Agent-Based Techniques For National Infrastructure Simulation

by

Kenny Lin

S.M. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 2002

B.S. Electrical Engineering
U.S. Naval Academy, 2000

Submitted to the Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Civil and Environmental Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

Author ....
Department of Civil and Environmental Engineering
August 16, 2002

Certified by ....
John R. Williams
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ....
Oral Buyukozturk
Chairman, Department Committee on Graduate Students

Agent-Based Techniques For National Infrastructure Simulation

by

Kenny Lin

Submitted to the Department of Civil and Environmental Engineering
on August 16, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Civil and Environmental Engineering

Abstract

Modern society is dependent upon its networks of infrastructure. These networks have grown in size and complexity to become interdependent, creating within them hidden vulnerabilities. The critical nature of these infrastructures has led to the establishment of the National Infrastructure Simulation and Analysis Center (NISAC) by the United States Government. The goal of NISAC is to provide the simulation capability to understand infrastructure interdependencies, detect vulnerabilities, and provide infrastructure planning and crises response assistance.

This thesis examines recent techniques for simulation and analyzes their suitability for the national infrastructure simulation problem. Variable and agent-based simulation models are described and compared. The bottom-up approach of the agent-based model is found to be more suitable than the top-down approach of the variable-based model. Supercomputer and distributed, or grid computing solutions are explored. Both are found to be valid solutions and have complimentary strengths. Software architectures for implementation such as the traditional object-oriented approach and the web service model are examined. Solutions to meet NISAC objectives using the agent-based simulation model implemented with web services and a combination of hardware configurations are proposed.

Thesis Supervisor: John R. Williams
Title: Associate Professor of Civil and Environmental Engineering

Acknowledgments

I would like to thank my advisor, Professor John R. Williams, for his ideas and time spent working with me on this project at MIT. His flexibility has made this thesis possible. I would like to thank Professor John B. Miller and Professor Feniosky Pena-Mora for their tutelage and guidance within the Civil and Environmental Engineering Department. I would also like thank the Department's administrative staff for making it possible to complete this degree within the given time constraints.

I would like to thank Mike Crafts and David Lee for their advice and friendship as we shared the MIT experience. Finally, I would like to thank my girlfriend, Yingli Zhu, and my family, Chiun-wen, Su-ching, and Frank for their love and support in all my endeavors.

Contents

List of Figures

Chapter 1

Introduction

Our modern society and economy is built upon a foundation of infrastructure. This foundation provides food, water, energy, transportation, communications, and countless other services, many of which are critical to the livelihood of a nation and its peoples. As infrastructure develops to support population and economic growth, historically separate systems have expanded to merge into networks of interdependent systems. These combined systems and a growing dependence on technology create new vulnerabilities in critical areas which need to be addressed [4].

1.1 Infrastructure Vulnerabilities

Critical infrastructure is defined by the United States Government in Presidential Decision Directive 63 as "physical and cyber-based systems essential to the minimum operations of the economy and government [12]."

Executive Order 13010, signed by President Clinton in 1996, emphasizes eight sectors whose services are vital to the defense and economic security of the United States. These are electrical power, gas and oil production, storage and delivery, telecommunications, banking and finance, water supply systems, transportation, emergency services, and government operations [11].

By this definition, infrastructure directly affects the lives of every citizen and almost every sector of the economy.

Historically, infrastructure systems have been physically separated and independent of each other. The need and capability to share resource demand and supply have been nonexistent. But increasing demand and improvements in technology have led to many geographic systems merging for greater efficiency and automation across distance, resulting in horizontal interdependencies within an infrastructure type. In addition, the growing complexity of our infrastructure needed to feed the multiple requirements of our modern industrial society has created vertical interdependencies of complimentary resources and services across different infrastructure types.

For example, resources such as electricity are easily transportable to fulfill non-local demands and have become networked with distributed power technologies to form the power grid. The ability to control power distribution over a wide geographical area creates a vulnerability and potential for power disruption over that same area by malicious intent. Other sectors such as banking and finance are dependent upon electricity for their operation and would consequently suffer from any disruption of service.

The high level of horizontal and vertical interdependencies within a nation's infrastructure creates new vulnerabilities, many of which are not fully recognized. These vulnerabilities are compounded by the reliance of our livelihoods on a stable and functional infrastructure. The potential for damage caused by unintentional events or an intentional attack is greatly magnified by these dependencies. This creates very attractive targets for low-risk, high-return attacks by hostile forces.

1.2 Need for Infrastructure Simulation

The vertical and horizontal complexity of our critical infrastructure systems are not fully understood. There may exist hidden interdependencies where the effects of failures or a crisis in one sector on others may surface only when a crisis occurs. Thus, the extent of threats against infrastructure are not fully recognized.

The ability to simulate large infrastructure systems can potentially reveal these hidden interdependencies. Such a simulation could identify new vulnerabilities and threats. This information would be a valuable planning tool. An extension of this ability would be the capability to update and perform the simulation in real-time. This creates an interative process and adds decision-making support functionality to provide a more effective crises response ability.

A strategic simulation involving government and industry leaders to examine and evaluate proposed policies for the U.S. President's Commission on Critical Infrastructure Protection (PCCIP) was carried out in 1997 [24]. This interaction showed that more government regulation and involvement in the daily task of infrastructure protection was not desired by industry. Efforts to protect infrastructure systems were being taken. However, increased education and communication on threat specifics and vulnerabilities from the government should be provided, as well as research and development of infrastructure protection technology. "Industry invests in security and protective mechanism to meet existing threats, but does not have the information to reduce vulnerabilities to future threats [24]." Infrastructure simulation partially satisfies this need.

1.3 Contributions of this Thesis

This thesis explores solutions for the implementation of a comprehensive national infrastructure simulation. This simulation will serve as an integral part of the U.S. Government's critical infrastructure protection effort. The simulation problem is defined and different simulation techniques analyzed for suitability.

The creation of variable and agent-based simulation models is described. The top-down approach of the variable-based model is compared to the bottom-up approach of the agent-based model to determine their suitability to the infrastructure simulation problem. Hardware requirements and different implementation configurations to meet the simulation's computational needs are explored. Supercomputer and distributed, or grid computing solutions are described and compared. Software architectures for implementation such as the traditional object-oriented approach and the web service model are examined. Finally, several implementation strategies using an agent-based simulation with web services technology are proposed for creating a flexible and capable national infrastructure simulation.

Chapter 2

National Infrastructure Simulation Efforts

The September 11, 2001 attacks on the New York City World Trade Center in the United States of America have shifted political attitudes towards placing a greater importance on national infrastructure surety in the United States. This has accelerated existing efforts to develop a comprehensive infrastructure simulation capability. This chapter will describe these efforts and their goals.

2.1 National Infrastructure Simulation and Analysis Center

Executive Order 13228 signed by President George W. Bush on October 8, 2001, established the Office of Homeland Security within the Executive Office of the President [13]. Current efforts are being taken with the Homeland Security Act of 2002 to elevate this to a cabinet-level agency named the Department of Homeland Security to unify government efforts in domestic security under one organization [15].

Currently, responsibility for domestic security is dispersed amongst over 100 different organizations [14]. The Department of Homeland Security seeks to consolidate and coordinate these efforts within four divisions, one of which is responsible for Information Analysis and Infrastructure Protection as seen in Figure 2-1 [15]. This division would be responsible for the comprehensive evaluation of infrastructure vulnerabilities. From this initiative, the National Infrastructure Simulation and Analysis Center (NISAC) was funded and created with a charter "To serve as a source of national competence to address critical infrastructure protection and continuity through support for activities related to counterterrorism, threat assessment, and risk mitigation [23]."

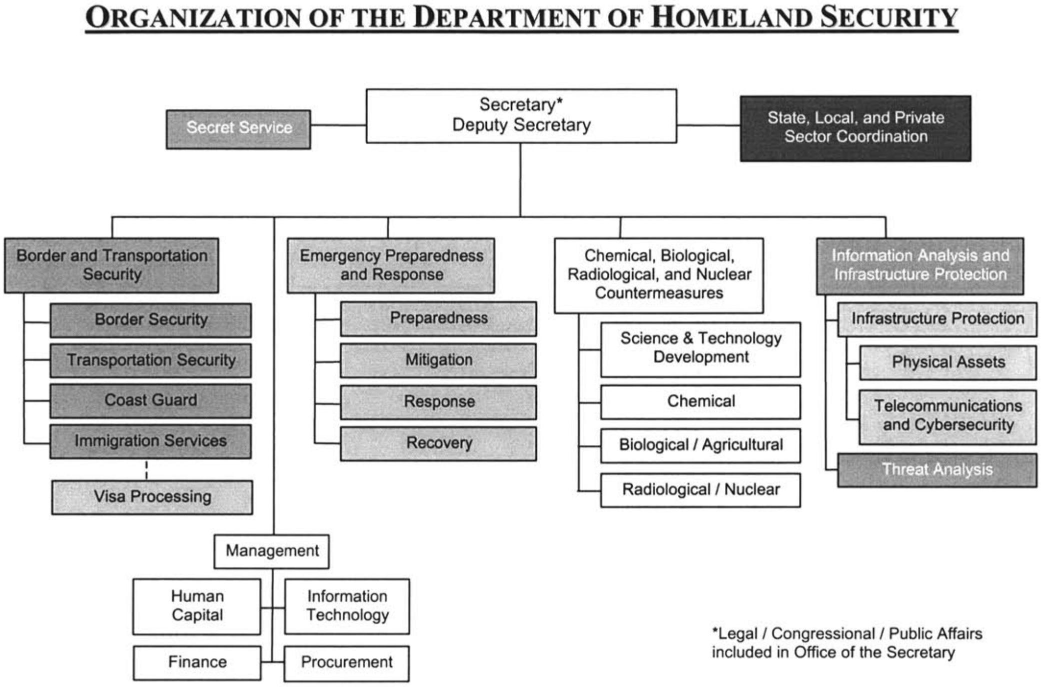

Figure 2-1: Department of Homeland Security Organization Structure

2.1.1 Mission and Objectives

NISAC is comprised of a partnership between Sandia National Laboratories and Los Alamos National Laboratory with the possible addition of partners from other national labs, private industry, and universities. Both partners have an established modeling and simulation capability as well as high performance computing resources to support this [18]. Their simulation expertise is well established with current projects to simulate specific infrastructures, but none offer a comprehensive capability to simulate all critical infrastructures.

NISAC's mission is to provide simulation capability to understand how infrastructure operates, identify vulnerabilities, determine consequences of outages, and optimize protection strategies [17].

More specific objectives [17] [19] as seen by Sandia and Los Alamos are to:

- Leverage the existing capabilities of the NISAC partners to provide leadership in critical infrastructure interdependencies modeling, simulation, and analysis;
- Establish a virtual capability that will provide a portal for nation-wide remote access and communications to infrastructure-related modeling, simulation, and analysis capabilities for the nation's leaders, policy makers, and infrastructure owners;
- Move toward a predictive capability that uses science-based tools to understand the expected performance of interrelated infrastructures under various conditions;
- Provide simulation and analysis capabilities to a wide range of users that will enhance the understanding of interdependencies and vulnerabilities of the national infrastructures and establish priorities and optimized mitigation strategies for protecting the infrastructures;
- Provide an early indications warning system to identify immediate threats, proactively protect threatened infrastructures, and provide information to first responders;
- Provide decision-makers the ability to assess policy and investment options that address near and long-term critical infrastructure needs;
- Provide education and training of public and private decision makers on how to cope effectively with crisis events through war-gaming and interagency planning and rehearsal;

- Provide reconstruction planning and real time crisis support in times of emergency;
- Provide an integrating function that includes interdependencies; bring disparate users and information providers and individual infrastructure sector leaders together.

These objectives show the potential value of NISAC and are important for the analysis of simulation technologies and implementation configurations and architectures.

Chapter 3

Techniques for Infrastructure Simulation

The computer simulation of infrastructure can be simplified into three steps: creating a model, establishing relations and rules between parts of the model, and performing calculations to yield desired results. However, there are several approaches for each step, each with different advantages and disadvantages for modeling a comprehensive infrastructure system. This chapter will discuss different ways to model complex infrastructure networks.

3.1 Choosing a Simulation Model

There are many possible models for any given system. They differ in the level of abstraction and mathematical or computational approach. All models abstract features of the system, losing some information. Thus, a more complex model will be more difficult to build but will result in greater accuracy [30].

The difficulties in constructing a complex model include greater raw data requirements, and greater difficulty in validating, verifying, and calibrating the model. Validation checks for accuracy, to see if the simulation produces realistic results. Verification checks for precision, to see if the simulation is implemented as desired. Calibration optimizes the simulation for more realistic results [1].

3.2 Direct Simulation

One simulation approach is the direct, or variable-based, simulation. This approach establishes a system that reacts to inputs based on its system dynamics to produce an output. The variable-based simulation is a macro, or top-down, approach to modeling a system, meaning that the system is explicitly defined beginning at the highest level of abstraction. Detail is added to the system model until the desired level of abstraction is reached.

The input variables to this system may be controllable or uncontrollable. Decision variables are created and given constraints. A measure of system performance is defined, and the system is given an objective function. This function is composed of equations of hypothesized relations and computes the output of the simulation [1]. This simple architecture is shown in Figure 3-1.

The variable-based approach is well suited for systems driven by discrete events and can be implemented at different levels of abstraction. For example, a traffic system may be modeled to compute the outcomes of interactions between individual vehicles. Characteristics such as speed, direction, and position for each vehicle would be required and each iteration would output a new speed, direction, and position. However, it is also possible to abstract individual vehicles and simulate traffic as shifting traffic densities as vehicles move from one area to another. This abstraction is advantageous when faced with simulating a large network.

3.2.1 Infrastructure Simulation Suitability

The national infrastructure problem requires the simulation of even larger networks. A variable-based model of the national infrastructure is extraordinarily complex. To build this model, it is necessary to define all contributing variables at each abstraction level, define decision variables and constraints, and define all relationships and objective functions within the system.

This model would be difficult to create because the characteristics and behavior of the system need to be known in order to process the inputs. This includes

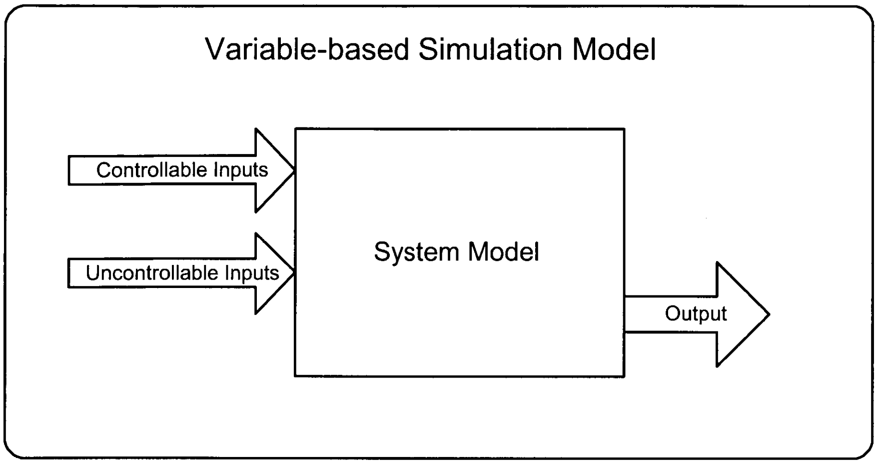

Figure 3-1: Variable-based Simulation Architecture

interdependencies that are unknown and that the simulation is tasked to discover. Thus, performing this simulation using variable-based methods without prior knowledge would require either a very high level of abstraction or a very complex, yet possibly inaccurate, system definition requiring enormous amounts of data. Neither is desirable.

A high level of abstraction will hide some interdependencies and vulnerabilities below that level. Discovery of these relationships is a primary objective of NISAC. When the building block is too large, anything smaller become folded into higher-level results. Thus, any interdependencies will be lost in the output.

Attempting to simulate infrastructure directly at a low level of abstraction creates an enormous data collection problem. It would be difficult to use the simulation in a dynamic decision making process or war-gaming if each iteration required a significant amount of effort to customize the data set. Therefore, the variable-based approach does not scale well to handle the national infrastructure simulation problem.

3.3 Agent-Based Simulation

Another approach to modeling national infrastructure is the use of agents, in an agent-based simulation model. The agent-based model is a micro, or bottom-up, approach to modeling a system, meaning that the system is implicitly defined through the definition of its subparts. Although the system itself is not defined, the model is formed by a collection of agents, all of which are well defined. An agent is an entity which exists and interacts in the simulation space. It is autonomous and controls its own actions and state. It is capable of interaction with other agents and can react to changes in its environment. An agent may also be programmed to be pro-active and initiate action [10].

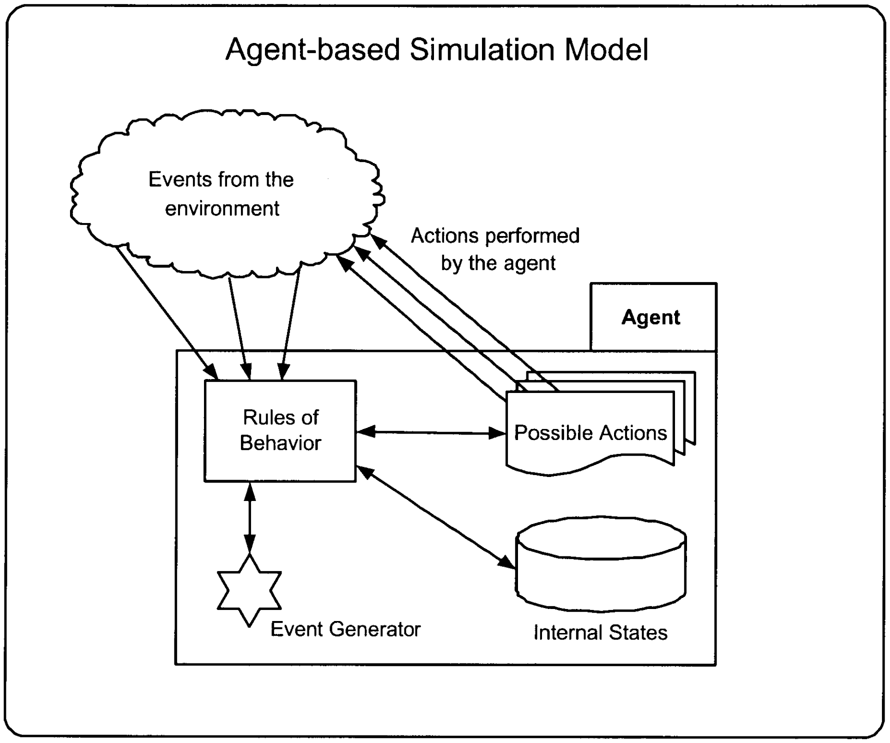

Figure 3-2: Agent-based Simulation Architecture

The agent-based model is constructed by first identifying the entities in the system that should be modeled to meet the needs of the simulation. Once the agent is identified, it is given an initial state and rules for its behavior are established to formulate methods of interaction and action within the simulation space. There is no global input or output although events may affect agents within the entire simulation space. Inputs to each agent are what it is able to perceive and its actions are its outputs. The interactions between agents and the simulation space are the computations. When the simulation is stopped, the final result is given by the state of the agents and simulation space. The agent-based simulation architecture is shown in Figure 3-2 [10].

Several toolkits exist to facilitate the construction of agent-based simulations. Swarm, originally developed at the Santa Fe Institute, is an object oriented library for building, analyzing, displaying, and controlling an agent-based model [9]. It is widely used for agent-based simulation and serves as the foundation of several other tools such as the Multi-Agent Modeling Language (MAML) [20] and Evo [29]. An evolution of these tools from the object oriented approach to a web service based architecture will be discussed in Chapter 5.

The primary difference between the agent and the variable-based model is autonomy. The entity in the variable-based model is the entire simulation and is controlled by a global set of rules, whereas the agent is only a player in the simulation governed by its own rules. While the variable-base model can access global information, the agent cannot. It is only able to use information that it perceives. This can be an advantage when attempting to model individual entities, resulting in more accurate behavior, but places the agent at a disadvantage when faced with the need for aggregate information.

3.3.1 Infrastructure Simulation Suitability

The autonomy of the agent-based model makes it highly suitable for simulating networks of interactions where system behavior is too complex to be predicted by statistical or qualitative analysis. This type of system has been called a "messy system" in a MABS 2000 paper by S. Moss [22] that compares different agent design techniques for agent-based social simulations. Large infrastructure networks can be classified as messy systems. Though the user may not be able to predict system interdependencies, it is possible that they will be revealed through the simulation of interactions between is subparts. This bottom-up approach of agent-based simulation is highly suitable for simulating infrastructure networks.

The modular property of the agent will also change the data collection requirements, scalability, and reusability of the simulation. Rather than collect large amounts of data to feed thousands of system variables, only the state and set of behavioral rules need to be established for each agent. When the simulation needs to be expanded, more agents are simply added to the environment to be incorporated by the system dynamic. This is much simpler than with a variable-based simulation where the system dynamic must be redefined when more variables are added. The behavioral rules of each agent will most likely be unchanged from one run to the next, leaving only the state to be refreshed with each iteration. The increased reusability facilitates the iterative planning and war-gaming objective of NISAC.

Although both variable- and agent-based simulation of an entire national infrastructure would involve an enormous data set and be computationally intensive, the latter method is a simpler model to create due to its bottom-up approach. The ability to define agents at a low level will reveal interdependencies and vulnerabilities through their interaction within the system, rather than having to explicitly define these as is required with the variable-based simulation. The interaction between agents offers the potential to meet the predictive capability objective given the accuracy of the agent behaviors.

As we will see in Chapter 5, the modular capability of an agent-based simulation with the right implementation architecture will be able to provide simulation capabilities to a wide range of users with different needs and in different locations. All of the characteristics of agent-based simulation examined above point to its greater suitability for national infrastructure simulation.

Chapter 4

Simulation Implementation Configurations

The agent-based simulation of the scale required to simulate national infrastructure is computationally intensive. The last step of the simulation process, performing calculations to yield desired results, may involve thousands or millions of agents and an exponential number of interactions. The software architecture required to implement this step has several solutions that can be implemented on competing hardware configurations and will be discussed in Chapter 5.

Here I define the configuration of the infrastructure simulation as its hardware implementation. The complex model required to accurately simulate a national infrastructure system to a useful degree of detail has a high computational requirement limited only by hardware constraints. There are two approaches to meeting its computational needs. The traditional approach has been to build a faster supercomputer. An alternative is to combine the computing power of many less powerful computers to form a virtual supercomputer. This approach is called grid computing.

4.1 Supercomputer Implementation

Both Los Alamos and Sandia National Laboratories have world-class supercomputers capable of multi-teraflop (trillion floating-point operations per second) performance.

Efforts are currently underway by a tri-lab coalition between Los Alamos, Lawrence Livermore, and Sandia National Laboratories to reach 100 teraflops combined computing capability as part of the Accelerated Strategic Computing Initiative (ASCI) [2].

These computers are used for complex simulations such as modeling the explosion of a nuclear weapon and could be used for infrastructure simulation. Their high-performance is a combination of raw speed from thousands of microprocessors, memory and storage capable of handling multiple-terabyte sized files, and high-speed interconnects to provide the bandwidth able to support its operation.

4.1.1 Infrastructure Simulation Suitability

The supercomputers owned by the NISAC partners are designed for the type of complex, large-scale models required to simulate infrastructure. They are highly suitable for this type of computing task, providing unmatched speed and throughput. This capability is costly, but is within the reach of a national laboratory.

Supercomputers are not widely accessible. Thus, a disadvantage of using these supercomputers is platform dependence. Any failure or inaccessibility to these systems results in the loss of the national infrastructure simulation capability. There is an additional disadvantage if we consider NISAC's objective of establishing a portal for nation-wide use of its simulation and analysis capabilities. Ideally, verified users from government and industry would be able to access these capabilities from distant locations for use in their planning and decision-making. Supercomputers are not typically configured for remote service requests. However, the right implementation architecture with front-end servers to configure and schedule simulation requests may solve these problems. This will be discussed further in Chapter 5.

4.2 Grid Computing Implementation

A lower cost alternative to achieving terascale levels of computer power is distributed, or grid computing. Grid computing is capable of achieving such performance by harnessing the combined computing power of many individual workstations over a local area network (LAN) or across the internet to form a virtual supercomputer. It can be defined as "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations (VO) [8]." This approach distributes the data and computing requirements, supported by the grid. The scalability and potential computing power of a grid has made this approach very attractive. Although it is a recent initiative, there are already many grid computing projects sponsored by industry, government labs, and academic institutions.

Grid computing is typically associated with resource sharing amongst lesser hardware, but it is also possible to include supercomputers in a virtual organization. The ASCI Grid will connect the supercomputing resources of the two NISAC partners, Sandia and Los Alamos, with a third Department of Energy Lab, Lawrence Livermore, to form the world's largest grid.

Another large-scale grid computing project in the United States is the Teragrid [27]. The Teragrid aims to create a distributed infrastructure for open scientific research. It is funded by the National Science Foundation and includes as one of its partners the Argonne National Laboratory which is also developing an infrastructure assurance center [3].

These projects all require middleware control systems to create and deploy the advanced network services required to coordinate communication within the virtual organization. One notable package is the Globus Toolkit™ [25] which was recently selected by Sandia for use with the ASCI Grid [26].

This toolkit provides security, an information infrastructure, data management, resource management, information services, and an advanced packaging technology. Security is provided using the Grid Security Infrastructure (GSI) based on public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) communication protocol to provide secure authentication and communication for the grid. To manage the grid infrastructure, the Grid Resource Information Service (GRIS) is used to determine configuration, capability, and status. These results are then aggregated by the Grid Index Information Service (GIIS) to present a coherent system image of the grid's resources. Resource management is performed by the Globus Resource

Allocation Manager (GRAM) which handles job submission and distribution. Data distribution and delivery, the Monitoring and Discovery Service (MDS) for information management, and packing to support a wide variety of platforms and uses are also part of the toolkit [25].

These middleware functions for implementing a grid are not without its difficulties. One of the biggest issues is the development of effective techniques for the distribution of processes. A large-scale simulation problem consists of solving many subproblems of different granularity, parallelism, computation, and communications requirements. Thus, a simple parallel model of computation will not result in an efficient solution [7]. The problem is to schedule the processes among elements of the grid to minimize communication delays and execution time, and maximize resource utilization. This distribution of work can be performed using a static schedule where processes are assigned before program execution begins. This minimizes communication delay at the cost of potentially inefficient imbalances in the workload. Processes may also be distributed dynamically at execution time to perform load balancing so that tasks from heavily loaded nodes are redistributed to lightly loaded ones. The advantage is the flexibility to adapt to unforseen computational requirements, but at the cost of communication delays, lateral data transfer delays, and decision making overhead.

The modular nature of the grid allows for ease of scalability within the virtual organization. Adding a new node is simply a matter of verifying its security permissions and communicating with the proper protocols. With thousands or even millions of computers, the mean time between failure (MTBF) of a node on a grid will be high. Grid scheduling and control will be faced with the overhead of verifying results and rescheduling in the case of a failure.

Another concern is security and protection. Each node on the grid shares not only files, but full use of its resources. This sharing must be highly controlled with clear definitions of what, who, and when sharing is allowed. The set of nodes defined by such sharing rules forms the virtual organization [8]. Communication within this virtual organization needs to be verified and validated to ensure that work to be distributed has securely travelled to its destination computer, been correctly processed, and securely returned.

Finally, it is desirable for the grid to be platform independent to capture the greatest number of potential nodes. High interoperability can be achieved by using common protocols not bound to a specific platform, language, or programming environment. This is a software solution and will be further explored in Chapter 5.

4.2.1 Infrastructure Simulation Suitability

The technology exists to create a virtual organization for a grid computing solution to the national infrastructure simulation problem. There are proven solutions to similar problems. Although feasible, this type of implementation configuration is currently unable to compete with supercomputers in throughput, bandwidth, and in efficient use of computing power. However, ease of accessibility, scalability, cost, and robustness are its advantages.

Both configurations for implementing the national infrastructure simulation are suitable and complimentary where the disadvantage of one is the advantage of the other. Using a supercomputer is highly suitable for time-sensitive critical crisis and decision making simulations. The grid computing solution is a suitable low-cost alternative and can take advantage of the computing power of the large number of users who would benefit from the simulation to form a virtual organization. The use of both configurations matches needs with the best resource for the task in meeting NISAC objectives.

Chapter 5

Simulation Implementation Architectures

Here I define the architecture of the infrastructure simulation as its software implementation. Until recently, most simulations have been written using object-oriented libraries for interaction with one user as a single program on a locally controlled machine. Given the infrastructure simulation's high computational requirement and accessibility objectives, this approach could limit the simulation's compatibility and interoperability with other hardware. It potentially limits the accessibility of the simulation. An alternative is to use a modular and more flexible service-based approach to the simulation architecture.

5.1 Object-Oriented Implementation

The traditional object-oriented approach is very well suited to forming agent-based models. This is the approach currently used by Swarm, the library for building agent-based models introduced in Section 3.3. In addition to model and environment creation, the Swarm toolkit adds memory management, list maintenance, and scheduling among other features [9].

Like an agent, objects are self-contained and may be designed to interact with other objects through the exchange and processing of information. In the agent-based simulation application, objects are used in a hierarchical process to represent agents within the environment, facilitate the modeling process, and present information about the simulation to the user.

5.1.1 Infrastructure Simulation Suitability

The object-oriented approach is suitable for the creation of a national infrastructure simulation targeted to be run on a supercomputer since it is centered around a local user. Thus, it holds many of the same advantages and disadvantages as the supercomputer configuration. Limitation to a specific platform may decrease accessibility. If inputs from outside sources are required, refreshing the large amount of data for each simulation run becomes a slow and tedious process, limiting its reusability.

Although this approach is bound by some limitations when faced with NISAC's objectives, a locally controlled simulation would be easier to create and manage. To capture these advantages and overcome its limitations, it is possible to provide the simulation as a service rather than running it as a program. A service-based simulation could meet the accessibility objective with greater flexibility, but not without its share of problems.

5.2 Web Service Implementation

A service in this context is an application with some functionality offered as a service to the user. For example, rather than running a local word processing application, a word processing service can be requested from a remote provider. The user would not need to install any software locally. A web service provides services over standard World Wide Web protocols. This combines the advantages of the componentization of software with the highest level of interoperability.

Componentization of software breaks down software into reusable building blocks. Until the development of web services, software developers adopted proprietary componentized software methodologies, such as DCOM [28]. This limited compatibility and thus reuse since different vendors used different interface protocols. By substituting standard internet protocols such as SOAP (Simple Object Access Protocol), web services expand interoperability of software components across different development languages and platforms.

5.2.1 Web Service Framework

The web service framework as seen in Figure 5-1 combines the best of both distributed componentization and the World Wide Web to offer interoperability, flexibility, evolvability and extensibility, and scalability. It is capable of performing platform support services such as discovery, transactions, security, and authentication.

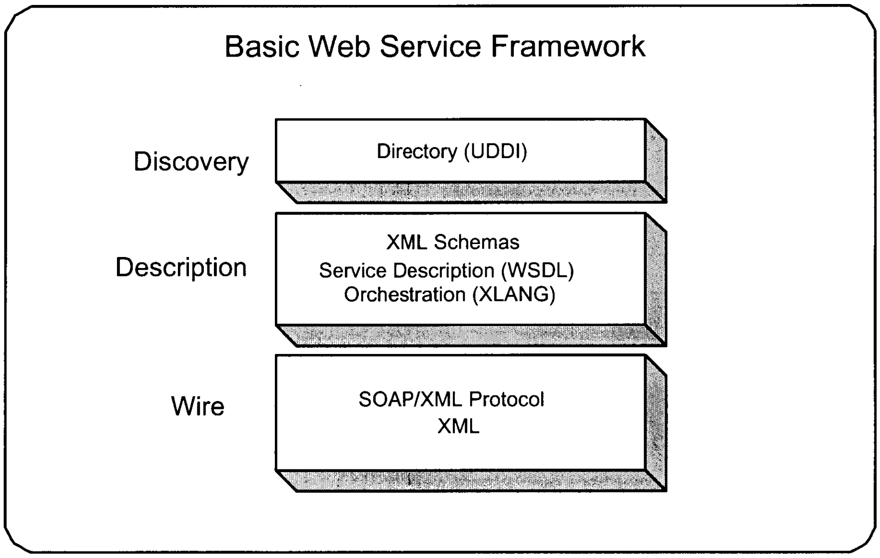

Figure 5-1: The Web Service Framework

Extensible Markup Language (XML) is the key technology used in web services. Through the use of markup tags, XML allows two systems to be loosely coupled over the ubiquitous Hyper Text Transfer Protocol (HTTP). It provides a metalanguage to create specialized languages for interactions between clients and services or between components. This gives the web service tremendous flexibility and interoperability. Behind the web server, an XML message is converted to a middleware request. The request is carried out and the results converted back to XML to be returned to the user. The actual application may be any software running on any hardware platform as long as the middleware is able to perform the conversion. This process is shown in Figure 5-2 [31].

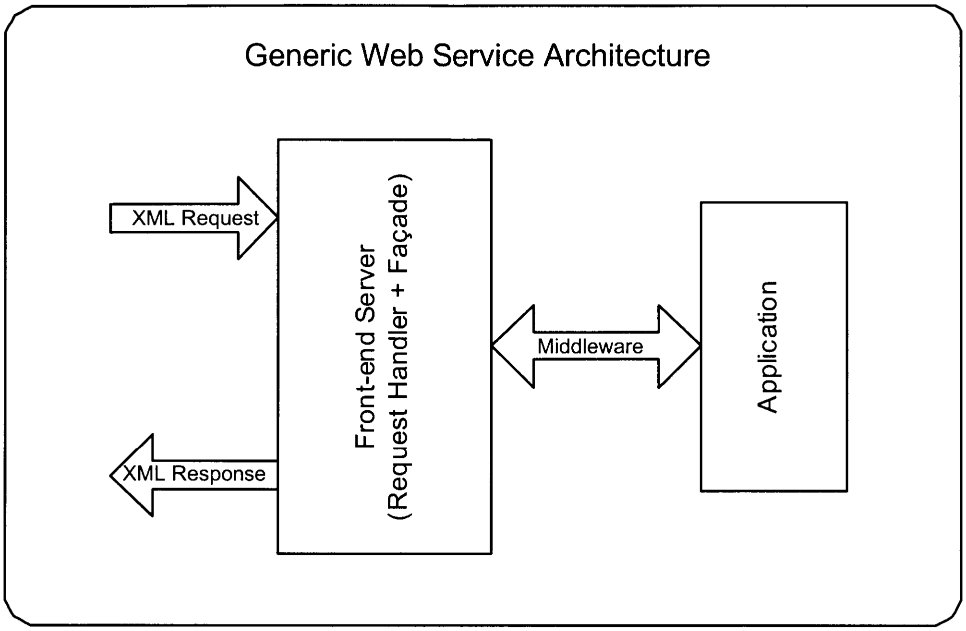

Figure 5-2: Generic Web Service Architecture

Simple Object Access Protocol (SOAP) defines a uniform way of passing XML-encoded data. A client will send an XML-encoded request to a server to get a service. It will receive another XML-encoded message in return. The format of these XML transactions is defined by SOAP. SOAP may also be used on top of other transport protocols to perform remote procedure calls.

UDDI (Universal Description, Discovery and Integration Service) provides a mechanism for clients to find web services. It is layered over SOAP and broadly supported although it is not yet an open standard.

Once a desired service has been located, the client needs a way to find out how to interact with the service. Web Services Description Language (WSDL) provides this information. It describes what a service can do, where it resides, and how to invoke it. Any client can use the information contained within the WSDL file to understand the service interface and invoke the service [28].

This XML/HTTP/SOAP/WSDL/UDDI architecture provides the framework for software to be offered as a service while solving the interoperability and reusability problems encountered in previous solutions. At higher levels, additional functionality for more complex transactions can be attained by adding optional technologies such as XAML (Transaction Authority Markup Language), XLANG (expresses compensatory actions), XKMS (XML Key Management Specification), and XFS (XMethods Filse System). Additional security initiatives include S2ML (Security Services Markup Language) and AuthXML [31].

5.2.2 Infrastructure Simulation Suitability

The interoperability and flexibility of web services makes it an excellent candidate for the delivery of infrastructure simulation services. It facilitates platform independence and scalability. A service can be created using standard protocols as described in Section 5.2.1. This service would serve as a portal to the simulation data, functions, and results.

The client requesting a simulation run would use UDDI to find the service, WSDL to find out how to invoke the service, and XKMS for verification, all encoded with XML and sent over HTTP. The request would be interpreted by NISAC front-end servers and sent to the core simulation program which would in turn deliver the service. This architecture is hardware independent and available to any user able to formulate a request using standard protocols.

The infrastructure simulation web service might allow the client to send in simulation parameters, situational data on the status of relevant infrastructures, assign processing priority, and deliver results, all from a remote location. For real-time data, a reciprocal service could be established at major client sites to deliver status updates to the infrastructure simulation control program. This would all be performed automatically through the use of web services.

5.3 Web Service Implementation of an Agent-Based Simulation

It is possible to utilize web services one level deeper to form the actual simulation architecture. The web service framework offers many useful advantages over the traditional object-oriented framework to describe an agent and its behaviors.

This evolution in agent-based simulation has been proposed by M. Daniels of the Swarm Development Group [9] and by M.N. Huhns of the University of South Carolina [16]. Daniels has recognized the need for a modular and extensible simulation [5] with the "ability to easily configure and share computing resources for Swarm via the web [6]." He has proposed making Swarm a web component based application.

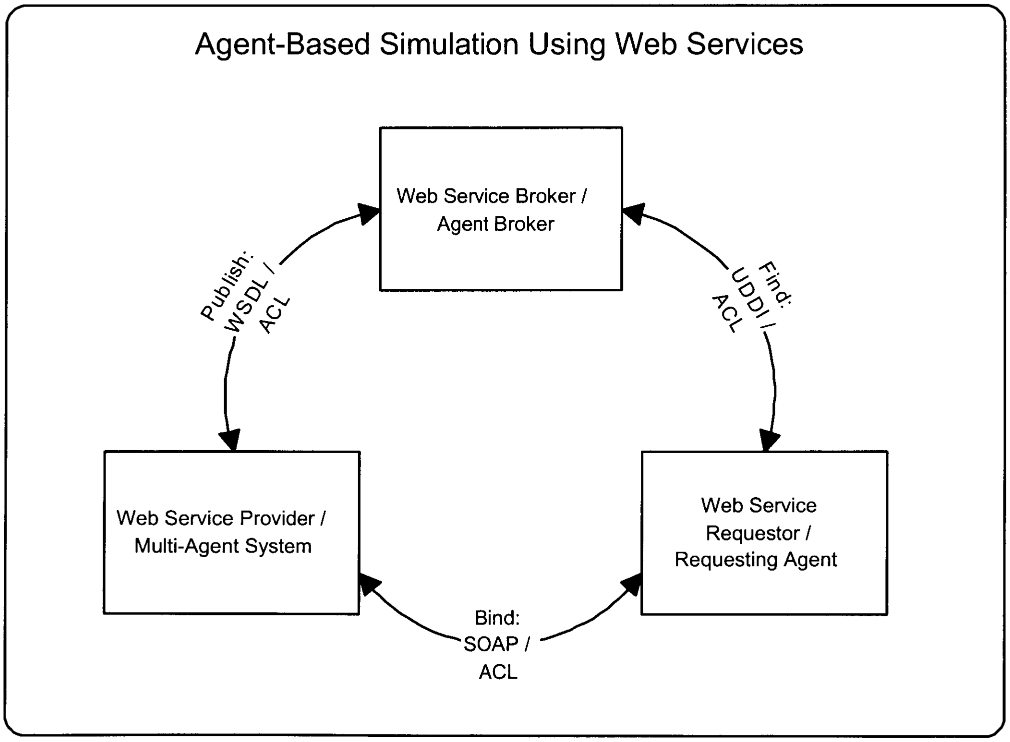

Figure 5-3: Agent Functions Using Web Services

The agent can be viewed as a software engineering unit that encapsulates functions of behavior describing its roles and goals. Agents can negotiate and collaborate in their interaction to adapt to a changing environment. Their interactions are performed using a common structured language, generically termed an agent communication language (ACL). These properties also apply to a web service. Huhns proposes that a parallel can be made between the two as shown in Figure 5-3 [16] and it is possible to see how an agent based simulation may be implemented as a group of mutually interactive web services by mapping communication between agents to the XML protocol [21].

This web service implementation of the agent-based simulation also resembles the grid computing configuration. Similar advantages and disadvantages apply towards its suitability for infrastructure simulation.

Chapter 6

Summary and Conclusion

Modern society is highly dependent upon its networks of infrastructure. The ability to manage food distribution, water resources, energy resources, communications, transportation, banking and finance, emergency services, and government operations is necessary to maintain our way of life.

The growth of our complex infrastructure systems have formed horizontal and vertical interdependencies that disguise potential vulnerabilities. These cannot be discovered by inspection and thus there is a need for national infrastructure simulation capability. In the United States, NISAC was created to provide this capability.

There are two approaches to creating a model of a complex infrastructure system. The variable-based approach was found unsuitable due to the lack of prior knowledge of interdependencies required to define the functional relationships central to this type of model. This approach also lacks scalability. The agent-based approach was found to be highly suitable in simulating networks of interactions due to its bottom-up approach and modular construction. This results in greater scalability, ease of implementation, and greater potential for the discover of infrastructure interdependencies and vulnerabilities.

An agent-based simulation could be implemented on a supercomputer or by a virtual organization formed by a computing grid. Both configurations were found to be suitable and are in fact, complimentary. The supercomputer configuration offers greater power and throughput to handle the large data sets required by the simulation.

It can also be incorporated into a grid as demonstrated by the ASCI Grid program. The grid computing configuration offers ease of accessibility, scalability, lower cost, and robustness as its advantages. Utilizing both configuration options would be optimum.

The object-oriented architecture is highly suitable for agent-based simulations and could meet NISAC's objectives in combination with web services. The use of web services as an implementation architecture was explored and found to be highly flexible in allowing a combination of different implementation configurations. Using the web service architecture would facilitate the creation of the national infrastructure simulation portal as envisioned by NISAC, resulting in greater accessibility and usefulness. It was also recognized that the agent-based simulation model shares many parallels with the web service and could perhaps be implemented using the web service framework.

6.1 Future Work

This thesis has presented many options and approaches to implementing the national infrastructure simulation problem. Several of these approaches are ready for further testing and evaluation. The use of web services as a front-end to a simulation performed on both supercomputers and a virtual organization is an attractive solution. However, creating the agent-based simulation itself using web services also shows great potential. Both of these ideas merit further development.

Bibliography

[1] H. Arsham. *Systems Simulation: The Shortest Path from Learning to Applications.* http://ubmail.ubalt.edu/ harsham/simulation/sim.htm, July 2002.

[2] ASCI at Sandia National Laboratories and ASCI at Los Alamos national Laboratory. http://www.sandia.gov/asci/ and http://www.lanl.gov/asci/asci.html, July 2002.

[3] Argonne National Laboratory Infrastructure Assurance Center. http://www.iac.anl.gov/index.html, July 2002.

[4] R. Cerveny and S. Stephenson. Managing technology of America's infrastructure. *Proceedings PICMET*, pages 458–459, July 1999.

[5] M. Daniels. *An open framework for agent-based modeling.* http://www.santafe.edu/ mgd/lanl/framework.html, April 2000.

[6] M. Daniels. *Next Generation Swarm.* http://www.swarm.org/alife7/img0.htm, August 2002.

[7] J.J. Dongarra and P. Raghavan. A grid computing environment for enabling large scale quantum methanical simulations. In *Proc. 1st IEEE/ACM Internation Workshop on Grid Computing*, 2000.

[8] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid. In *Supercomputer Applications*, 2001.

[9] Swarm Development Group. http://www.swarm.org/, July 2002.

[10] L. Gulys. *A Framework for Classifying Modeling Approaches.* http://www.syslab.ceu.hu/ gulya/IntroToABM/index.html, October 1997.

[11] U.S. White House. *Executive Order 13010-Critical Infrastructure Protection.* Washington: Government Printing Office, July 1996.

[12] U.S. White House. *Critical Infrastructure Protection: Presidential Decision Directive 63.* Washington: Government Printing Office, May 1998.

[13] U.S. White House. *Executive Order 13228-Establishing the Office of Homeland Security and the Homeland Security Council.* Washington: Government Printing Office, October 2001.

[14] U.S. White House. *Department of Homeland Security Information Pamphlet.* Washington: Government Printing Office, June 2002.

[15] U.S. White House. *Proposed Homeland Security Act of 2002.* Washington: Government Printing Office, 2002.

[16] M.N. Huhns. Agents as Web Services. *IEEE Internet Computing*, July 2002.

[17] Sandia National Laboratories. *National Infrastructure and Simulation Center Brochure.* http://www.sandia.gov/CIS/nisac-brochure.pdf, July 2002.

[18] Sandia National Laboratories. *National Infrastructure and Simulation Center Factsheet.* http://www.sandia.gov/CIS/nisac-factsheet.doc, July 2002.

[19] Los Alamos National Laboratory. *Counter Terrorism, Mapping the Vulnerabilities.* http://www.lanl.gov/worldview/news/pdf/TRCounterTerror_NISAC.pdf, July 2002.

[20] Multi-Agent Modeling Language. http://www.maml.hu/, July 2002.

[21] L. Moreau. Agents for the grid: A comparison with web services. In *Proc. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2002.

[22] Scott Moss. Messy Systems-The Target for Multi Agent-Based Simulation. *MABS*, 2000.

[23] U.S. House of Representatives. *Section 1016 of H.R. 3162-USA Patriot Act.* Washington: Government Printing Office, October 2001.

[24] U.S. President's Commision on Critical Infrastructure Protection. *Critical Infrastructure Protection Simulation Report.* Washington: Government Printing Office, 1997.

[25] The Globus Project. http://www.globus.org, July 2002.

[26] Platform Computing News Release. http://www.platform.com/newsevents/, July 2002.

[27] Teragrid.org. http://www.teragrid.org/, July 2002.

[28] T. Thai and H.Q. Lam. *.Net Framework Essentials.* O'Reilly & Associates, June 2001.

[29] EVO Artificial Life Framework The Omnicron Group. http://omicrongroup.org/evo/, July 2002.

[30] K. Troitzsch and N. Gilbert. *Simulation For The Social Scientiest.* Open University Press, 1999.

[31] V. Vasudevan. *A Web Services Primer.* http://www.xml.com/pub/a/2001/04/04/webservices/index.html, July 2001.