NRC Publications Archive Archives des publications du CNRC

A Framework for the Manifestation of Tacit Critical Infrastructure Knowledge

Bagheri, Ebrahim; Ghorbani, Ali A.

Publisher's version / Version de l'éditeur:

Sustainable and Resilient Critical Infrastructure Systems, pp. 139-159, 2010-01-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=a63e9f71-f386-4100-8d73-36d4aab0cb73>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=a63e9f71-f386-4100-8d73-36d4aab0cb73>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

A Framework for the Manifestation of Tacit Critical Infrastructure Knowledge

Ebrahim Bagheri¹ and Ali A. Ghorbani

Institute for Information Technology, National Research Council Canada

ebrahim.bagheri@nrc-cnrc.gc.ca

Faculty of Computer Science, University of New Brunswick, Canada

ghorbani@unb.ca

Abstract Critical infrastructure systems are tightly-coupled socio-technical systems with complicated behavior. They have emerged as an important focal point of research due to both their vital role in the normal conduct of societal activities as well as their inherent appealing complications for researchers. In this chapter, we will report on our experience in developing techniques, tools and algorithms for revealing and interpreting the hidden intricacies of such systems. The chapter will include the description of several of our technologies that allow for the guided understanding of the current status quo of infrastructure systems through the Astrolabe methodology, the formal profiling of infrastructure systems using the UML-CI meta-modeling mechanism, and also observing the emergent behavior of these complex systems through the application of the agent-based AIMS simulation suite.

1 Introduction

Critical infrastructures are among the most significant technical systems that influence the ordinary life of any person or the normal operation of any industrial sector. Their importance is mainly due to the type of facilities/utilities that they provide. These facilities (either in the form of asset supply or service provisioning) serve as the building block for any other simple or complex functionality of the society. The outputs of the infrastructures although complex in nature, can be

¹ This work was conducted while the first author was with the University of New Brunswick.

thought of as the essential atomic inputs to other more complex systems. Apparently, without the proper operation of infrastructure systems, the function of other dependant systems would be disrupted.

The very interesting fact is that throughout the years, infrastructure systems themselves have become dependent on each others' outputs, turning the vertically integrated systems with only a few points of communication, into horizontally integrated systems with various points of interaction in many of their dimensions [1]. Analogous to the dependency of other systems on infrastructures, it can be observed that infrastructure systems themselves are inter-reliant or in other words tightly coupled. As has been extensively studied in the field of fault tolerant computing, a complex system built from interacting components is exposed to a high risk of failure derived from the possibility of mal-function in any of its components. The degree of effectiveness of the failing component in the overall architecture (e.g. in a digital circuit it can be thought of as the number of input/output connections of a specific component), suggests an estimate of the degree of damage or harm that its failure will cause.

The high interdependency of infrastructures makes their characteristics somewhat similar to the explained systems. Their extreme inter-connectedness makes one think of them as different components of a single network. A failure in a node of this complex network of interdependent infrastructures, results in catastrophic failures; many of which had not been foreseen. These failures are in many cases the result of the propagation of failure through these interconnected systems. Failure propagation is known as the *cascading effect* or *ripple effect* and has been the inspiration for many fruitful research efforts [2].

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. The tragic 9/11 terrorist attacks in New York City are clear depictions of threat towards the most influential and operational elements of both the society and industry.

Researchers have pursued two main directions of investigation for studying the structure and behavior of critical infrastructures, each of which has been to our belief successful. The researchers in the first group have been mainly involved with the study, analysis, and understanding of the infrastructures' current makeup. Their goal has been to identify methods, techniques, tools and schemes for describing the current status of an infrastructure. Based on this understanding, these researchers exploit various vulnerability, risk and/or threat assessment methods to gain more insight into the operation of an infrastructure. A fine-grained deployment of this process reveals many of the possible causes of failure and to a great extent their consequences. It is understandable that the result of this process would only shed light on the types of failure that are a clear result of a breakdown or

malfunction of a system. It should also be noted that although many of the possible roots for failure are detected in these approaches, but not all of their consequences are visibly perceived and understood.

Fig. 1. The high-level structure of the framework developed for critical infrastructure modeling, simulation and profiling.

From the understanding gained from the study of the infrastructure organization, and the identification of their points of weakness, proper risk mitigation strategies can be proposed and ranked based on, e.g., three metrics namely costliness, time-consuming, and effectiveness [3]. Each of these metrics can be weighted and suitable mitigation strategies can be selected to enhance infrastructure safety and protection according to the priorities of the infrastructure management and their strategic directions.

The other direction of research has mainly focused on the understanding of the dynamic behavior of infrastructure systems. In this route, the investigators attempt to explore the many paths of infrastructure process and operation through which they will try to identify any causes for instability. The search mainly revolves around the discovery of the paths that introduce more risk of catastrophic failures into the system. Usually such a process is supported by agent-based simulations or differential and/or algebraic-differential equations modeling. Some of the well-known examples of such approaches are the Agent-based Interdependency Model-

ing (AIMS) suite [4], Aspen [5], the inoperability input-output model, which has been based upon Leontief's I/O model [6].

In this chapter, we include the description of a platform that consists of several of our technologies that allow for the guided understanding, classification and decision making based on the current status quo of infrastructure systems. As depicted in Figure 1, the platform constitutes the following interacting components:

1. *Astrolabe* [15] is a collaborative goal-based risk analysis methodology, which is based on the causal analysis of system risks. It allows the analysts to both align the current standpoint of the system with its intentions and identify any vulnerabilities or hazards that threaten the systems stability;
2. *AIMS* [4] is a multi-agent modeling and simulation suite that allows its users to create models of an infrastructure and observe the behavior of the modeled system through actual simulations. AIMS provides the means for inserting transients into a running simulation for observing the outcome of the occurrence of an unexpected event;
3. *UML-CI* [2] is a model driven architecture-based profiling method that provides its users with the required high level meta-models to create a profile of an infrastructure system. Its main advantages are that it gives initial insight for infrastructure analysis and system identification, provides sound basis for common understanding, communication, and knowledge transfer, it also allows the documentation of best practices.

In the rest of this chapter, we will provide more detailed information about each of these components and technologies, discuss how they are able to collectively form a framework for modeling, simulating and profiling critical infrastructure systems.

2 Risk Analysis for Critical Infrastructure Systems

Risks are the likelihood and the degree of severity of unplanned or undesirable states. Analogous to its nature, the definition of risk is very much dependant on context and contextual factors. What might not be considered as risk in one context may be identified as a major risk in the other. Even in the same context, in cases where these factors are qualitatively analyzed, different points of view may rank the severity or likelihood of a risk with dissimilar values, which results in more ambiguity. However, currently the common understanding is that analyzing risk requires methods to identify the sources of the events that drive a system or an organization towards the states that can expose it to risk. Therefore, besides the direct events that lead to unsafe conditions, the courses of action guiding these

events and even, more importantly, the purpose of these actions need to be identified and well understood.

It makes sense to track the sources of risk back to their origins (intentional causes), because without the proper adjustment of these roots, it would be impossible to change the outcomes. In other words, the probable formation of the branches of a behavior tree in a system is dependent upon the arrangement of its roots. It is undeniable that local changes made to the branches of this tree can have quick and even spontaneous effects, but they do not have long-term durability.

From a systems engineering perspective, the roots of the behavior of a system relate to the goals that it pursues. Parson argues that goal attainment is an indispensable aspect of any system [7]. Let's consider two very different systems, each formed based on communal and associational relationships [8]. A system developed for a communal relationship focuses more on mere member gratification. An example of such systems can be the formation of a student association that organizes student activities. In a system of associational relationships the membership of the participants is no longer because of the importance or pleasure of a relationship and is more or less attained so that the results of this relationship can indirectly help the participants create other systems based on communal relationships. For instance, employment in a job is a type of associational relationship that is accepted by a person so that he/she can establish his/her own family (a system of communal relationships). There are fundamental differences between the natures of the systems developed from these two types of relationships. However, one common factor exists in both of them and that is goal attainment. Even in a student association that has been established for a very informal cause, the inability to cater its objectives may result in the discontinuity of the relationship. Therefore, goal attainment has primacy over all other activities of any type of system.

Goals are often the result of the strategy selection process through which system stakeholders identify its direction and decision making criteria [9]. To achieve system's goals, the stakeholders devise plans to undertake a series of actions. The implementation of the course of these actions situates the system under various states and conditions among which unsafe states may also be found. The existence of these states depends on the degree of willingness of the system to take risks. If the risk is outweighed by the benefits perceived by system stakeholders, then that specific action may be performed. Based on this description, system goals cater related criteria and metrics for action generation and selection. This means that goals are the driving force of system behavior. Hence, the behavior of a system can be justified by its goals. Goals can also be used to appraise current system performance. The appraisal can be based on the gap between the desired system derived from its initial set of goals and its current standing [10]. The iterative process of re-appraisal can be employed to adjust a system, such that it moves towards its initially planned ends.

More specifically, a system can be described by its goals and objectives, and the set of scenarios that it undertakes to support the operationalization of its goals. Studies show that very few systems actually achieve their intended goals to the ex-

tent of their initial desire [11]. This may be due to several factors. It may be either because the scenarios that have been supporting the attainment of system goals are not in full alignment with these goals, or it may be due to the incomplete or incorrect undertaking of these scenarios. Empirical evidences from the current behavior of a system can help identify the gap between system goals and the present practice. The existence of such a gap is not a rare incident in many systems. Even in political systems, the leaders initially acknowledge the goals of their party, but over time, as they acquire power, become rather conservative in order to maintain the current situation and as a consequence the initial goals of the party may be sacrificed.

Fig. 2. Key concepts of the Astrolabe methodology.

A system can also be caught in a situation where the rapid change of its context leads to the requirement of goal amendment. The need for a quick adjustment can result in a condition where the goals of a system are no longer well defined. This

situation can be described with the *Garbage Can theory* [12]. This theory describes a condition where a system has been left with a set of outdated solutions, which are the remainder of the previous goals of the system and as the goals of the system are amended and the problem statement changes, the system starts looking for a suitable problem statement to match the solutions that it has to offer. Therefore, the risks associated with this state should be also analyzed.

Systems that need to incorporate the role of human resources into their structure also face a different kind of risk. *Motivational theory* formalizes the reasons behind the involvement of any human resource into a system through inducements and contributions [13]. Inducements are desired aspects of participation. For instance, inducements of working for a company are a suitable salary along with insurance options. Contributions on the other hand, have negative utility from the human resource perspective, but are the requirements for participation. Constant traveling for a salesperson is a type of contribution that he/she has to make in that position. In cases where the contributions and inducements of a position in a system contradict each other, risks may arise for the system, since the human resource may not adhere to the requirements of the contribution (This has also been addressed as orthogonal goals of organizations and individuals in the related literature).

Given these issue and important consideration, Astrolabe provides a collaborative multi-perspective risk analysis methodology that identifies complex systems, e.g. critical infrastructure systems, risks and vulnerabilities based on systems operational activities and its strategic goals and objectives.

2.1 Astrolabe Key Concepts

The risk analysis process in Astrolabe is based on five key concepts shown in Figure 2. Astrolabe aims to fully identify instances of these concepts for any target system. This information can then collectively describe a system and its status. The key concepts defined in Astrolabe are:

1. *Perspective* is the mental and conceptional standpoint of the representative of a group of related individuals through which they examine the universe of discourse (e.g. the target system being examined).
2. *Goal* is the conceptualization of the ideal state of affair for a system. A system may pursue multiple goals.
3. *Evidence* is an activity currently practiced in the universe of discourse for the attainment of one or more goals.
4. *Obstacle* is a goal or evidence, which can be from the outside of the universe of discourse that obstructs or delays the achievement of one or more system goals.
5. *Hindrance* is evidence, from within or outside the universe of discourse that disrupts the normal operation of one or more system activities.

2.2 Astrolabe Process Overview

The Astrolabe methodology is intended to support the process of identifying the risks that threaten a complex system such as critical infrastructure systems and provide the means to trace the roots of these risks. It aims to provide means for analyzing the sources of risk so that proper mitigation strategies can be selected. Moreover, it can be used concurrently with system design methodologies to assist risk identification throughout the complex system design process.

The Astrolabe methodology has an iterative nature in which all of its phases can be re-visited at any time. That is, once the steps in any of the phases have been performed, there may be a need to go back to the previous phases and refine some of the information.

The Astrolabe methodology has seven major phases. This does not mean that the completion of these phases ends its lifecycle. As it is inherent in the nature of risk, analyzing and managing risk is a non-stop activity that requires constant re-visits and refinements. Within our proposed methodology, after the completion of each iteration, regular examination of the deliverables and products of each phase is required, so that any change in system goals and activities can be captured and suitable risk identification and analysis activities can be performed. The major phases of the Astrolabe methodology are:

1. *Boundary Specification*: The functional and intentional spaces of a system usually span multiple domains; therefore, risk analysts should initially specify which one of the aspects of the system attracts their attention the most and is going to be the target of investigation. Boundary specification should also consist of the identification of the sub-systems of a larger system that are of interest.
2. *Perspective Identification and System Analysis*: In this phase, risk analysts decide on the parties that are going to be involved in the information elicitation process. For instance, they may decide that only two points of view, one from the CEO and one from the marketing representative, satisfies their needs and Requirements.
3. *Preliminary Hazard Identification*: Threats and vulnerabilities of a system can be identified by a close examination of the goals and evidences that have been identified up to the current point. For example, the representative of a perspective can look at the set of goals and evidences that he/she has identified and think of the risks that threaten their operation. This hazard identification process does not produce a complete list of all system hazards, since neither each perspective is complete nor the elaboration has been extensive enough yet.
4. *Perspective Integration*: Having identified a set of goals and evidences in each perspective, risk analysts should consolidate all this information into a unique representation. Within this unique representation, conflicts between the statements of the different perspectives should be resolved.

5. *Process Refinement*: The concentration of each perspective on the issues more relevant to its position may result in a sub-optimized view of the problem domain. This issue can be handled by aggregating and refining the information provided by all of the perspectives. In this way, each perspective will become aware of the goals or evidences that he/she may have missed by viewing the information provided by other perspectives.
6. *Risk Analysis*: Identification of risk in Astrolabe does not occur in a single phase and crosscuts all of the phases. The risks that are identified throughout all phases are analyzed in this phase. This analysis includes ranking goals, evidences, capabilities, and resources based on the degree of the threats that they pose. Based on this ranking, risk analysts will be able to concentrate on risk filtering and proper risk mitigation strategy selection.
7. *Quality Measurement and Validation*: The quality of a risk analysis process is very much subject to the expectations and needs of risk analysts and system administrators; however, in many cases, the integrity and correctness of the risk analysis process needs to be validated and the quality of the deliverables be assessed. In Astrolabe, the quality of the products, deliverables, and the analysis process is evaluated based on five metrics. These metrics are namely *accuracy, consistency, completeness, traceability and unambiguity*.

In principle, Astrolabe mainly focuses on catering a formal framework for multi-perspective goal and risk negotiation, and in particular, providing a structured approach to multi-perspective risk identification, prioritization and mitigation. These features of Astrolabe provide several benefits for both experienced and inexperienced risk analysts. Experienced risk analysts can exploit Astrolabe's risk classification and ranking methods to semi-automate the risk analysis procedure and hence decrease the required risk analysis time and increase risk analysis procedure for complex interconnected systems such as critical infrastructure systems. On the other hand, inexperienced analysts can gain benefits from the process, by analyzing the information provided by Astrolabe. For instance, Astrolabe provides the degree of cross-perspective consistency and information alignment, which can help inexperienced analysts in seeing and evaluating their own performance. Inexperienced risk analysts can benefit from these two features of Astrolabe, by comparing themselves within the framework of Astrolabe with an experienced risk analyst. The comparison would allow them to observe the degree of deviation of their risk analysis procedure with that of an experienced analyst.

Summarily, the most outstanding features of Astrolabe that make it a very suitable option for risk analysis in critical infrastructure systems are: 1) Identifying risk from its system origins and stakeholders strategic objectives; 2) Incorporating multiple perspectives on system intention and structure for risk analysis that helps avoid any information misinterpretation and overlook that is common while analyzing large and complex systems such as critical infrastructures; 3) Quality measurement metrics for process validation purposes; and 4) An iterative model for risk management with clearly enumerated phases, steps and deliverables.

3 Critical Infrastructure Systems Behavior Analysis

Multi-agent systems provide a collective understanding of a very complex system through a unified view on the aggregate behavior of their constituent elements (agents). Infrastructure systems can hence be very well modeled by multi-agent systems. Depending on the modelers understanding of the structure and organization of an infrastructure, appropriate multi-agent systems can be created to mimic the actual behavior of an infrastructure system.

Fig. 3. An agent-based model of an infrastructure system.

There are two main issues to a proper design for a multi-agent based architecture: agents' internal design, and agents' environmental commitments and collaborations (See Figure 3). The first issue addresses the internal representation of each agent within the multi-agent architecture. This aspect specifies what functionalities/services an agent requires/provides, what resources it owns/consumes, and what goals it pursues. These specifications would allow an agent to decide on its reactions under various conditions. For example, an electricity generator can be modeled as an agent in a multi-agent architecture. If the generator goal is to maximize its profit, it would regardless of its current load, accept new requests, but if

its objective is to provide a very high quality of service to its customers, it would prefer to accept a lower number of requests but provide them with a better service.

Investigators can use a very well known multi-agent architecture known as BDI, which specifies the internal structure of an agent through three concepts: Belief, Desire and Intention [14]. An agent's belief is its perception of the surrounding environment. Desire is the agent's motivational states or goals, and an agent's intention is its current decision in order to reach his goals (desires). Agent intentions are realized through intelligent planning.

Following the previous example, an electricity generator can forecast power market's future, and plan not to forward sale its electricity supplies because of a rise in electricity prices in the near future. As this example portrays, the sort of planning problems in a multi-agent system that simulates an infrastructure system is usually complicated and have to be discovered and optimized in a multidimensional space and hence requires complex techniques.

Fig. 4. Infrastructure service exchange in the multi-agent architecture of AIMS.

An agent's societal interaction within a multi-agent architecture is the other point that should be specified when modeling an infrastructure system. This would include features such as *Negotiation*, *Cooperation*, and *Coordination*. Negotiation

is a social practice through communication that is experienced for regulating power, resources and commitments among different agents.

Agents are autonomous by their definition (as their corresponding infrastructure systems are in the real world), but require interaction to reach their goals. As a result of a successful negotiation, cooperation can be achieved. Negotiation and cooperation procedures appear to be able to reveal a set of infrastructure interdependencies. The negotiation of agents (each agent being an infrastructure system) over each others' resources depicts their interdependencies. This allows the revelation of many physical or cyber interdependencies. However, still logical interdependencies are not revealed.

Logical interdependencies are related to a more sophisticated notion in multi-agent systems: *Coordination*. Coordination is conceptually different from cooperation, since two coordinated agents may not in fact cooperate with each other, however make coordinated decisions. Suppose that the price of oil is strictly dependent upon the peace process in the Middle East. Therefore, major oil producers and consumers, try to coordinate their decisions with the political developments in that region. It is visible that oil companies have no direct cooperation with the political side of the issue, but they are highly coordinated with their decisions. This coordinated behavior can be considered as logical interdependency between two infrastructure systems.

Given these important issues relevant to the multi-agent design for critical infrastructure behavior analysis, we have designed an Agent-based Interdependency Simulation Suite (AIMS) that supports for the understanding of the behavior of complex interdependent systems. The major advantage of AIMS over similar existing simulation suites such as CISIA [16] and ASPEN [5] is the following:

1. 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 subsystems, or even its resources. These component templates are properly stored in AIMS repositories and can be used in various modeling procedures.
2. Secondly, besides the concept of an agent, 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. Figure 4 shows how this process is supported in AIMS.

Fig. 5. The overall structure of the AIMS metamodel.

3.1 AIMS 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 those 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 5, a model instance of the AIMS metamodel can contain as many component templates, and contracts (binding) instances as required, but can only be associated with a single scenario instance each time. More formally these four components are described below:

1. *Component template* 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 Mobile Switching Center can be considered as one of its sub-systems; therefore, it can be modeled through a component template in AIMS;
2. *Contract (Binding)* provides the means for two infrastructure systems modeled as component templates in the model agree to share their goods through the exchange of their services. For instance in Figure 4, a contract has been established between the electricity infrastructure and the telecommunication infrastructure for the exchange of electricity;
3. *Scenarios* specify the initial setting of the simulation model, and provide guidelines on what paths the simulation has to follow during the course of its execution. Since many of the interesting (and at many times harmful) events within an infrastructure system are unpredictable and rare, scenarios provide the possibility to design simulation routes that will give rise to problematic events;
4. *Model instance* is global metaclass that incorporates all the required information for the execution of a proper infrastructure simulation process including information about what component templates are involved, which scenario will be executed, and what contracts will be evaluated between the simulation components.

3.1 AIMS Suite Structure

The AIMS suite 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:

1. *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 Facade.
2. *Visualization, Manipulation, and Analysis (VMA) 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.

3. *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.
4. *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.

Fig. 6. A four-layered meta-modeling architecture extending UML for the water and sewage infrastructure.

Summarily, AIMS is a multi-agent modeling and simulation suite that allows its users to create models of an infrastructure and observe the behavior of the modeled system through simulations. One of the distinguishing features of AIMS is that it provides its users with a set of predefined component templates (e.g. pipes, switches, etc). The organization of an infrastructure can be built through the instantiation of these component templates into actual software agents. AIMS also provides the means for inserting transients into a running simulation for observing the outcome of the occurrence of an unexpected event. The set of these transients called scenarios, control and hence effect the operation of the simulation. Visualization and analysis modules can also be plugged into a running simulation to view and study the structure and behavior of the simulated infrastructure system.

4 Profiling Critical Infrastructure Systems

UML-CI is a meta-modeling language that provides the required constructs to formulate and properly structure and represent information related to and about a critical infrastructure system. It is based on the profiling mechanism provided by the UML modeling language. A UML profile allows the selection of a subset of the UML base metamodel, denotes the common model elements required for the specific modeling domain, and specifies a set of required well-formedness rules. Well-formedness rules are a set of constraints that accompany a family of models to show their proper composition. Object Constraint Language (OCL) is a strongly typed declarative language that is based on the mathematical set theory and predicate logic and is extensively used with UML for this purpose. Figure 6 depicts an example of how a base UML class can be gradually extended to provide more specific models for two different instances of a single water and sewage infrastructure metamodel.

The major stimuli for devising a reference model for critical infrastructure modeling, UML-CI, are multifold. Here, we elaborate more on some of the most important advantages of an infrastructure profile that have led to the proposal of UML-CI.

1. *Base Recognition and System Identification*: A unified platform with extensive components allows an initial understanding of the organization of an infrastructure. Novice modelers can exploit this reference model to plan the modeling process. It also suggests the type of information that needs to be collected throughout the modeling practice.
2. *Common Understanding and Communication*: The elements of the reference model bring about a shared conception of the modeling task between the members of the team. This consequently eases communication and collaboration among the involved people. One of the other major advantages of the employment of UML-CI would be that a mutual understanding between the modelers and the infrastructure stakeholders can be reached more easily.
3. *Current Understanding (Knowledge)*: A completed profile (an instantiated version of UML-CI) can provide a detailed view of the present infrastructure setting; however the level of detail of this completed profile depends vastly on the granularity of the information collected through the information acquisition phase. Many of the interdependencies (e.g. physical, cyber, or geographical) between different infrastructures may be illuminated through this process. The current understanding can also aid in infrastructure management by providing more detailed understanding of the current situation.
4. *Knowledge Transfer*: Infrastructure modeling projects are extremely specialized, requiring the team members to be acquainted with both the modeling tasks and infrastructure domain. As a result, those involved are highly skilled people that pose a great risk if they decide to leave the team. Having a standard modeling notation reduces this threat by providing the opportunity for the

group to add new members. The new members can quickly grasp the problem domain using the detailed reference model.

5. *Best practices and New Understanding*: Different modeling teams based on the same framework can communicate and transfer their experience. This exchange of knowledge can occur through the attachment of thoughts, ideas, recommendations or even standards to the metamodel or its elements. The proper transfer of the best practices would bring about a more concrete understanding of infrastructure organization and behavior.
6. *Documentation and Re-use*: A reference model such as UML-CI can provide a highly readable and semantically rich documentation of the infrastructure that is being modeled. The created document can be easily understood by both the stakeholders and the modeling team for it has a graphical notation. The models created based on this reference model can be further re-used if need be. Similar infrastructures can also be modeled through the refinement of an existing model without the need for starting from scratch.

Fig. 7. The five high-level critical infrastructure models of UML-CI.

4.1 High-level Critical Infrastructure Models

The UML-CI critical infrastructure reference model consists of five main models, each of which addresses a different issue. Each of the metaclasses within a model is relevant to the concept that rules the model. These five models are briefly explained in the following lines and shown in Figure 7:

1. *Ownership and Management Model*: The elements within this model provide the means for the identification of the managerial aspects of an infrastructure. These characteristics include the specification of the infrastructure stakeholders, the government(s), and the geographical span of the infrastructure. It also includes features for defining the policymakers, regulations, and the roles they play in the infrastructure operation.
2. *Structure and Organization Model*: This model provides the means for specifying the makeup of an infrastructure system. It has three major metaclasses namely infrastructure, system, and task.
3. *Resource Model*: Resources are the raw processing materials that are required for the operation of an infrastructure. They are consumed, produced or processed within the operations of an infrastructure system. This model provides the metaclasses for defining different types of resources that may be encountered in a critical infrastructure environment.
4. *Threat, Risk, Vulnerability (TRV) Model*: One of the main reasons for modeling infrastructure systems is to identify the hazards that threaten its operation. The TRV model, provides various metaclasses so that these hazards can be categorized and their causes, consequences and possible mitigation strategies be clearly specified.
5. *Relationship Model*: The relationship model provides many different metaclasses (derived from KernelAssociation) for connecting and joining the concepts that have been identified in the previous models. For example, although various hazards that threaten the operation of an infrastructure can be specified in the TRV model, but they are not specifically attached to the system, or task that they are actually threatening. To address this concern, the relationship model provides suitable metaclasses, so that all of the created classes in the previous models can be integrated into one unique model.

To make the employment of the proposed infrastructure reference model easier, the metaclasses and well-formedness rules of UML-CI have been added to the Together Architect 2006 CASE tool. This facility provides the modelers with the choice of easily creating a model based on the proposed reference model in a graphical environment with drag and drop features. UML-CI metaclasses integrated in this tool can be easily selected and instantiated, and joined together in an integrated environment. The integration of UML-CI into this CASE tool allows the end-users to reap extra facilities such as automatic document generation, and

model checking (based on the UML-CI well-formedness rules) that are already available in Together Architect.

Summarily, UML-CI is a reference model for profiling and modeling different aspects of a critical infrastructure system. The metaclasses in this reference model are categorized in five major high-level models that address various aspects of infrastructure organization and behavior. The most important concerns that have been addressed in this reference model are the issues of critical infrastructure ownership and management, their internal organization and system structure, asset classification and identification, and risk profiling. Other than the process of modeling and profiling critical infrastructure systems, UML-CI also concentrates on subjects that are of high importance to the management of critical infrastructure systems such as providing ground for creating common understanding and communication between infrastructure stakeholders, knowledge transfer, and documentation of best practices. Based on the models within UML-CI, infrastructure system knowledge bases can be built to aid the process of infrastructure system modeling, profiling, and management.

5 Concluding Remarks

Critical infrastructures are networks of interdependent, mostly privately-owned, man-made systems and processes that function collaboratively and synergistically to produce and distribute a continuous flow of essential goods and services [1]. These highly complex systems can be classified as socio-technical organisms that have some sort of hidden consciousness. Due to the potentially severe repercussions that infrastructures failure can induce, they have been the target of many fruitful research activities, which attempts to model, simulate and understand their behavior. In the past few years, we have focused on analyzing various aspects of critical infrastructure systems, developing a variety of technologies ranging from the analysis of the status quo of infrastructure systems through devising a goal-based risk analysis methodology to dynamic analysis of the behavior of critical infrastructure system using the agent-based interdependency modeling and simulation suite. Also, we have investigated the important issue of profiling and codifying the gained knowledge of infrastructure systems in a structured and formal representation model called UML-CI.

The shining aspect of our work is that our developed technologies form a complete all-round framework that provides most of the necessary tools for analyzing, understanding, storing and reporting on critical infrastructure systems. The developed technologies, namely AIMS, Astrolabe, and UML-CI form a comprehensive framework whose products can be exchanged between each technology and be further processed to induce and reveal hidden aspects of the critical infrastructures behavior and structure.

6 References

- [1] Ebrahim Bagheri and Ali A. Ghorbani, The state of the art in critical infrastructure protection: a framework for convergence, *International Journal of Critical Infrastructures*, Vol. 4, No.3 pp. 215 – 244, 2008.
- [2] Ebrahim Bagheri and Ali A. Ghorbani, UML-CI: A reference model for profiling critical infrastructure systems, *Information Systems Frontiers*, Published online: 10 July 2008, in press, 2008.
- [3] Turner, J. V., Hunsucker, J. L., Effective risk management: a goal based approach. *International Journal of Technology Management* 17 (4), 438–458, 1999.
- [4] Ebrahim Bagheri, Hamidreza Baghi, Ali A. Ghorbani, Alireza Yari, An agent-based service-oriented simulation suite for critical infrastructure behaviour analysis, *International Journal of Business Process Integration and Management*, Volume 2, Number 4, pp 312 – 326, 2007.
- [5] N. Basu, R. Pryor and T. Quint, ASPEN: A Microsimulation Model of the Economy, *Computational Economics*, Volume 12, Number 3, 1998.
- [6] Leontief, W., *Input-output economics*. Oxford University Press, New York, 1966.
- [7] Parsons, T. *The System of Modern Societies*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [8] Curry, J. a. *Sociology for the Twenty-First Century*, 3rd ed. Prentice-Hall, 2002.
- [9] Scott, W. R. *Organizations: Rational, Natural, and Open Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1992.
- [10] Scott, W. R. Technology and structure: An organization-level perspective. In *Technology and Organizations*, P. S. Goodman and L. S. Sproull, Eds. Jossey-Bass, San Francisco, 1990.
- [11] Gross, E. The definition of organizational goals. *British Journal of Sociology* 20(3), 277-294, 1969.
- [12] Cohen, M., March, J., and Olsen, J. A garbage can theory of organizational choice. *Administrative science quarterly* , 1972.
- [13] Simon, H. A. Rational decision making in business organizations. *American Economic Review* 69(4), 493-513, 1979, available at <http://ideas.repec.org/a/aea/aecrev/v69y1979i4p493-513.html>.
- [14] Rao, M. P. Georgeff, BDI-agents: From Theory to Practice, *Proceedings of the First International Conference on Multiagent Systems (ICMAS'95)*, 1995.
- [15] Ebrahim Bagheri, Ali A. Ghorbani: Astrolabe: A Collaborative Multiperspective Goal-Oriented Risk Analysis Methodology. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 39(1): 66-85 (2009)

- [16] Roberto Setola, Sandro Bologna, Emiliano Casalicchio and Vincenzo Masucci, An Integrated Approach For Simulating Interdependencies, Critical Infrastructure Protection II, Springer, pp 229-239, 2008.