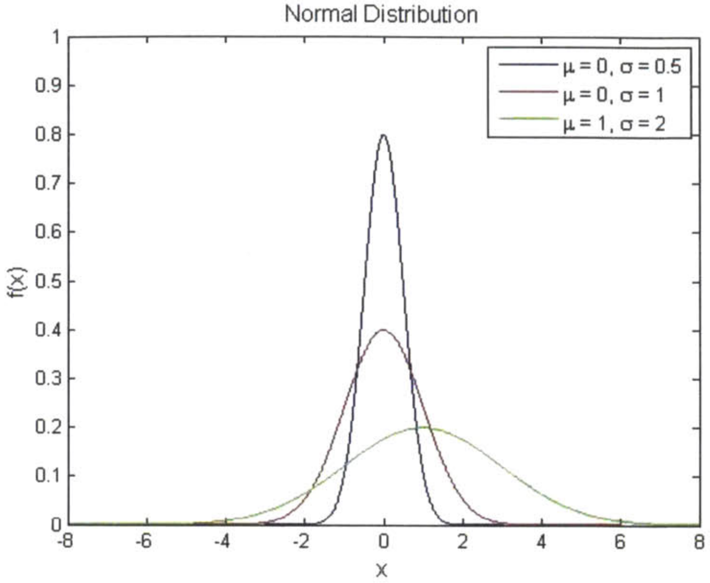


Figure 24. Normal Distributions in SoSAT

LogNormal

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(lnx-\mu)^2}{2\sigma^2}} \qquad (3)$$

$$\mu = mean$$

$$\sigma = standard\ deivation$$

SoSAT Applications:

- Delay time
- Personnel

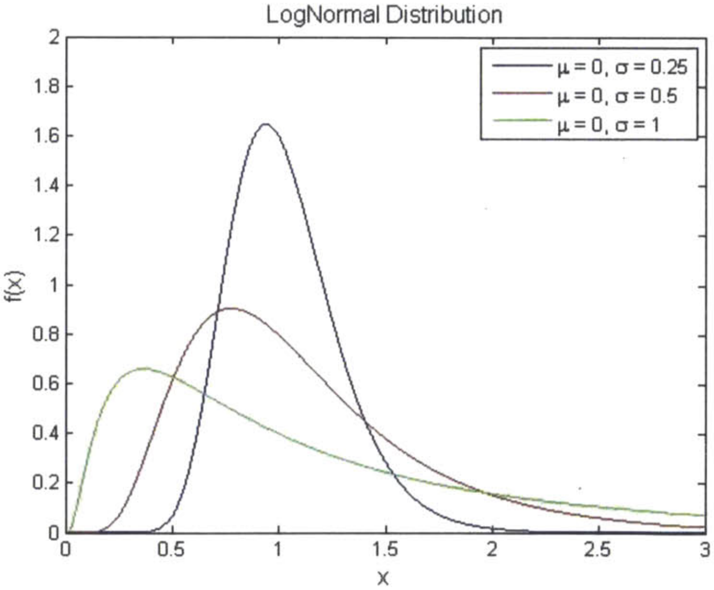


Figure 25. LogNormal Distributions in SoSAT

Uniform

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a \leq x \leq b, \\ 0, & otherwise. \end{cases} \quad (4)$$

$$a = min$$

$$b = max$$

SoSAT Applications:

- Delay time
- Personnel
- Multiplier

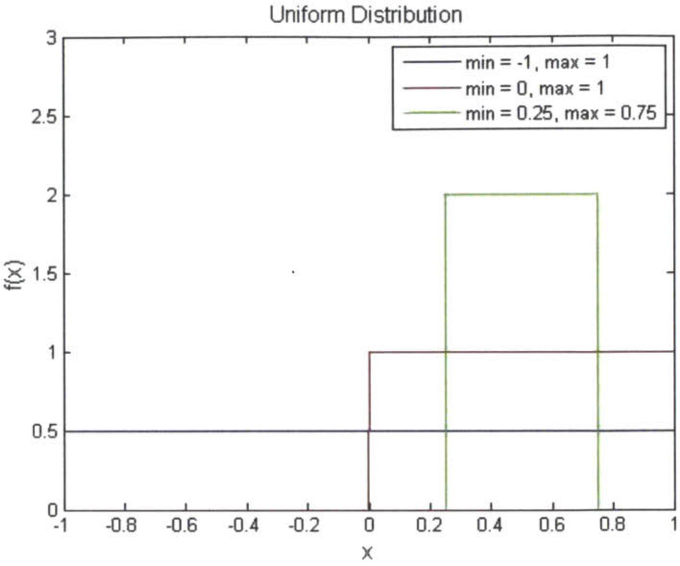


Figure 26. Uniform Distributions in SoSAT

Triangular

$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)}, & a \leq x \leq c \\ \dfrac{2(b-x)}{(b-a)(b-c)}, & c < x \leq b \\ 0, & otherwise \end{cases} \quad (5)$$

$$a = min$$

$$b = max$$

$$c = \frac{(a+b)}{2}$$

SoSAT Applications:

- Delay time
- Personnel
- Multiplier

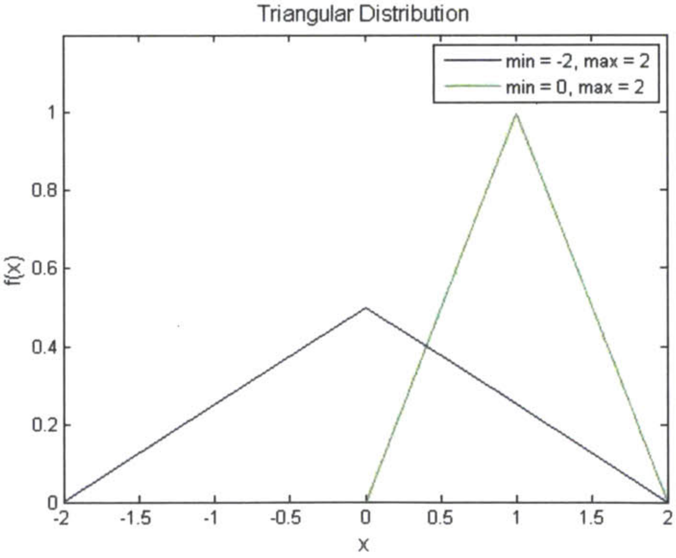


Figure 27. Triangular Distributions in SoSAT

Weibull

$$f(x) = \begin{cases} \frac{\kappa}{\lambda}\left(\frac{x}{\lambda}\right)^{\kappa-1} e^{-(x/\lambda)^{\kappa}}, x \geq 0, \\ 0, \qquad x < 0. \end{cases} \tag{6}$$

$$\lambda = scale\ parameter$$

$$\kappa = shape\ parameter$$

SoSAT Applications:

- Time-to-fail

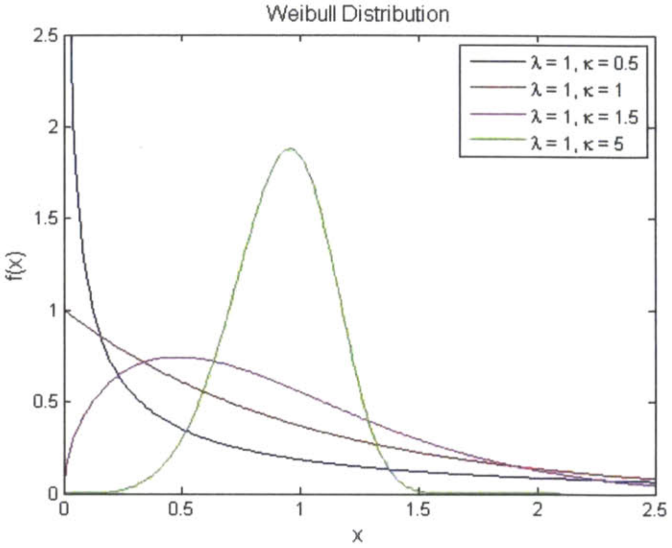


Figure 28. Weibull Distributions in SoSAT

Time-to-fail distributions are applied to component systems to specify how often they are expected to experience failures. Delay time distributions can be applied to repair times, service delivery times, and service performance times. Personnel distributions can be used to specify error rates, reset times, and replacement times. Lastly, multiplier distributions can be used to add realistic variability to resource consumption, resource generation, and time-to-fail for component systems.

If the stakeholders desire to examine the ability of a city or base camp to perform qualitative tasks, functions can defined and added to the model. For example, a function of a base camp could be to “feed the troops,” requiring the component systems in the kitchen to be operating. A function of a city could be to “provide public transportation via bus,” necessitating an adequate subset of the city’s buses be operating during the desired time frame. Functions in SoSAT specify which systems are required for the function to be accomplished, and whether all of the required systems need to be operating, or if only a subset is needed (i.e. whether the systems for the function operate in series or in parallel). The importance of such functions lies in their accessibility to those with non-technical backgrounds. While terms such as “operational availability” may not be clear, the average person can immediately understand functions such as the two listed above. Similar to visualization, functions are another key tool in conveying the information contained within the SoS city model to a general audience.

The number of personnel on base is assigned to be 150. Human behavior is not explicitly represented in this SoSAT model, though there are some calculations capabilities to include such variation. Some of these attributes include what kinds of skills each person has, what level of authority they have (ex. Can they order supplies for the camp?), and fatigue rates which influence their availability to perform daily duties. As with other attributes, human attributes can be assigned probability distributions to introduce variation and more realistically represent behavior. However, SoSAT is not a behavioral model, and people's behavior over time does not evolve as with other types of modeling techniques.

Most actions of the personnel on base are considered to be relatively uniform. For example, the potable water consumption for the showers is based on the assumption that each person takes a 10 minute shower. The lighting in the tents and the use of the kitchen component systems is also based on a strict schedule. These are safe assumptions for a military base camp, where personnel are in tents, eating, showering, etc. according to rigid timetables. These assumptions would not however be appropriate for traditional cities, where residents are for the most part free to do what they please.

A major source of variation in the model due to human behavior is how many personnel are on base at any given time. The baseline model includes changes in how many people are on base to mimic personnel leaving for missions while in deployment. These variations cause the usage of component systems on base to be scaled by the ratio of personnel remaining over total personnel, resulting in resources being used at a reduced rate. A scaling factor is applied to the number of personnel for each time interval of the simulation to determine how many personnel of the total 150 personnel are still on base during that time interval. This can be represented by the following equation,

$$PAX_i = s_i * x_{total} \tag{7}$$

where PAX_i represents the number of personnel on base during time interval i, s_i represents the scaling factor applied during interval i, and x_{total} represents the total possible number of personnel on the base camp (150 for this case study). The number of personnel on the base camp at each time interval of the simulation is shown in Figure 29 below.

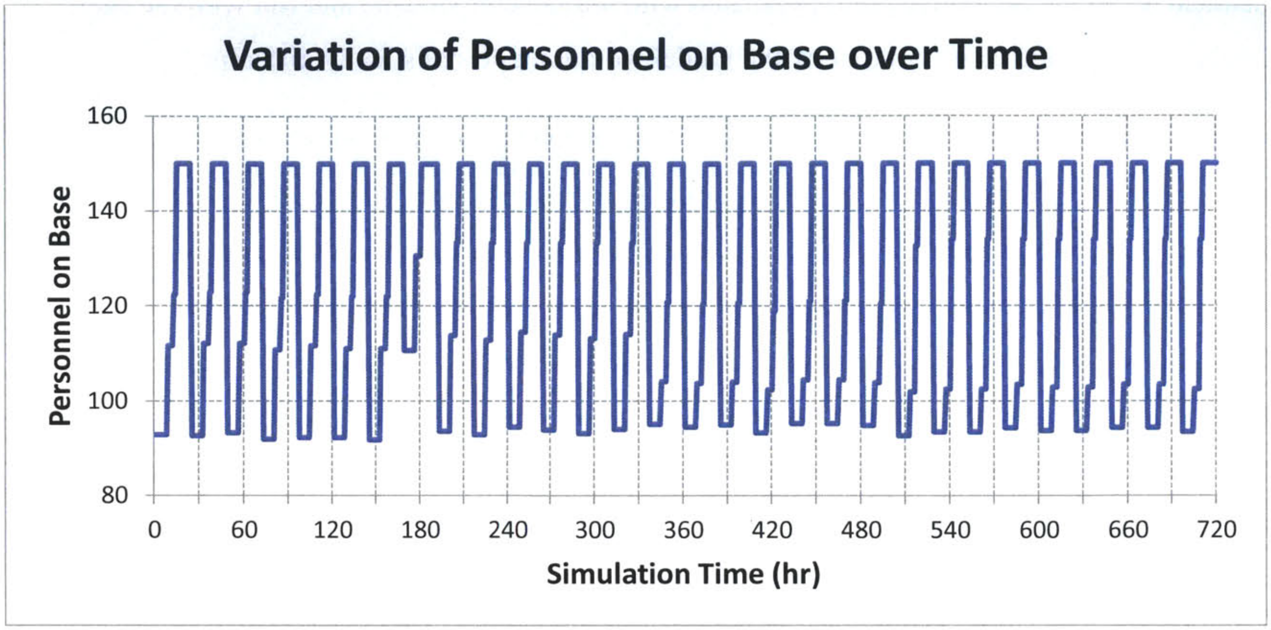


Figure 29. Personnel on Base

The scaling factors were determined through consultation with stakeholders. Their knowledge of how and when missions are carried out during deployment was the foundation for the scaling factors. Note that while the scaling factors produce a mostly cyclical daily pattern, there is still variation within each cycle.

5.5 Running the Simulation

Once the model is completed, a time-stepped simulation is run to generate results. The SoSAT simulation can be run for any desired period of time, with user-specified time intervals. For the BCIL case study, the baseline models were run for 30 days with one hour time intervals. The simulations are typically conducted using multiple trials with results based on the average values of the trials to smooth variation caused by the stochastic nature of the model, though individual trial results are still available. The simulations were run with 10 trials for validation purposes and with 100 trials for the detailed analysis. Using 10 trials is generally enough to spot most errors for validation purposes, but more trials are needed to achieve accurate results for the analysis.

Choosing the number of trials for the simulation and detailed analysis was based on the acceptable amount of variability in the results. Variability should be small enough that if an identical SoSAT model were built, populated with the same input data, and run with the same number of trials; the results output by the model would not be statistically different.

Normally the number of trials needed could be calculated based on the number of elements in the population that exhibit variability. This is virtually impossible for a systems of systems structure because of the interconnectedness between systems. Various attributes of component systems can exhibit different types of variability, as can the many supply connections and network distributions. Additionally, elements with no defined variability may still exhibit variability during the simulation because of the influence of other elements that do include variability. Therefore, the number of trials needed to achieve results within a certain acceptable level of variability cannot be determined a priori for SoS models. We instead depend on an iterative process to determine the number of trials needed.

100 trials were selected for the initial evaluation, as this is typically the minimum number needed for reliable results which can be presented to stakeholders. The output data used to analyze whether or not 100 trials minimized variation satisfactorily was the mission capable rate, or MC Rate. This rate is calculated using the following equation:

$$MC\ Rate = \frac{time\ operating + time\ operable}{time\ operating + time\ operable + time\ inoperable + time\ down} \quad (8)$$

The MC Rate was calculated for each of the 100 trials. The mean and standard deviation was then calculated. To really understand the variability between the trials, the standard error of the mean was also calculated as follows:

$$st\ error\ of\ the\ mean = \frac{\sigma}{\sqrt{n}} \quad (9)$$

where σ is the standard deviation and n is the number of trials. Using 100 trails, the standard error was calculated to be 0.000296. There is no standard acceptable level of variation for systems of systems models and the amount desired will vary based on stakeholders and applications. For the purposes of the case study, a standard error equal to or less than 0.05% was desired. The 100 trials put us within this range and were thus used for the analysis.

Had the amount of variability produced by the 100 trials not fallen within the acceptable range, we would then have begun an iterative process of progressively adding more trials and recalculating the standard error until acceptable. For more than 100 trials, we would expect the standard deviation to stay relatively stable. Therefore if more accuracy is desired, we could use the standard error calculation to determine approximately how many trials are needed to achieve that reduced level of variability. The approximate relationship between number of trials and the standard error of the mean is given in the table below for various numbers of trials.

Table 3. Standard Error vs. Number of Trials

	Number of Trials			
	100	250	500	1000
Standard Error of the Mean	0.000296	~0.000186988	~0.00013222	~0.000093494

Note that the number of trials needed to achieve output within the acceptable range of variability is SoS model dependent. Both the number of component systems and also the level of interconnectedness within a SoS model will affect the number of trials necessary.

5.6 Validating the Baseline Models

There are a few key ways of checking the model for implementation errors. The first includes running the model. Without properly defined systems and connections, the model will fail to run and will produce a list of errors to correct. These systematic errors are the simplest to fix, since in most cases the model structure dictates and explains how they need to be adjusted.

Once the model runs, the next area to check is availability over time for the entire SoS structure. Availability is calculated by SoSAT using the following equation:

$$availability = \frac{time\ operating}{time\ operating + time\ down} \tag{10}$$

We know the BCIL baseline camp is functional in real-life implementation, so we would expect to see only slight variations in availability over time. Sharp drops or a steady decline in availability would indicate an error in the model, most likely in the supply connections or distribution networks. For example, upon initially running the SoSAT model for the BCIL baseline for the summer configuration, the availability of the micro grid steadily decreased over time as shown in Figure 30.

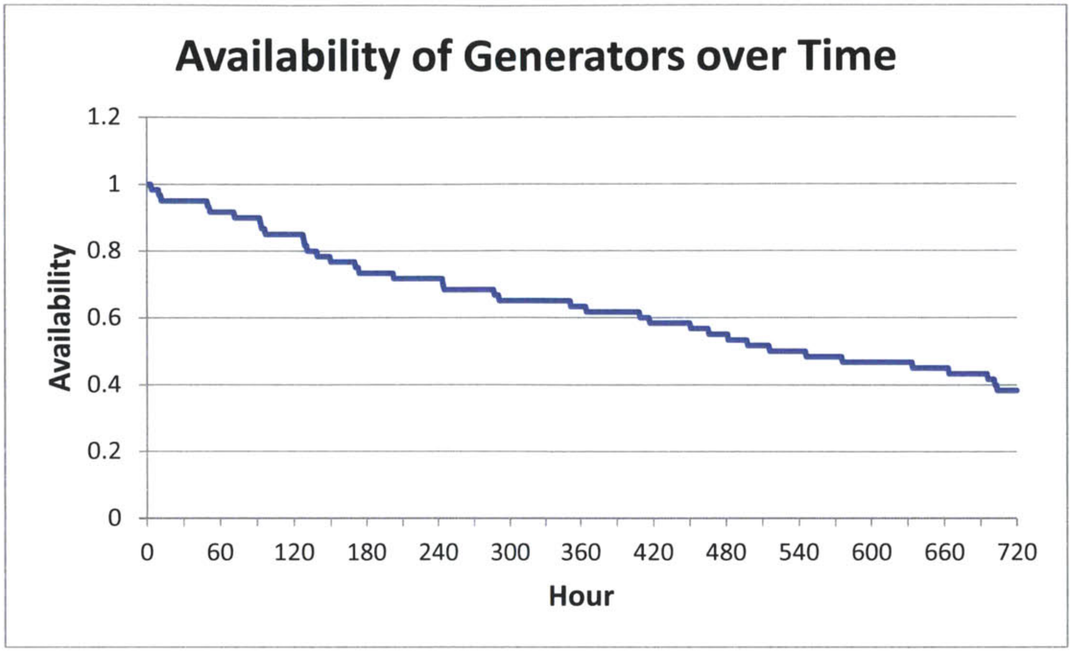


Figure 30. Example of Availability over Time Error

After examining the results from the individual generators powering the micro grid, it was observed that generators were failing during the simulation and never regained operability after that point. Though the systems were designed to age and fail, no maintenance personnel/actions were included in the model to service the systems. Maintenance had not been discussed while originally defining the baseline model and required a follow-up conversation with the customer to determine how many maintenance personnel would be present in a 150 person camp and how long an average repair would take depending on the type of failure.

Maintenance personnel are considered to be included in the 150 personnel present on the base camp. The maintenance personnel are included in the model according to the attributes related to their technical skills. This includes what types of maintenance and repair they are able to perform, as well as how long each type of maintenance and repair will take on average. To account for variations in human performance, probability distributions were included in the maintenance and repair times. Once the maintenance aspect was added to the SoSAT model, the simulation was rerun and the availability of the micro grid appeared reasonable as shown in Figure 31.

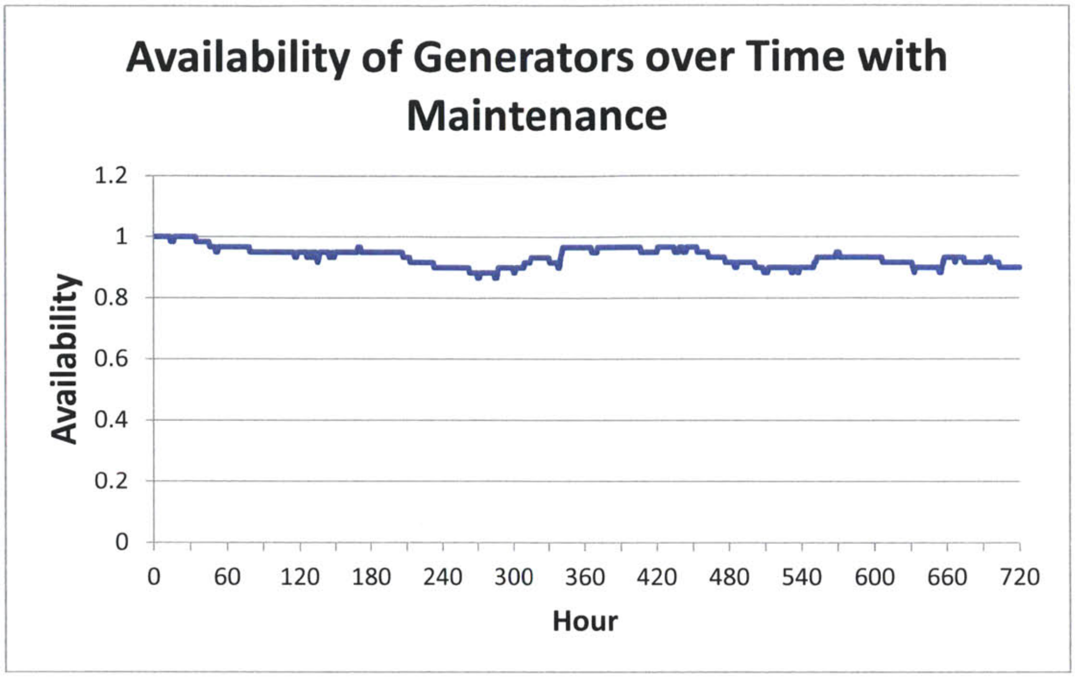


Figure 31. Availability of Generators over Time, BCIL Summer Baseline

Viewing availability results of various systems within the camp provides a good way to quickly find large scale errors in the implementation of modeling component systems or in assumptions that have been made. However, aggregated availability results do not reveal smaller errors whose presence may impact the SoS model in other aspects.

The third type of results that need to be examined to verify the performance accuracy of the SoSAT model are the quantities of resources used and consumed by systems over the course of the simulation. These quantities should be directly related to the attributes entered for the system. By taking the hourly rates of consumption or generation of each resource per component system, we can multiply by the number of hours each system operates per day and multiply again by the length of the simulation to get the consumption/generation quantity for the entire simulation. We want to first examine results aggregated to Level II of the structure, which contains the layers of interest. Therefore we sum the simulation-long results of component systems consuming/generating each type of resource and compare these quantities to the SoSAT model output for a simulation of the same duration.

To perform this initial comparison may involve temporarily altering the model in certain areas. For example, the generators supplying the micro grid are set up to age as the simulation progresses and fail or need repairs based on a given probability distribution. These failures are not accounted for in the spreadsheet calculations using the technical specifications, meaning the electricity consumed by the component systems in the SoSAT model is lower than calculated from the spreadsheet, since not enough generators are operating to provide sufficient electricity to the micro grid. As discussed, the baseline model includes changes in how many people are on base to mimic personnel leaving for missions while in deployment. These variations cause the usage of component systems on base to be scaled by the ratio of personnel remaining over total personnel, resulting in resources being used at a reduced rate. This variation in personnel was also temporarily removed when comparing the SoSAT output to the spreadsheet calculations.

Removing the failures from the generator systems and the variability from the number of personnel on base, and rerunning the simulation yielded the same quantities of resources generated and consumed as in the spreadsheet calculations. Temporarily making these small model alterations is the simplest way to determine whether differences between the SoSAT model output and the spreadsheet calculations are caused by the stochastic nature of the model and interconnections between systems, or if the differences result from error in data entry within the model. In the second case, more in-depth investigation is typically required to find the source of the error. Usually this is accomplished by comparing resource quantities used and generated at Level III of the structure, which are the containerized systems. Once a difference is found at that level, the resource consumption/production of the component systems can be compared to the spreadsheet calculations to determine which component system was improperly defined while creating the model.

The SoSAT model for the BCIL baseline for the summer configuration was run with slight changes to remove any random variability. The resource consumption and production produced by the SoSAT model for each type of resource was then compared with the spreadsheet calculations as described above. Each resource was calculated for the simulation by using the following equation,

$$R_{total} = \sum_{i=1}^{d} \sum_{j=1}^{n} r_j * o_i \tag{11}$$

where R_{total} represents the total amount of resource used and produced during the simulation, *d* represents the number of days in the simulation, *n* represents the number of systems that consume/produce the resource, r_j represents the hourly rate of resource consumption/production for system *j*, and o_i represents how many hours the system operates during day *d*.

Person-hours were also calculated based on how many personnel were on base during each time interval of the simulation, according to the following equation,

$$person \cdot hrs = \sum_{i=1}^{n} PAX_i = \sum_{i=1}^{n} s_i * x_{total} \tag{12}$$

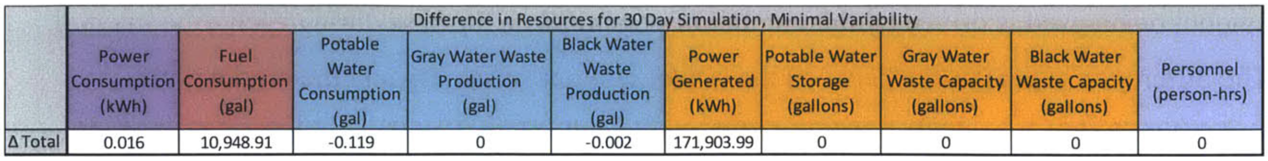
where n represents the number of hours in the simulation, PAX_i represents the number of personnel on base during hour *i*, s_i represents the scaling factor applied during hour *i*, and x_{total} represents the total number of personnel in the base camp (150 for this case study). To remove variability associated with personnel, s_i was set to 1 for all time intervals. When variability was included, s_i was assigned as seen in Figure 29.

The difference between the spreadsheet calculated quantity value and the SoSAT generated output quantity value is shown in Table 4 below for each resource type of interest. The difference between the two values was calculated using the following equation for each resource of interest:

$$\Delta\, Total = v_s - v_{SoSAT} \tag{13}$$

where v_s represents the value calculated from spreadsheet data and v_{SoSAT} represents the value generated from the output of the SoSAT model.

Table 4. Resource Comparison, Minimal Variability

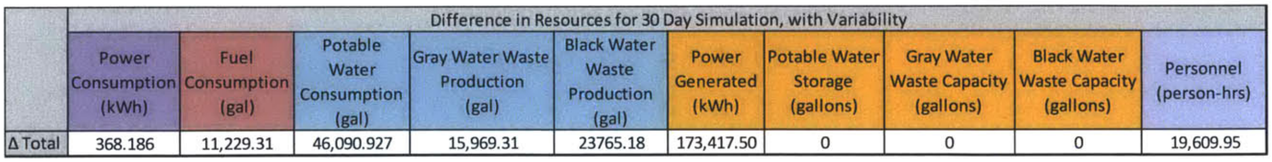


	Difference in Resources for 30 Day Simulation, Minimal Variability									
	Power Consumption (kWh)	Fuel Consumption (gal)	Potable Water Consumption (gal)	Gray Water Waste Production (gal)	Black Water Waste Production (gal)	Power Generated (kWh)	Potable Water Storage (gallons)	Gray Water Waste Capacity (gallons)	Black Water Waste Capacity (gallons)	Personnel (person-hrs)
Δ Total	0.016	10,948.91	-0.119	0	-0.002	171,903.99	0	0	0	0

From the table, we see that all quantities are approximately equal, except for the amount of fuel consumed and the amount of power generated. The difference in fuel consumption was caused by the micro grid implementation. Generators connected to the micro grid only operate as needed based on the energy consumption of the other component systems on the base camp. This varies from a spot generation configuration, where specific component systems are connected to a single generator, which remains on despite sometimes being underutilized. If we were using the spot generation configuration in the SoSAT model, we would expect the fuel consumption to be the same for the model output and the spreadsheet calculation. There is no way to switch between the two generator configurations in this BCIL model. Aside from significant alterations to the model, implementing spot generation requires the modeler to know the mapping between generators and component systems. This information cannot be obtained for this case study, since the micro grid configuration is the one implemented in reality and any mapping would have to be hypothetical. Though the fuel consumption quantities are difficult to compare, we can check that an appropriate number of generators are operating at various points in time according to the power demand on base.

Once the resource quantities output by the SoSAT model had been validated by the comparison process, the changes to the baseline model were removed and the simulation was re-run. The differences between the SoSAT model outputs (with variability included) and the spreadsheet calculations are given in Table 5 below.

Table 5. Resource Comparison, with Variability



	Difference in Resources for 30 Day Simulation, with Variability									
	Power Consumption (kWh)	Fuel Consumption (gal)	Potable Water Consumption (gal)	Gray Water Waste Production (gal)	Black Water Waste Production (gal)	Power Generated (kWh)	Potable Water Storage (gallons)	Gray Water Waste Capacity (gallons)	Black Water Waste Capacity (gallons)	Personnel (person-hrs)
Δ Total	368.186	11,229.31	46,090.927	15,969.31	23765.18	173,417.50	0	0	0	19,609.95

This table shows large differences between the expected consumption/production and that observed in the model. However, when we check the individual systems, we can verify that the amount being used is driven by the number of personnel on the base during each time interval. The difference in person-hrs between the model without variability and the model with variability is shown in the last column of the table and is based on the number of personnel present during each time interval.

This fully accounts for the differences seen in potable power consumption, water consumption, gray water waste production, and black water waste production. Note that the fuel consumption and power generated is slightly lower (i.e. the difference is larger) for this SoSAT model than in the one where variances were minimized. We are now seeing slight differences due to the use of the stochastic elements within the SoSAT model. This variation is an important component of having simulations mimic real life, where system operations are non-deterministic.

5.7 Baseline Simulations and Results

The BCIL baseline models were run for 720 hours (30 days) to simulate the use of the camp by 150 personnel over the course of a month. 100 trials were run and the results presented represent the mean value of those 100 trials unless otherwise noted. Results are presented in graphical format, though tabular output is also produced by the SoSAT model.

5.7.1 Availability Over Time Results

After the simulation is run, the availability of all component systems over time can be analyzed. These results can be looked at for individual systems, as well as at various levels of aggregation. Results and analysis are presented here for the entire camp, the Level III containerized systems and other aggregated groups, as well as for select individual component systems.

The first "availability over time" result of interest is that of the entire base camp. Given all of the interconnections among component systems, differing operating scenarios, system failures, resource consumption, etc., we want to know if the camp is able to function successfully over time or if adjustments may need to be made. The availability over time of the camp is given in Figure 32 below.

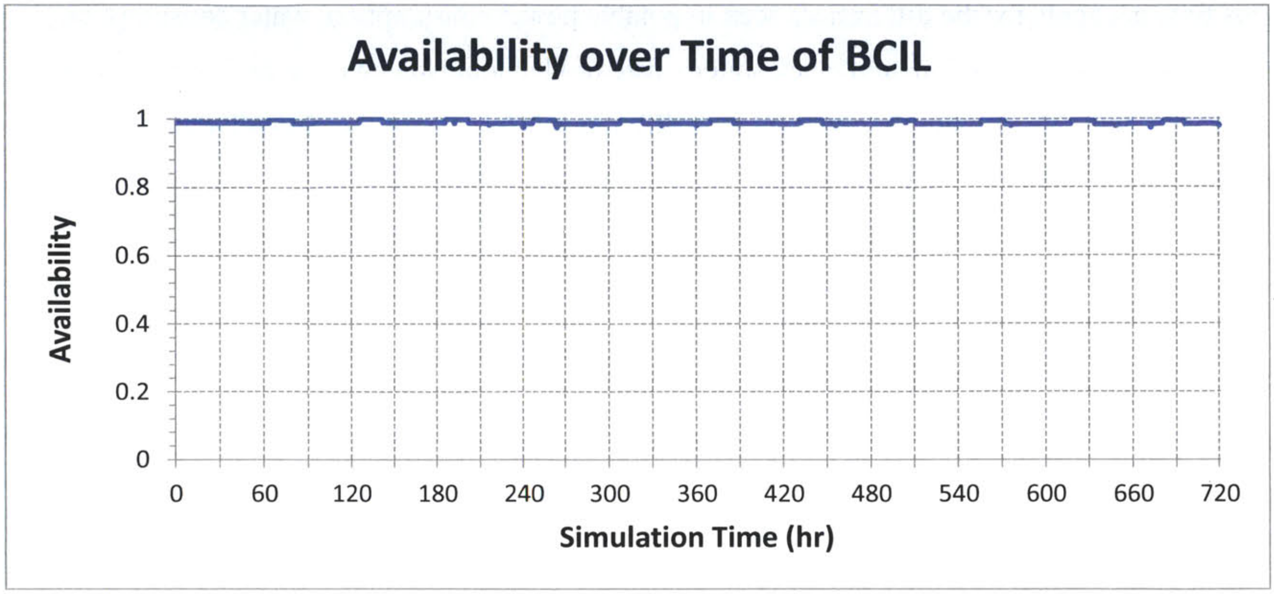


Figure 32. Availability over Time, BCIL

As seen in the plot, the availability of the base camp over the entire simulation remains close to 1.0, or 100%. The cause of the slight variations in availability will become clear as we analyze other component systems. The high availability of the base camp corresponds to our knowledge of the physical base camp in reality, which we already know is able to function over time under normal operating conditions. The availability results presented are the averaged values of the 100 trials run for the simulation. Results can also be viewed by individual trials. The availability over time results for each of the 100 trials are shown below in Figure 33.

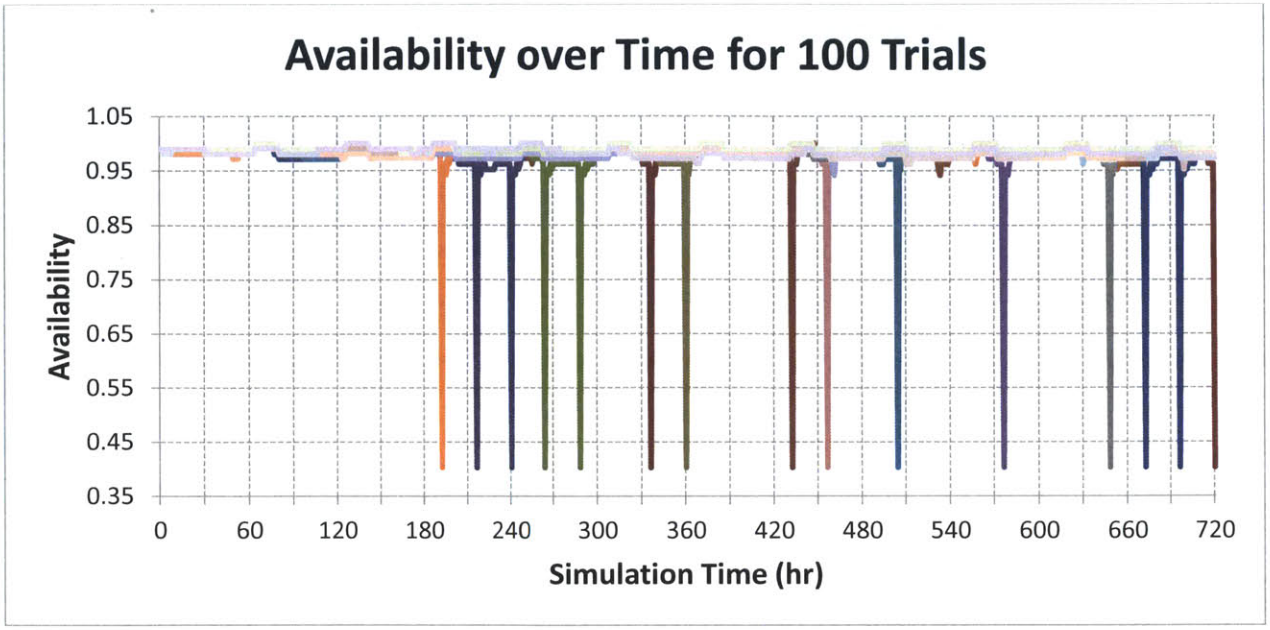


Figure 33. Individual Trial Availability Results

The most likely source of the variability in the base camp availability is the micro grid, since the majority of the component systems in the camp require electricity to operate. The availability over time of the generators is given in Figure 34. The results shown are the average of all the generator component systems that comprise the micro grid.

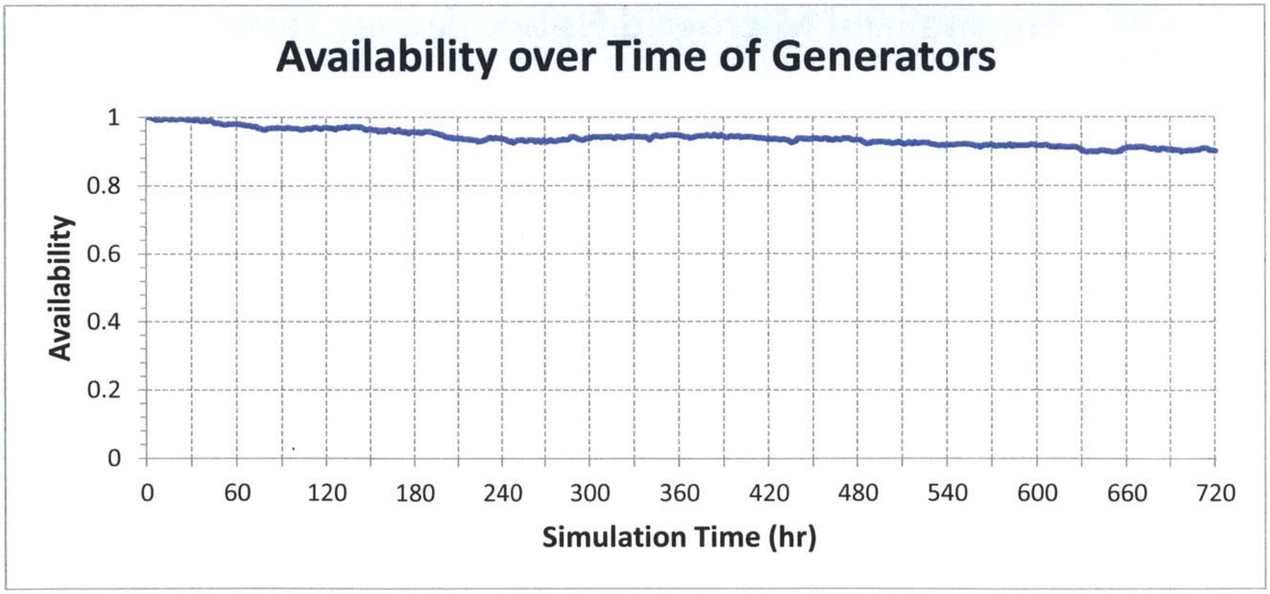


Figure 34. Availability over Time, Generators

The availability of the generator systems is clearly low (around 90%) as the simulation progresses. However, this is not a clear indication of how it affects the performance of the base camp and its component systems since this is the availability of the generators *individually*, rather than when they are combined to form the micro grid. Implementing the micro grid allows electricity to be distributed to the power-consuming component systems, even if generator(s) have failed and are temporarily unavailable. Because many component systems are operating less than 24 hours a day, the energy demand is almost always less than the peak demand would be if all the component systems were operating simultaneously.

A more appropriate way to judge the effectiveness of the micro grid is to look at the micro grid network over the course of the simulation. By looking at the kW requested by the component systems and the kW delivered by the micro grid, we can determine how well the micro grid is meeting the camp's energy requirements. These quantities are shown in Figure 35 with a zoomed-in version shown in Figure 36 for each time step of the simulation. kW usage at each time interval was normalized by dividing the quantity used by the peak demand energy demand for the base camp.

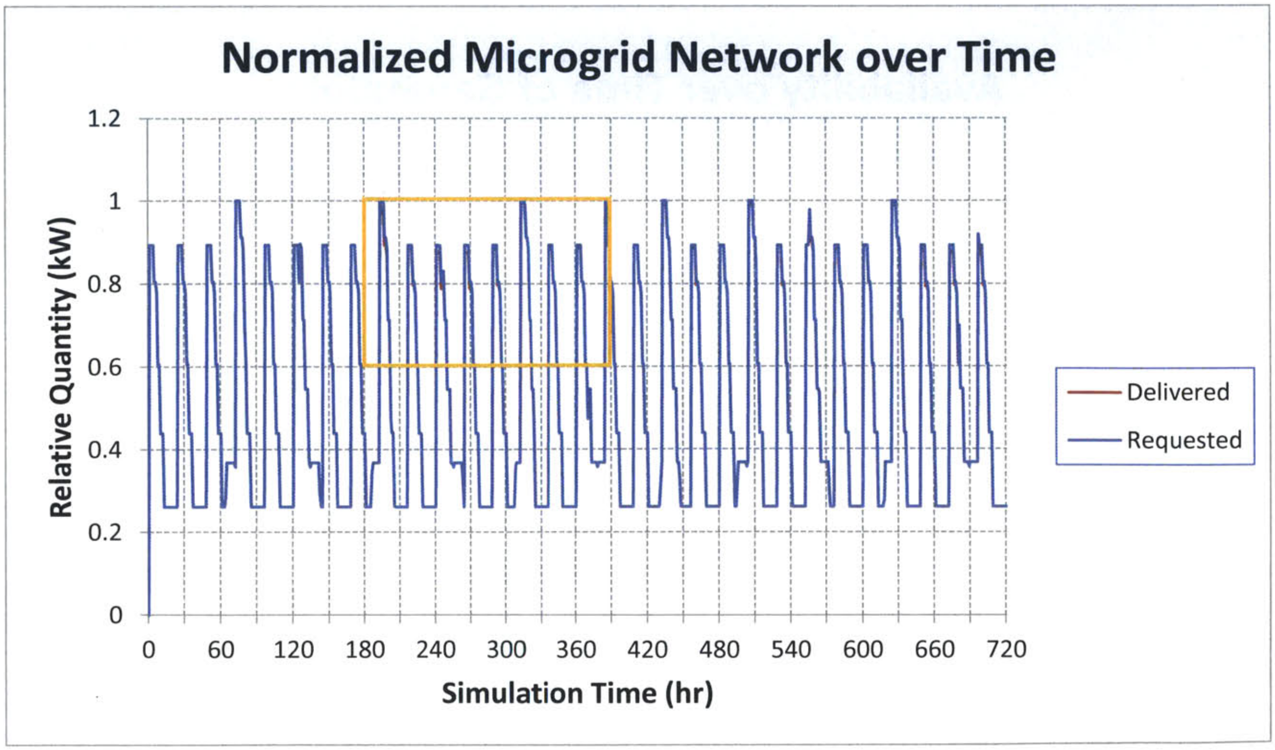


Figure 35. Micro Grid over Time

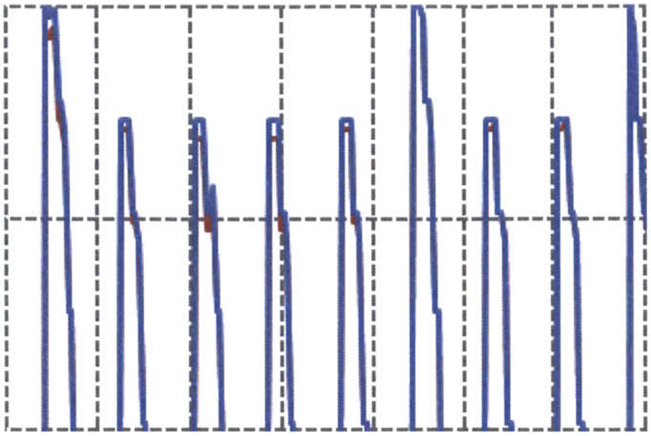


Figure 36. Zoomed View of Micro Grid Results

From the plot of the micro grid network over time, we see that the amount of energy requested was fulfilled at almost every time step of the simulation. Only during a few intervals did the quantity delivered by the micro grid fall short of the amount required. These discrepancies are difficult to see in the main plot, so a zoomed in version of the area within the yellow box of the main plot is given for clarity. Differences between the amount required and the amount delivered result in the temporary unavailability of certain component systems, which will be seen in later results. This also accounts for the slightly reduced availability seen at the base camp level.

We can also look at the availability over time results for the containerized systems and the main groupings of component systems of interest. The first result of interest is the kitchen containerized system with the attached serving/dining tent. This result does not contain the water-consuming kitchen component system. The customer did not want any of the component systems to be constrained by the water supply in the baseline model. Water is therefore delivered frequently enough to prevent any of the systems from experiencing a water resource shortage. With the flexibility of the SoSAT model, the water supply assumption can always be changed in the future if stakeholders desire a sensitivity analysis of how often additional water supplies need to be delivered to maintain functionality of component systems. The results for the water-consuming portion of the kitchen are shown in Figure 37 and the results over time for the kitchen and dining component systems are shown in Figure 38.

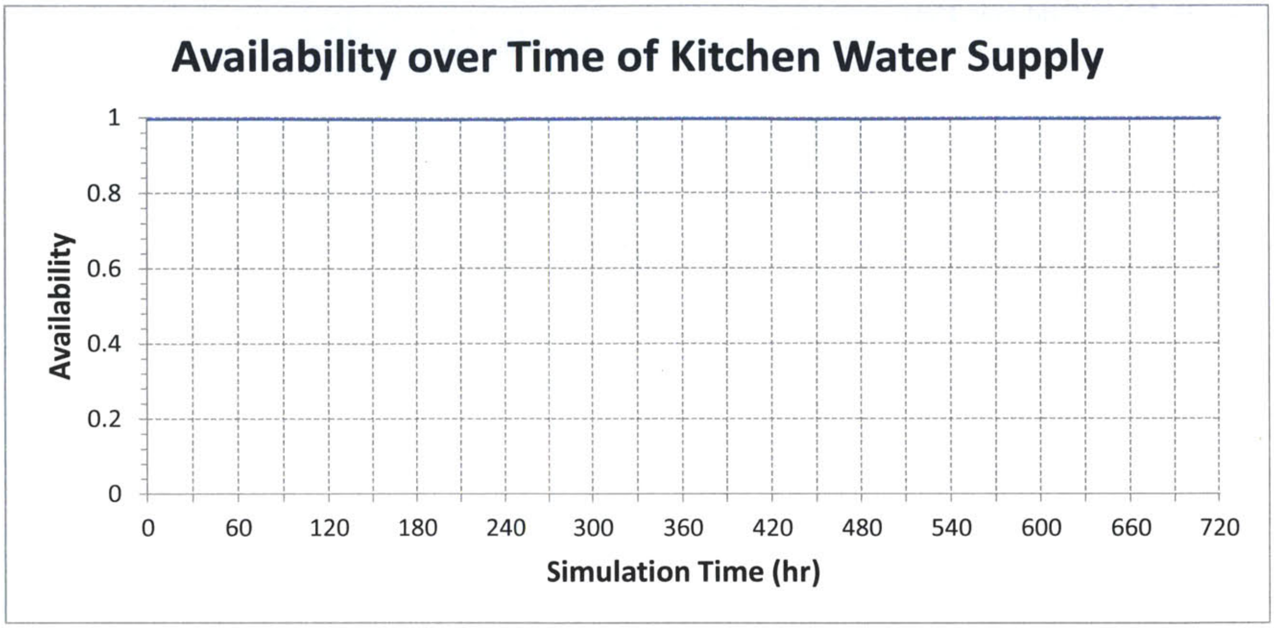


Figure 37. Availability over Time, Kitchen Water Supply

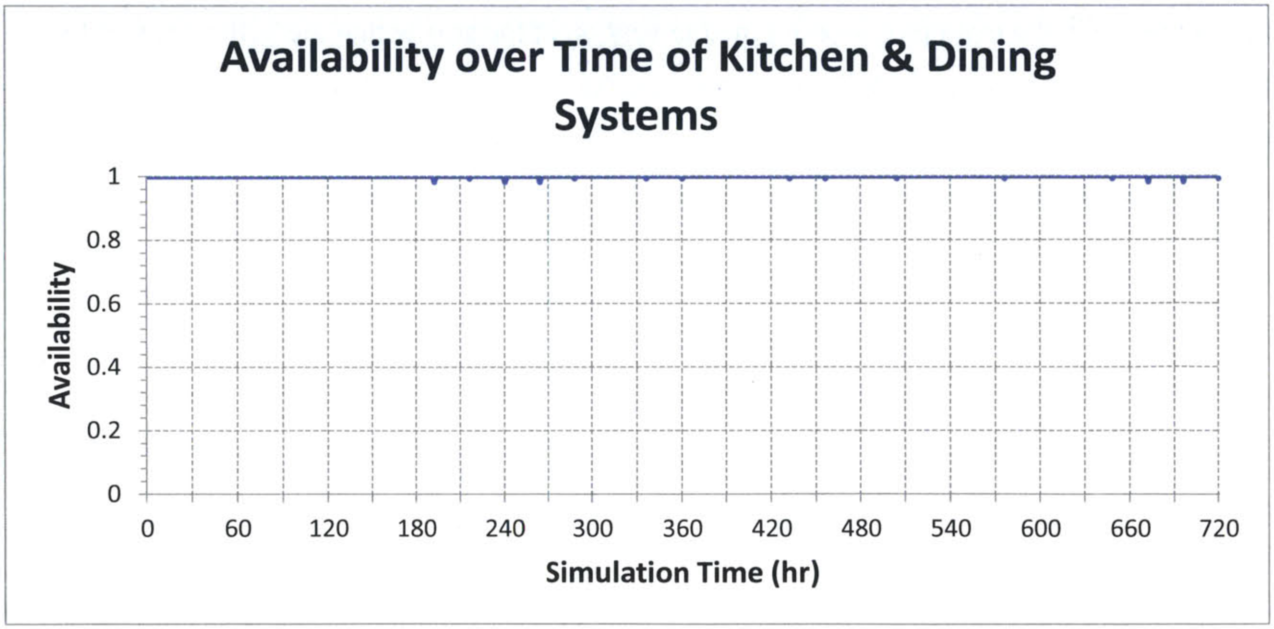


Figure 38. Availability over Time, Kitchen

As described, the kitchen water supply is always functioning at 100% since the component systems are assumed to never experience potable water shortages. For the other kitchen and dining component systems, we see that there are slight decreases in availability over time. These correspond to the time intervals when the amount of electricity provided by the micro grid failed to meet the amount required by the base camp power-consuming systems. The impacts to availability are slight and temporary.

Availability over time results are also examined for the latrine component systems. Once again, we omit the water supply since we have already demonstrated that component systems are not resource constrained by the availability of potable water. In Figure 39, the same pattern of availability is seen for the latrine containerized systems as was seen with the kitchen and dining systems. Availability is at 100% for the majority of the simulation with slight decreases in availability caused by the micro grid. The magnitude of the decreases is slightly larger for the latrine component systems than for the kitchen component systems.

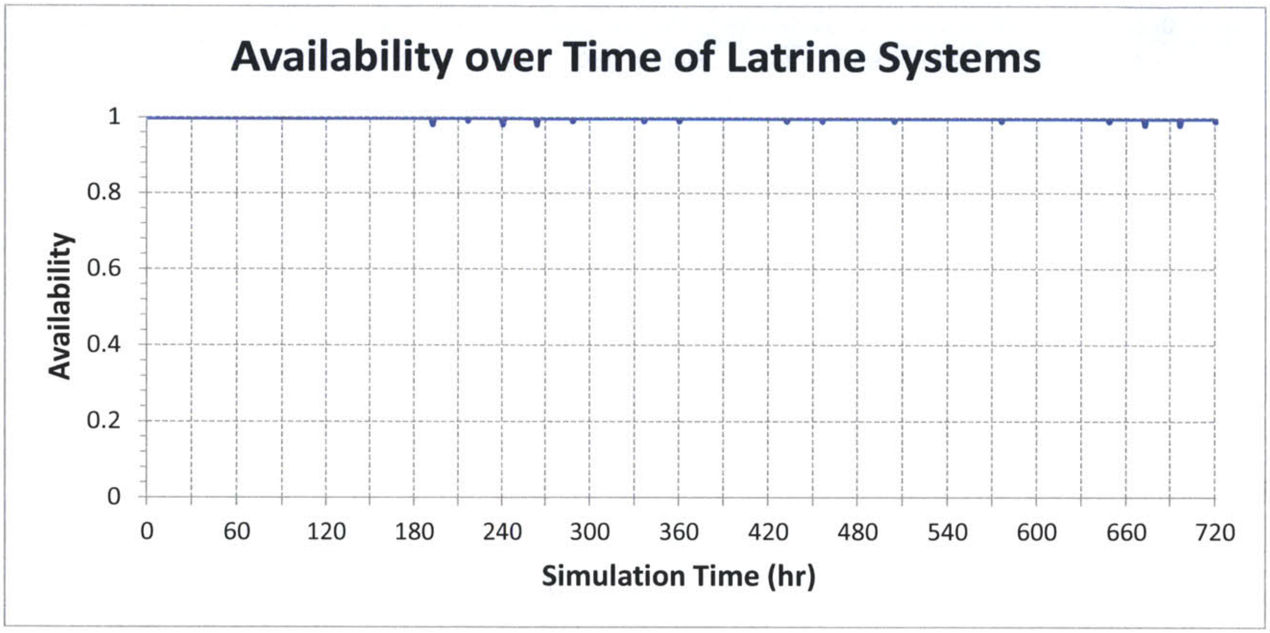


Figure 39. Availability over Time, Latrines

The laundry containerized system is the next system to be analyzed. Once again, we see approximately the same pattern of availability, which can be seen in Figure 40.

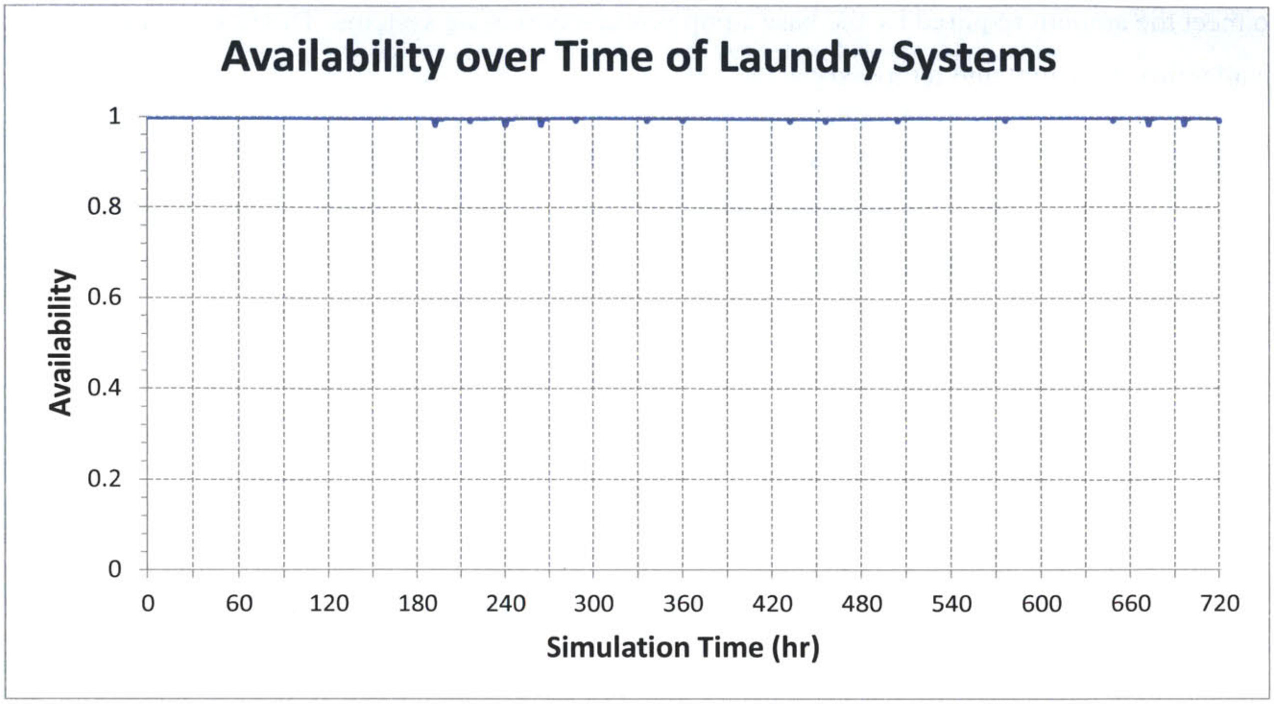


Figure 40. Availability over Time, Laundry

Unlike the results for the kitchen and latrines, the results for the laundry containerized system must include the water-consuming component system represented by the washer. While the washer consumes potable water, it also requires electricity to operate, meaning that its availability will potentially be impacted by changes in the availability of the micro grid. The availability of just the washer system is shown below in Figure 41. As predicted, the availability of the washing machine decreases during the time intervals when the micro grid does not generate enough electricity to meet the demand of the camp.

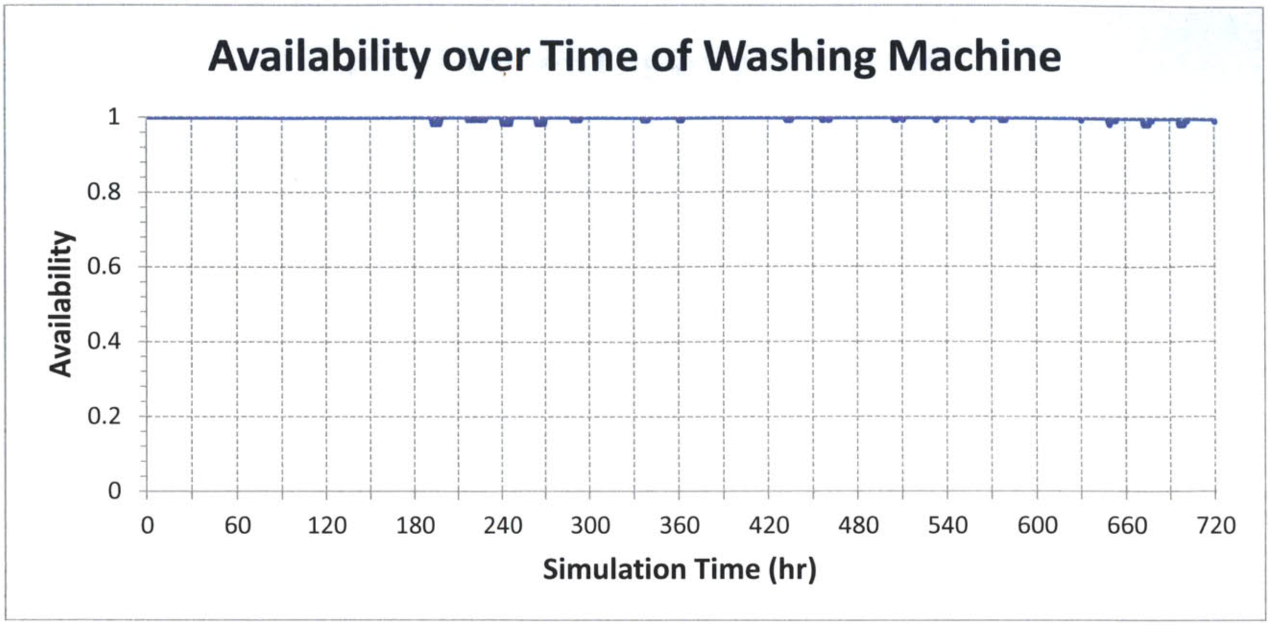


Figure 41. Availability over Time, Washer

Continuing with the analysis, the same pattern in availability is also observed for the shower containerized systems and the tents, whose availability over time results are shown in Figure 42 and Figure 43, respectively.

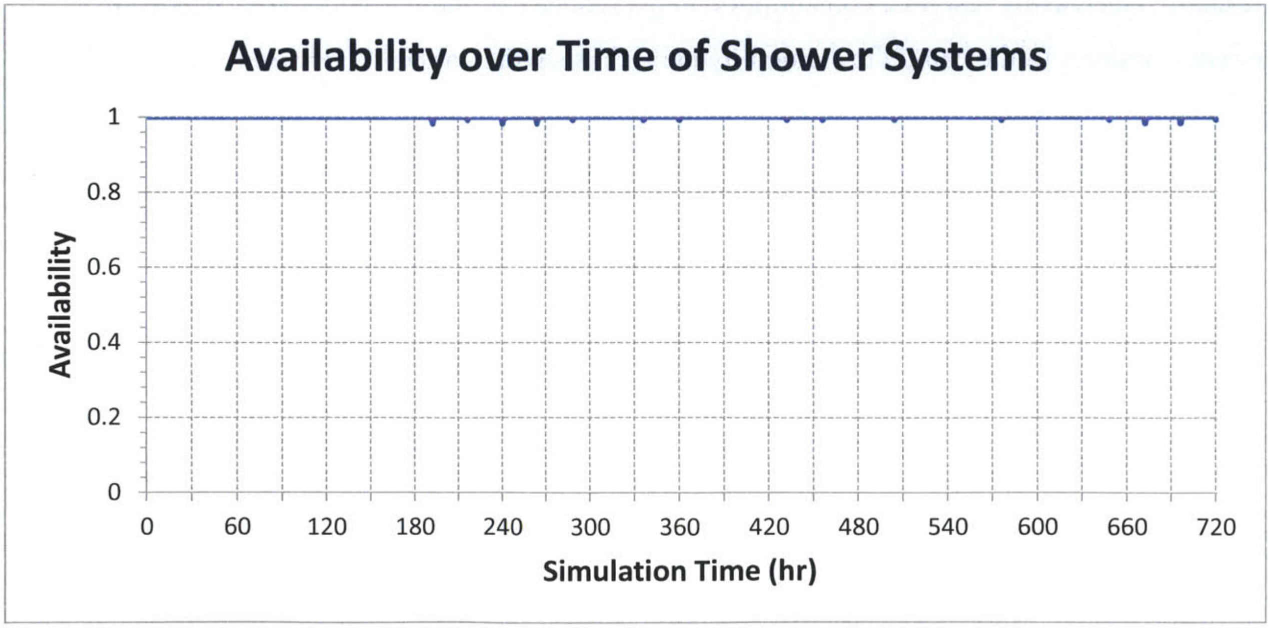


Figure 42. Availability over Time, Showers

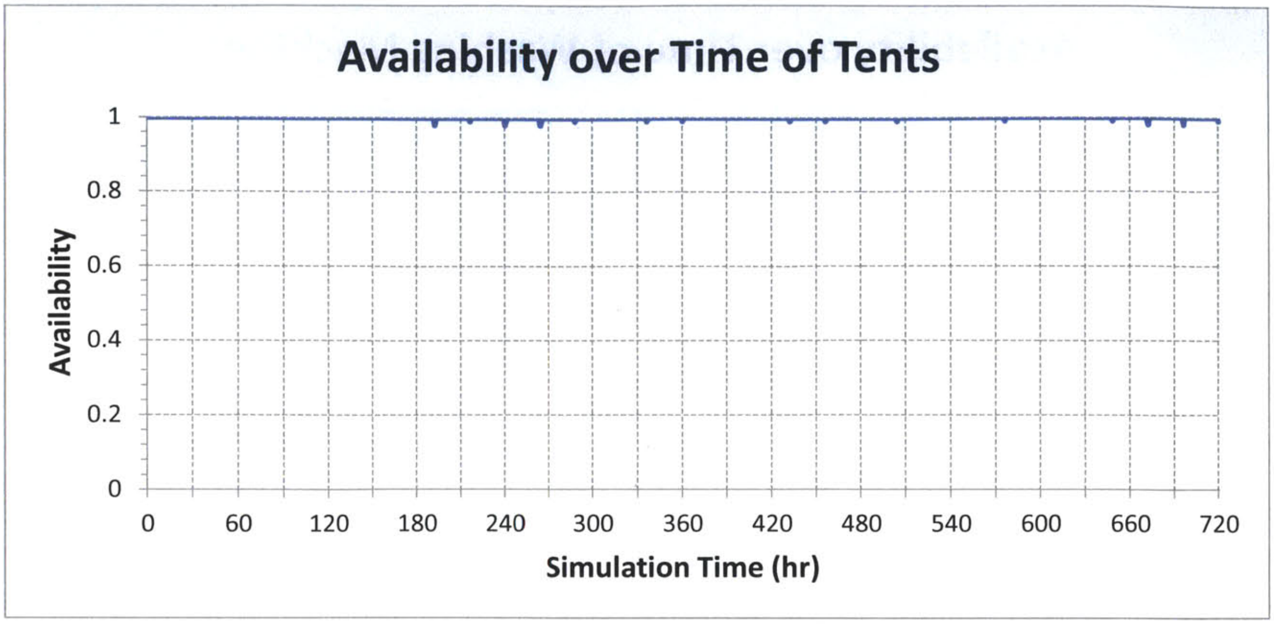


Figure 43. Availability over Time, Tents

The last system of interest is the shower water reuse system (SWRS) which processes gray water from the showers and filters out contaminants to produce potable water. The potable water is then reused by the shower systems to decrease the required amount of potable water delivered to the camp. Decreasing resource consumption helps reduce the number of convoys needed to deliver resources to the camp during deployment, thus saving valuable fuel and reducing the frequency of potentially dangerous outings for personnel. The SWRS is capable of recycling 75% of the gray water generated by the shower systems. Its availability over time is shown in Figure 44.

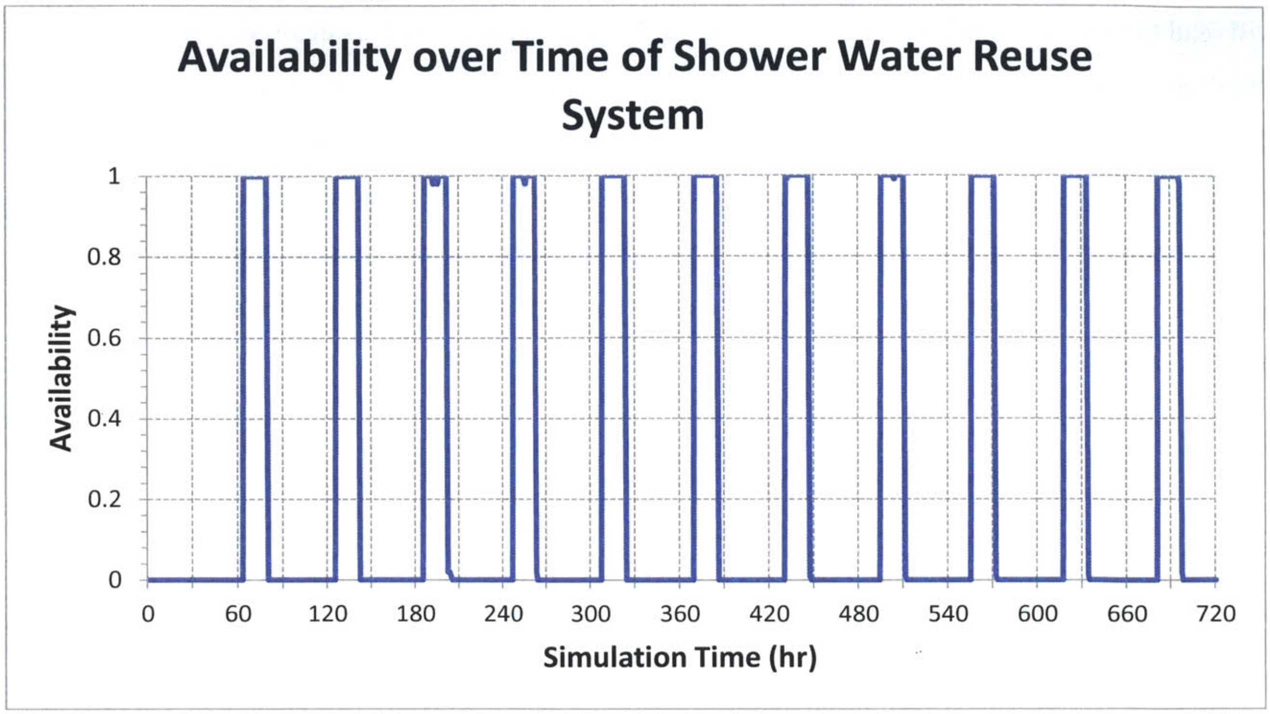


Figure 44. Availability over Time, SWRS

The SWRS is only considered operational when there is enough gray water to process. A full batch of gray water must be stored before the SWRS will turn on and begin filtering the water. Each batch required 8 hours of processing time. There are two containerized shower systems being used approximately equally, so two batches of gray water are available around the same time. Therefore, the SWRS operates for approximately 16 hours at a time to process the two batches. The time while the SWRS is processing the gray water is shown as ~100% availability in the plot. According to the plot, we can determine that the SWRS goes through 22 processing cycles during the 30 day simulation.

It is also important to realize that the way individual component systems are implemented within the SoSAT model can impact the output for certain results. For the SWRS specifically, the system is considered to have 0% availability unless the system is operating. This is because the system is not defined by a specific operating scenario telling the model how many hours per day it operates, but rather by the inventory levels of the gray water collection containers it processes water from. Earlier we looked at the availability over time of the BCIL structure, which is the aggregation of the availability over time of all the component systems—including the SWRS. Yet including the SWRS in this aggregated result may not be an accurate representation, since it will lead the viewer to believe the base camp isn't functioning as well as desired. For results which more accurately capture BCIL performance, the availability of time results for the SWRS should only be included in the aggregated BCIL availability for time intervals during which the SWRS is processing gray water. Correcting the BCIL level availability over time results for the SWRS component system implementation gives the availability over time seen in Figure 45. The original BCIL availability over time results is repeated directly below in Figure 46 for comparison.

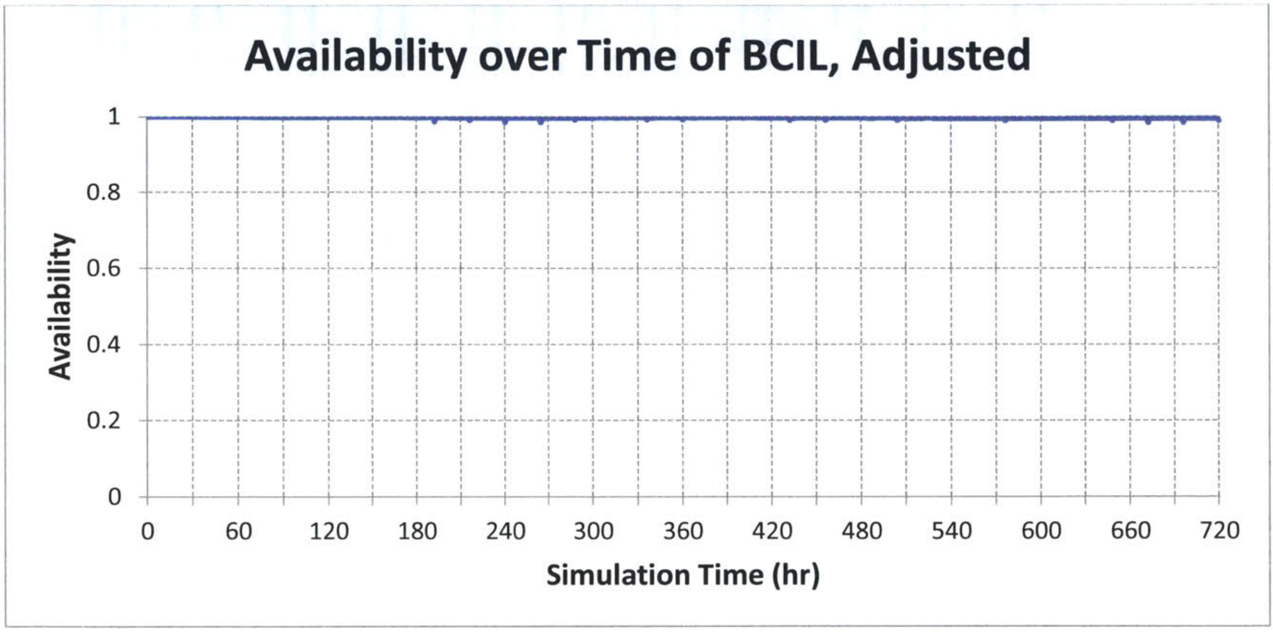


Figure 45. Availability over Time of BCIL, Adjusted

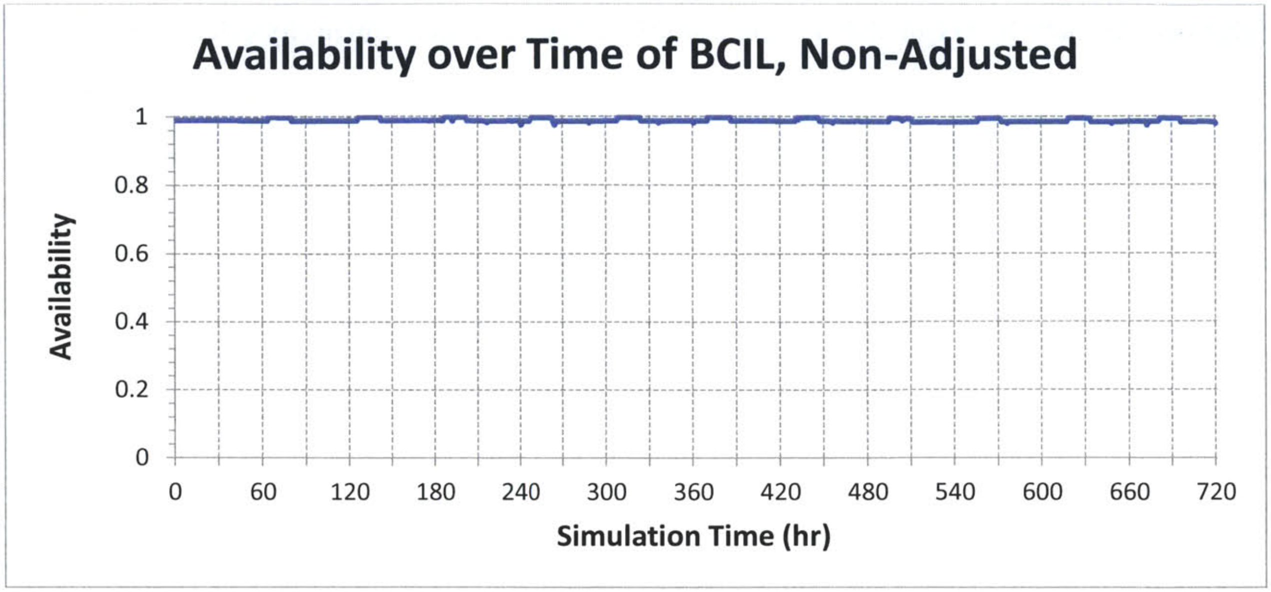


Figure 46. Availability over Time of BCIL, Non-Adjusted

From the comparison of the two aggregated availability over time results, we can see that even the availability of a single component system can change the overall availability of the entire structure. When building SoS city models, care should be taken to represent systems as consistently as possible. The impact of using non-standard representations should always be examined and output should be adjusted if necessary to prevent incorrect results. Any availability over time results aggregated to the base camp level will be adjusted using the procedure described for the remainder of the case study.

5.7.2 Functional Availability over Time

We can also look at "availability over time" in terms of the functions the base camp provides, rather than looking at groupings of component systems. Defining availability by functions helps to provide stakeholders and those with non-technical backgrounds with a clearer picture of how the camp is functioning. Each function is comprised of a component system or grouping of component systems, whose operability determines the "availability" of the function. These functional availability results provide us with the same results we would observe by looking at the availability over time of the component systems comprising the function.

Functional availability is calculated using a similar equation to the availability calculation. Time operating and time down now refer to the function rather than component systems. The equation for functional availability can be defined as

$$functional\ availability = \frac{time\ operating_f}{time\ operating_f + time\ down_f} \tag{14}$$

where $time\ operating_f$ is defined as the amount of time that the function is operating and $time\ down_f$ is defined as the amount of time the function is down due to failures, resource depletions, etc. Each of these quantities may correspond to a grouping of multiple component systems, since each function may comprise one of more component systems.

For example, consider the function "Provide Electrical Power." This function's availability is determined by the availability of the generators tied to the base camp's micro grid. The availability over time of this function is shown below in Figure 47. Notice that this is the exact same result we obtained from looking at the availability over time of the generators as shown in Figure 34. Functional availability provides perhaps a more intuitive way of presenting results, though the results themselves are identical to their component system results presented earlier.

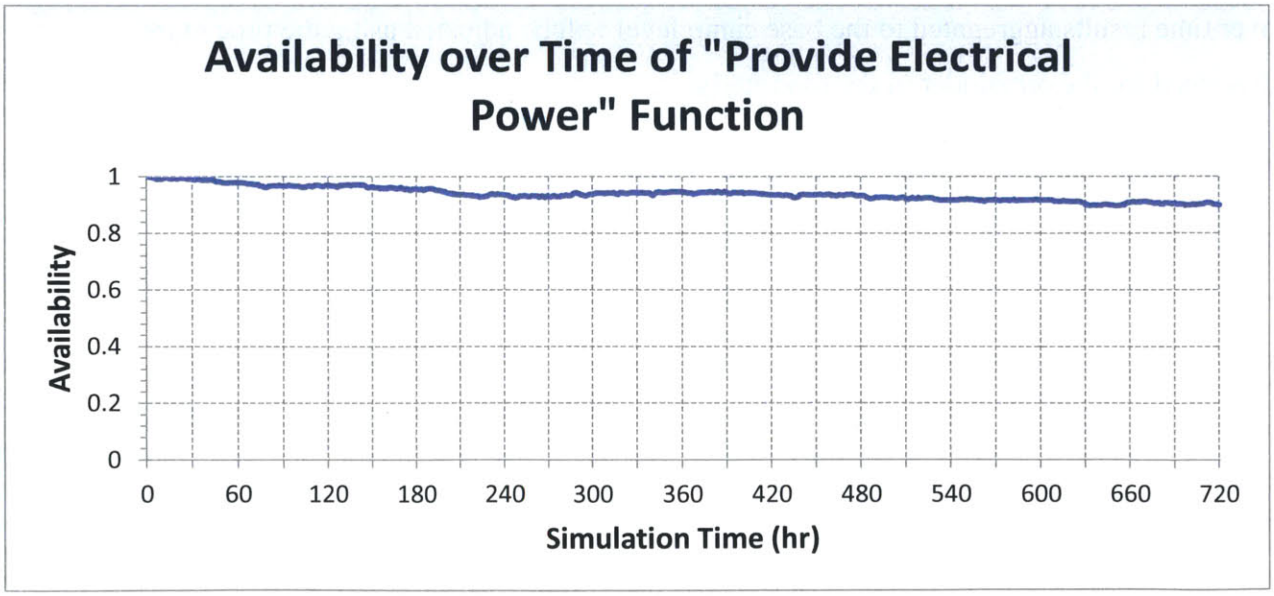


Figure 47. Functional Availability over Time, "Provide Electrical Power"

Since results for component systems and groupings of component systems have already been presented, we will focus instead on the mapping between systems and functions. The functions used in the BCIL baseline SoSAT model are given in Table 6, along with the component systems that define the functions. Note that the component systems are listed in groupings where appropriate. Containerized systems are comprised of the relevant component systems, such as lighting, outlets, air conditioners, etc. This is equivalent to looking at availability over time results for systems aggregated to Level III of the SoSAT structure for all functions except for "Operability." The "Operability" function depends on all component systems and is equivalent to looking at the availability over time results aggregated to the level of the entire BCIL base camp. When presenting results to stakeholders, care should be taken in deciding which types of results are most useful and whether it makes sense to present results as functions or as their component systems.

Table 6. SoSAT Function Mapping

Function	Component Systems
Collect Black Waste Water	Black Water Collection Blivets
Collect Gray Waste Water	Gray Water Collection Blivets
Provide Assigned Tenant Billeting	Billeting Tents
Provide Electrical Power	Generators
Provide Field Services (Clean Clothes)	Laundry Containerized System
Provide Field Services (Latrine)	Latrine Containerized Systems
Provide Field Services (Personal Hygiene)	Shower Containerized Systems
Provide Subsistence	Kitchen Containerized System
Store Supply (Fuel)	Fuel Supply System
Store Water (Potable)	Potable Water Blivets
Operability	All Component Systems

5.7.3 Consumption and Generation

Often we are interested in the use of resources over the course of the simulation. Especially in resource-constrained environments or in situations where obtaining additional resources is costly or dangerous, the amount of each resource used and the remaining inventory is crucially important to understand. Planners, managers, etc. need to plan how to distribute resources in the most efficient manner possible, while still maintaining operability and quality of life. For the BCIL case study, we have already shown that availability requirements of the component systems are adequate to sustain camp operations. Here we show how resources are used and additional insight that can be gained from further exploration.

All quantities for consumed and generated have been normalized for this case study. The goal of this thesis is to illustrate how systems of systems modeling is useful and applicable to city systems. Therefore we are more interested in comparing relative quantities and seeing how various quantities are generally impacted by the interconnectedness of the systems, rather than in the numerical quantities themselves. Normalizing the values aids the reader in quickly comprehending trends and changes in different resource consumption and production rates. For resources consumed or produced by component systems, including resources measured through supply connections and networks, the quantities were normalized by dividing the quantity at each time interval by the maximum time-interval quantity observed during the simulation as shown in the following equation,

$$relative\ x_i = \frac{x_i}{max(x)} \qquad (15)$$

where x_i represents the original resource quantity consumed or produced during time interval *i* and *max(x)* represents the maximum resource quantity observed at any time interval during the simulation. All inventory levels are represented as percentages of the total inventory capacity.

Potable water is undoubtedly a key resource for any human habitation, providing water for cooking, hygiene, and laundry. In the base camp, component systems quickly become inoperable without frequent water resupplies. The potable water consumed by component systems of the BCIL is shown in Figure 48 for every hour of the 30 day simulation. Consumption rates vary due to both changes in personnel on base and also the individual operating profiles of the various systems since some systems operate only a subset of the day.

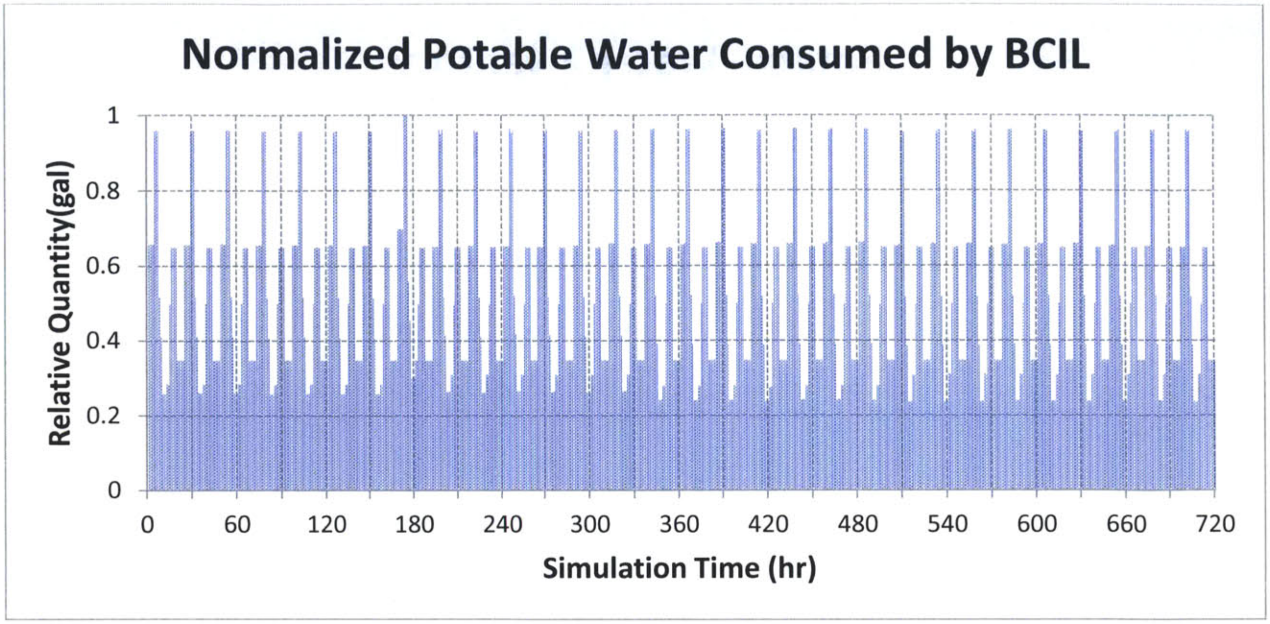


Figure 48. Potable Water Consumption

While knowing the consumption rate of systems is important, it provides only one half of a crucial equation. To fully understand how the base is operating, we must also examine what quantity of the available resource inventory remains throughout the simulation. If resource inventory continually dips toward zero, the operating structure of the base may not be robust enough to function under conditions that are slightly varied from the baseline conditions. Particularly in unknown operating conditions or unstable environments, resource inventory becomes a primary concern. The potable water inventory for the BCIL is given below in Figure 49.

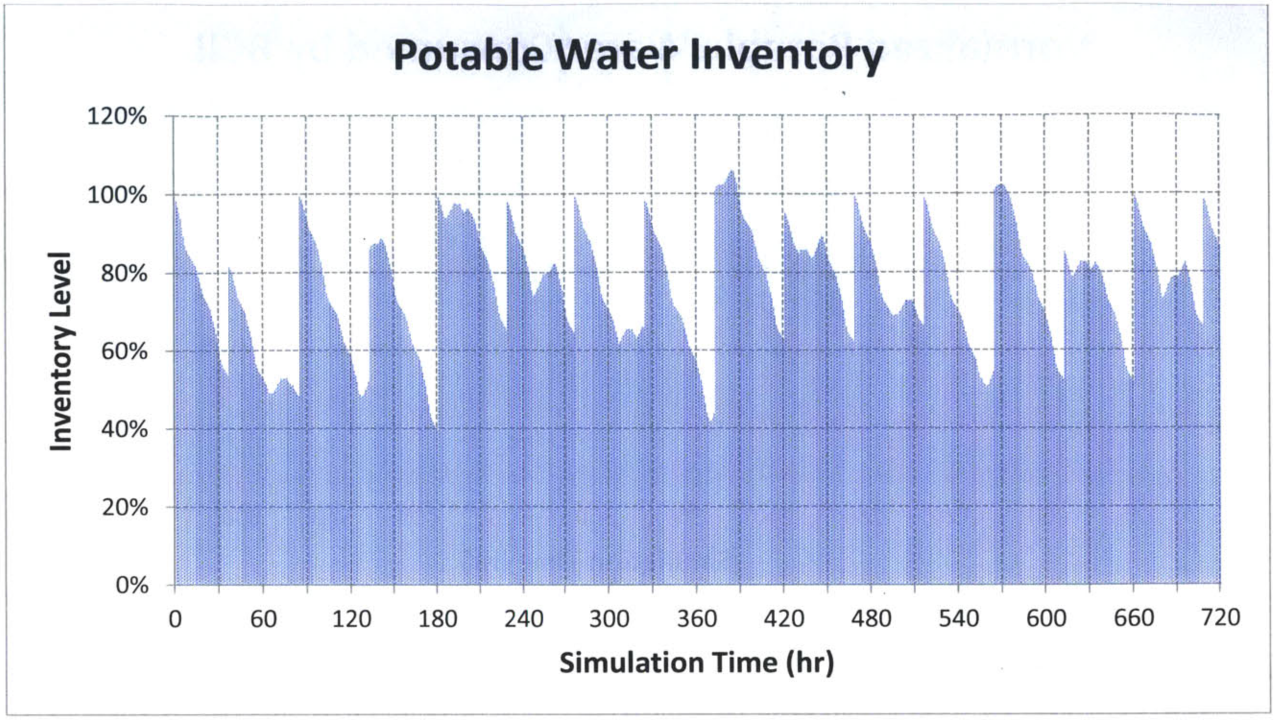


Figure 49. Potable Water Inventory

The potable water inventory is augmented throughout the simulation by deliveries of additional potable water and by potable water produced by the shower water reuse system which filters gray water produced by the showers into potable water. Resupplies from delivery are seen in the figure as steep vertical lines, where the supply is dramatically increased during a single time period. Resupply from the shower water reuse system is not as apparent, since potable water is fed back into the shower systems continuously during the eight hour processing time. However, during two points of the simulation, all potable water storage containers are at capacity and the potable water produced by the shower water reuse system remains in the system until it can be transferred into the storage containers. This accounts for the two areas during the simulation where the inventory level appears to be above 100%. From the plot, we can see that potable water inventories remain high at all time intervals of the simulation. The effect of increasing the time between potable water deliveries will be examined later during a sensitivity analysis.

Another key resource for the base camp is fuel, which supplies the generators that comprise the micro grid. Every containerized system and grouping of component systems at least partially relies on the micro grid to provide electricity. Without fuel for the generators, the generators cease to operate and the micro grid fails, essentially taking down the operability of the entire camp. The fuel consumed by all of the BCIL component system during the 30 day simulation is shown in Figure 50. As was seen with base potable water consumption, base fuel consumption also varies by hour. While the number of personnel on base changes, so does the utilization of component systems. This in turn changes the base power demand, changing the number of generators operating to meet that demand. Fuel consumed each hour of the simulation is driven by the number of generators operating each hour to fulfill the demands placed upon the micro grid by the component systems.

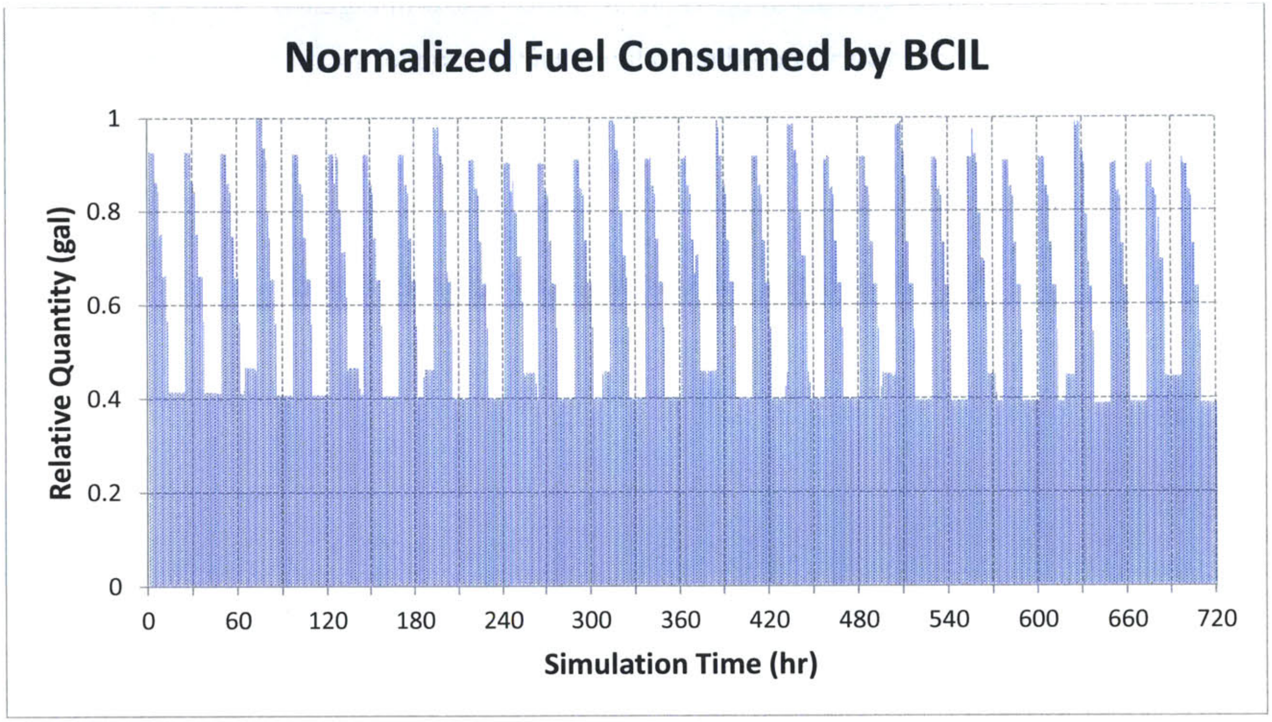


Figure 50. Fuel Consumption

For the BCIL case study, the customer was not concerned with tracking how much fuel remained in inventory. Regardless the SoSAT model was set up with a set fuel supply and the ability to request deliveries of additional fuel once inventory levels decrease to a certain point. Currently the fuel supply is large enough to last the duration of the simulation, and delivery requests are not initiated. Should the stakeholders wish to include fuel inventory considerations in the future, the initial fuel supply can be lowered, and fuel deliveries will take place at specified intervals. Inventory levels can then be examined to determine whether the delivery schedule meets the fuel demand of the base.

Besides looking at consumption or inventory quantities, we can also look at how much of a given entity is produced by the base camp. Notably, black water and gray water are produced through normal operation and must be collected in storage containers. From the perspective of the base camp, black water is waste water that is considered too contaminated to filter and reuse. Gray water contains contaminants that can be filtered out to return the water back to a potable water state. Only the gray water produced by the shower containerized systems is processed for reuse. All other gray water remains in a waste water state. The black water and gray water generated each hour by the component systems of the BCIL is shown below in Figure 51 and Figure 52, respectively. Generation rates vary according to how many personnel are on base during each time interval and which component systems are operating. From the figures, we clearly see that the component systems operate in a cyclical fashion over the course of the simulation.

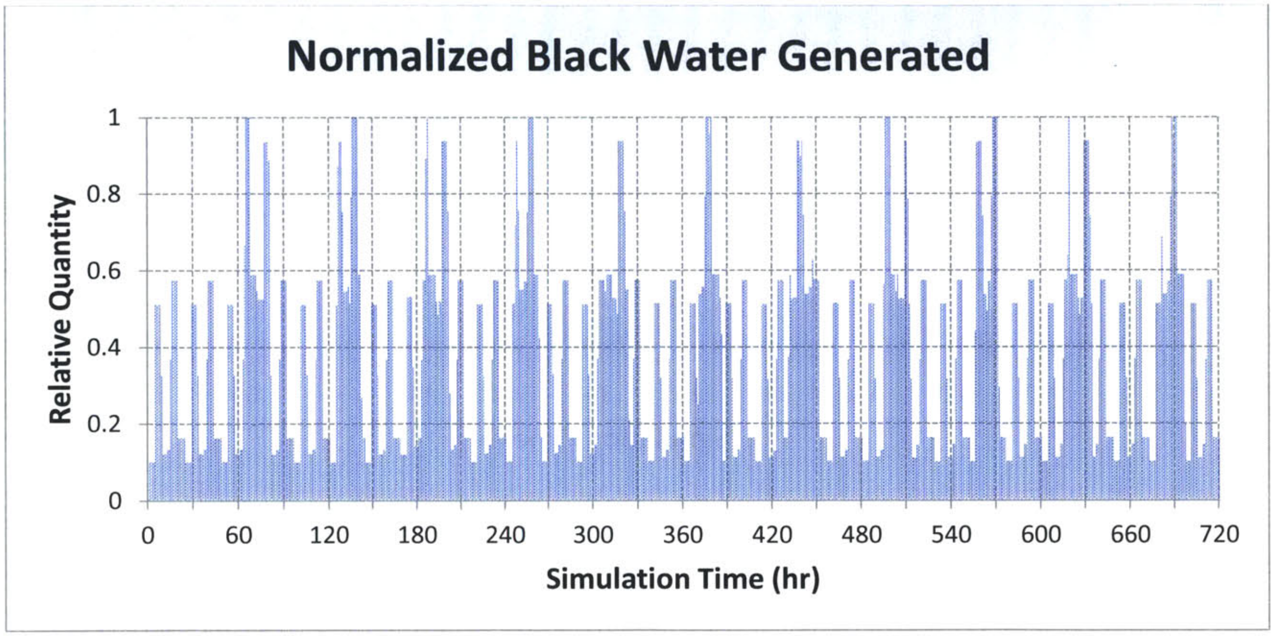


Figure 51. Black Water Generation

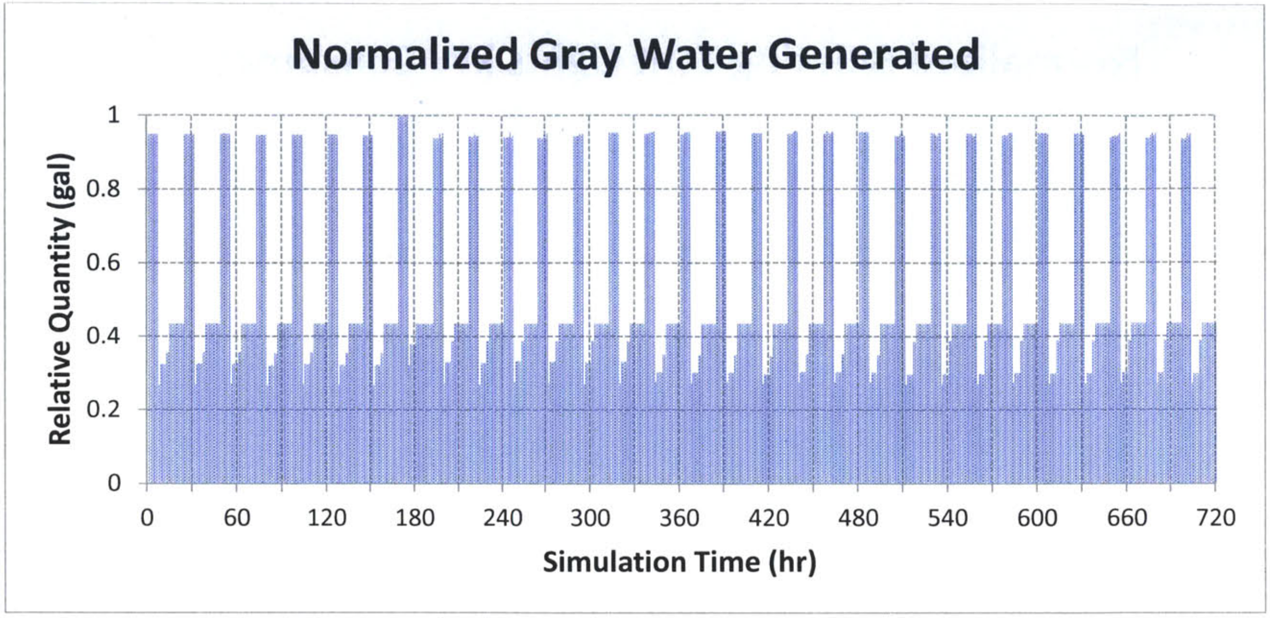


Figure 52. Gray Water Generation

Consumption and generation rates over the course of the simulation are also useful for verifying that the correct scenarios have been assigned to component systems. If a component system requires electricity to operate, and only operates four hours a day, then that system should only be consuming electricity during four hours out of each 24 hour period. Alternatively, if a component system is supposed to operate 24/7 but displays time intervals where the system is not operating, then the cause of that inoperability must be examined.

Consider the lights in billeting tents used for housing personnel. These lights are assigned a scenario where they operate 12 hours a day and are off the other 12 hours. Looking at the kW consumption for these lights verifies the scenario is working as desired, as can be seen in Figure 53. Slight variations in the quantity consumed while operating can be attributed to changes in the availability of the micro grid.

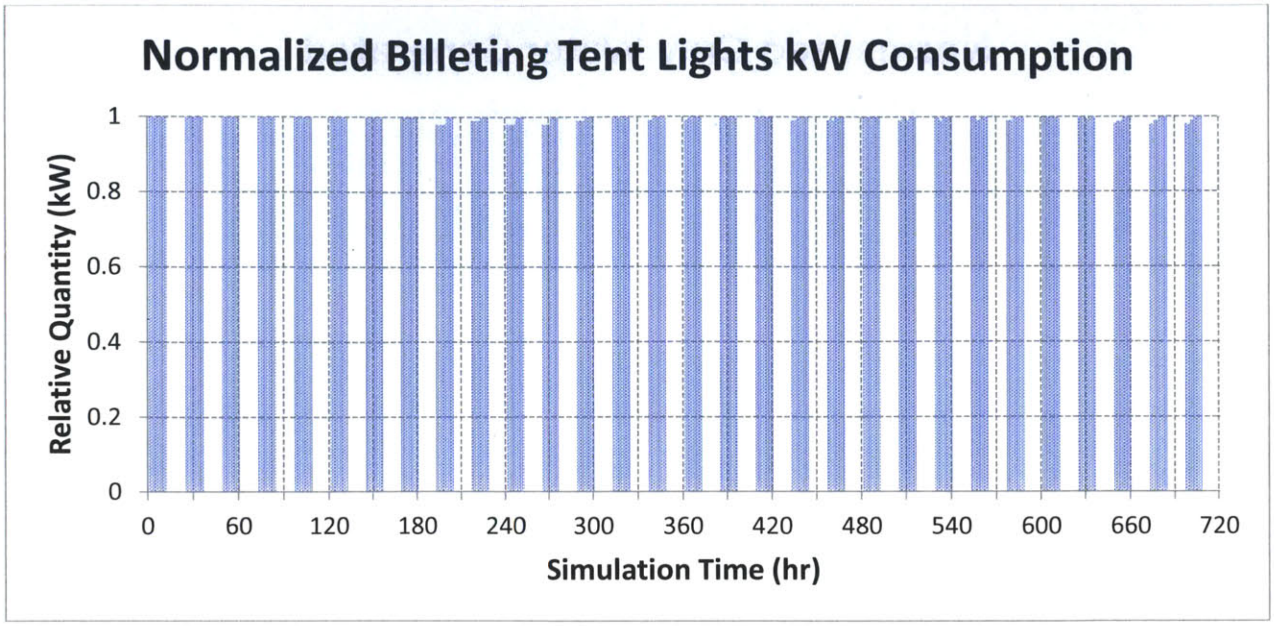


Figure 53. Tent Lights kW Consumption

Meanwhile, the latrines need to be operating 24/7, during which time they consume potable water. Again, we can verify this scenario implementation by looking at the quantity of potable water consumed over the course of the simulation as shown in Figure 54. This also shows how the quantity of potable water consumed changes over the course of the day. We attribute this to the change in the number of personnel on base. The fewer personnel present, the less the component systems will be utilized. We do not observe any periods during the simulation where consumption is zero, which is what we expect for systems operating 24/7.

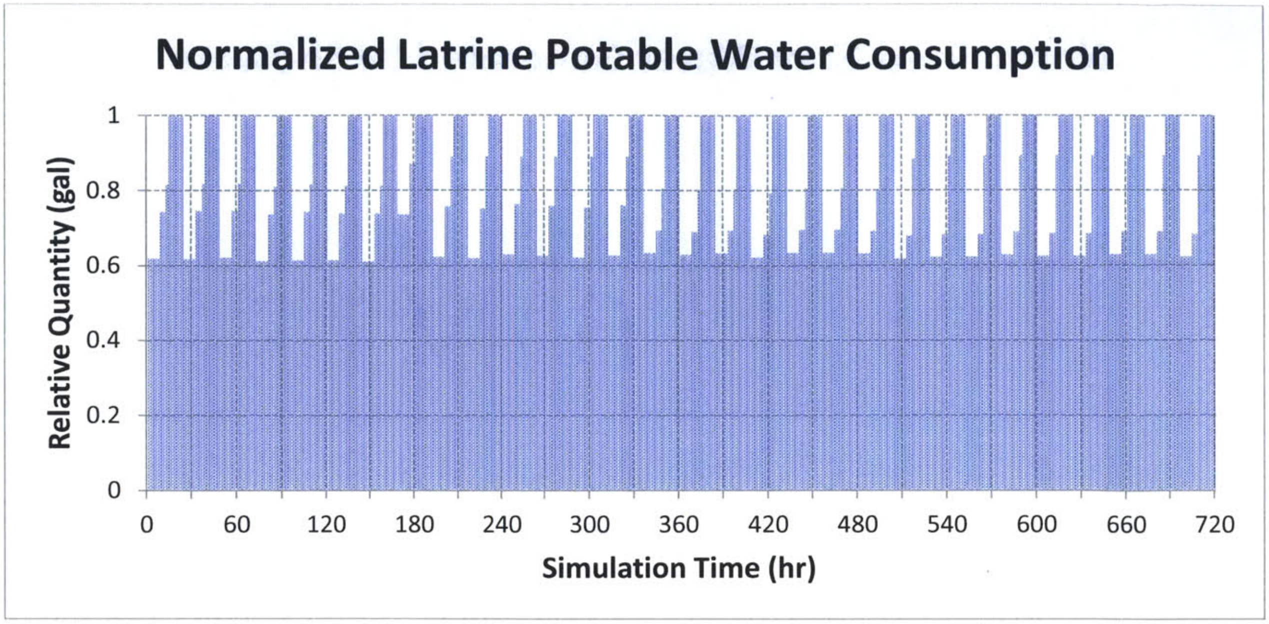


Figure 54. Latrine Potable Water Consumption

Up to this point, the quantities consumed or generated have been obtained by looking at the output from component systems. We can also obtain this same information by looking at the output of the resource networks built into the SoSAT model. The networks are defined at Level III of the SoSAT structure, which is the level that specifies the various types of containerized systems and main groups of component systems such as housing.

For an illustrative example, consider the quantity of gray water produced by the shower containerized systems. Selecting the shower component systems and analyzing the output gives us the gray water generation quantities shown in Figure 55. Once again, the quantity produced each hour varies according to the number of personnel on base and the consequent utilization of the shower systems. Notice how this figure varies from the gray water generation of Figure 52, since we are now selecting only the shower component systems, rather than all gray-water producing systems on the camp.

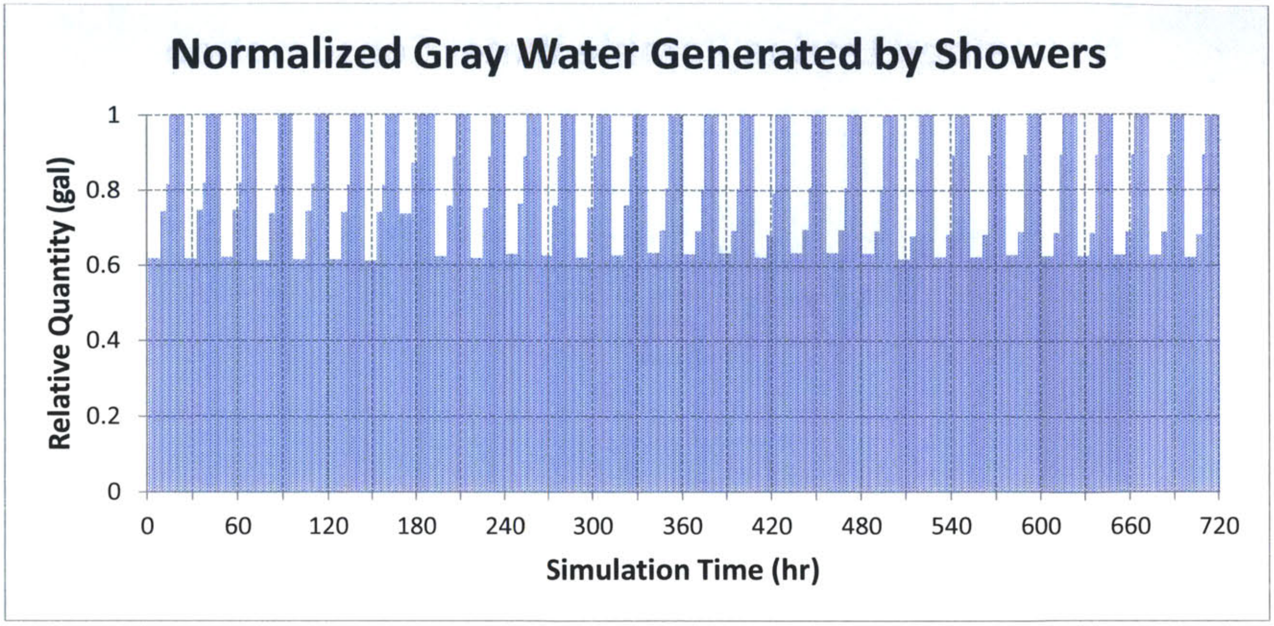


Figure 55. Gray Water Generated by Showers

We can now compare this result to the result obtained from the output of the gray water shower network. The gray water shower network connects the gray water producing component systems from the containerized showers to the gray water collection containers. Looking at the output of the network tells us how much gray water is transferred from the component systems to the holding containers. This result is given below in Figure 56.

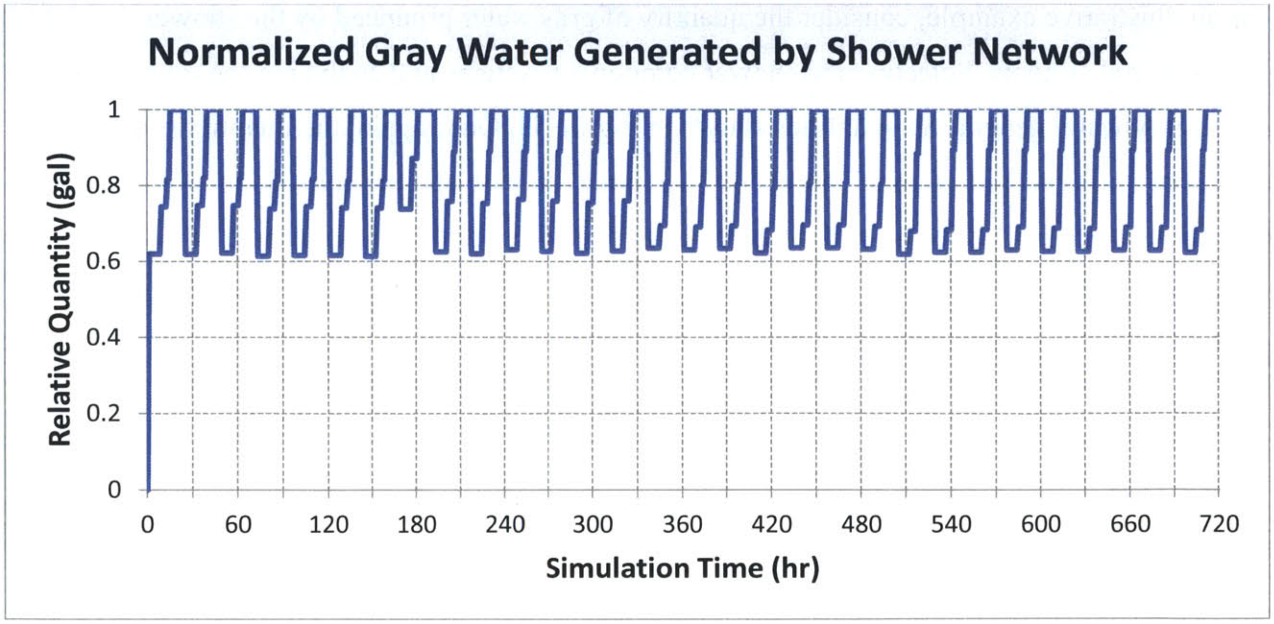


Figure 56. Gray Water Generated by Shower Network

These results can be obtained for all containerized systems as well as for the micro grid and base camp billeting tents. The networks define potable water and electricity delivered to the all of the various containerized systems, and the black and gray water generated by these containerized systems and stored in containers. For the sake of brevity, these comparisons are omitted.

5.7.4 Aggregated Results

Results over time are not always appropriate for all types of analysis. Often the stakeholders want to know the total amount of various consumables which were consumed or generated for the entire simulation. Time based results are cumbersome to interpret in these cases. Besides time-based output, SoSAT also provides aggregated results for the full simulation. These results are useful for quick comparisons, such as the ones used to validate that the model was producing and consuming the correct quantities of resources in 5.6 Validating the Baseline Models.

The first aggregated result of interest is provided in the system state summary. This output allows the user to view the amount of time that individual component systems are operating, operable, and down. The system state summaries are given in Figure 57 below for four systems, including the waste pump for the showers, the washing machine, the oven in the kitchen, and the lights in the billeting tents. Results are shown as a summary of the entire 30 day (720 hour) simulation.

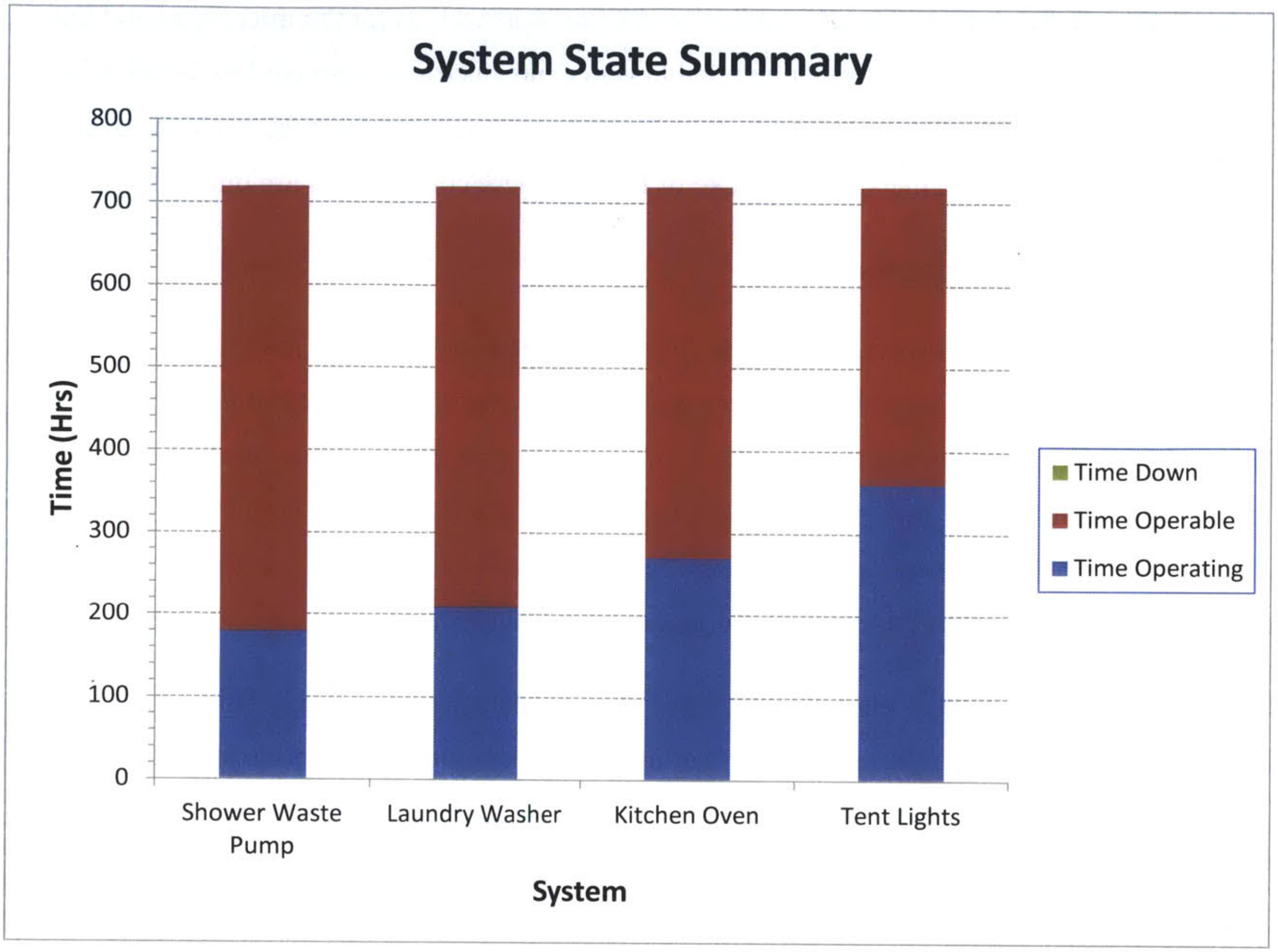


Figure 57. System State Summary

The four systems summarized in the state summary graph all operated on scenarios which are less than 24/7. "Time Operating" is the number of hours the system is on and producing and/or consuming resources. This number should be equal to the number of operating hours specified in the system's scenario, minus the amount of time the system is down. "Time Operable" refers to the number of hours the system is not on, but is still capable of functioning. Time during which the system cannot operate due to failures of either that system or a system upon which that system depends is recorded as "Time Down." These three quantities help give a visual depiction of how the system is operating over the duration of the simulation.

Just as we examined the availability over time of various systems, we can also analyze the aggregated availability of individual systems for the entire simulation. The availability of three of the generators comprising the micro grid is shown in Figure 58.

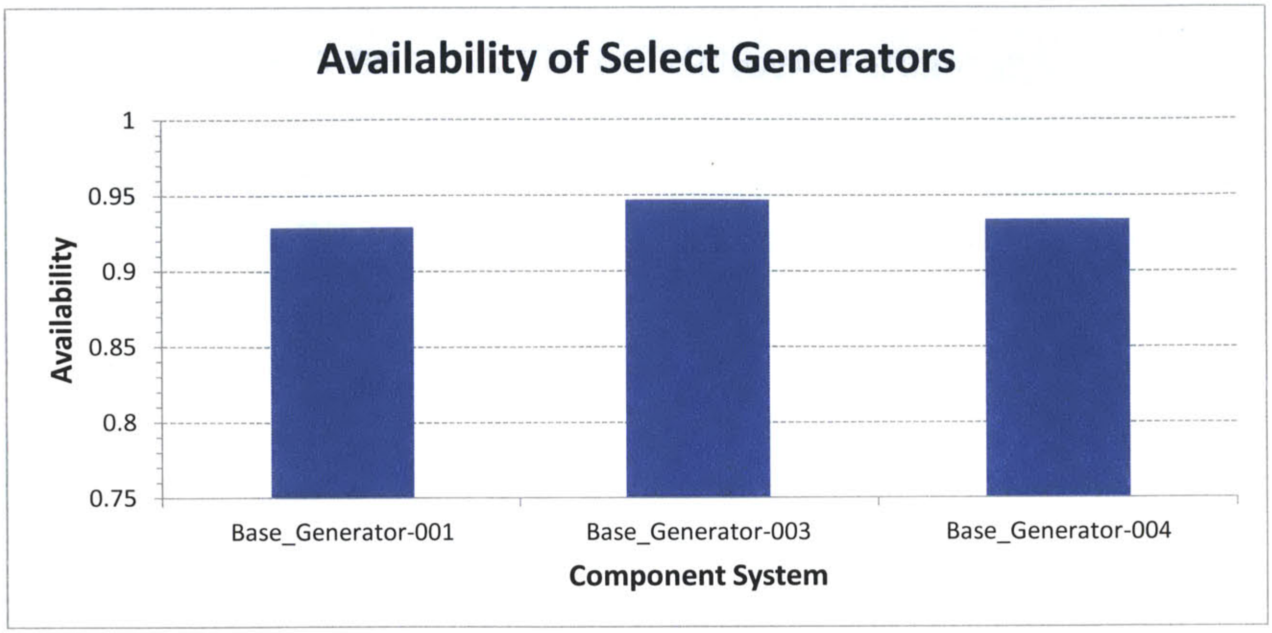


Figure 58. Availability of Select Generators

From the figure, we can see that availability varies slightly between generator systems. Those that experience a larger number of fails due to the stochastic nature of the SoSAT model will have reduced availability. Service time could also potentially impact availability, since if more generators fail than there are maintenance personnel available to perform repairs, the failed systems remain down longer. This in time decreases overall availability when aggregated over the 30 day simulation.

Similar to the availability over time results, we can also look at aggregated results broken out by each of the 100 trials. From the availability over time results for the BCIL, we know that some trials experienced time intervals of low availability. We may wish to assess the aggregated availability for each of the trials and see how those results compared for the 100 trials. A histogram of the availability of the BCIL for each trial is shown in Figure 59.

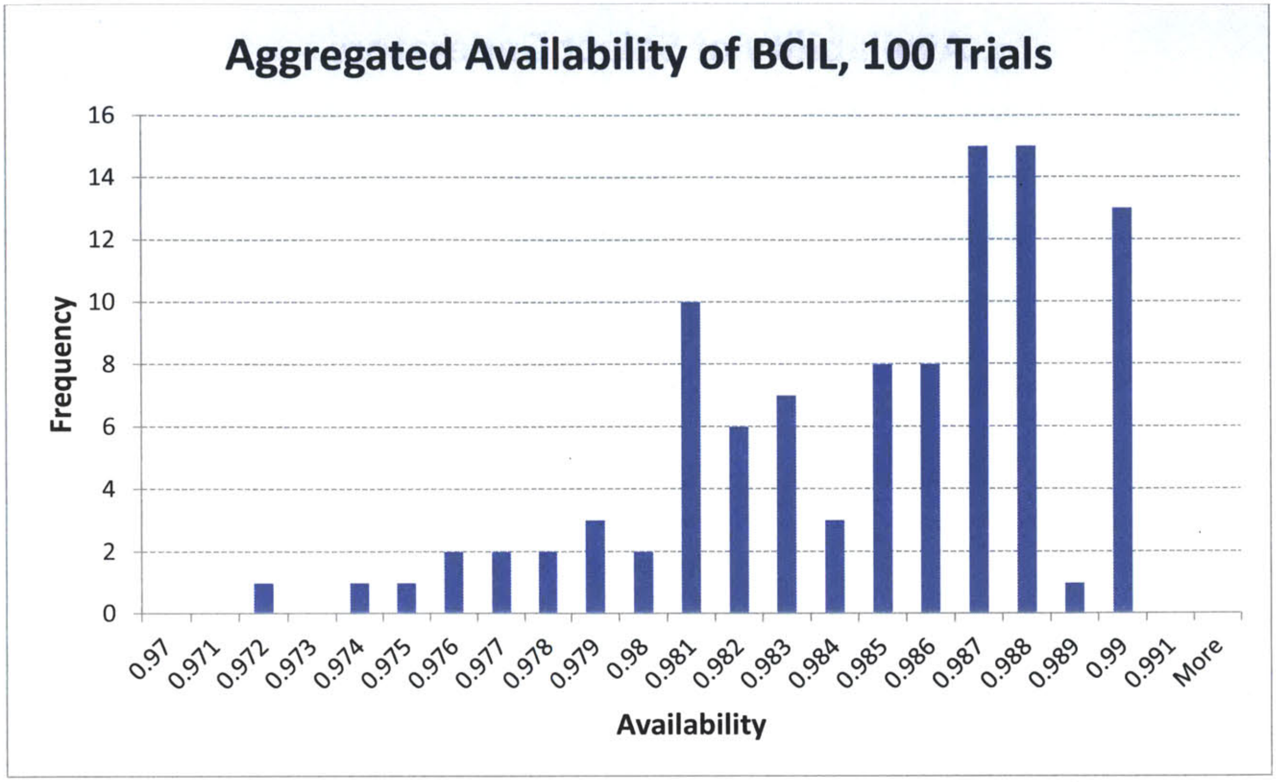


Figure 59. Aggregated BCIL Availability Histogram

Just as with the results shown over time, we can also look at aggregated functional availability. This functional availability is given at the individual system level rather than at the base camp level or containerized system level. The functional availability for the washer and dryer is shown in Figure 60. These systems are both part of the function "Provide Field Services (Clean Clothes)." Even though these systems are both mapped to the same function, individually they can have different levels of availability since each has its own dependencies and resource needs.

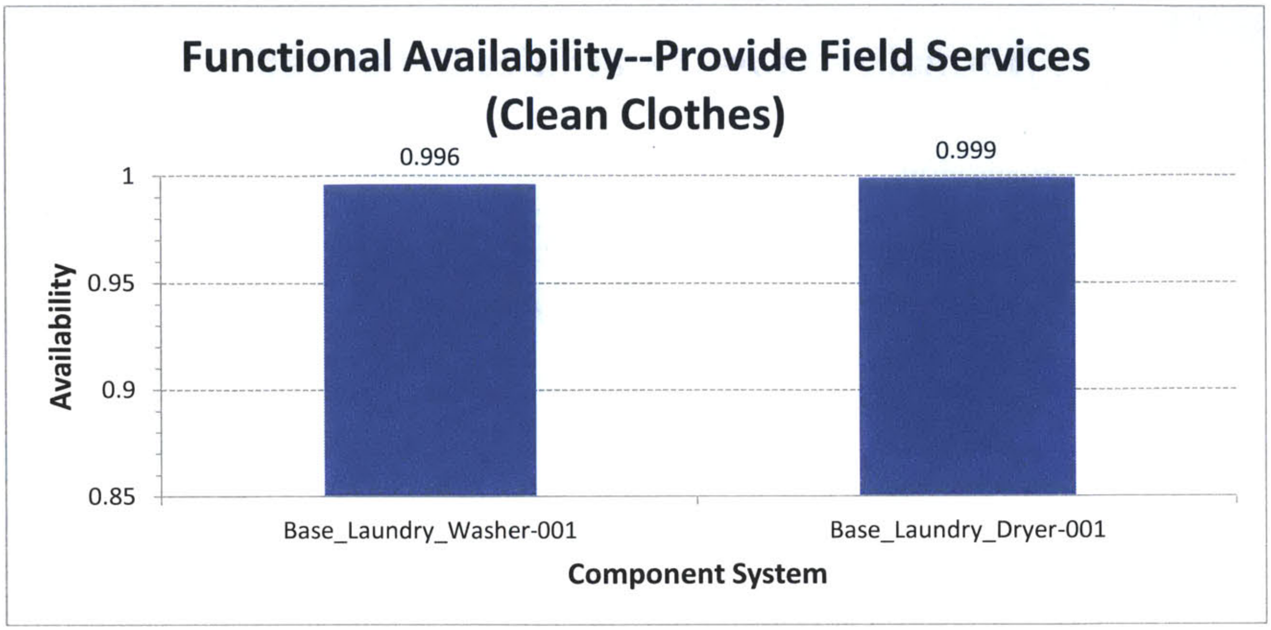


Figure 60. Functional Availability, Clean Clothes

We may also desire to know the total amount of resources generated and consumed during the simulation. These are the same quantities that were used for validation purposes to confirm the output of the model matched the expected output calculated from the spreadsheet values of technical and historical performance. The relative quantities of black water, gray water, and potable water generated are given in Figure 61 below. The relative quantities of fuel and gray water consumed are also analyzed and appear in Figure 62. Note that gray water is both produced and consumed. The containerized shower systems produce gray water as a by-product while operating. Meanwhile, the shower water reuse system consumes gray water as it processes and filters the water back to a potable water state for reuse in the containerized shower systems.

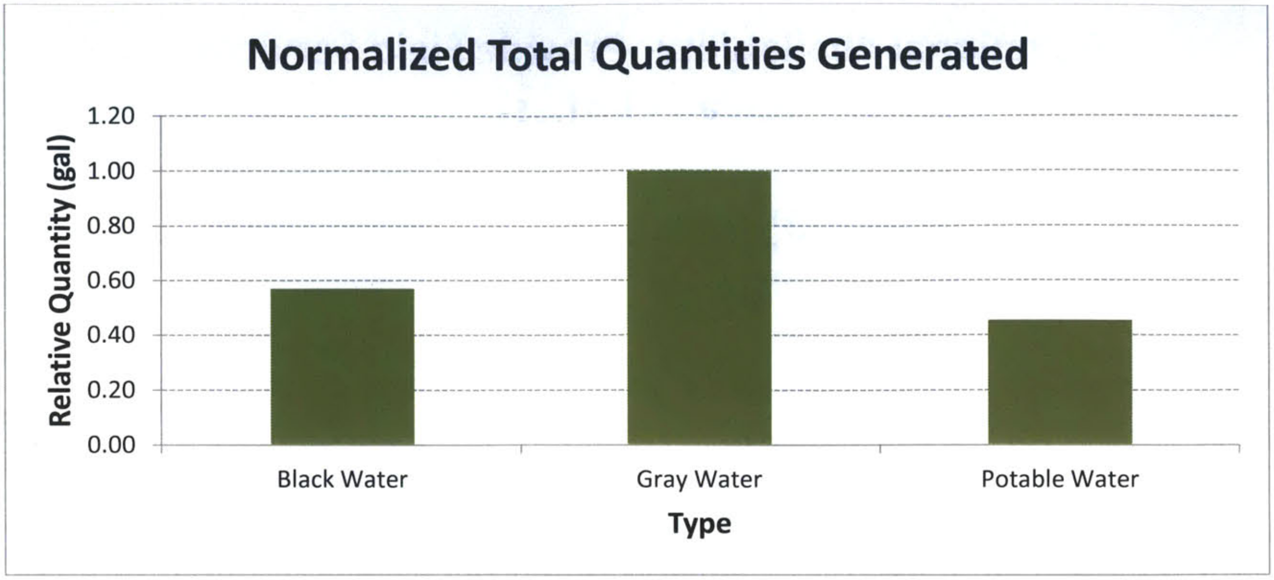


Figure 61. Total Quantities Generated

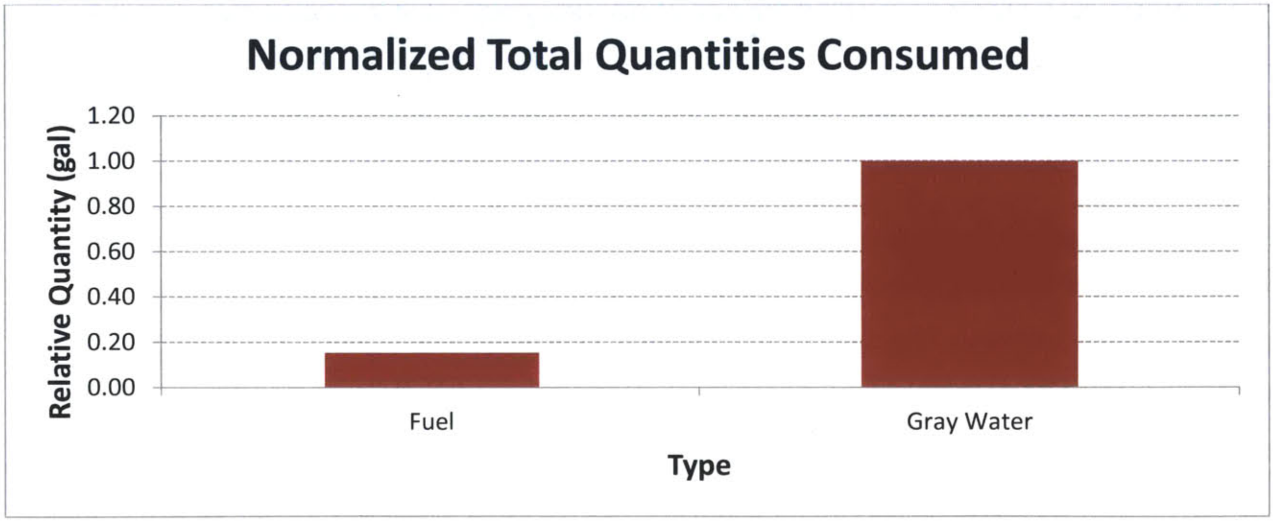


Figure 62. Total Quantities Consumed

The results shown for quantities consumed and produced are the totals for the duration of the simulation. These same results can be generated as an average for all systems rather than a total. For example, black water generated would be the average amount of black water generated by all component systems capable of black water generation. While averaged results may be useful in some situations, they can also be misleading if component systems using or producing a given resource have vastly different rates of consumption or production.

Resource consumption and production can also be examined at the individual component system level or at level of containerized systems. For these systems or groups, we can analyze the quantity of resource used or consumed and the quantity remaining. The average supply inventory of the potable water blivets is shown in Figure 63. The relative average quantity used and relative average quantity remaining is calculated based on the component systems.

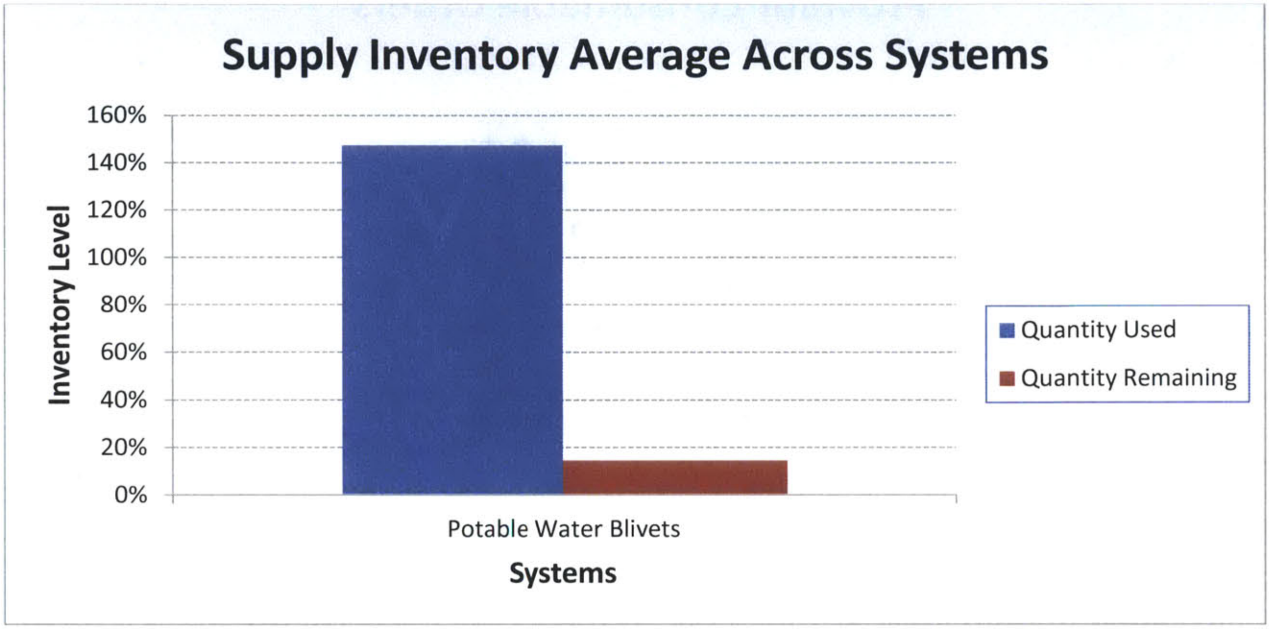


Figure 63. Supply Inventory of Potable Water Blivets

On first glance, the relatively low quantity remaining as compared to the quantity used seems a cause for concern. We must remember however that the potable water blivets are refilled repeatedly during the simulation by deliveries of additional potable water. This is why the quantity used is greater than the maximum inventory capacity of the storage containers. Taking this into account, we would then expect the quantity used to be many times greater than any quantity remaining. The quantity remaining must logically be less than or equal to the capacity of the component potable water blivets, while the quantity used may be many times the capacity since they are being refilled.

Another useful output is the number of provider consumable orders. From the results over time, we determined the shower water reuse system processes 22 batches of gray water from the showers. Therefore we would expect the shower water reuse component system to submit 22 order requests to the gray water blivets and for those gray water blivets to deliver those 22 orders to the shower water reuse system. SoSAT output can be viewed from both the user and provider perspective. Figure 64 shows the number of orders submitted to the gray water blivets and the number of orders they were able to deliver. These verify the 22 processing cycles we observed for the shower water reuse system.

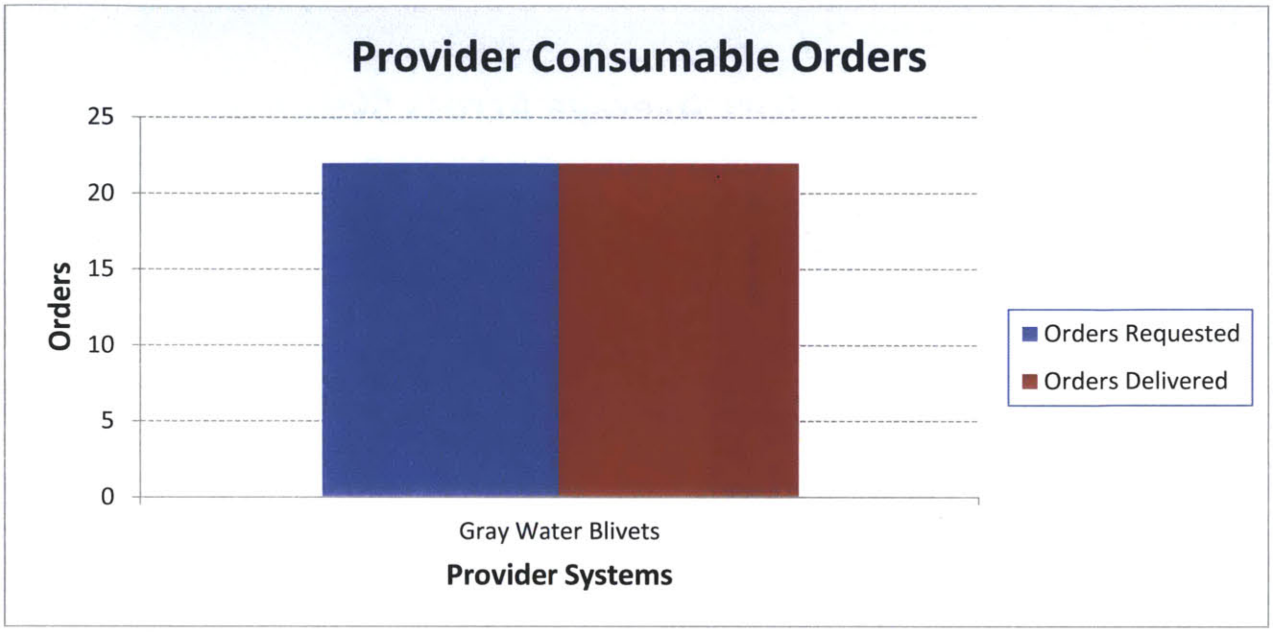


Figure 64. Provider Consumable Orders, Gray Water Blivets

Aside from system performance, the stakeholders may also seek information on how the network groups function during the simulation. Network performance can be analyzed at the network level or at the level of the component systems comprising the network. The first network of interest is the micro grid network. kWh generation by three of the generator component systems is shown in Figure 65. Generation varies slightly due to differences in availability between the generator systems. If the base were using generators specifically designated as back-up generators, we would expect to see significantly lower generation for those generator systems. Additionally, if the micro grid specifies an order for which generators are turned on/off as base power demand increases/decreases, we would also expect to see the generation vary in response.

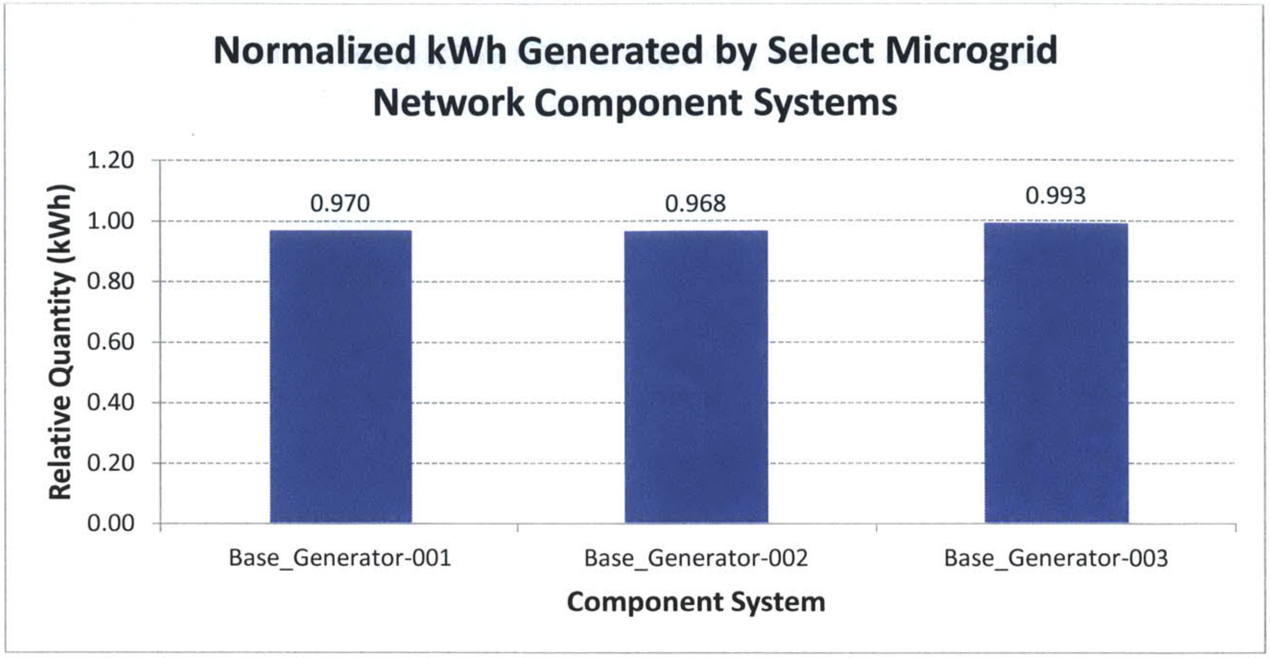


Figure 65. Generation of Power Network

Other types of networks in the model are those defined as "distribution networks" and which provide resources to other systems or groups of systems. The distribution networks which deliver potable water are shown in Figure 66. For example the "Laundry" distribution network connects the potable water blivets which store the potable water to the component systems within the containerized laundry system which use the potable water. Networks are used to model quantities as continuous flows, which is particularly useful for resources such as water, electricity, and waste. The figure shows relative quantities of potable water used by each network.

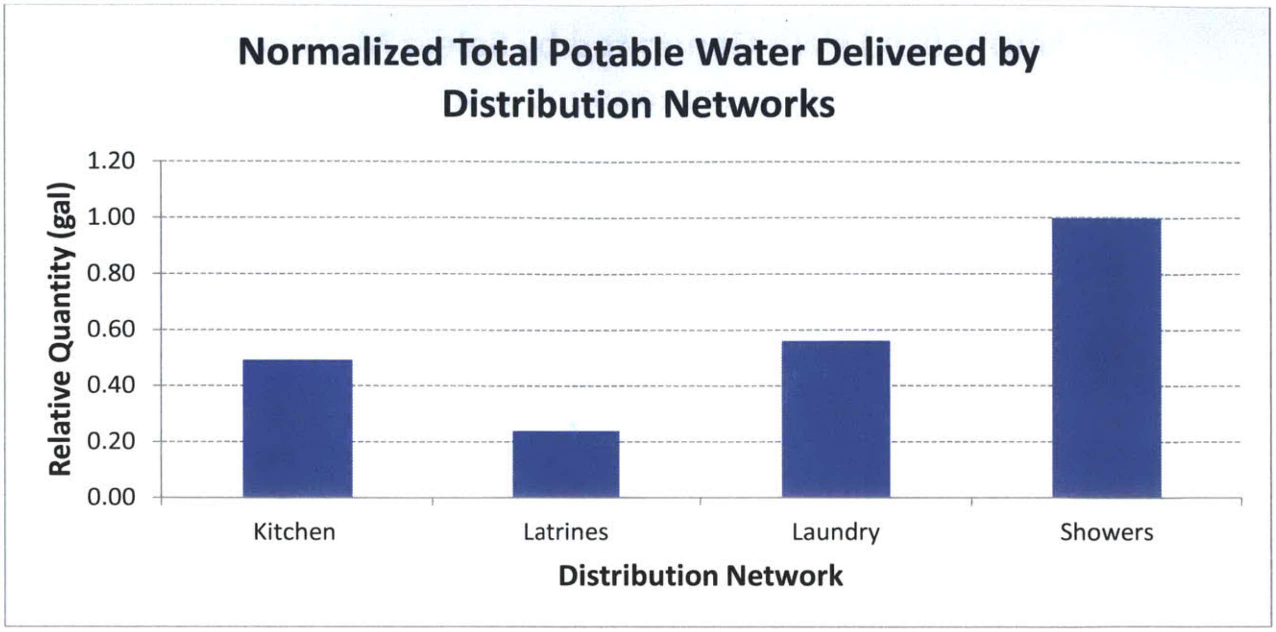


Figure 66. Potable Water Delivered by Distribution Network

From the figure we notice the shower containerized systems consume significantly more potable water, as indicated by the quantity of potable water used by the "Showers" distribution network. This was the primary motivating factor for adding the shower water reuse system to the baseline configuration of the BCIL. The potable water consumption shown includes the potable water fed back into the system after gray water is processed by the shower water reuse system.

The last type of network analyzed is the collector network. "Collector Networks" represent the flow of resources from component systems to container systems for storage. Figure 67 shows five collector networks, which each collect either potable water, gray water, or black water. The relative quantities collected are shown for each collector network. The gray water shower collection network includes gray water that is produced by the shower containerized systems and later processed by the shower water reuse system. The potable water produced by the shower water reuse system and fed back into the shower systems is modeled as a collection network rather than a distribution network because the potable water produced is stored in the potable water supply blivet for the shower before it is used. In general, if the destination of a resource is some type of storage container, then the network is modeled as a collection network.

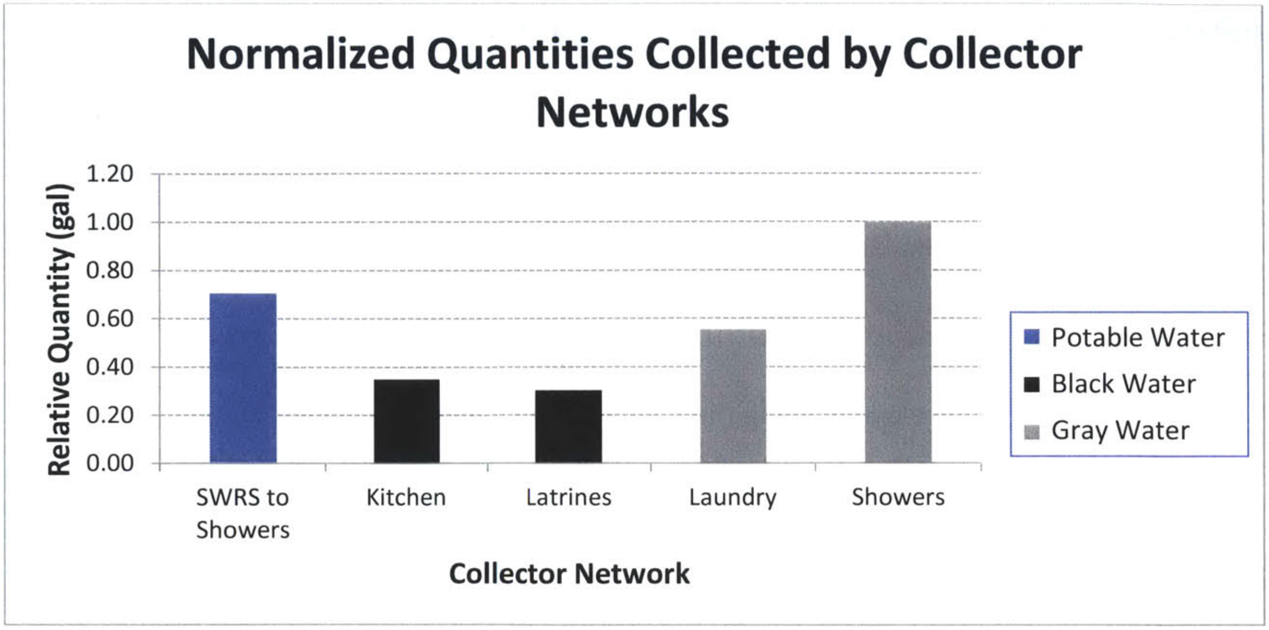


Figure 67. Collector Networks

As illustrated, a properly constructed model is capable of producing a wide variety of outputs translatable to virtually any result of interest. The structure of this SoSAT BCIL baseline model imbues the model with the flexibility to conduct various types of analysis without necessitating changing the model or rerunning the simulation. Results presented to the stakeholder should be chosen by their applicability in addressing the specific concerns of the modeling effort. Additionally, giving stakeholders the opportunity to generate results of interest leads to stronger model buy-in since they can experience how different results are related and develop further insight into how the model functions.

5.8 Sensitivity Analyses

One of the many benefits of a properly structured model is the flexibility to make small changes and analyze the impact on the performance of the SoS model. This is especially useful when determining operational needs for real-life operations of cities or base camps. For the BCIL base camp case study, we are particularly interested in how the availability of the micro grid is impacted by the number of maintenance personnel on base. We also wish to examine how often additional potable water supplies need to be delivered to meet base demand.

For each of the two sensitivity analyses, the respective changes were applied to the baseline model. All other attributes of the systems and interconnections were held constant during the analysis. Values for the sensitivity analyses were determined based on stakeholder input and based on values that the base camp could realistically employ. Non-realistic values were not considered for the analysis since they were considered infeasible for the purposes of this case study. Results are given for the average output of 100 simulation trials run for 720 hours each, as was seen in the baseline simulation.

5.8.1 Analysis 1: Number of Maintenance Personnel

The number of maintenance personnel in the baseline model was determined through consultations with the customer. While adding more maintenance personnel inarguably increases the availability and performance of the generators (and thus the micro grid), the number added to the model must be representative of reality. For the sensitivity analysis, the number of maintenance personal was varied by plus and minus one from the baseline number. Maintenance personnel are included in the number of people on base, which remains fixed at 150. The main focus was to evaluate how small changes in maintenance personnel would impact the availability of the generator component systems over the course of the simulation. Additionally, the performance of the micro grid was examined for each variation, as was the impact to the containerized systems on the base camp.

The average availability over time for the generator component systems for each of the variations in maintenance personnel is shown in Figure 68. As expected, as the number of maintenance personnel increases, the average availability of the generator systems also increases. This occurs because maintenance and repairs of the generator systems can be performed in a timelier manner when there are additional personnel. If multiple generators require attention at the same time, there can be service delays when not enough maintenance personnel are on the base. Increasing the number of personnel prevents these delays, and generators are returned to an operating state more rapidly.

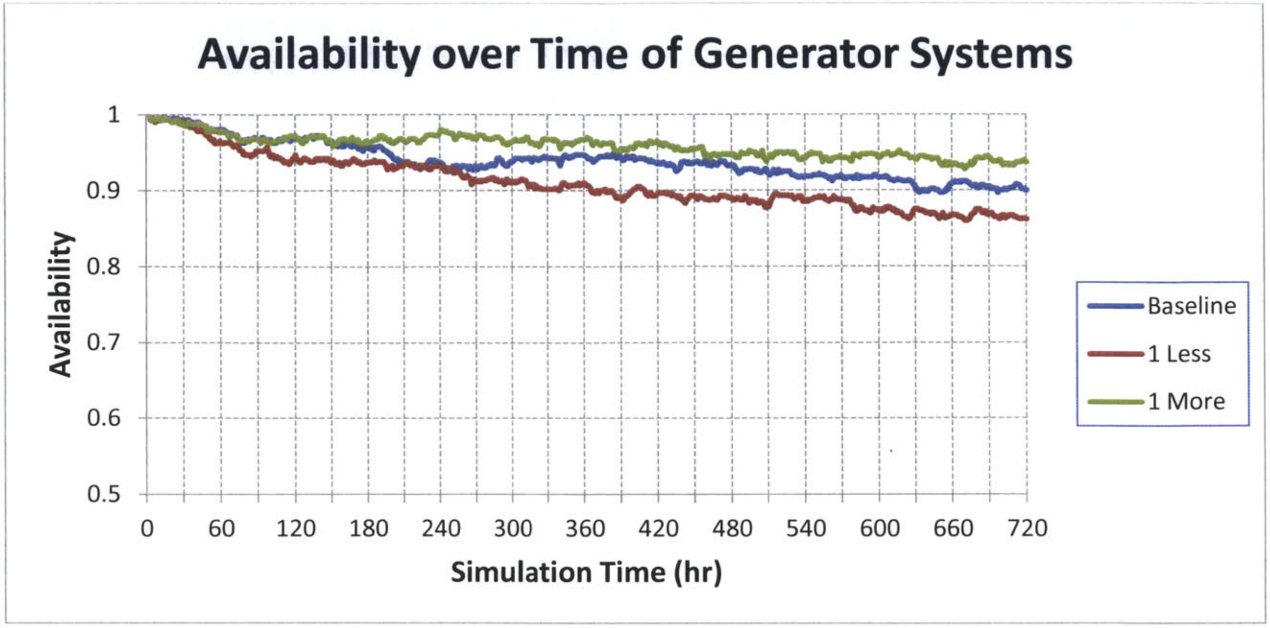


Figure 68. Effects of Maintenance, Generators

Though the base camp should not employ more maintenance personnel than necessary, it should have an adequate number to keep base camp component systems operating at a reasonable level of reliability. From the availability over time results, we question the prudence of removing one of the maintenance personnel. The availability of the generators is quite low when one is removed. Alternatively, adding one more maintenance person seems to provide higher levels of availability that may be more desirable for base operations.

As always, looking at the average availability of the generators does not provide a complete picture of the base's ability to provide power to component systems. We turn to an analysis of the micro grid network to get a better picture of how much electricity is alternatively being demanded by the base camp systems and supplied by the micro grid. Figure 69 shows the difference between the amount of electricity requested by the base camp component systems and the amount provided by the micro grid for each maintenance personnel variation.

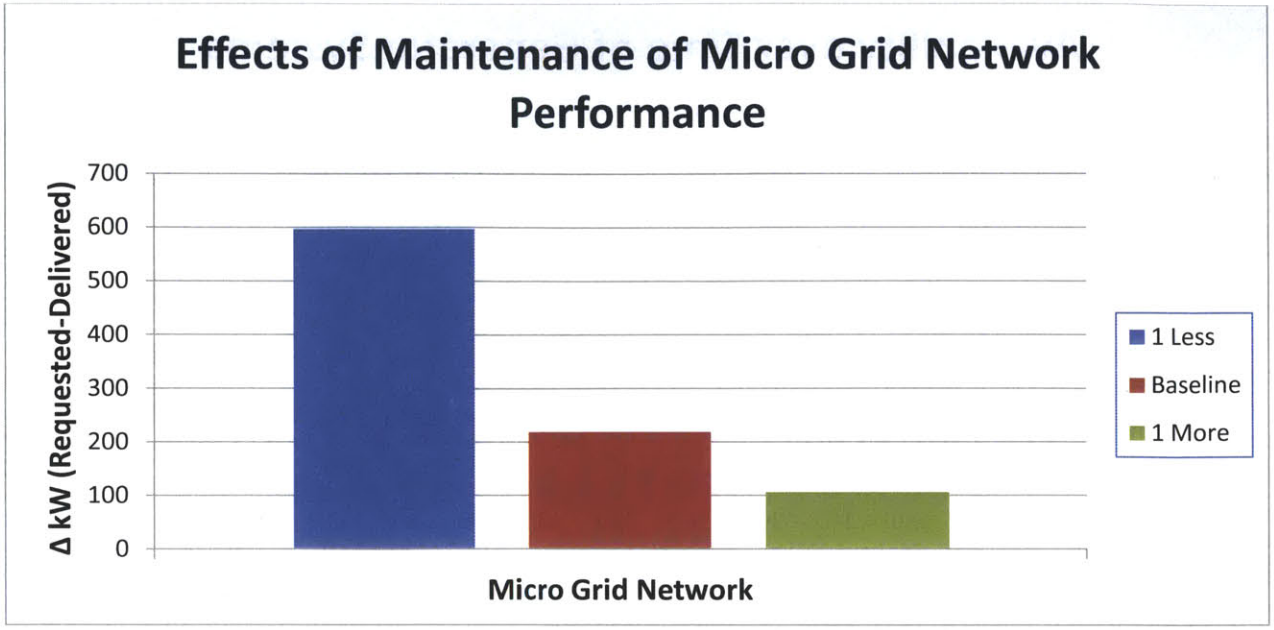


Figure 69. Effects of Maintenance on Micro Grid Performance

As expected, the more maintenance personnel present on base, the more efficiently the micro grid network is able to function. We would expect this trend to continue up to a certain upper limit of micro grid efficiency, with additional maintenance personnel providing less gain in efficiency as the upper limit is approached. Since a base camp of this size would not realistically employ more than a few maintenance personnel, the variation for the sensitivity analysis was limited to adding or subtracting one person from the baseline configuration. For larger base camps or for traditional cities, the sensitivity analysis would likely examine a larger range of potential values.

Aside from looking at the micro grid, we can also look at availability results for the containerized systems that consume electricity from the micro grid. The availability over time results for the laundry containerized system is given in Figure 70 below for the different maintenance variations. Taking away a maintenance person clearly has a negative impact on availability. Adding a maintenance person improves availability slightly, though the effect is harder to discern from this type of result.

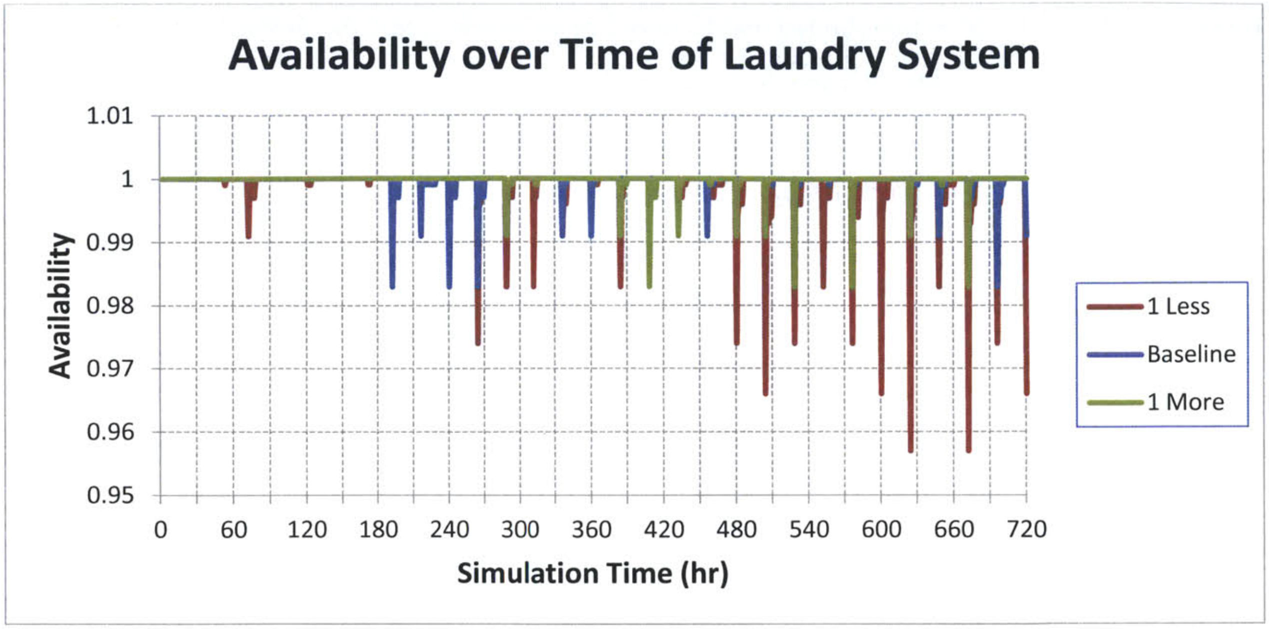


Figure 70. Effects of Maintenance, Laundry

From the figure, we see that the availability of the laundry containerized system decreases more and more drastically as the simulation progresses when there is one less maintenance person on base. This is not a reflection of the operability of the laundry system, but rather of its dependence upon the micro grid. From the analysis of the generator systems, we know that increasingly more generators fail as time progresses when maintenance and repairs cannot be performed in a timely manner. This is certainly the case when we remove a maintenance person. This in turn inhibits the micro grid from meeting the demand of energy-consuming systems (including the laundry containerized system) during all time intervals of the simulation. Though the laundry system itself is in working order and capable of operating, it cannot obtain the needed electricity from the micro grid and the system consequently goes down.

For a clearer analysis of how changes in the number of maintenance personnel impact systems on the base, we can look at the downtime of individual component systems. The downtime for the dryer, the water heater for the shower, and the waste pump for the latrine is given in Figure 71 for each maintenance personnel variation. This is the total downtime experienced by the systems during the 720 hour simulation. The downtime for these systems results from time intervals during the simulation during which the amount of electricity produced by the micro grid is not enough to meet base demand. As more maintenance personnel are added, the micro grid component systems can be maintained and repaired more quickly, and the down time of the individual power-consuming systems on base is reduced.

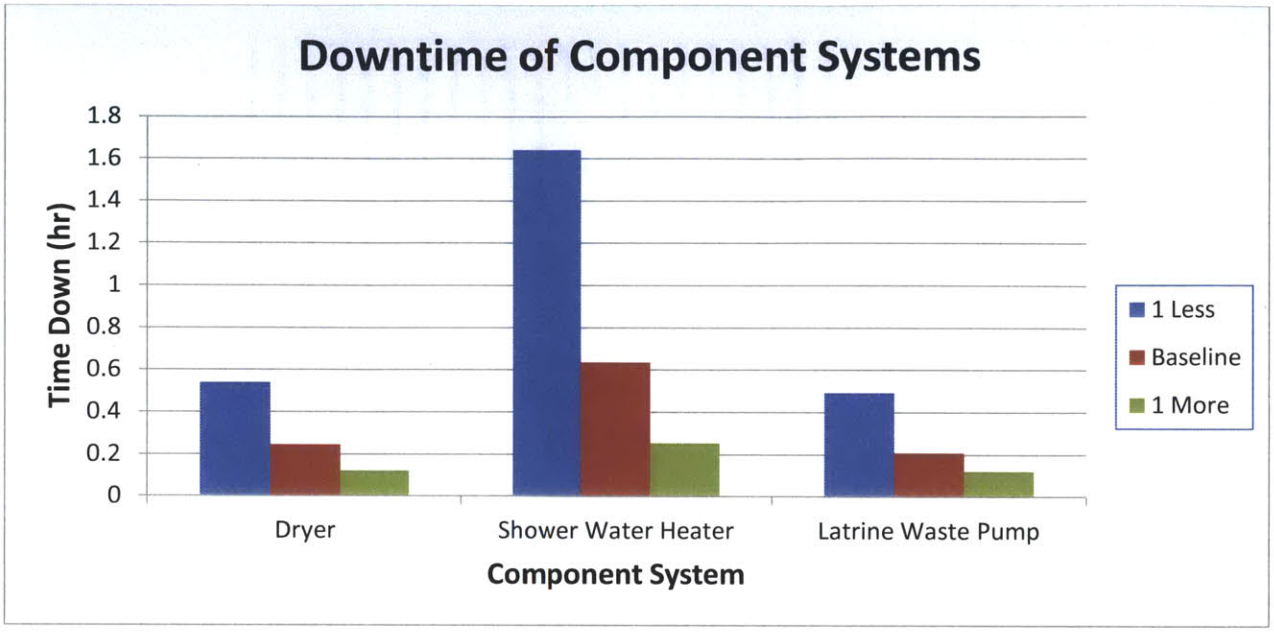


Figure 71. Effects of Maintenance, Downtime of Component Systems

5.8.2 Analysis 2: Frequency of LOGPAC Deliveries

Base camps also depend heavily on deliveries of potable water to maintain operability. Potable water is supplied via a logistics package (LOGPAC) which is delivered by truck to the base. Without timely resupplies, potable water-consuming component systems experience resource depletions and can no longer operate. This in turn diminishes the base camp's ability to provide personnel with laundry, hygiene, and meal needs. From the in-depth analysis of the SoSAT baseline model, we know the current delivery schedule is capable of meeting the base's potable water demand and there are no resource depletions. Potable water deliveries occur at regular intervals over the course of the simulation. In this sensitivity analysis, the time between deliveries is extended by 24 hours or 48 hours and the changes in availability of the component systems are examined.

The first result of interest is the average availability of all potable water-consuming systems on the base camp. Average availability over time results are shown in Figure 72 for the baseline BCIL model, and with the time intervals between deliveries extended by 24 hours and 48 hours.

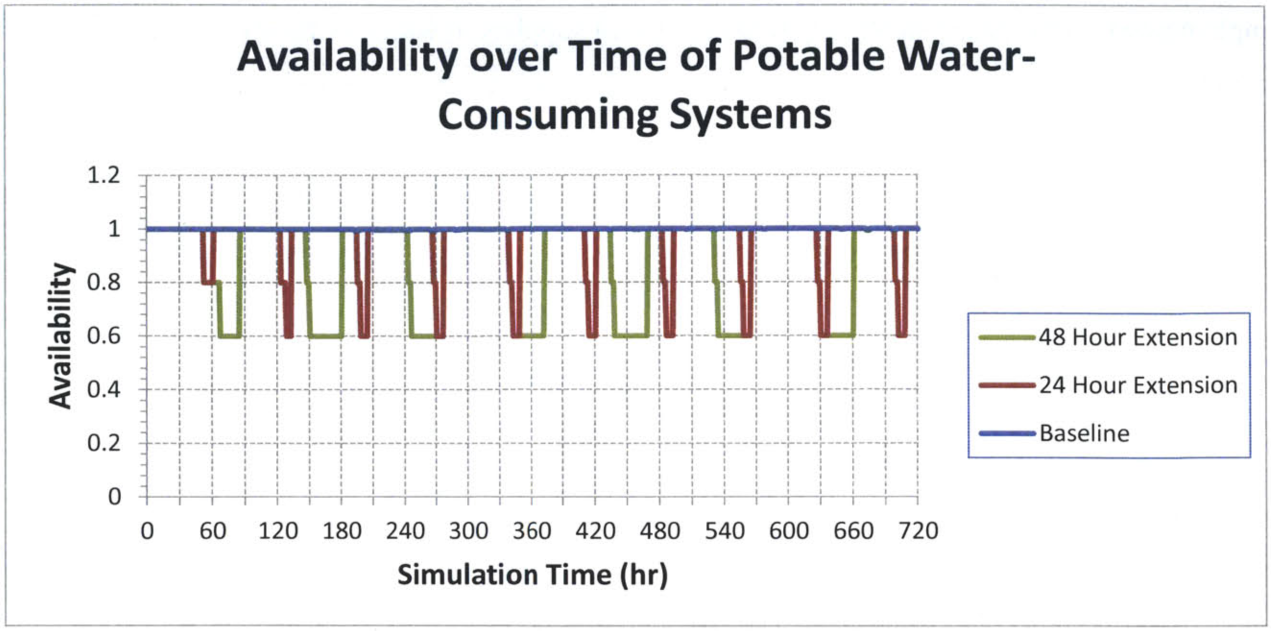


Figure 72. Effects of LOGPAC, Aggregated

The baseline model has a potable water delivery schedule that provides an adequate supply of potable water to keep base systems operating throughout the duration of the simulation. Once the time between deliveries is extended by even 24 hours though, some of the component systems begin to experience resource depletions, those systems stop operating, and the overall availability decreases. Extending the time between deliveries to 48 hours exacerbates the situation. Since availability is going down to approximately 60% rather than 0%, it's clear that only select component systems are experiencing resource depletions caused by the change in frequency of LOGPAC deliveries to the base. If all potable water-consuming component systems experienced resource depletions, then availability could go down to zero were those resource depletions to occur simultaneously.

Looking at results for individual systems revealed that only the kitchen containerized system and the laundry containerized system experience depletions of their respective potable water supplies during the simulation as a result of extending the time between LOGPAC deliveries. Results for the supply inventory over time of potable water for the kitchen and laundry systems are shown in Figure 73 and Figure 74 respectively. Both had enough potable water in the baseline model, but experience resource depletion with the extended delivery scenarios. All other potable water-consuming systems had enough inventories to operate, regardless of the delivery schedule implemented. Either the systems required only small supplies of water, or their potable water supply was augmented by an additional source, such as the shower water reuse system which supplies potable water to the shower systems.

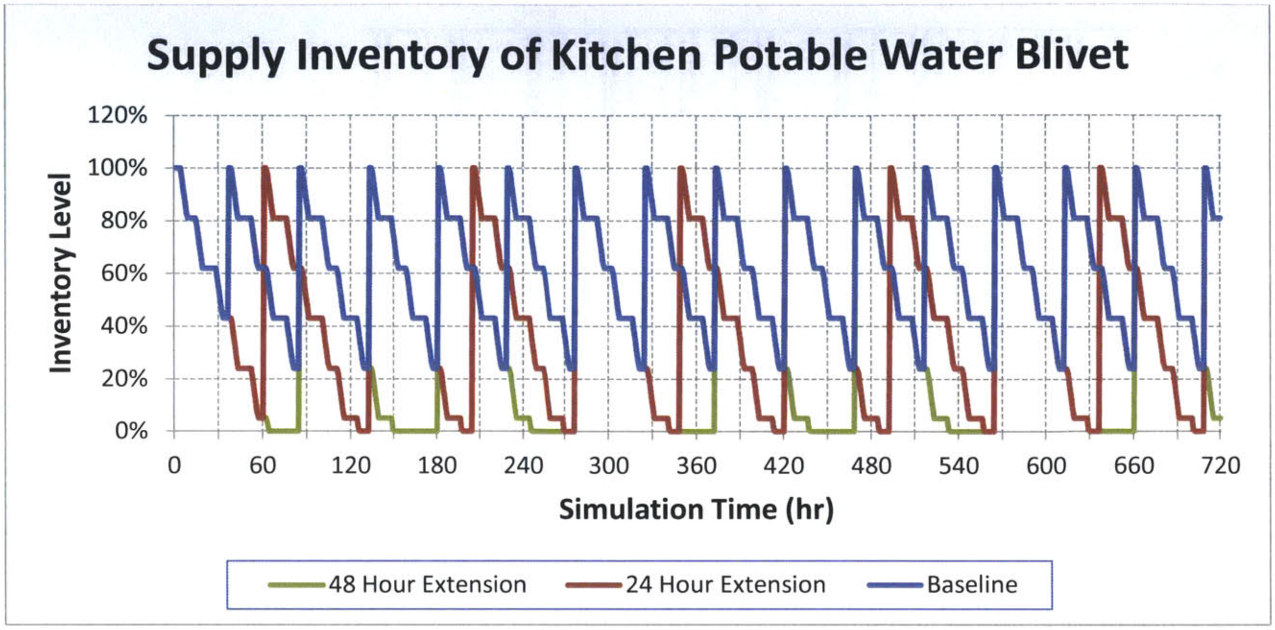


Figure 73. Supply Inventory of Potable Water for Kitchen

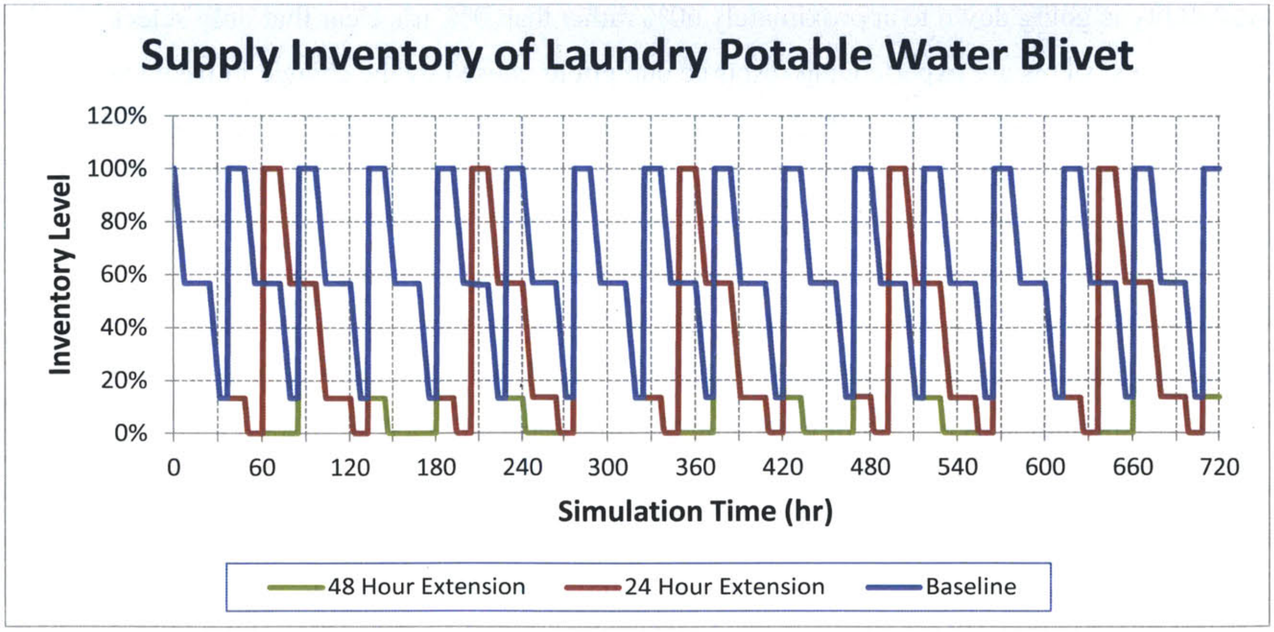


Figure 74. Supply Inventory of Potable Water for Laundry

Neither the kitchen system nor the laundry system experienced potable water shortages under the baseline configuration. Both repeatedly experienced resource depletions once the delivery time between LOGPACs was increased. We therefore consider these systems to be highly sensitive to the potable water delivery schedule and to provide a constraint on the LOGPAC schedule. If any variation in the delivery schedule is desired (particularly an extension) in the future, the potable water needs of these systems must be further addressed.

5.9 Integrating Energy Sensor Data

All component systems on the BCIL are equipped with sensors which record information such as weather conditions, energy usage, water usage, etc. The frequency at which this information is recorded varies according to the type of data being collected. Typically the frequency varies from every second to every few minutes.

Sensor data in general represents a unique challenge for cities as planners and officials strive to move toward smart cities with thousands or even millions of sensors generating real-time data about how systems are operating. Storage, processing, and usage of this data are already key concerns being researched. Another area of interest is how this sensor data could be used in systems of systems models of cities. Ideally, there should be a method of blending sensor data and SoS city modeling to improve knowledge of key component systems and provide insight as to how the city is functioning over time.

When thinking about how sensor data would be used in an SoS city model, it becomes crucial to differentiate between input data and output data. The type of data generated by sensors is equivalent to the output data produced by SoSAT or other types of SoS modeling software. The data retrieved is already capturing interactions between component systems and is affected by downtime and failures of the systems. This data should never be used as an input to the SoS model since input data is system or interconnection specific. Mistaking output data for input data would result in an unintelligible and incorrect model.

Since senor data is equivalent to the output data from an SoS model, it would be sensible to use sensor data as another way of validating the SoS model. Once the SoS city model is validated through a variety of means as have been explored in this case study, sensor data could potentially be used to test whether systems are performing as expected in reality. For example, sensor data from component systems can be aggregated and compared to the output from the SoSAT model. Large discrepancies could indicate that component systems are not functioning as expected, or that some type of failure has occurred that has not been captured by the model or understood by those analyzing the structure. This comparison would likely not occur at a real time rate, but instead occur at a delay of a few time intervals, based upon the time interval used in the simulation model. Care should be taken when aggregating the sensor data to make sure it both matches and realistically represents the time interval used in the SoS model. Alternatively, the model could be used to check city sensor data after post processing if the comparison is not time critical.

The implementation of comparing sensor data to the SoSAT BCIL model output will not be addressed as part of this thesis. While there are functioning sensors on the BCIL component systems, the data retrieved from the sensors is only applicable during trial periods when the BCIL is being utilized as it would be if the base camp were actually deployed, operating, and inhabited. The current baseline configuration is no longer identical to the baseline configuration during the last trial run, and cannot readily be altered to match. Future BCIL trials are scheduled for a time frame which occurs after the completion of this thesis. Sensor data usage will be studied further at that time, but will not be included as part of this research effort.

5.10 Technology Insertion

New technologies on their own rarely contribute to the value of a systems of systems framework. After they are integrated into the structure though, they can have significant impact on the structure at multiple levels of aggregation. As with any component system within a SoS model, the operation of a new technology system can be well understood when the system is operating in isolation, but it's performance in reality is both affected by and influences other component systems in the city or base camp. The consequences of these interactions are rarely straightforward and therefore not easily comprehended without the use of computational tools. SoS models provide a way to analyze the impacts of these new technologies and understand the implications of their inclusion without having to physically implement the system in reality. Testing new technologies through SoS modeling and simulation saves time and money, while quickly revealing any misconceptions about how the new technology will perform when interacting with other component systems.

Traditional technology insertion cases have focused on the addition or advancement of a component part within a complex system and how the presence of the new technology impacts potential performance [25], [26]. Technology insertion from a systems of systems perspective differs slightly in that rather than adding a part to a component system, an entirely new or revised component system is added. The scope between modeling complex systems and SoS therefore varies in the level of detail examined. For systems of systems models, the smallest item considered for technology insertion purposes will always be a component system. In the case where a change is made to a component system, it will still be entered into the model as a revised component system, where its attributes will vary from the original system, yet the system is still modeled from the perspective of a component system rather than from the perspective of individual parts of a system.

The BCIL case study presents certain limiting factors on the normal technology insertion framework. Potential new technologies must follow the force provider base camp structure in that they are pre-packaged as containerized systems. These potential technologies are well-defined for this purpose and already under development. In other situations we might wish to examine the impact of new technologies before they reach the development stage. This would be particularly important for traditional cities, where multiple types and variations of new technologies are under consideration. The distinction between base camp technology insertion and traditional city technology insertion will be examined in more detail in 6 Conclusions and Future Work.

5.10.1 Case 1: Reusing Laundry Water

From the sensitivity analysis examining the effects of the delivery schedule of the LOGPACs on the availability of potable water-consuming systems on base, we know the laundry containerized system was highly sensitive to additional deliveries of potable water. In deployment, the delivery of supplies presents clear challenges, since convoys can be intercepted by enemy forces. The risk presented by these attacks not only impacts the operability of the base camp, but also poses significant risk to the lives of the personnel driving the delivery trucks. A key goal of future base camp configurations and management is to reduce the reliance upon outside supply deliveries. Designing more self-sufficient component systems or trying to configure groups of component systems to operate in a more closed-loop manner are primary ways of decreasing the need for LOGPAC deliveries.

The shower containerized system can be viewed as a relatively closed-loop system since the initial supply of potable water is used by the showers, processed by the shower water reuse system, and fed back into the shower system. This cyclical process operates for the duration of the simulation. Small amounts of potable water are lost since the shower water reuse system can only process 75% of the gray water from the showers. However, the dependence upon additional potable water supplies is still drastically reduced and the shower containerized system is largely self-sufficient when the shower water reuse system is included.

Since the laundry containerized system is highly sensitive to potable water deliveries, we wish to examine whether a similar technology could be used with gray water produced by the washing machine to reuse water supplies. A laundry water reuse system was added to the baseline BCIL model. The attributes and implementation of this component system mirror that of the shower water reuse system. Though the shower water reuse system is considered to be part of the baseline for the BCIL base camp, it is a relatively new technology which is usually considered as an add-on technology for other base camps. Thus we consider the addition of the laundry water reuse system to the laundry containerized system to be a technology insertion for the purposes of this case study.

When inserting a new technology into an SoS model, it is crucial to not only define the new system, but also to define any new supply connections or supply networks that may be associated with the new technology. The laundry water reuse system is physically connected to the gray water collection blivet for the washing machine and is also physically connected to the potable water supply blivet for the laundry containerized system. There are also resource supply connections that must be defined for this new technology. The laundry water reuse system is supplied with gray water from the gray water collection blivet once the blivet is filled to capacity. After processing the gray water, the laundry water reuse system then in turn supplies potable water to the potable water blivet for the laundry containerized system. Meanwhile, all original supply connections must also be retained, such as those between the potable water supply and the laundry containerized system, as well as between the washer and the gray water collection blivet. Finally, the laundry containerized system must still be able to receive additional potable water deliveries from the LOGPACs to make up for the small amount of water not processed by the laundry water reuse system.

Once the new technology and its interconnections with other component systems has been implemented within the SoSAT model, the output of the model needs to be examined to make sure there are no problems with the implementation and that the key features of the system are performing as expected. We would expect for the newly added laundry water reuse system to be off while the gray water storage blivet is filling with used water from the showers and then start operating once the gray water is transferred into the system. Like the shower water reuse system, the laundry water reuse system takes eight hours to process each batch of gray water. While the laundry water reuse system is operating, it consumes both gray water and electricity. The electricity powers the system to process the gray water from the gray water collection blivet. Both of these resources should be consumed by the system during operation. Resource consumption for the laundry water reuse system is shown in Figure 75 and Figure 76 below. As expected, the system consumed gray water and electricity during the eight hour operational cycles.

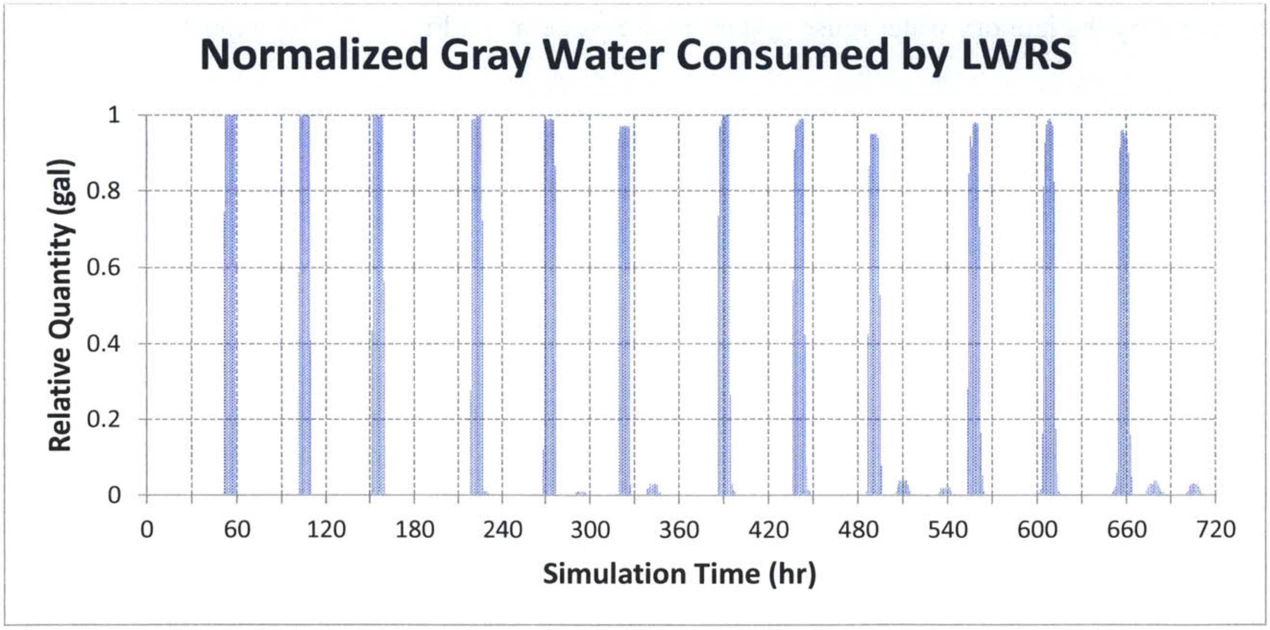


Figure 75. Gray Water Consumed by LWRS

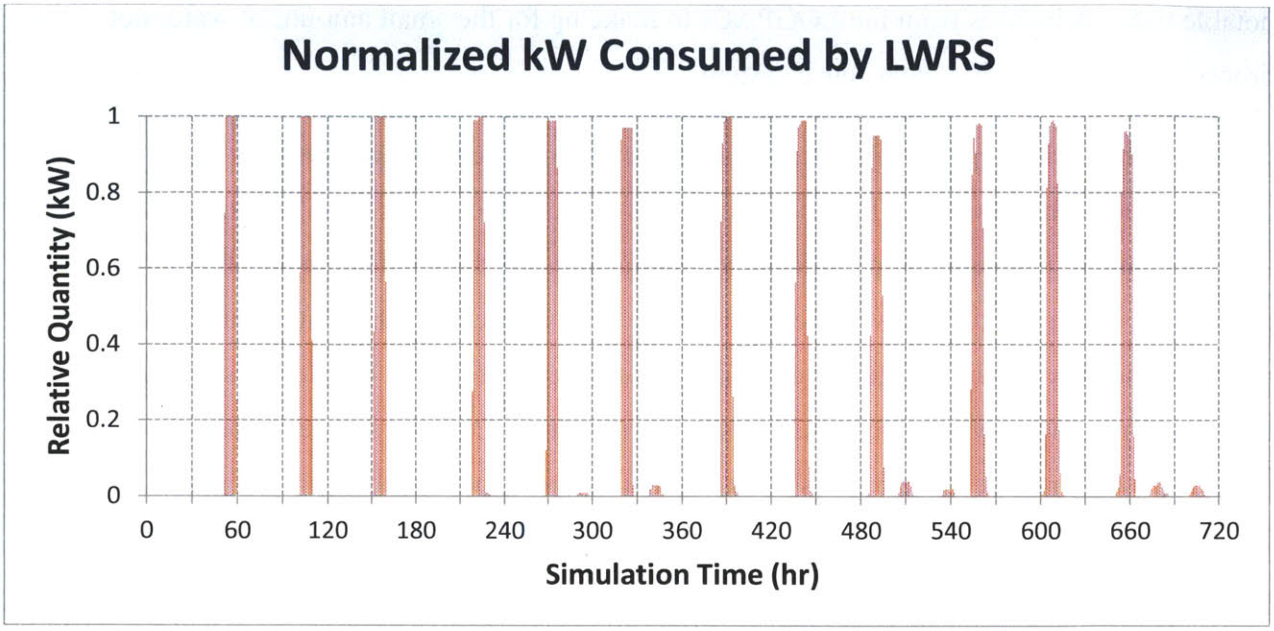


Figure 76. kW Consumed by LWRS

Besides checking to ensure that resources consumed match expected performance, we also need to check resource production. During the eight hours the laundry water reuse system is operating, it should be processing the gray water and in turn producing potable water. The potable water produced by the laundry water reuse system is shown below in Figure 77. As expected, the system is operating in a generally cyclical manner as gray water is fed into the system and processed back to potable water quality. The potable water generated represents the supply connection between the laundry water reuse system and the potable water blivet that supplies the laundry containerized system.

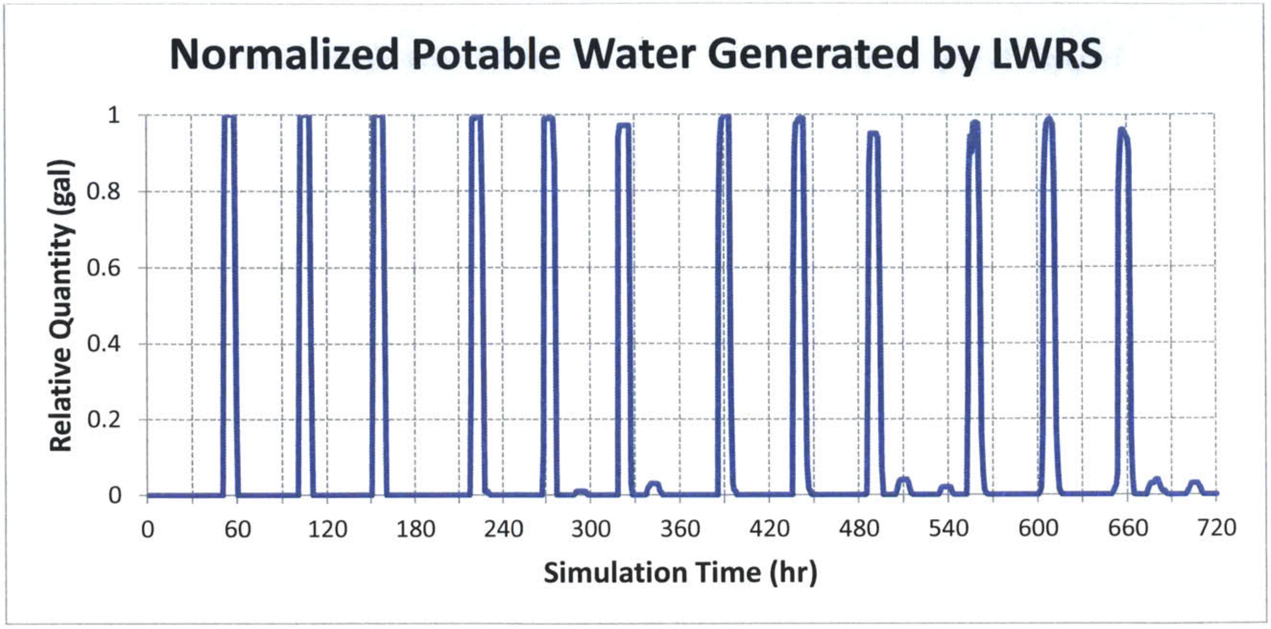


Figure 77. Potable Water Generated by LWRS

The component system is already shown to be consuming gray water during operational hours. A quick check should be done to ensure the gray water is being properly transferred from the gray water collection blivet. Gray water should only be transferred to the laundry water reuse system once the collection container has filled to capacity. The inventory of the gray water collection blivet that collects the used water from the washing machine is shown in Figure 78.

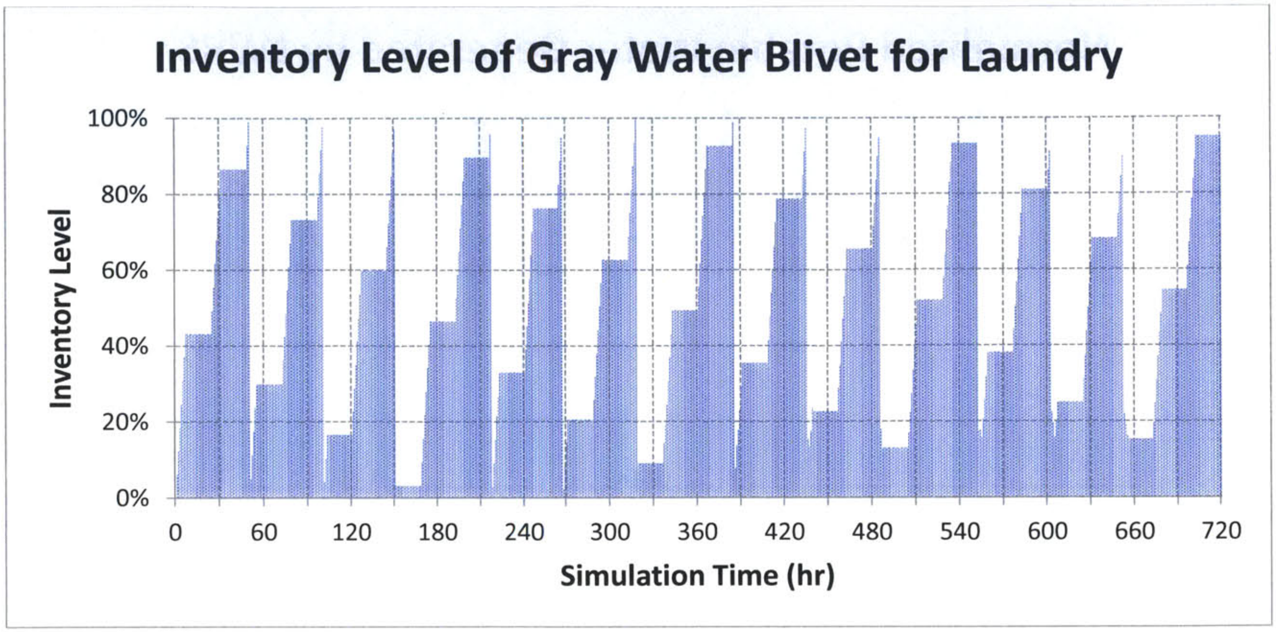


Figure 78. Inventory of Gray Water for Laundry

The gray water collection blivet fills gradually to capacity and is then emptied when the gray water is transferred into the laundry water reuse system. The rate at which the collection blivet fills is impacted by the washing machine utilization. The washing machine is only used seven hours per 24 hour period, which accounts for the flat-lined intervals during the simulation. The fill rate is also affected by the number of personnel on base, which again is varied during the simulation to mimic personnel leaving the base camp for various missions. Utilization of the washing machine varies according to how many personnel are still on base, which in turn determines the amount of gray water produced during the washing machine during each time interval of operation.

We should also check to make sure the potable water produced is being transferred into the potable water blivet that supplies the laundry containerized system. The supply inventory for this blivet is shown in Figure 79. Notice the blivet remains at maximum capacity during certain intervals, or rapidly returns to maximum or near-maximum capacity after short time periods. Two factors are playing a key role in this scenario. As mentioned, the washing machine which uses the potable water only operates seven hours per day. If the potable water is resupplied during the 17 hours during which the system is not operating, then no water is consumed until the system starts operating again. This accounts for the flat-lined areas seen in the inventory.

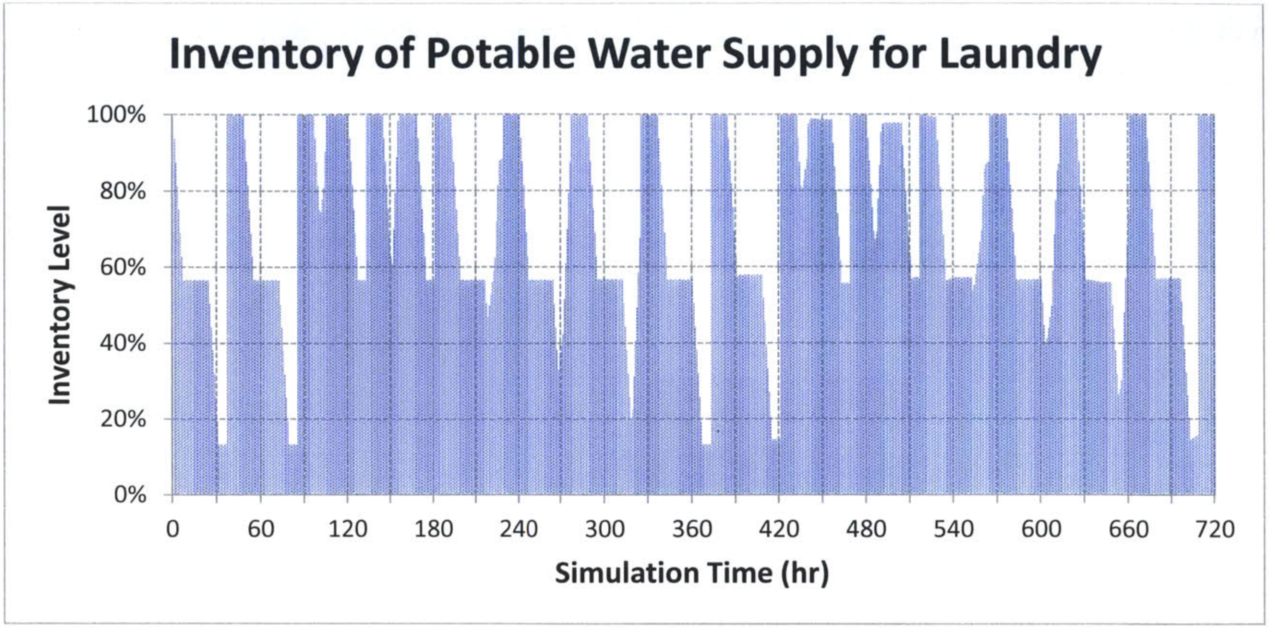


Figure 79. Inventory of Potable Water for Laundry

The potable water inventory does not appear as cyclical as the other component systems because it is being used at this variable rate, but also because it is being supplied by two sources—the LOGPAC and the laundry water reuse system—which both operate on strict schedules which may or may not align with how the washing machine is being used. In fact, we see the impact of these mismatched utilization schedules in the potable water produced by the laundry water reuse system which was shown earlier in Figure 77. Besides the main processing cycles during which the system is producing potable water, there are also small time intervals during which additional small amounts of potable water are produced. This is mainly because the potable water blivet that supplies the laundry containerized system was filled to capacity and the laundry water reuse system couldn't finish processing the batch until some of the inventory from the potable water blivet was used during the next time the washing machine became operational again.

The most telling analysis for assessing the impact of this new technology addition is to examine how much of the total potable water used by the laundry containerized system washing machine is being generated by the laundry water reuse system. The amount of potable water provided by the laundry water reuse system and the amount provided by additional LOGPAC deliveries are shown in Figure 80. Note that the original amount of water in the potable water blivet is included in the LOGPAC provided quantity, since that potable water was originally delivered to the base along with the component systems.

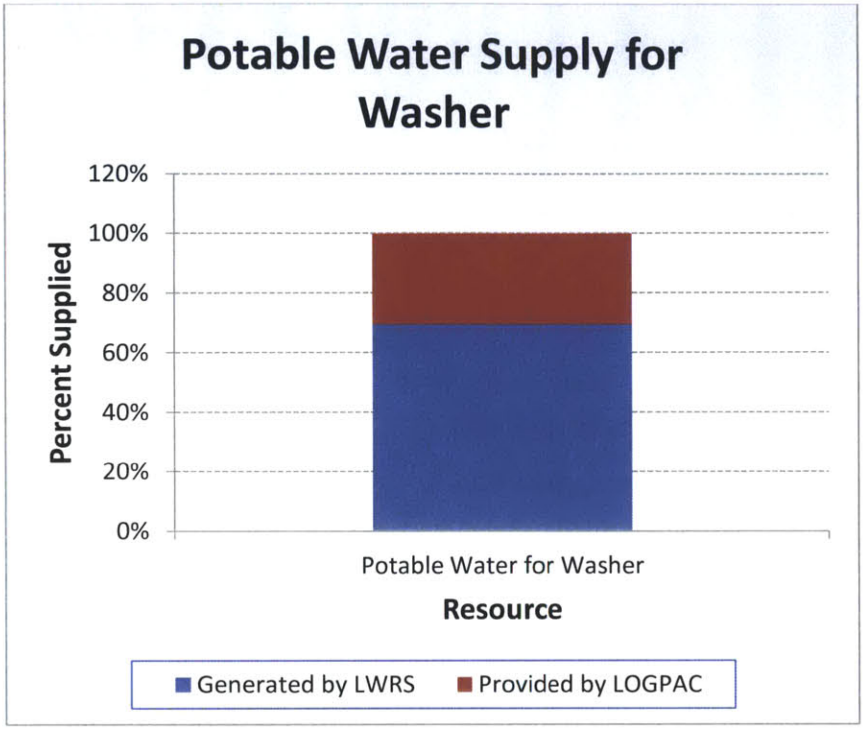


Figure 80. Potable Water Suppliers

From the figure we see that a large majority of the potable water being used by the washing machine is now being provided by the laundry water reuse system. In fact, this new technology provides approximately 70% of the total potable water used by the washing machine. Including this new technology thereby significantly reduces the dependency of the laundry containerized system on additional deliveries of potable water. The delivery schedule is still constrained by the potable water needs of the kitchen though. Since the kitchen has its own potable water supply blivet and cannot draw from other blivets, the number of LOGPAC deliveries cannot be decreased and remains the same as in the baseline model. If stakeholders wish to decrease the number of potable water deliveries to the base, changes will need to be made to the kitchen containerized system. New technologies to lessen the kitchen's potable water dependency will not be examined as part of this case study, but are an ongoing area of interest.

5.10.2 Case 2: Solar Powered Hot Water Heater

The second technology insertion case looked at for the BCIL base camp study was adding a solar powered hot water heater to heat the water used in the kitchen's sanitation system. The sanitation system consists of a 3-well sink for washing, rinsing, and sanitizing utensils and kitchen equipment [27]. The sanitation system for the kitchen containerized system is shown in Figure 81 for illustrative purposes.

Figure 81. Kitchen Sanitation System

Normally the kitchen sanitation system uses energy from the micro grid to heat water for the sink. The goal of inserting this new technology is to completely remove this energy dependency by instead using solar power to heat the water. The same potable water blivet that supplied the kitchen sanitation system is now used to supply the solar powered hot water heater. While the solar hot water heater eliminates the use of electricity by the sanitation system, it does require a small amount of fuel to operate. The tradeoff between this incurred fuel usage and the reduced energy demand should be considered when deciding whether to implement this new technology.

The first result of interest is to see how much the energy demand of the base camp has changed with the addition of this new solar technology. We would expect the energy requested by the component systems, and also the quantity delivered by the micro grid to decrease when the solar powered hot water heater is included in the model. The sanitation system should no longer be using any electricity from the micro grid and the demand should adjust accordingly. The comparison between the quantities requested and delivered for the baseline model and for the model including the solar powered hot water heater (SHWH) is shown in Figure 82 below.

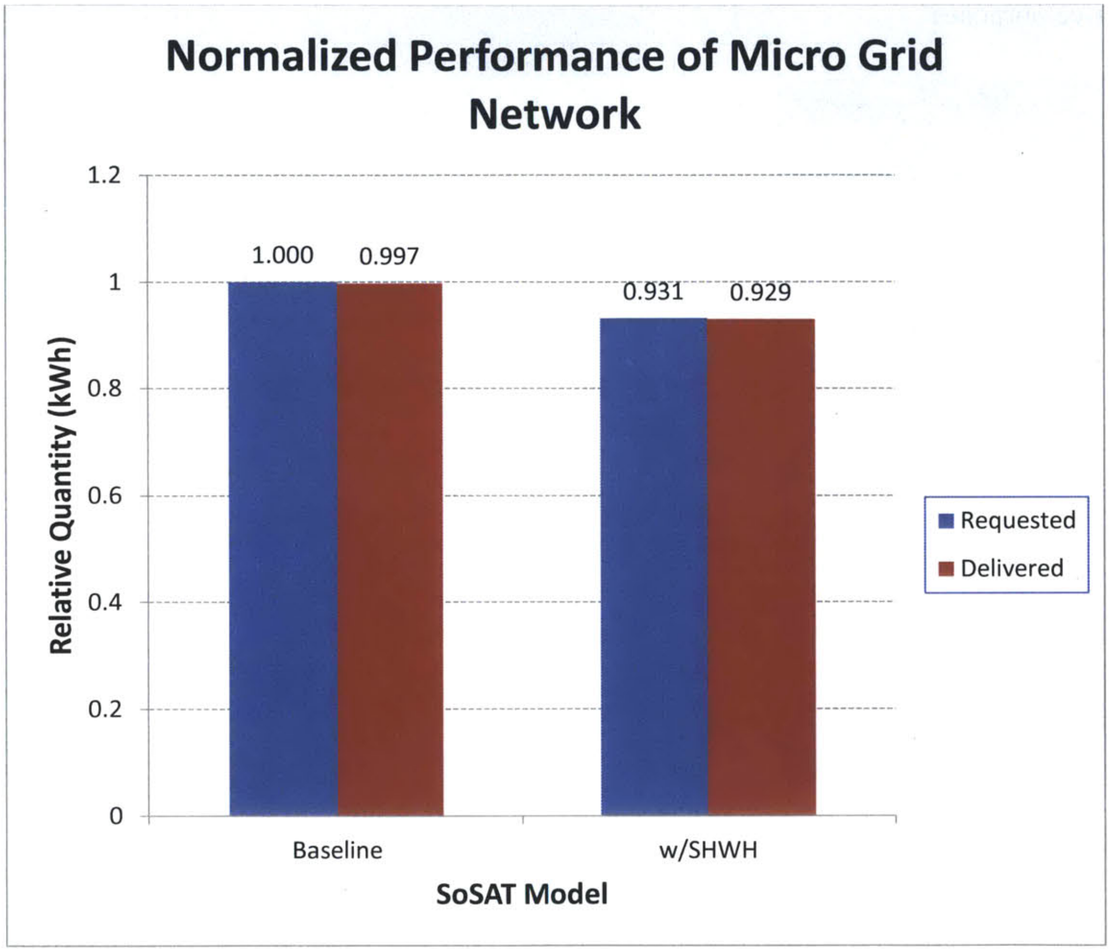


Figure 82. Comparison of Micro Grid Performance

The amount of electricity used in the model with the solar powered hot water heater decreases as expected. The amount requested still slightly exceeds the amount delivered by the micro grid during certain time intervals, though not as much as in the baseline model. Before normalizing the values, the difference in amount requested and amount delivered is 218.6 kWh for the baseline model and 168.3 kWh for the technology insertion model. This slightly increased performance for the technology insertion case stems from the fact that the base camp's peak power demand is lower when the solar powered hot water heater is included. As compared to the baseline model, when one or more generators is down, the micro grid of the technology insertion model still fulfills the power demand of a higher percentage of component systems since the overall demand has been reduced. For micro grids with the same configuration and properties, the power grid will perform better as the peak demand of the base camp is reduced.

Based on the specifications of the solar powered hot water heater, we expect the electricity consumed by the kitchen sanitation system to decrease to zero. Looking at the aggregated results for resources consumed over the simulation shows us that the total power consumption of the sanitation system for the technology insertion case is zero as expected. This result is shown in Figure 83.

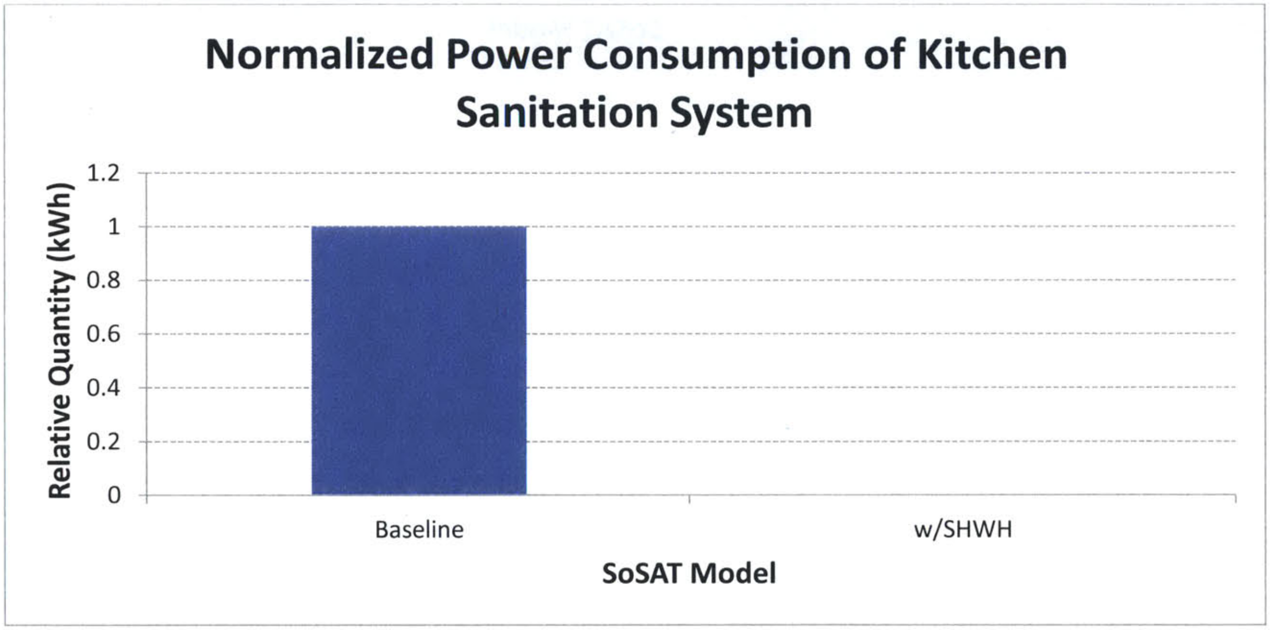


Figure 83. Comparison of Power Consumption of Sanitation System

The other analysis we are particularly interested in is analyzing the tradeoff between the reduced power consumption achieved by adding the solar powered hot water heater versus the fuel needed to run this additional component system. When considering only that a new system is being added to the baseline configuration which consumes fuel, we'd expect the total fuel consumption of the base to increase, while the electricity consumed decreases for the reasons already discussed. Yet comparing the total fuel consumed for the baseline model and the technology insertion model reveals a startling result as shown in Figure 84.

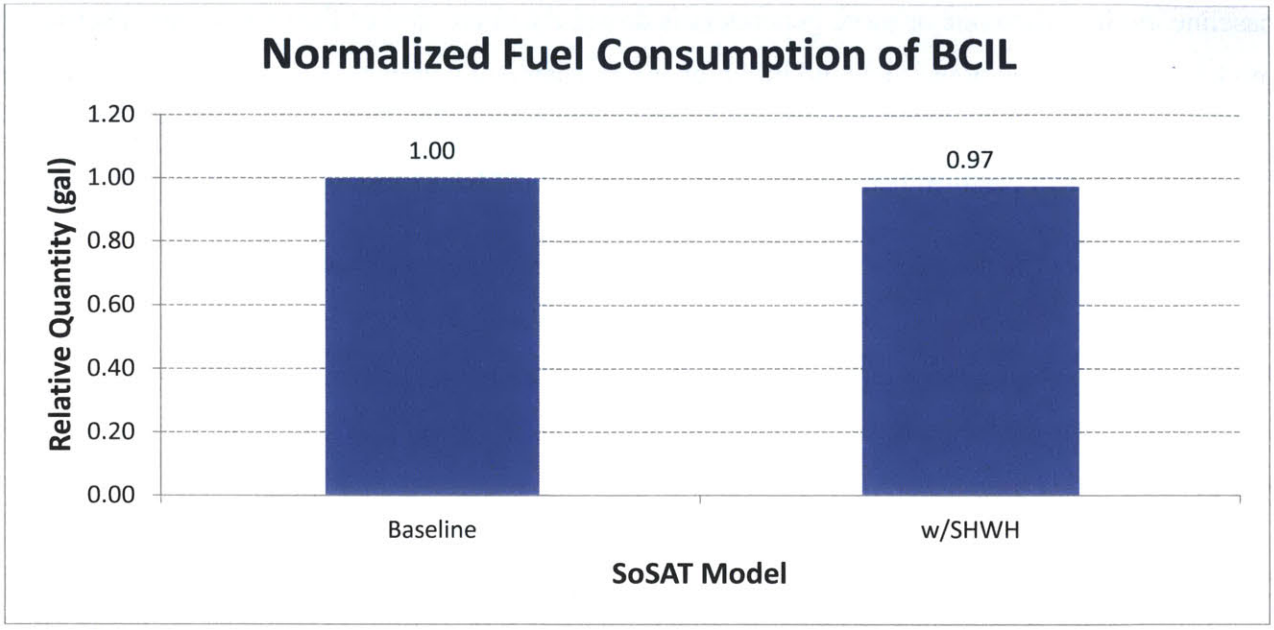


Figure 84. Comparison of Fuel Consumption of BCIL

While we would expect the total fuel consumption of the base camp to increase with the inclusion of an additional fuel-consuming component system, it has actually *decreased*. Herein lays the extreme power of modeling cities from a systems of systems perspective. While our basic intuition has led to an incorrect conclusion, the SoS model has successfully captured the intricacies of the interconnections between component systems and how they influence the functionality of the overall structure. By thinking instead about how the systems interface with each other, we can understand the difference between the output of the SoSAT model and the output of our mental model. Specific examples will illustrate this idea.

Looking at the fuel consumption of individual component systems gives a clearer picture of how adding the solar powered hot water heater actually contributes to overall base camp fuel consumption. The total fuel consumption during the 30 day simulation is shown in Figure 85 for the water heater for the shower and for three of the generators that help make up the micro grid.

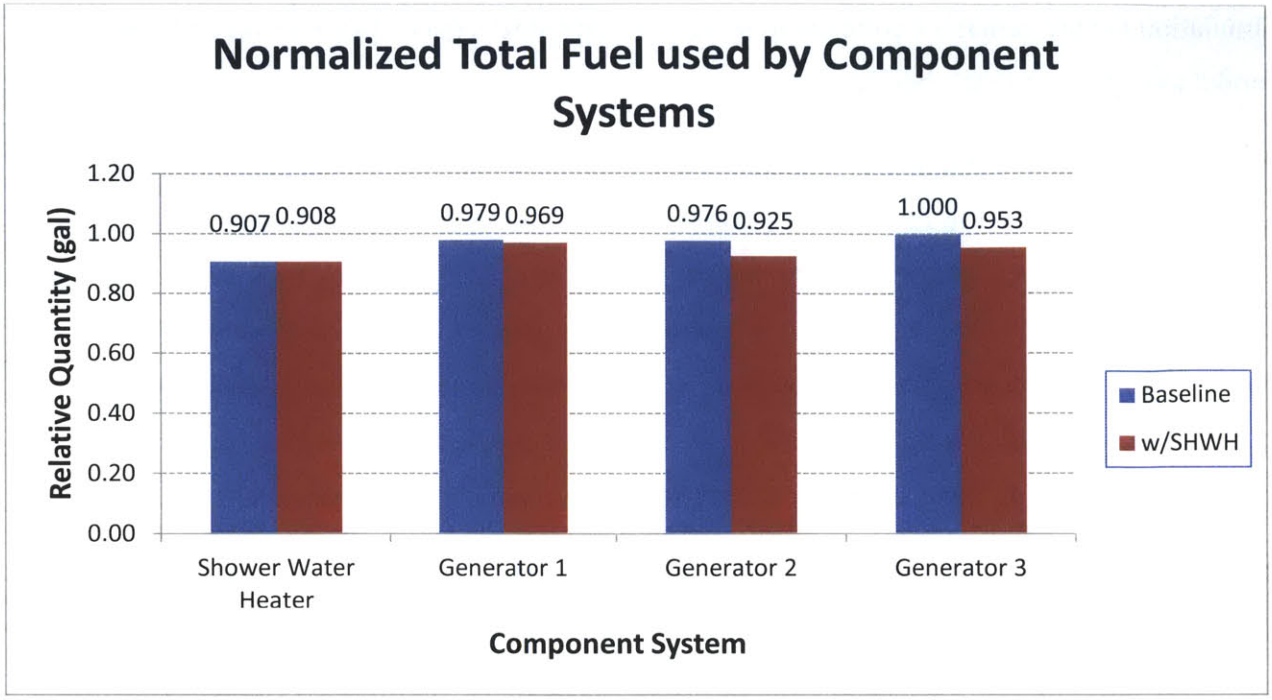


Figure 85. Comparison of Total Fuel Consumption, Component Systems

Earlier we acknowledged the increased performance exhibited by the micro grid when the solar powered hot water heater is included in the structure as a result of the reduced peak power demand of the base. The micro grid fulfilled a higher percentage of the component systems' power demand at any given time interval since the micro grid remained in its original configuration, with its original attributes. Therefore the component systems relying upon the micro grid for power also had increased availability over the duration of the simulation. Higher availability translated to a higher total operating time and consequently a higher consumption of any other resources they may have consumed. This behavior accounts for the increased fuel usage of the shower water heater (an approximately 0.24 gal increase before normalizing the values), which depends on electricity to run but uses fuel to heat the potable water.

While the fuel needed to operate the water heater for the shower increased with the new technology insertion, the fuel needed to power the generators actually decreases. Remember that we have essentially removed a power-consuming system when we replaced the sanitation system's need to heat potable water with the new solar powered hot water heater. This decrease in electrical demand was significant enough to lower the number of generators needed to power the micro grid during certain time intervals of the simulation. Total fuel consumption during the simulation for the generator component systems is therefore reduced in the technology insertion model as compared to the baseline model.

Now that the systems and interconnections have been explored from a systems of systems perspective, the impact to fuel consumption is easier to understand. By looking at component systems and how they interact with the micro grid, we can clearly see why total base camp fuel consumption is decreasing when the new technology is inserted into the baseline configuration. Yet though this behavior is obvious once the structure is examined from a systems of systems perspective, we would likely never have been able to determine this behavior from spreadsheet calculations which do not account for interactions between component systems.

5.11 Insights Gained from BCIL Case Study

From the modeling effort involved with the BCIL case study, a number of significant insights were gained into the process of systems of systems modeling and more importantly, the value provided by modeling cities from an SoS perspective. Crucially, the case study illustrated that modeling interconnections between systems reveals behavior that is sometimes counterintuitive at first glance, but sensible after thinking about the systems of systems structure. Without these types of SoS models, incorrect conclusions can easily be drawn through analyzing systems individually or in small groups, rather than understanding their place within the entire SoS structure. What follows is a concise list of these insights, in no particular order.

- The majority of system data is obtained through stakeholders, necessitating the need for sustained interaction.
- The bulk of the time needed to build a systems of systems city model is spent defining the baseline with stakeholders, obtaining adequate data, and creating the structure.
- The structure used in the SoS model dictates what kind of results will be available from the simulation (not taking into account post-processing efforts).
- It is helpful to also define systems by the functions they fulfill, since these functions are often more intuitive for stakeholders to understand than looking at results from component systems.
- The number of trials needed to minimize the variation in the simulations must be determined through an iterative process that is model specific.

- The impact of the availability of key resource-supplying systems (such as the micro grid) can clearly impact the operability of the systems to which it supplies resources.
- Adding new technologies can impact the overall SoS model in unforeseen ways, even when the resource production and consumption of the new component system is well understood. This was seen when the overall fuel usage of the base went down, even though the solar powered hot water heater that was added was a fuel-consuming system.
- Sensitivity analyses yield insight as to which component systems or groups of systems are constraining the overall structure.

6 Conclusions and Future Work

6.1 Summary of Findings

Findings from this research fall into two main categories—those obtained from the comparison of the city modeling software packages, and those gleaned from the BCIL case study. Both provide insight into the current state of city modeling, as well as critical aspects of SoS modeling and its useful features.

6.1.1 Comparison of City Modeling Software

Few software packages are available to model cities from a systems of systems perspective. Four programs (SimCity 4, CityOne, CityNet, and SoSAT) were evaluated from both a user perspective and a mathematical perspective to determine what characteristics and functionality are of utmost importance to city modeling software. These programs represented only a small effort of the current effort to model cities as a systems of systems. Other types of methods such as agent based models, cellular automata, and object oriented modeling are being used by researchers to model cities, though programs based around these tools are not yet available on the market.

Of those programs that are publicly available, some are designed as games, such as SimCity 4 and CityOne. While these games employ the desired systems of systems structure, they do not allow for enough user control to be used for serious modeling purposes. They also choose to forgo reality in certain instances in exchange for increased entertainment value. This was seen in the option for UFO attacks in SimCity 4. On the other hand, these games excel in their visual representation of cities. 3D graphics bring the cities to life and can help users understand connections between systems, particularly among the infrastructure layer.

Meanwhile, the programs that are designed specifically as modeling tools face their own challenges. User interfaces are often difficult to understand and require training to use properly. Component systems are usually defined by the user in these programs, which allows for increased flexibility and the power to model the city precisely as desired. However, this flexibility can easily lead to errors in user-entered data, particularly if the program doesn't have any way to check variable names or values for incorrect entries. Close attention must be paid to data entered by users to avoid these errors and maintain an accurate model.

The evaluation of the four programs yielded insight into which features are most necessary for successful SoS city modeling programs. The software must be able to handle multiple systems and equally importantly, be capable of modeling the interconnections between them. This includes transfer of resources such as water and power, but also physical connections and dependencies. While the ability to create models is important, there should also be a visualization component to aid in modeler and stakeholder understanding of how the SoS structure is functioning. The entire interface should be clear and easy to use, while still providing the flexibility needed for city modeling. User entered data must be validated somehow through the modeling process, and off course, the underlying mathematics of the software must also be correct and validated. A model without accurate equations and data is useless for analyzing cities and how changes may propagate through the systems of systems structure.

None of the programs evaluated provided a comprehensive solution for city modeling. SoSAT was chosen primarily for its systems of systems-specific modeling capabilities. The software has already been used and its underlying mathematics validated through applications to other systems of systems modeling efforts. While its use for modeling cities was novel, the software is designed to model *any* SoS structure and this was not a limitation. Though SoSAT lacked means of visualizing the structure, three visualization methods were developed for use with the model and successfully captured the various considerations of SoS modeling through graphical representations.

6.1.2 BCIL Case Study

The BCIL case study was an effort to model the Army Base Camp Integration Laboratory at Fort Devens, MA. The BCIL has two identical base camps, one held as a baseline and the other in which new technologies and configurations are tested. Each camp is designed to support 150 personnel and is based on the Force Provider base camp in which component systems are packaged into containerized systems for easy transport and set-up. The goal of the case study was to build a systems of systems model of the baseline camp. The case study involved working with the stakeholders to build the baseline model of the camp, two sensitivity analyses, and two technology insertion models.

The case study began with close consultation with the stakeholders. Working together, all of the component systems of the baseline model were defined and data was obtained that included historical data, technical specifications, and SME input. One advantage to considering base camps as small, temporary cities was that the model boundary was clearly defined. The stakeholders were also involved in the process of creating the baseline model's structure. The hybrid structure proposed in this thesis was used for the baseline. Though this structure is universally applicable, the final choice of structure (layered, geographical, or hybrid) is at the discretion of the modeler and should be based on how the software being used handles aggregated results. The hybrid structure allows resource consumption and production to be aggregated at every level of the structure, including at the level of the entire base camp.

The structure was defined within the SoSAT model and each system was added with appropriate attributes. All interconnections between component systems were defined through supply connections and distribution networks. Besides defining each component system, functions were also added to the model to represent the main contributions of the component systems or groupings of systems. These included functions such as providing electrical power to the base, providing personal hygiene, providing subsistence, etc. Functions were added since they offer a more intuitive way for stakeholders to understand how the model is performing. The results of functions are based on the component systems that define them.

Three types of visualization were used to graphically describe the SoSAT model. These included an aerial view of the base camp, an adjacency matrix of the interconnections between component systems, and a network representation of the base's component systems. Each visualization added critical information to the understanding of the model and ideally should be used in combination to create a complete representation. However, any of the three could be used to highlight particular aspects of the model, whether it be the level of interconnectivity among systems, the geographical relationships, or how systems are grouped together functionally.

Once the baseline model was built, the simulation was run for 720 hours (30 days). The model included variation achieved through applying distributions to various rates within the model (failures rates, consumption rates, production rates, etc.) and also by varying the number of personnel on base. Personnel variations were designed to mimic those leaving the base for missions. Varying personnel changed the usage of component systems, and in turn, the demand and production of subsequent resources. Types of distributions available included exponential, Weibull, normal, uniform, triangular, and lognormal distributions.

The model was validated by examining the output of the simulation. Excluding variability, SoSAT model outputs aligned with expected output based on spreadsheet calculations of the input data. Variability was examined and determined to be in line with values expected from varying the number of personnel on base and the stochastic nature of the model. An in-depth analysis of the baseline model followed, with 100 trials averaged to produce output with an acceptable amount of variability among trials. The measure of variability was based on calculating the standard error of the mean. For this simulation effort, a standard error of less than 0.05% was desired. The acceptable amount of variation considered acceptable is determined through consultation with stakeholders and based on the purpose of the model.

Many types of results were presented for the baseline model, including availability over time; functional availability over time; consumption, production, and inventory levels of resources over time; operability results; and aggregated results for the duration of the simulation. All results were examined at different levels of aggregation, including the individual component system level, groupings of component systems such as containerized systems, and at the overall base camp level. Availability of the BCIL was near 100% for each time interval during the simulation, which is expected since we know the base camp is up and functioning in real life. Small drops in availability in both the overall structure and at the level of individual systems were caused by changes in the availability of the micro grid. During a few time intervals of the simulation, the micro grid was incapable of producing enough power to meet the demands of component systems. Micro grid availability was in turn directly impacted by the number of maintenance personnel on base providing maintenance and services to the generators comprising the micro grid.

Two sensitivity analyses were conducted to examine the impact of adding or removing a maintenance person from the baseline configuration, and to assess the changes brought about by delaying the time between deliveries of potable water to the base camp. Removing a maintenance person lowered availability for the generator systems, the micro grid, and for component systems such as laundry systems. Adding a maintenance person improved availability of generator systems and led to energy-consuming component systems experiencing less down time during the simulation. Expanding the time between the deliveries of potable water (LOGPACs) revealed key constraints in the model. When the time was extended by just 24 hours, both the kitchen and laundry containerized systems experienced resource depletions. Lengthening the time interval by 48 hours only exacerbated the situation. Without potable water deliveries, these systems didn't have the necessary resources to operate and were essentially in a failed state until the next delivery. None of the other potable water-consuming systems on base experienced resource depletions. Knowing which systems provide constraints to the system is critical to knowing where to focus improvement efforts.

To address the constraint created by the laundry containerized system, a laundry water reuse system was added to the baseline model as the first of two technology insertion cases. The new system processed gray water from the washing machine, filtered it back to potable water quality, and fed it back into the laundry system. The laundry water reuse system fulfilled 70% of the potable water needs of the laundry containerized system, significantly reducing that system's dependence on potable water deliveries.

The other technology insertion case looked at the addition of a solar powered water heater to heat the water used in the kitchen's sanitation system. This technology insertion case in particular provided a perfect example of the value of systems of systems modeling. The solar powered hot water heater requires fuel to operate. Initially, it would seem that adding the system would increase the fuel usage of the base camp, something that must be considered in the tradeoff between using less energy to heat the water and having to use more fuel. However, once the new system was inserted into the baseline model, the overall fuel consumption actually *decreased.* The electricity saved by not having to heat the water by conventional means was enough that the energy demand at each time interval could be met by the micro grid employing fewer generator systems. Since fewer generators were operating, their fuel consumption decreased, and this decrease was greater than the additional fuel used by the solar powered hot water heater. Adding the new system therefore decreased both electricity usage and fuel usage with no negative impacts to the overall base camp. Without the systems of systems perspective, we would have still assumed a fuel increase and may have incorrectly assessed whether or not this system was worth adding to the baseline.

6.2 Lessons Learned

The BCIL case study yielded a number of important considerations for the future, including improvements to the SoSAT modeling software and how to apply the methodology from the case study to traditional cities.

6.2.1 SoSAT Improvements

SoSAT excelled in its designated ability to capture multiple component systems within a defined structure and the many and varied interconnections between those systems. The flexibility offered by utilizing the hybrid structure allowed small changes to be easily implemented and assessed for the sensitivity analysis. Likewise, adding systems for the technology insertion cases was also a simple matter of defining the new system's attributes and adding any necessary supply connections. The only real obstacle to using SoSAT remained in its lack of visualization capabilities.

The lack of visualization was dealt with by creating the three auxiliary visualization methods. While these methods are relatively straightforward to generate, they still require an extra time commitment and can introduce new errors if user data is entered incorrectly, particularly for the adjacency matrix and the network representation. Having these automatically generated by the SoSAT model would lessen the demand on the modeler as well as ensure the data used for both the model and for the visualizations aligns.

6.2.2 Applying SoS Modeling to Traditional Cities

The BCIL presented an idealized version of traditional city modeling. The systems were known and well-defined, the camp was isolated with a defined boundary, and the size of the camp/number of component systems was not prohibitively large. There are three main differences between the base camp study and traditional cities that need to be addressed in order to be able to apply this modeling methodology successfully to larger, permanent cities. These considerations include a building block approach for modeling large cities, accurately capturing the starting conditions of component systems for the simulation, and including human considerations.

SoS models have already been established as exhibiting a hierarchical structure. This hierarchy should be exploited when creating larger models. Modeling every component system and every interconnection would prove nearly impossible. However, using a modular approach to model large cities should prove useful. SoS models could be made for a residential home, factory, office building, etc. These models should capture critical component systems, their attributes, and the interconnections between the systems. They should also capture any interface between the structure and the outside world. For example, water and energy entering or leaving the structure needs to be defined. These models could then be used as building blocks to create city blocks, which in turn could be used to build out entire neighborhoods, and finally the entire city. Interconnections represented by supply connections and distribution networks must be properly maintained and aggregated through every step of the modular approach. This approach is a suggestion based on qualities observed in the BCIL case study, but has not been tested. More research would need to be done to see whether this method can accurately capture the complexity of a traditional city from an SoS perspective.

Every component system of the BCIL baseline model was considered to be at optimal operating conditions at the beginning of the simulation. All resources started at maximum capacity, all systems were operable and not experiencing failures, and all interconnections were functioning properly. This is an acceptable assumption for a recently deployed base camp, whose systems have all been tested before deployment and presumably set up according to specifications once on the site. However, these assumptions would not be appropriate for traditional cities. Cities are in a constant state of flux, and the simulation would almost undoubtedly start at some point while the city has already been operating for many years, decades, or even centuries. Care must be taken to set initial inventory levels and system operability to values that realistically represent the state of the city. Alternatively, a burn-in period can be added to the simulation to allow the systems to naturally reach a steady state more representative of how the city is actually functioning. Both varied initial conditions and burn-in periods may be implemented using SoSAT. These capabilities should be considered necessary if using an alternative software package.

As mentioned during the BCIL case study, human behavior was not explicitly included in the baseline model, except where it pertained to maintenance of component systems. The movement of humans on and off the base was included to show how the number of personnel present impacted system utilization, but no behavioral variation was addressed. This was mainly because of the rigid lifestyle of military personnel. In traditional cities, this rigid behavior does not exist and more aspects of human behavior must be included to more accurately capture how the city functions. Various daily scenarios should be created for different subsets of the population (i.e. office workers, night-shift personnel, students, individuals who work-from home, etc.). Percentages could then be assigned to each scenario to designate what fraction of the population follows that approximate schedule. Additionally, variation must be applied to these schedules to account for divergences from normal behavior. Including fatigue rates, assigning work skills, and how much each person can impact the city (i.e. what level of authority they possess) would also help to capture the human aspect of cities. This is another area of traditional city modeling that will require more research and assessment through additional case studies.

6.3 Future Work

The BCIL case study was the first step of an ongoing modeling project with the Army. After the conclusion of this thesis, more technology insertion cases will be examined, as well as case studies to examine how reconfiguring some of the systems would impact base camp operability.

An ongoing technology insertion case of interest is the use of solar panels. While the panels themselves are relatively straightforward to implement, and have in fact already been used as part of the solar powered hot water heater, the storage of generated energy presents a larger problem. Batteries would be needed to store the power and these would somehow need to be integrated into the micro grid network. Multiple technology insertion cases will be looked at to determine how to best include this new technology within the existing framework of the base camp systems of systems structure.

There is also interest in using natural resources to provide potable water to the base camp. This would eliminate the need for LOGPAC deliveries and significantly reduce the risk these deliveries place upon military personnel. Additionally, the fuel reduction from eliminating deliveries of potable water by truck would save on expenses. Alternative options to LOGPAC deliveries could include drilling wells and installing water filtration systems to purify the ground water. However, fuel savings and improved troop safety obtained from eliminating deliveries would need to be weighed against the increased risk of chemical warfare geared towards tainting water sources. Additionally, any technology and infrastructure needed to drill the wells would need to be small, portable, and easy to use in a variety of climates, including harsh weather conditions.

Stakeholders should also look at processing the waste water from the kitchen containerized system back to potable water quality, as is done for the shower systems and was examined as a technology insertion case for the laundry system. Additionally, they could work on improving the efficiency of these water reuse systems to further reduce the need for additional potable water deliveries. Finally, new configurations should be considered, including using larger water blivets for storage and creating networks to link multiple water blivets with multiple systems. With the current configuration where each water-consuming system is supplied by a unique potable water blivet, the SoS structure will always be limited by whatever system is consuming potable water at the highest rate. Linking high and low potable-water consuming systems to a bank of potable water blivets through a distribution network would allow water supplies to be used more efficiently. Another option would be to use gray water directly for latrines, thereby freeing up both potable water supplies and also processing space in the water reuse systems.

From a software perspective, visualization methods should continue to be assessed. In addition to the visualizations previously discussed, models may be able to take advantage of existing technologies such as Google Earth. Using the Keyhole Markup Language (KML) in conjunction with Google Earth or Google Maps could provide enhanced aerial visualizations for city modeling. Basic features implementable through KML include adding placemarks, descriptions, ground overlays, paths, and polygons to images generated by Google Earth or Google Maps [28]. These visualizations should be a key consideration regardless of the software being used as they present a powerful method of understanding how the system of systems structure is functioning and are often easier than trying to sort through the complexly detailed SoS mathematical model.

Lastly, as more SoS city models are built, we need a way to compare the complexity of the models and thereby the cities themselves. This can be accomplished by computing a structural complexity metric which takes into account the complexity of each individual component system, the complexity of each connection between component systems, and the topological complexity [29]. The complexity metric can then be calculated as

$$C = c_1 + c_2 c_3 \tag{16}$$

where c_1 represents the summation of component complexities, c_2 represents the summation of the connection complexities, and c_3 represents the topological complexity. The topological complexity can be found by calculating the graph energy of the binary adjacency matrix that was constructed as part of the visualization effort. Calculating component and connection complexities however will require extensive consultation with stakeholders and subject matter experts. In the future, guidelines for classifying the complexity of components and connections should be explicitly developed and adhered to in order to facilitate comparisons between multiple systems of systems city models.

Appendix A-Current Calculation Capability in CityNet

Table 7. CityNet Calculation Capability

Layer	Calculation	Matlab Name
Transportation	Emissions produced by transportation	TransportationEmissionsProduction
Transportation	Amount of energy used by transportation system	TransportationEnergyUse
Transportation	Counts the cost of fixed expenses for the transportation system	TransportationFixedExpense
Transportation	Amount of land used by transportation system	TransportationLandUse
Transportation	Recurring cost for transportation system	TransportationRecurringExpense
Transportation	Amount of water used within transportation system	TransportationWaterUse
Waste	Usable compost and biogas extracted from restwaste output	BiologicalTreatment
Waste	Amount of leachate and gas generated by the landfill process	Landfill
Waste	Amount of materials recovered from waste management process	Materials
Waste	Usable recyclable materials available and amount of residues	MRFSorting
Waste	Usable cRDF and dRDF extracted from restwaste output and amount of residues	RDFSorting
Waste	Outputs of the thermal treatment (incineration, RDF burning, and PPDF)	ThermalTreatment
Waste	Total amount of commercial waste generated by specific waste stream	TotalCommercialWaste
Waste	Total amount of bulky waste delivered by residents to system	TotalDeliveredWaste
Waste	Total amount of residential waste generated by the city	TotalResidentialWaste
	Number of residents in a cell	NumberResidentsCell
	Number of residents in the city	NumberResidentsCity
Water	Wastewater from the commercial water output	CommercialWasteWater
Water	Cost of water produced by a desalination plant	computeCostofWater
Water	Energy, land, and cost requirements for membrane biological reactor	MBRfacility
Water	Plots cost of water for varying inputs	plotWaterCostAnalysis
Water	Wastewater from the residential water output	ResidentialWasteWater
Water	Energy, land, and cost requirements for desalination plant	SWROfacility

Water	Total amount of commercial water demand in city	TotalCommercialWater
Water	Total amount of residential water demand in city	TotalResidentialWater
Energy	Energy generated by biomass power plant, capacity, and fuel required	BiomassEnergy
Energy	Energy generated by CSP station and capacity	CSPStation
Energy	Energy generation by cell?	EnergyGeneration
Energy	Land area used by energy system	EnergyLandUse
Energy	Energy generated by hydropower station and capacity	HydropowerStation
Energy	Energy generated by natural gas plant, capacity, and fuel required	NaturalGas
Energy	Energy generated by PV station and capacity	PVStation
Energy	Energy generated by wind farm and capacity	WindFarm

References

[1] O. L. de Weck, D. Roos, and C. L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World.* Cambridge: The MIT Press, 2011, p. 213.

[2] J. Johnson, H. Meyer, E. Stolk, and E. T. Springer, "Complexity Theories of Cities Have Come of Age," *Cities*, pp. 153–172, 2012.

[3] E. Taniguchi, R. G. Thompson, and T. Yamada, "Emerging Techniques for Enhancing the Practical Application of City Logistics Models," *Procedia - Social and Behavioral Sciences*, vol. 39, pp. 3–18, Jan. 2012.

[4] D. M. Gann, M. Dodgson, and D. Bhardwaj, "Physical-digital integration in city infrastructure," *IBM Journal of Research and Development*, vol. 55, no. 1&2, pp. 90–99, 2011.

[5] D. L. Alderson and J. C. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 4, pp. 839–852, Jul. 2010.

[6] A. Attwood, M. Merabti, P. Fergus, and O. Abuelmaatti, "SCCIR: Smart Cities Critical Infrastructure Response Framework," *2011 Developments in E-systems Engineering*, pp. 460–464, Dec. 2011.

[7] T. L. Friesz, R. Mookherjee, and S. Peeta, "Modeling Large Scale and Complex Infrastructure Systems as Computable Games," *Network*.

[8] L. Gao, B. Durnota, Y. Ding, and H. Dai, "An agent-based simulation system for evaluating gridding urban management strategies," *Knowledge-Based Systems*, vol. 26, pp. 174–184, 2012.

[9] A. Crooks, C. Castle, and M. Batty, "Key challenges in agent-based modelling for geo-spatial simulation," *Computers, Environment and Urban Systems*, vol. 32, no. 6, pp. 417–430, Nov. 2008.

[10] Y. Chen and J. Lin, "Modeling the self-affine structure and optimization conditions of city systems using the idea from fractals," *Chaos, Solitons & Fractals*, vol. 41, no. 2, pp. 615–629, Jul. 2009.

[11] P. S. Satsangi, D. S. Mishra, S. K. Gaur, B. K. Singh, and D. K. Jain, "Systems dynamics modelling, simulation and optimization of integrated urban systems: a soft computing approach," *Kybernetes*, vol. 32, no. 5/6, pp. 808–817, 2003.

[12] M. Pooyandeh, S. Mesgari, A. Alimohammadi, and R. Shad, "A Comparison Between Complexity and Temporal GIS Models for Spatio-temporal Urban Applications," *Complexity*, pp. 308–321, 2007.

[13] M. Al-Hader, A. Rodzi, A. R. Sharif, and N. Ahmad, "SOA of Smart City Geospatial Management," *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, pp. 6–10, 2009.

[14] E. M. Huestis and J. L. Snowdon, "Complexity of legacy city resource management and value modeling of interagency response," *IBM journal of research and development*, vol. 55, no. 1–2, 2011.

[15] J. D. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World.* Irwin/McGraw-Hill, 2000, p. 982.

[16] D. J. Marceau and I. Benenson, *Advanced Geosimulation Models*. Bentham Science Publishers, 2011, p. 155.

[17] F. Tsai, T. Teo, L. Chen, and S. Chen, "Construction and Visualization of Photo-Realistic Three-Dimensional Digital City," *Agricultural Economics*, 2009.

[18] "IBM INNOV8: CityOne," 22-Apr-2010. [Online]. Available: http://www-01.ibm.com/software/solutions/soa/innov8/cityone/index.html. [Accessed: 04-Mar-2012].

[19] Reikhardt, "St Andrews," 2011. [Online]. Available: http://sc4devotion.com/forums/index.php?topic=12679.20.

[20] Electronic Arts Inc., "Software and Documentation." p. 88, 2003.

[21] B. (USAG-N. P. A. Reinert, "Base Camp Integration Lab opens at Fort Devens," 2011. [Online]. Available: http://www.army.mil/article/60473/.

[22] "Product Manager Force Sustainment Systems," 2013. [Online]. Available: http://peocscss.tacom.army.mil/PdMFSS.html. [Accessed: 14-Mar-2013].

[23] B. (PEO C. & C. Good, "Shower Water Reuse System to be force multiplier at FOBs," 2011. [Online]. Available: http://www.army.mil/article/65812/. [Accessed: 14-Mar-2013].

[24] M. E. Smoot, K. Ono, J. Ruscheinski, P.-L. Wang, and T. Ideker, "Cytoscape 2.8: new features for data integration and network visualization.," *Bioinformatics (Oxford, England)*, vol. 27, no. 3, pp. 431–2, Feb. 2011.

[25] R. Smaling and O. de Weck, "Assessing risks and opportunities of technology infusion in system design," *Systems Engineering*, vol. 10, no. 1, pp. 1–25, 2007.

[26] E. S. Suh, M. R. Furst, K. J. Mihalyov, and O. De Weck, “Technology Infusion for Complex Systems : A Framework and Case Study *,” *Systems Engineering*, vol. 13, no. 2, pp. 186–203, 2010.

[27] Eemax, “Eemax Electric Tankless Water Heaters Venture into the Warzone.” [Online]. Available: http://eemax.com/assets/files/BattleReady_FINAL.pdf. [Accessed: 18-Apr-2013].

[28] G. Developers, “Keyhole Markup Language: KML Tutorial,” 2012. [Online]. Available: https://developers.google.com/kml/documentation/kml_tut. [Accessed: 28-Apr-2013].

[29] K. Sinha and O. De Weck, “Structural Complexity Metric for Engineered Complex Systems and its Application,” in *In 14th International Design Structure Matrix (DSM) Conference*, 2012, pp. 1–6.