

# An Agent-based Service-Oriented Simulation Suite for Critical Infrastructure Behavior Analysis

**Ebrahim Bagheri, Hamidreza Baghi and Ali A. Ghorbani**

Intelligent and Adaptive Systems Lab, Faculty of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, {e.bagheri, hamid.baghi, ghorbani}@unb.ca

**Alireza Yari**

Iran Telecommunication Research Center (ITRC), Tehran, Iran, a\_yari@itrc.ac.ir

**Abstract:** Critical infrastructure systems are complex networks of adaptive socio-technical systems that provide the most fundamental requirements of the society. Their importance in the smooth conduct of the society has made their role more and more prominent. A failure in any of these important components of today's industrial society can well affect the lives of millions of people. It is not only their individual break down that raises serious concerns, but their mutual reliance (interdependency) is even more threatening. Although interdependency in these infrastructure systems provides many benefits for their operation, a failure in one can ripple down to the others and cause a catastrophic irremunerable event. In this paper, we have introduced a simulation suite for analyzing the behavior of interdependent critical infrastructure systems. The simulation suite focuses on the types of services that are provided by infrastructure components. Each infrastructure system component is modeled as an agent and its services as its behavior. We believe that this simulation suite can assist researchers in better understanding critical infrastructure behavior and hence prevent catastrophic failures.

**Keywords:** Critical Infrastructure Systems, Modeling, Simulation, Interdependency

---

## 1 Introduction

---

The California power crisis in the year 2000 (Purdum (2001)) and the tragic 9/11 terrorist attacks in New York (Bram et al. (2002)) are clear depictions of threat towards the most influential and operational elements of both the society and industry: Critical Infrastructures. Critical infrastructures are fundamentally considered to be those national or even internationally moderated systems whose prolonged disruption could cause significant military, economic, or social disturbance (CIP-Commission (1997)). Their prominent but concealed role in the ordinary every day life of each citizen signifies their importance and devastates the state of their disruption. Imagining the direct consequences of a slight interruption in the correct operation of only the electricity infrastructure in a small city, clarifies to what extent the normal pace of life is distressed.

One of the major threats towards the normal operation

of these complicated systems is their extreme internal complexity. These infrastructures mostly consist of a diverse range of systems and operational elements that form a very intricate structure. It will be inevitable to see undesired or even sprouting internal interdependencies between the interior systems of an infrastructure system.

Critical infrastructures are also vulnerable to terrorist or malicious attacks. Exploring different reasons or intentions of the launchers of such spiteful activities is important for the prevention of their actual occurrence. Classifying infrastructures through a terrorist attack vulnerability analysis provides a good understanding of the threats and hazards that infrastructure systems may be facing. It would be much easier to avoid such incidents based on this analysis or its least benefit would be the possibility of creating appropriate recovery plans.

Critical infrastructure's mutual interdependency is the

other source of threat that may jeopardize their normal course of operation in the case of failure in one infrastructure systems (Thissen and Herder (2003)). Similar to the internal dependency between different systems of a single infrastructure, different infrastructures can be interdependent from four aspects (Rinaldi et al. (2001)). In the first model of interdependency, infrastructures can rely upon each others services that may be in the form of commodity trade or service provisioning. This type of interdependency, Physical interdependency, is the most apparent form and can be detected by analyzing the inter-infrastructure contracts or relationships. The other three subtle interdependencies are Geographical, Cyber and Logical. In the geographical interdependency, infrastructures are not directly reliant, but have overlapping operational bounds. Infrastructures with either shared information states or dependent knowledge bases can be classified as cyber interdependent. Logical interdependency between two or more infrastructures is the most concealed form of interdependency, which is very difficult to spot. War for example, is a phenomenon that has no direct relationship with gold or oil markets; however a war can have repercussions on oil or gold markets causing unpredictable fluctuations in their price.

It is the interdependencies of critical infrastructures that has brought about serious concerns. A failure in one critical infrastructure can amazingly ripple through to other infrastructures producing a cascading effect (Bagheri and Ghorbani (2007)). It is the low probability - high impact events that can seriously damage the well being of a society through a sudden failure in one single infrastructure (Dunn (2005)). As the description for such events depicts, their low probability causes an underestimate of their side effects. It had never been estimated that an attack on the twin towers in New York would cause so many disruptions. A very simple example of rippling failure under tense conditions was the disorder in the emergency crew communication lines in the 9/11 incident. Since it had never been estimated that the telephone lines would be so much highly occupied that the emergency crusade cannot use it, no other means of communication had been provisioned; however, reality told a different story leaving the rescue team dismantled with no means of communication and hence no way of coordination.

---

## 2 Related Work

---

The shocking side effects of critical infrastructures' interdependencies has provoked many researchers to devise suitable models for understanding the behavior of infrastructure systems. These attempts can be classified into two major categories. The first category addresses the issue from a pure mathematical perspective and models the behavior of infrastructure systems through the employment of differential and/or algebraic-differential equations. One of the well-known examples of such approach is the inoperability input-output model proposed by Haimés et al

(Haimés et al. (2005), Haimés et al. (2005)), which has been based upon Leontief's I-O model (Leontief (1966), Leontief (1951)).

The second branch of research focuses on modeling infrastructure systems and their interdependencies through the exploitation of intelligent software agents. The rationale behind such approach is that each infrastructure system or any of its sub-systems can be modeled as a software agent. Therefore, the aggregate behavior of these agents within a unified framework (as a multiagent system) can reveal their actual behavior under various conditions. The benefits of this approach as compared with the methods used in the previous category is that these systems are on the one hand much simpler to formulate and on the other hand more flexible and easily extendible. CISIA (Panzieri et al. (2004), Panzieri et al. (2005)) and Aspen (Basu et al. (1998), Barton and Stamber (2000)) are two of the well-known examples of the systems that have been developed based on the concept of agent-based systems. Here, we briefly introduce these systems and point out some of their most outstanding features.

**CISIA** models the behavior of an infrastructure systems (or a set of interacting infrastructures) through a set of non-linear interdependent agents. Each of the agents represents a macro component of the modeled system. For the sake of generality, and to be able to encompass more infrastructure systems, CISIA employs a very high level description of the internal logic of an agent. This description consists only of the specification of the agents' operative level (agents stamina), requirements (agents needs), and faults. To model the interaction of the agents (provide mutual requirements or disseminate failure), three types of matrices namely Operative Level Incidence Matrix, Requirement Incidence Matrix, and Fault Incidence Matrices are devised. Fault incidence matrices are further refined to allow the analysis of different types of failure propagation (geographical, physical, and cyber).

**Aspen** is an agent based micro-analytic simulation model designed specifically at the SANDIA national laboratories to simulate the US economy. Based upon evolutionary learning algorithms within the agent architecture, Aspen has been able to simulate the behavior of simple decision making agents of the economy. The agents that have been modeled in Aspen have been households, banks, companies and the government. It also provides means for analyzing the impact of various policies and regulations on the economy. Different economical sectors can be modeled independently or as an integrated whole within Aspen. More recent attempts have extended Aspen in various directions that have resulted in the creation of Aspen-EE, N-ABLE, and CommAspen.

The modeling and simulation suite that we propose in this paper is based on the Agent-based Interdependency modeling and Simulation (AIMS) architecture (Ghorbani

et al. (2006)). However, there are various differences between AIMS and the systems introduced in this section. These differences are discussed in the following lines:

- Firstly, based on the understanding that simulating infrastructure systems requires an in-depth study of their structure, AIMS provides suitable means for initially modeling infrastructure systems. Therefore, modeling an infrastructure system in AIMS is a prerequisite for its simulation. The modeling process is supported by static *component templates* that represent an infrastructure, its sub-systems, or even its resources. These component templates are properly stored in AIMS repositories and can be used in various modeling procedures.
- Secondly, besides the concept of agents, AIMS centers around the type of services that infrastructure systems can provide. Therefore, the modeling activity in AIMS needs to take a service oriented approach in defining the parties that participate in the simulation. This is basically due to the fact that real-world infrastructure systems interact in a service centric fashion. As an example, let's consider a subset of the interactions between the electricity and telecommunication infrastructures. The telecommunication infrastructure relies on the electricity infrastructure for its power needs. In order for the telecommunication infrastructure to reach its needs, it has to negotiate with various electricity providers. These providers, based on this request, provide the telecommunication infrastructure with a proposal for service provisioning. This proposal may consist of information on the Quality of the Service (QoS), their cost and conditions of use. The telecommunication infrastructure can then decide on the choice it wants to make. If the environment in which the infrastructure systems operate gets more complicated, service brokers can assist in finding their requirements. This process is analogous to what is currently practiced in service oriented architectures (Bagheri and Ghorbani (2006a)). Figure 1 shows why service provisioning is the central concept in the modeling process of AIMS.

---

### 3 Motivations

---

The AIMS system is designed to aid research into critical infrastructures and the interdependencies between them. The complexity of critical infrastructures has made it very difficult for stakeholders to obtain accurate data to use when making decisions. Stakeholders can include infrastructure owners, managers, and operators, regulatory agencies, security agencies, government, and many others. In most cases (though not exclusively) stakeholders are interested in knowing how a system of infrastructures will react under a particular set of conditions. These conditions can range from the failure of components of an infrastructure to a change in the supply and cost of raw material

inputs. While individual infrastructures often have tools that can assess changes within a single infrastructure, the AIMS system is designed to extend the effects of changes across a system of interconnected infrastructures.

Choosing a hypothetical example, if a government was interested in simulating the effects of increased electrical consumption across a geographical area the question could be delegated to a group for study. This group would collect as much data as available and needed relating to the various infrastructures and direct both the question and the data to a researcher familiar with the AIMS system. The researcher would then need to model the infrastructures in question, perhaps taking advantage of previously created models. The modeling process could require the creation of new component templates. With the completed model the researcher would then set about running a series of scenarios designed to highlight the effects of varying levels of increased electrical consumption. The results of the simulations would then be passed up to the requesting parties.

It is important to realize that the results generated in the previous example are not restricted to the electricity infrastructure. A lack of electricity at certain times may cause a failure in the telecommunications infrastructure indicating a need for greater backup power capability for telecommunications. Perhaps the lack of electricity disables the dispatching capabilities of emergency response teams or the computers that run major banking institutions. These are the results that are currently under-represented when simulation tools for individual infrastructures are used.

Using the more comprehensive view of infrastructures and their interdependencies allows for greater accuracy in decision making related to regulation, investment, security, development, etc. by government agencies and by infrastructure owners and managers. If the population is growing rapidly then the government may mandate a higher capacity for an existing electrical grid. If an ambulance company is heavily reliant on a flimsy telecommunications system it may invest in private telephony equipment. If electricity generation uses a single transportation route to receive its raw materials it may try to diversify its supply (or increase its stockpiles) to reduce the risk of being unable to have raw materials delivered. By expanding the scope of simulation to include other infrastructures, AIMS simulation programs are able to expose the effects of "what-if" questions that are missed when infrastructure inter-connectivity is ignored.

---

### 4 Major Design Criteria

---

When designing the AIMS system there were a number of criteria that guided the design decisions. Some of the criteria were general and reflected good programming practice; some criteria focused on the reusability of the system to be built; and some of the criteria were specific to the fact that the AIMS system was designed as a simulator of a specific type of system, namely critical infrastructures.

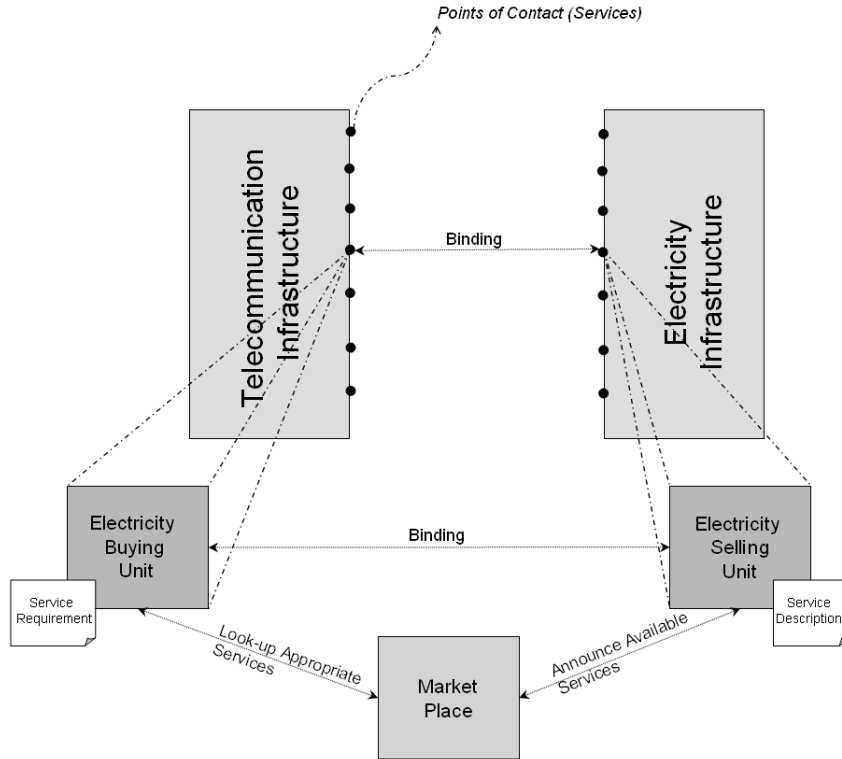


Figure 1: AIMS Models Explicit Infrastructure Interdependencies with a Service Centric Architecture.

#### 4.1 Portability and Reusability Criteria

One of the first, and most general, criteria was the desire to have the AIMS system be portable and to the greatest extent platform independent. This led to the choice of Java as the programming language of choice. As the AIMS system was relatively large, the subsystems were to be well defined and encapsulated to provide blocks that were as functionally independent as possible. A related criterion was the desire to allow the system to be augmented through a plug-in mechanism that was able to add new blocks of functionality to the existing system. Owing to the desire to be able to distribute the AIMS system on a single CD or over the network, it was to be as small and modular as possible; this is related to the desire to have the overall system divided into well-defined functional blocks.

#### 4.2 Simulator-Related Criteria

The fact that the AIMS suite was supposed to be designed as a simulator led to a number of other criteria for the design. The simulation programs created by the AIMS system needed to be run in both batch (where a number of runs could be done with predefined parameters) and interactive modes (where a user can manipulate the values in a running simulation). This led to a desire for a scenario-centered design where simulation programs needed to have a scenario to describe their execution parameters. Like the separation of the functional blocks within the AIMS system, a logical separation needed to be kept between the system being simulated, the scenario under which it

would be run, and the manner in which the output would be viewed and analyzed. Though it would be possible to create the AIMS system without a graphical interface for the user, this is archaic and a graphical interface was also deemed necessary.

#### 4.3 Critical Infrastructure-Related Criteria

While the previous set of design criteria would suit many simulation systems, the fact that the AIMS system was designed specifically for simulating infrastructures and the interdependencies between infrastructures led to some additional requirements. One of the most important criteria was the need to use an agent-based system for the simulation programs. Software agents provide excellent means to model the autonomous actions of infrastructures and the subsystems within infrastructures. Additionally, the systems being simulated are typically built of systems of interconnected components. This leads to the need for a component-based simulation system that allows for generic components to be used as building blocks for larger systems. These components should be easy to build for users and creators of new components should be able to use existing, possibly generic, components as a foundation. The components need to be reusable and can be combined to create (encapsulate) more complex components. Reusability of all the aspects of the AIMS system inputs (documents, configurations, components, VMA specifications, etc.) was held as an important feature to ensure a useful final product.

---

## 5 The Metamodel

---

The AIMS metamodel consists of four major metaclasses: *Model Instance*, *Component Template*, *Contract(Binding)*, and *Scenario*. The model instance metaclass incorporates all other three metaclasses into a whole and shapes the overall design of the infrastructure that needs to be analyzed and simulated. Suppose that we intend to investigate the behavior of an electricity infrastructure. In this case, the model instance will represent that specific infrastructure systems, while the other three metaclasses depict its internal setting, external relationships and the context of its operation. As it is seen in Figure 2, a model instance can contain as many component templates, and contract (binding) instances as is required, but can only be associated with a single scenario instance each time.

### 5.1 Component Template

A component template metaclass is an abstract entity, that can be instantiated to represent any infrastructure system, sub-system or resource. Its design is mainly focused around the type of services that it provides. For instance, in a cellular network, an MSC (Mobile Switching Center) can be considered as one of its sub-systems; therefore, it can be modeled through a component template. This component template should incorporate the type of real-world services that are provided by a typical MSC. Hence, the design of an MSC should include the following services:

- *Routing*: Routing calls to mobile users
- *PSTN Connection*: Connecting calls from the mobile network to the PSTN network
- *Handover*: Managing handovers from one BSC to another
- *Interoperability*: Interoperating with other mobile switching centers

In our synthesis of a service, we define it as a logical entity that can be only deployed under certain circumstances. These conditions are described in the service *deployment description* class. The deployment description class can define any sort of constraint that applies either on the service provider itself or the service requestors/consumers (e.g. the service can only be offered within a certain geographical region, provided only to a set of defined customers, or sold with a minimum price of  $\alpha$ ). To specify the behavior of a service under various conditions we define a logic class inside each service. Each logic class has a set of operations that themselves can have different states of operation. Related to the different states of an operation are a set of instructions that are executed whenever the service is performing that operation and is in that particular state.

As an example, let's consider the routing process of an MSC device as a service. Basically this service has two major operations: receiving an incoming message (RIM), and

passing over the received message (PRM). Each of these operations may themselves have different states of operation. The RIM operation may have two states, one in which it is expecting new messages, and the other where it processes the received message before passing it to the PRM operation. The respective execution procedure that needs to be pursued in any of these states is separately defined in the related instruction classes. The same procedure also applies for the PRM operation.

Beside the services that each component template can offer, it also has a set of unique attributes that distinguishes it from the other similar component templates and allows the instantiation of that component template. For example, an MSC has various attributes such as the set of services that it supports (since not all MSC systems have the same set of functionality), its geographical location and etc. These attributes should be inclusive enough so that most types of systems based on that component template can be created.

The infrastructure systems that are modeled based on this metamodel will be stored in a component template repository within AIMS and can be used in other modeling practices as well. In our example of the cellular network, all of the available sub-systems (e.g. MSC, BSC, BSS, and etc) should be defined based on this metamodel and integrated into the model instance metaclass in order to create a unique representation.

As we will further show in the following sections of the paper, after the end of the modeling process, each component template will be converted into a running agent. This agent will be responsible for performing the set of services that it possesses. In other words, the behavior of each agent is based on the logic which is incorporated in the agent. In AIMS this logic is the set of services that each agent has. For example, in the mobile switching center component template shown in Figure 3, the agent which will be created based on this component template will have a logic that reacts to requests for two types of services: *handOver*, and *queryUserLocation*. The agents that are created based on this component template can be initialized by setting the values of the specific component template features: *MessageCapacity*, and *Neighbors*; therefore, different agent instances can be created from a single component template.

### 5.2 Contract (Binding)

Since the focus of infrastructure systems is very fine and is centered around a specialized product (e.g. electricity, water, etc.), they require interaction with each other to obtain their functional requirements. This sort of requirement obtainment which is known as commodity trade is achieved through the exchange of goods (either in the form of physical or non-physical products) via infrastructure services.

In AIMS, the possibility of commodity trade is provided through the use of the contract metaclass. In a contract, two components of the model agree to share their goods through their services. For example, in Figure 1 the elec-

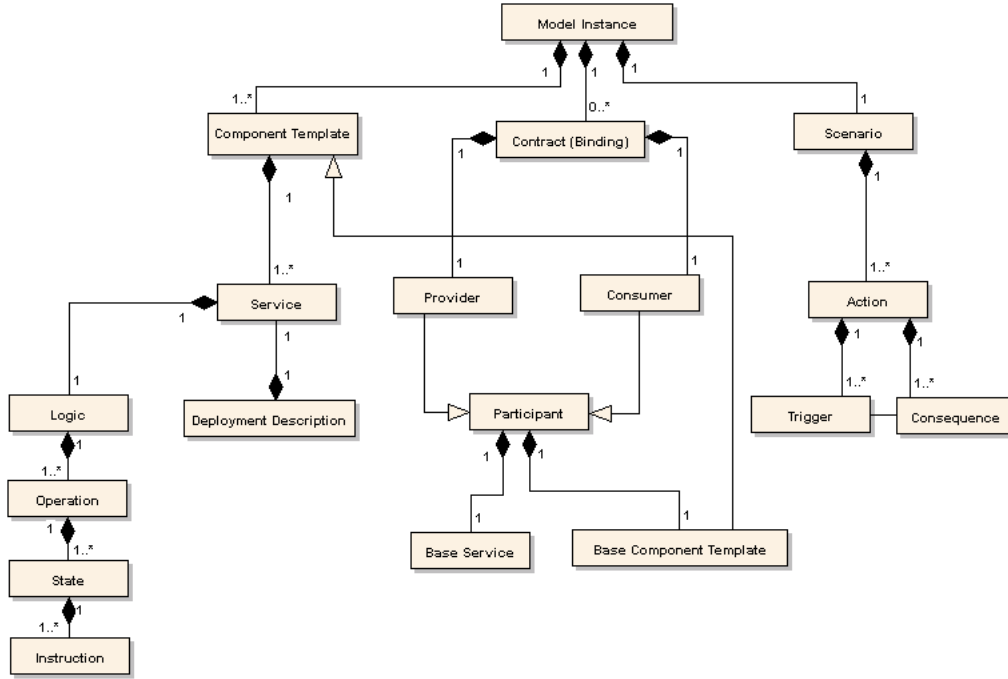


Figure 2: The UML Class Diagram Specifying the AIMS Metamodel.

tricity infrastructure has agreed to provide the telecommunication infrastructure with its required electricity. This shows that each contract has two parties involved. Each one of these parties is responsible for specifying the service through which it intends to participate in the contract. Any established contract in AIMS has a set of attributes that need to be conformed by the two participating parties. These constraints are initially proposed by the service provider component. If the consumer agrees to conform to the set of constraints, a contract is established and service exchange will take place between the two parties.

The defined contracts in AIMS do not have a static nature. By this we mean that the contracts that have been defined though the modeling process may expire during a simulation, while other new contracts may be established between the participating components. This process is developed in a market place. A market place in AIMS is similar to a service registry where all of the components introduce their services along with its description. This may contain information such as proposed SLAs and conditions of use. Other components can find their requirements in the market place and start a negotiation process with the specific service provider and ultimately receive their required goods.

### 5.3 Scenario

Scenarios are tools for creating alternative scenes based on the different arrangements of the initial hypotheses about environmental parameters and system settings (Zhang and Bose (1989)). They are usually built in cases where more knowledge or understanding of a situation or system is

required. They allow the observation of different perspectives on the same system by giving new viewpoints. In a computer simulation, a scenario specifies the initial setting of the model, and provides guidelines on what paths the simulation has to follow. Since many of the interesting (and at many times harmful) events within an infrastructure system are unpredictable and rare, scenarios should be able to design simulation routes that cover most of these events.

In AIMS, scenarios are composed of a set of actions. Each action in a scenario has two constituent elements: trigger and consequence(s). The trigger of an action is based on time (e.g. 4<sup>th</sup> minute of the simulation) and causes the consequences of that action to happen at that point of time. The consequence of an action is basically the manipulation of the component services (e.g. Starting/stopping/freezing a service of a component) or component attribute values. Figure 4 shows a sample scenario in XML format.

The metamodel introduced in this section provides the means for modeling the internal structure (the attributes and services) and external relationships (contracts/bindings) of an infrastructure system. It also caters an appropriate format for defining a scenario for the runtime analysis of infrastructure system behavior. In the next section, we will visit the AIMS suite itself, and introduce its capabilities.

```

ComponentTemplate MobileSwitchingCenter {
  Features {
    MessageCapacity : Integer
    Neighbors : Array of Integer
  }

  {java
    class UserInfo {
      public int userID;
      public String userLocation;
    }

    ArrayList<UserInfo> usersDatabase;
  java}

  Service handOver(AIMSPacket pkt) {
    int address = pkt.header.getDestinationAddress();
    String data = pkt.getBody();

    if ( findInArray(address, Neighbors) ) {
      pkt.header.insertRoutePath(this.getID());
      sendMessage(address, "queryUserLocation", pkt);
    }
  }

  Service queryUserLocation(AIMSPacket pkt) {
    String data = pkt.getBody();
    int userID = Integer.parseInt(data);

    for ( UserInfo info : usersDatabase ) {
      if ( info.userID == userID ) {
        int address = pkt.header.getSourceAddress();
        AIMSPacket newPacket = new AIMSPacket();
        newPacket.header.setSourceAddress(this.getID());
        newPacket.header.setDestinationAddress(address);
        String temp = ""+info.userID+" "+info.userLocation;
        newPacket.setBody(temp);

        sendMessage(address, "setUserLocation", newPacket);
        break;
      }
    }
  }
  ...
}

```

Figure 3: Segments of a Component Template Representing a mobile Switching Center with Two of its Services: handOver, and queryUserLocation.

## 6 AIMS Suite

The AIMS suite (Ghorbani et al. (2006)) has a modular design in which all of its modules interact through appropriate message passing schemes. This suite is composed of four major modules namely *AIMS Core Module*, *Visualization, Manipulation, and Analysis Module*, *Scenario Handler Module*, and the *User Interface Module*. Before going into the details of each module, we briefly introduce each of them in the following lines:

- **AIMS Core Module:** The AIMS core module is responsible for creating an active simulation from the models that have been created based on the AIMS metamodel. It is also in charge of controlling the simulation process, and managing the interaction of the running simulation with the other three modules. Internal to the design of this module, we have incorporated four operating units: AIMS Controller, Simulation Controller, Market Place, and JADE Façade.
- **Visualization, Manipulation, and Analysis (VMA)**

```

<Scenario Name="TestingMSCRoutingScenario">
  <Trigger Time="050001"/>
  <List_Of_Consequences>
    <Consequence>
      <Agent GUID="MSC02" />
      <Service Name="Routing" >
        <Parameter Name="RoutingMode" Value="OLSR" />
        <Parameter Name="MaxHops" Value="3" />
      </Service>
    </Consequence>
  </List_Of_Consequences>
</Scenario>

```

Figure 4: A Sample Scenario in XML Format.

**Module:** This module is responsible for providing AIMS users with proper means to analyze the active simulation. Various VMA modules can be introduced into AIMS to show different perspectives and interpretations of a single simulation.

- **Scenario Handler Module:** Once an actual simulation has begun, The scenario handler module will parse the scenario section of the model instance. Based on the scenario action information, it will then send appropriate instructions to the AIMS core module at specific points of time in the simulation.
- **User Interface Module:** The user interface module provides various graphical user interfaces for the AIMS suite modules. These interfaces include the front-end user interface, which is responsible for creating the model instance and executing the model through the AIMS simulator, and administration user interface that is in charge of configuring the AIMS suite.

In the remaining parts of this section, we minutely visit each of these modules and explain their role.

### 6.1 AIMS Core Module

The major role of the AIMS core module is to coordinate the activity of the other modules and administer the complete execution of a simulation. The AIMS controller unit is the point of interaction of the AIMS simulator with the outside world and the other modules. Whenever a user intends to perform a simulation, the AIMS controller unit is launched. This unit will then fork the simulation controller unit and the other three modules.

Within the simulation controller, the model instance that has been provided to the AIMS controller will be parsed. The information gained from this static description of a set of infrastructure systems is transformed into a multi-agent model. The generated model is in accordance with the description of an agent in JADE. In the transformation process, each component template instance is transformed into an active agent. The services provided by each component template are incorporated into the respective agents as their behavior modules. Figure 5 shows how a static model instance that consists of various component templates and contracts (bindings) can be transformed into a multi-agent system.

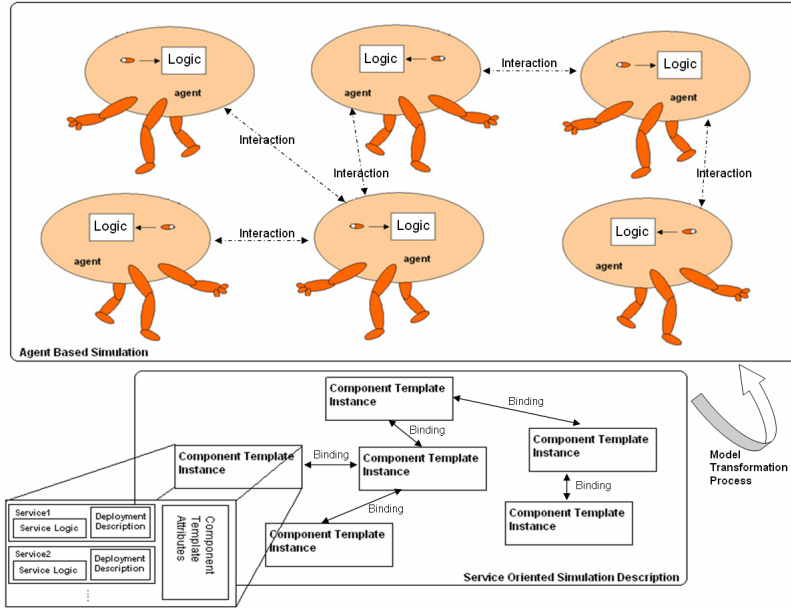


Figure 5: Transforming a Static Model Instance into an Agent-based Model.

The transformation process creates a set of agents based on the available component template instances that are available in the model instance. In order for the agents to be able to interact and create new contracts they require some form of capability advertisement (analogous to service advertisement in SOA) mechanism. This functionality has been incorporated into the AIMS core module through the market place. Agents can register their set of available services in the market place. Each announced service can have various access control descriptions, quality of service proposals, and pricing tables attached to it in the market place. Agents that are searching for a specific service can query the market place. For each received query the market place automatically filters out the set of services that should not be known to the requesting agent. The proposed services by the market place are hence all utilizable by the requestor, but still he has the chance to choose between the proposed options.

The agents that are created in the simulation controller are passed over to the JADE environment (Bellifemine et al. (2001)) for operation. Since the JADE environment does not provide a unified way of extracting environmental information and most of the information are encoded within the structure of the agents, we exploit a complementary agent called the JADE Façade. The JADE Façade is automatically created in the JADE environment whenever a simulation process is embarked. It has the role of gathering the required information from within the multi-agent environment. These information are processed by the VMA modules. The JADE Façade can also be used for changing a parameter in the JADE environment. For example, it can be used to change an attribute of a specific agent. This is achieved by sending a request to the JADE Façade. This request is then sent from the JADE Façade agent to the target agent. All of the agents in the envi-

ronment are obliged to conform to the orders that are sent from the JADE Façade agent. In this way, we have the option of inserting transients into the simulation through the change of agents' attributes values and services status.

## 6.2 Scenario Handler Module

The scenario handler module has close interaction with the AIMS core module. It is responsible for monitoring the simulation clock. Whenever a match is found for the trigger section of an action in the scenario with the current point of time in the simulation clock, it will interrupt the normal course of simulation. In such a situation, the scenario handler sends the consequence part of that specific action to the JADE Façade. The JADE Façade would then based on this request, send a message off to the specific agent in the consequence requesting the changes to happen.

Lets consider the scenario shown in Figure 4. This scenario only consists of one action. The action requires the simulator to start the Routing service of the MSC02 agent (e.g. the second mobile switching center in the model instance). The initial parameters that should be set when initializing the service are routingmode and maxhops. For this example, the scenario handler will send a request to the JADE Façade consisting of the name of the agent (MSC02), the service that needs to be invoked (Routing), and the parameter-value pair that are needed for initialization (RoutingMode=OLSR, MaxHops=3). Upon receiving this message, the JADE Façade will redirect the message to the destination agent (MSC02), and requests the action to be performed.

There are two major concerns related to the concept of scenario that need to be explicitly addressed. These issues are:



1. How do we simulate and control the behavior of the environment (e.g. earthquakes, floods, thunders,etc.)?
2. How does the human resource (the role of people) factor affect our simulation and how do we interact with it?

To address the first question, AIMS defines a separate internal agent called the ‘*environment*’. Manipulating the services of this agent, or changing its parameters through the JADE Façade allows us to apply changes on the environmental conditions and hence study the behavior of our model under different environmental conditions. For example, we can insert an earthquake as a transient into our simulation, by invoking the earthquake service of the environment agent (There is a need for other input parameters to the earthquake service such as geographical location and its degree).

The second dilemma has been addressed differently in other systems. In AIMS, we intend to define human resources as a sort of system that affects the balance of the infrastructure system operations; therefore, we leave the modeling process of these entities to the users of the AIMS suite. For example, one can define a component template named citizen that can act as a consumer of electricity and telecommunication services. Based on this, he/she can study the market behavior by instantiating the citizen component template with different initial parameters (i.e. defining a citizen with interest in using high speed internet, and another with high electricity consumption).

### 6.3 Visualization, Manipulation, and Analysis Module

Beyond the concept of simulation, lies the analysis of the results obtained from this process. For the complete understanding of the critical infrastructure systems behavior and interdependencies the data produced from a simulation need to be processed in both on-line and off-line modes. In the on-line mode the data should be collected in a real-time fashion. These data should be then analyzed and presented to the user. The only point that should be considered in this approach is that the analysis results do not present the complete picture of the system behavior in all points of time during the simulation.

Off-line data analysis can be used in addition to the on-line mode to create a collective understanding of the overall simulation. The major advantage of using the on-line mode is that the results of the data analysis provides the researchers with the possibility to alter their simulation scenarios based on the obtained results. For example, in the electricity and telecommunication (See Figure 1) simulation instance, the modelers may have never anticipated a case where an epidemic disease would paralyze the human resource of the electricity infrastructure. Therefore, by monitoring the simulation in an on-line mode, they can detect such a case and insert further transients into the running simulation through scenarios. They can for instance, enable the electricity infrastructure to hire more

employees and observe its repercussions on the overall simulation.

This process is supported by external entities outside of the AIMS suite that conform to a given protocol named VMA entities. The VMA module is responsible for providing these entities with proper interfaces so that they can connect to a running simulation and request a specific information, or send a request for a change. The VMA module acts as a relay between these entities and the JADE Façade within the AIMS core module. It also performs registration, authorization and authentication checks on the VMA entities that send their request to the AIMS suite.

A sample VMA entity for the AIMS suite can be written in a third-party software such as Matlab. It can periodically request the number of the electricity vendor agents in the environment. Receiving these information it can plot a chart for the researchers to see how this number changes over time and under various conditions. It can also send a request to the VMA module asking it to insert a certain transient into the simulation. The implementation details and capabilities of the VMA entities is independent of the details of the AIMS suite. The only point that should be observed is that their interfaces with the AIMS suite should conform with the AIMS standard protocol. Figure 6 shows how VMA entities can interact with the AIMS suite through the VMA module.

### 6.4 User Interface Module

The AIMS suite provides different graphical user interfaces for various purposes. The main features that are provided through these graphical user interfaces are:

- Creation of New or Manipulation of Existing Component Templates
- Instantiation of Component Template Attributes
- Creation/Loading/Modification of Simulation Scenarios
- Registration/Addition of Visualization, Manipulation and Analysis Entities
- Assembly of Model Instances
- Initialization/Execution of the AIMS Simulator
- Providing Guidelines to Users and Inconsistency Checking

AIMS suite user interfaces are Web applications (See Figure 7); therefore, a modeler can access his model repository or running simulation from any location through the Web. This provides the possibility of running a simulation on a server, and connecting to it from multiple other clients through different VMA entities. This layered architecture allows us to provide the modelers with two options:

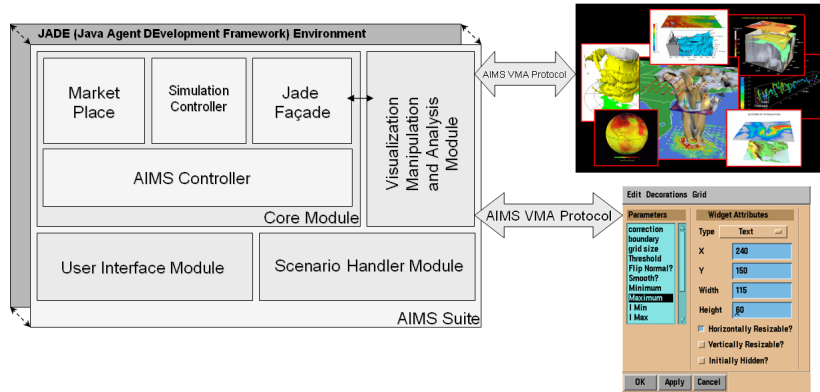


Figure 6: The Interaction between VMA Entities and the AIMS Suite.

1. The modelers can setup workshop like settings for critical infrastructure technicians, or even stakeholders and allow them to connect to a running simulation from their own machines. They can also provide the attendants with control panel like VMA entities. These VMA entities would allow the manipulation of the running simulation which results in a deep understanding of the current system for its operators. It is through such training sessions that the operators of an infrastructure get the chance of actually manipulating the system without the risk of encountering hazardous failures and therefore more easily identify the causes of possible threats.
2. From the point of view of computing resource consumption, the process of simulating multiple infrastructure systems is a compute-extensive task. Through the exploitation of a layered approach we can separate the interfaces (which are mostly thin clients) from the actual simulation server.

In the following section, we will show how the different components of the AIMS suite come together to create a unique modeling and simulation structure for critical infrastructure systems.

## 7 AIMS Workflow

The workflow of AIMS can be looked upon from two perspectives: end-users' perspective, and simulation controller perspective. On the one hand, the internal process of AIMS is completely irrelevant to the end-users; however, the AIMS controller is very much concerned about this procedure since it needs to control its proper execution, while on the other hand, the end-users are more concerned with the steps that they should take to be able to run a correct and consistent critical infrastructure simulation. We explain each of these perspectives in the following.

### 7.1 End-user Perspective

The interaction of the end-users with the AIMS suite mainly revolves around modeling a set of interacting infrastructure systems along with their internal sub-systems, and defining a scenario under which the analysis is performed (See Figure 8). For this purpose, the end-users should initially create the models of the component templates that are going to be used in the modeling procedure. Based on these component templates, they can then develop an integrated specification of all infrastructure systems that are going to participate in the simulation. This specification includes the set of initial parameters used for instantiating the component templates and the contracts (bindings) between them.

The specification of the available infrastructure systems in the simulation will allow the modelers to compose an appropriate scenario under which the simulation will operate. The collection of these information will create the model instance. Before commencing the simulation process, the modelers should specify what VMA entities they intend to employ during the simulation. All these information will be sent to the AIMS controller as a single entity named the Simulation Instance Document (SID). Upon the start of the simulation the end-users can observe the course of actions within the simulation and insert transients when(wh)ever they deem appropriate.

### 7.2 AIMS Controller Perspective

The AIMS controller is responsible for initializing and coordinating the actions of the AIMS suite modules. When the end-user submits a simulation instance document to the AIMS controller, this document is parsed and the set of component templates used in the model instance are identified. The specification of these component templates along with their initial parameters are sent to the JADE Façade where peer agents are created for each of them. Each of the components and the services (capabilities) that they provide are registered in the market place.

The AIMS controller spawns a scenario handler instance to monitor and apply the simulation scenario and also launches the appropriate VMA entities that have been de-

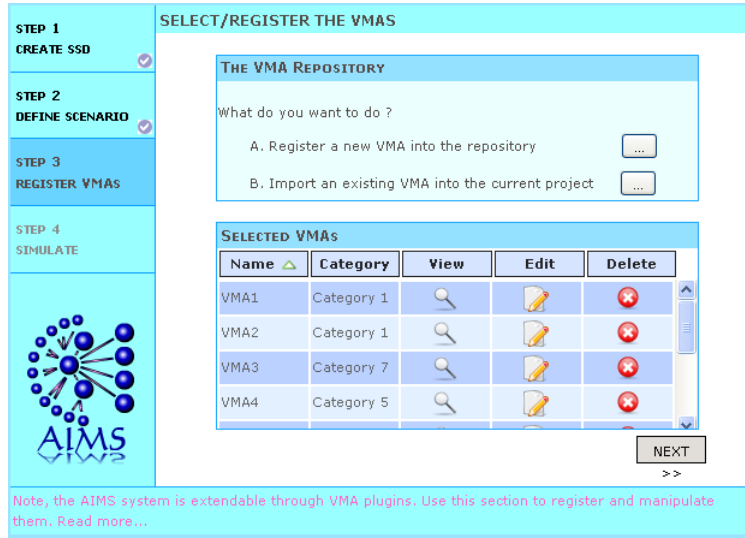


Figure 7: Registration of Visualization, Manipulation and Analysis Entities through an AIMS User Interface.

fined in the simulation instance document. The simulation starts with the commencement of the execution of the scenario handler module, and ends either with the direct interference of the end-user or the expiration of the specified simulation time in the simulation scenario.

## 8 Case Study

To show how the structure and behavior of various systems can be modeled and simulated, we employ the physical setting of an electronic service provider as a case study (See Figure 9). The electronic service provider is catering three different types of services for its end-users. It is hosting its applications on three servers in a proprietary network. To make its services more accessible, it has duplicated each of them on two different servers. For example, Service 1 has been both hosted on Server 1, and Server 2. In this case, if one of the servers fails the other will be able to provide the rest of the customers with the appropriate service.

As it can be seen in Figure 9, the servers are receiving their required electricity power from one of the two electricity suppliers in the network. The servers are hence dependant on the energy provided by these generators and will go down if the corresponding supplier fails to provide sufficient energy. Furthermore, the services on each server are only able to serve a certain amount of customers at a time; therefore, any extra request will be discarded. A proper response to each customer request will financially benefit the service provider, and if the request is dropped the provider will loose some money.

The service provider is also dependant upon an Internet Service Provider (ISP) for its Internet connection. The connection provided by the ISP can effect the number of requests that the service provider receives. If the connection line is of high quality, the service provider may receive

up to twice as many requests as it receives when the connection is poor.

The owners of the service provider are willing to find out the relationship between the total financial benefit of the company and the quality of the services that they receive from the electricity suppliers and the Internet service provider. For this purpose they create the proper simulation models and run four major scenarios:

1. In the first scenario, the services are regularly provided to the end-customers until the 200<sup>th</sup> cycle when the first electricity supplier is shut down. This discontinuity of service provisioning from the first electricity provider continues until the 250<sup>th</sup> cycle.
2. The second scenario is very much similar to the first scenario with the slight difference that the second electricity supplier is shut down instead of the first one in this case.
3. To investigate the effect of change in the bandwidth provided by the ISP, the bandwidth is increased between the 200 – 250 time cycles.
4. The last scenario is a complex combination of several actions. The bandwidth provided by the ISP increases between the cycles 150 to 250. The electricity suppliers one and two are shut down at 350 and 200, and are restarted at 400 and 250, respectively.

Through simulation, the owners of the infrastructure attempt to understand the degree of interdependency of their firm and the other services that they receive. The major benefit of a simulation developed through the AIMS suite is that the outcome of the simulation can be analyzed using the analysis functionalities that are provided in the VMA modules. The study of the outcome of the simulation can help identify any hidden interdependency and the degree of their affect on the infrastructure system.

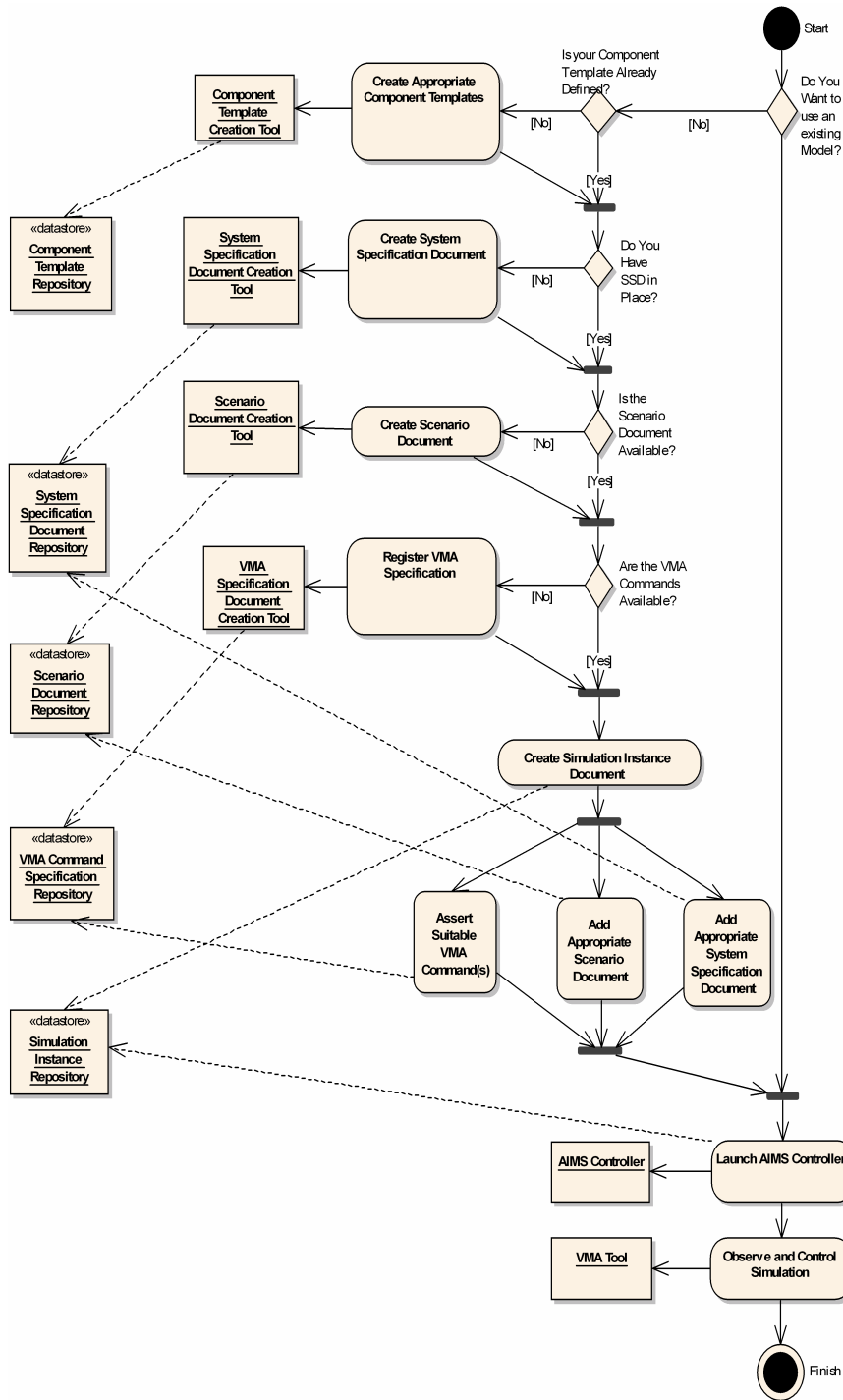


Figure 8: AIMS Workflow from the Perspective of End-user.

In this example, the apparent result of the executed scenarios is that the increase of bandwidth does not significantly affect the obtained benefit of the system (See Figure 12); therefore, the infrastructure owners can conclude not to invest a lot in increasing the amount of bandwidth that they currently have, since their infrastructure is not highly reliant on bandwidth. In other words, the simulation results show that the infrastructure has a very slight

dependency on bandwidth that can be neglected.

In contrast, the effect of the malfunction of any of the electricity suppliers is significant. As is visible in Figures 10, 11, and 13 the overall infrastructure benefit sharply decreases in the time cycles that the electricity suppliers have been shut down. Even more, the effect of the second supplier is greater than the first one. It can be inferred from the simulation results that it would be highly justifi-

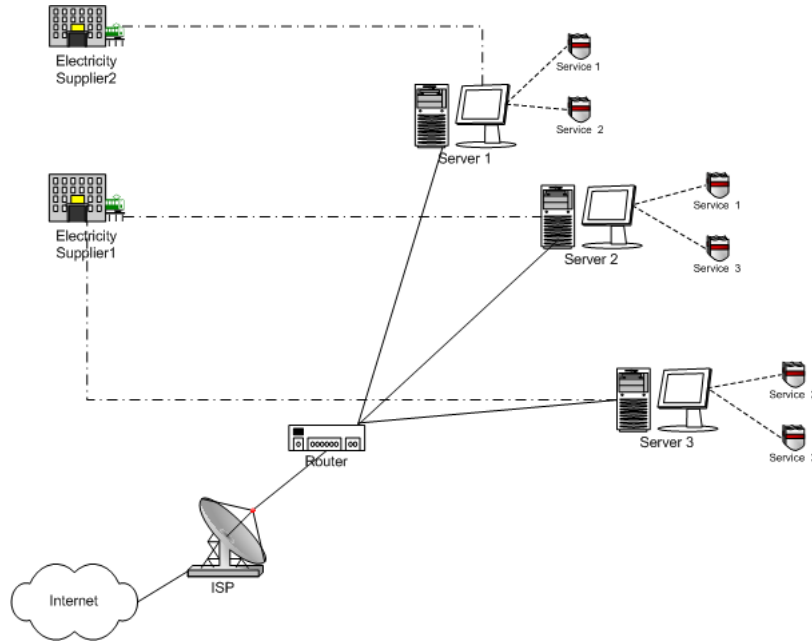


Figure 9: The Sample Topology Employed in the Performed Case Study.

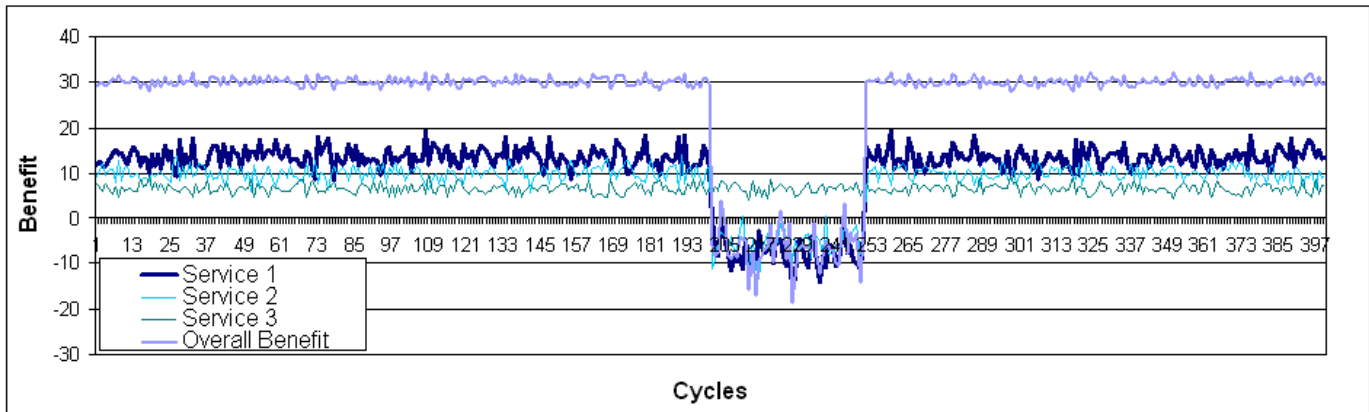


Figure 10: Simulation Results Obtained from Scenario 1.

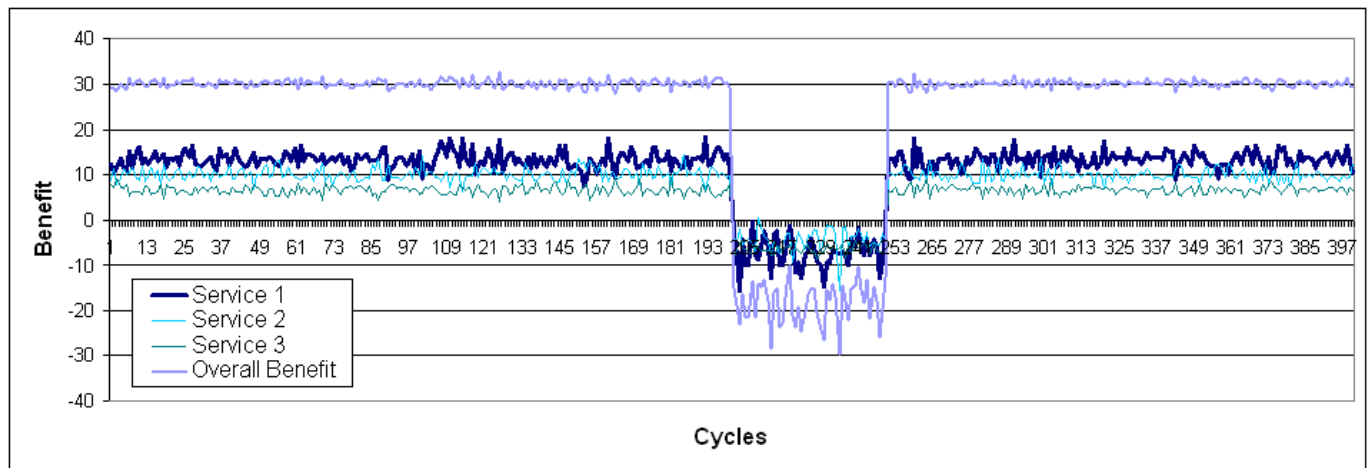


Figure 11: Simulation Results Obtained from Scenario 2.

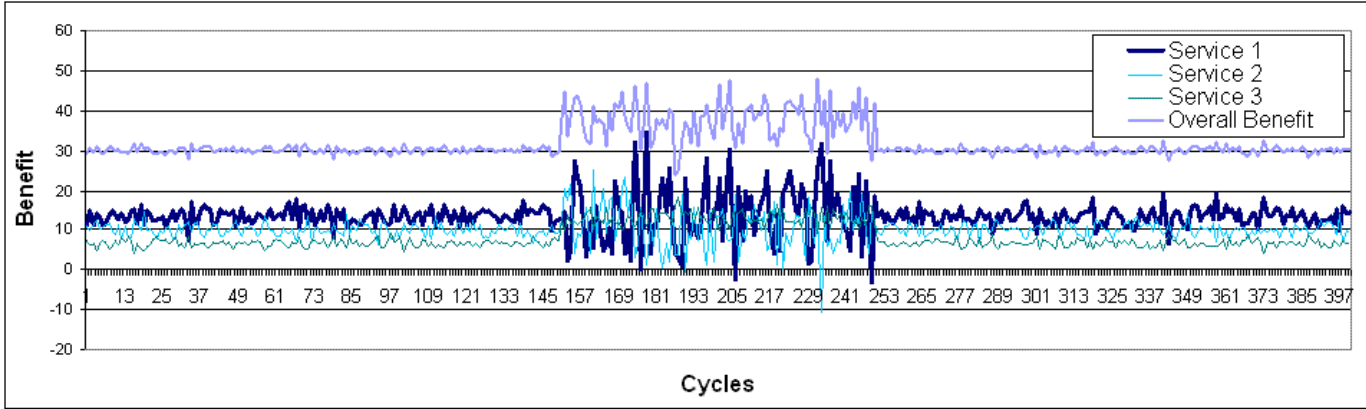


Figure 12: Simulation Results Obtained from Scenario 3.

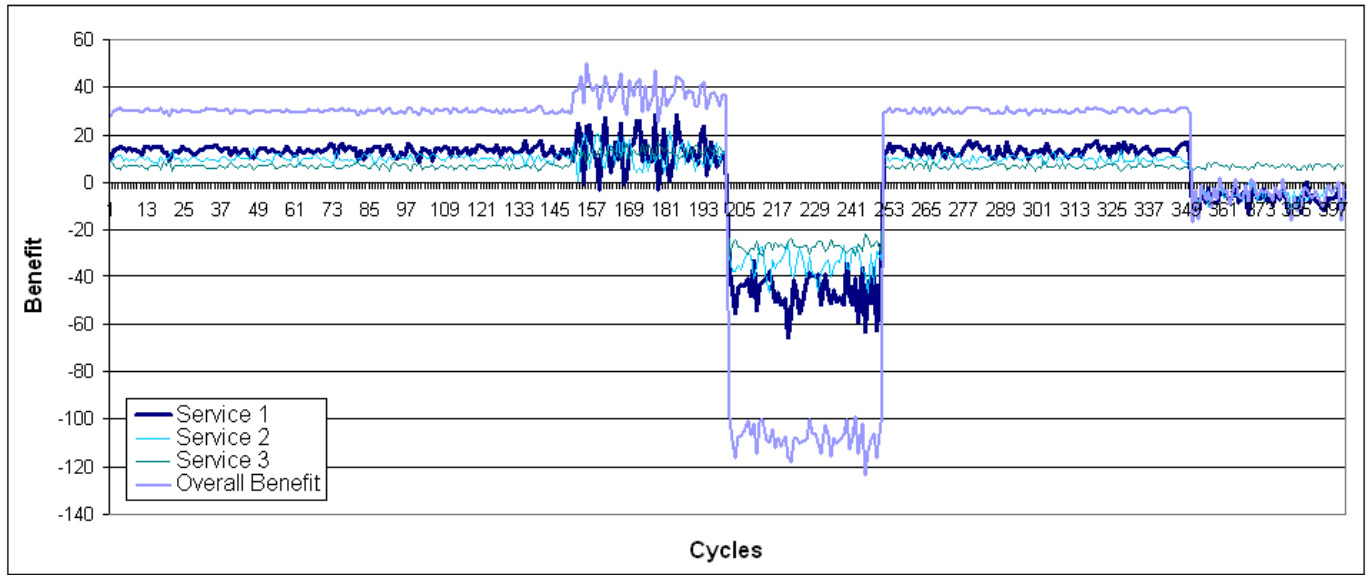


Figure 13: Simulation Results Obtained from Scenario 4.

fied to mitigate the risks posed by the second electricity supplier as compared with the first one due to the high dependency of the infrastructure benefits on the second electricity supplier.

The results of this case study show that the information revealed through controlled scenario-based simulation can be useful for inferring hidden information in the structure of interdependent infrastructure systems and hence evaluating the degree of their effect on the performance of an infrastructure system. These sort of simulations can be helpful in identifying blind interdependencies and their degree of severity.

## 9 Conclusions

In this paper, we have introduced an appropriate simulation suite for analyzing the behavior of interdependent crit-

ical infrastructure systems. The simulation suite, AIMS, is centered around the services that are provided by critical infrastructure systems and their sub-systems. Throughout the paper, we have explained the structure and functionality of the simulation suite modules, as well as the metamodel that is driving the suite. In the future, we aim to extend the modeling features of the AIMS suite by adding the UML-CI modeling profile (Bagheri and Ghorbani (2006b)) to its current metamodel. UML-CI is an MDA-oriented metamodel specifically designed for modeling critical infrastructure systems. This integration would assist the users of the AIMS suite in making appropriate choices for modeling their intended systems. We are also interested in creating a set of initial component template libraries for the ICT sector, so that we can easily model and simulate various settings of the ICT infrastructure systems.

---

## Acknowledgements

---

The authors generously acknowledge the funding from Iran Telecommunication Research Center to Dr. Ghorbani. They would also like to acknowledge the contribution of Mr. Gordie Noye, Iosif-viorel Onut, and Reza Zafarani to the development of the AIMS suite.

---

## REFERENCES

---

- Bagheri, E. and Ghorbani, A. (2006a). A service oriented approach to critical infrastructure modeling. In *Workshop on Service Oriented Techniques*. National Research Council, Canada.
- Bagheri, E. and Ghorbani, A. (2006b). Towards an mda-oriented uml profile for critical infrastructure modeling. In *International Conference on Privacy, Security and Trust*. ACM.
- Bagheri, E. and Ghorbani, A. (2007). Conceptualizing critical infrastructures as service oriented complex interdependent systems. In *International Conference of Information Technology and Management*. ACM.
- Barton, D. C. and Stamber, K. L. (2000). An agent-based microsimulation of critical infrastructure systems. Technical Report SAND2000-0808C.
- Basu, N., Pryor, R., and Quint, T. (1998). Aspen: A microsimulation model of the economy. *Computational Economics*, 12(3):223–41.
- Bellifemine, F., Poggi, A., and Rimassa, G. (2001). Developing multi-agent systems with jade. In *ATAL '00: Proceedings of the 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages*, pages 89–103, London, UK. Springer-Verlag.
- Bram, J., Orr, J., and Rapaport, C. (2002). Measuring the effects of the september 11 attack on new york city. *Economic Policy Review*, (Nov):5–20.
- CIP-Commission (1997). Critical foundations: Protecting americas infrastructures. Technical report.
- Dunn, M. (2005). The socio-political dimensions of critical information infrastructure protection (ciip). *International Journal of Critical Infrastructures*, 1:258–268.
- Ghorbani, A., Bagheri, E., Zafarani, R., Baghi, H., Noye, G., and Onut, I. (2006). Agent-based interdependencies modeling and simulation (aims). Technical report, Technical Rep. No. IAS-TR01-06, Intelligent and Adaptive Systems Research Group, Faculty of Computer Science, UNB (for a copy of the report email ghorbani@unb.ca).
- Haimes, Y. Y., Horowitz, B. M., Lambert, J. H., Santos, J. R., Lian, C., and Crowther, K. G. (2005). Inoperability input-output model for interdependent infrastructure sectors. ii: Case studies. *Journal of Infrastructure Systems*, 11(2):80–92.
- Leontief, W. (1951). *The structure of the American economy*. Oxford University Press, New York.
- Leontief, W. (1966). *Input-output economics*. Oxford University Press, New York.
- Panzieri, S., Setola, R., and Ulivi, G. (2004). An agent based simulator for critical interdependent infrastructures. In *Securing Critical Infrastructures*.
- Panzieri, S., Setola, R., and Ulivi, G. (2005). An approach to model complex interdependent infrastructures. In *16th IFAC World Congress*.
- Purdum, T. (2001). Rolling blackout affects a million across california.
- Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems Magazine, IEEE*, 21(6):11–25.
- Thissen, W. A. and Herder, P. M. (2003). Critical infrastructures: challenges for systems engineering. In *IEEE International Conference on Systems, Man and Cybernetics*.
- Zhang, G. and Bose, A. (1989). Scenario building for operator training simulators using a transient stability program. *Power Systems, IEEE Transactions on*, 4(4):1542–1549.