Inf Syst Front (2010) 12:115–139
DOI 10.1007/s10796-008-9127-y

UML-CI: A reference model for profiling critical infrastructure systems

Ebrahim Bagheri · Ali A. Ghorbani

Published online: 10 July 2008

Abstract The study of critical infrastructure systems organization and behavior has drawn great attention in the recent years. This is in part due to their great influence on the ordinary life of every citizen. In this paper, we study critical infrastructures' characteristics and propose a reference model based on the Unified Modeling Language (UML). This reference model attempts to provide suitable means for the task of modeling an infrastructure system through offering five major metamodels. We introduce each of these metamodels and explain how it is possible to integrate them into a unique representation to characterize various aspects of an infrastructure system. Based on the metamodels of UML-CI, infrastructure system knowledge bases can be built to aid the process of infrastructure system modeling, profiling, and management.

Keywords Modeling and profiling · Critical infrastructure systems · UML profiles · Infrastructure system knowledge bases

1 Introduction

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 (CIP-Commission 1997). These highly complex systems can be classified as socio-technical organisms that have some sort of hidden consciousness. They can undergo aging in their lifecycle and hence experience different operational states. Complexness is an inherent property of these systems that stems from the nature of their tasks. These complex systems are mostly in charge of catering specialized services to a wide range of different parties. The type of collective services that is provided through these systems is so immense that the ordinary lives of all citizens are tightly coupled with their functionality. It is hence taken for granted that the sort of services that they provide should be ubiquitous, reliable, affordable and conveniently accessible (Jonsson 2005). Ubiquity implies that all citizens will have an equal opportunity in accessing these resources from any location. The penetration rate of many of these services is quite acceptable in many western countries. The Internet as one of the critical infrastructures in the information technology field has a high penetration rate in many countries. The significant growth of Internet can be taken as a measure for estimating the pervasiveness of other infrastructures that serve as a basis for the operation of the Internet. Electricity is probably the most significant infrastructure that affects the Internet. One can imagine that the penetration rate of electricity is far beyond the statistics that have been provided for the Internet. Electricity, Telecommunication, Water and Sewage, and Transport are among the most prevalent infrastructures that can be considered as omnipresent in many developed or even developing countries.

Reliability, as the other feature of infrastructure systems, indicates that the state of infrastructure operation should not be dependent on stochastic real world

E. Bagheri (✉) · A. A. Ghorbani
Faculty of Computer Science, University of New Brunswick,
Fredericton, NB, Canada
e-mail: e.bagheri@unb.ca

A. A. Ghorbani
e-mail: ghorbani@unb.ca

incidences. It is therefore assumed that the same services will be delivered with the same quality regardless of the location, time, or the end-customer. These services should also be economically affordable for the average citizen as they play a major role in every ones' life. It should also be convenient for new customers to join the network of the infrastructure customers and start receiving appropriate services.

The lifecycle of each infrastructure consists of various phases. In its lifecycle, an infrastructure has to deal with a diverse range of social, political, technical, environmental, security and human concerns. For this reason, after having established themselves, infrastructure systems need to expand, improve, repair, and even reconfigure to continuously adjust with the context that they are operating in. In order to compare an infrastructure with a living being, we assume that an organism is primarily born, nurtured, and raised. It then undergoes the maturity phase and finally dies. Infrastructures have a similar lifecycle with subtle differences. They are initially created for a certain purpose, and evolve as the circumstances change. Evolution of an infrastructure is pursued to allow it to survive in a very tough infrastructure market. The difference between living organisms and infrastructures is that infrastructures can experience several maturity phases. As it was previously explained, infrastructures evolve to fit new situations and conditions. Having changed their status and while evolving, they may loose their solidity and hence suffer from instability until they reach their new maturity level.

The hidden consciousness of these types of systems lies beyond their definition. Although these systems are structurally independent of any outside component, they collaborate with each other to provide their end-customers with suitable services. It is obvious from the functionality of the Internet that the performance of each infrastructure is at its simplest form dependant on the resources that are provided by other infrastructures. This type of interdependency, mainly known as commodity trade, can be considered as the most obvious type of infrastructure interdependency. There are many other types of mutual interdependencies between infrastructures that are explained in the forthcoming sections.

Due to the potentially severe repercussions of infrastructures' interdependencies, there have been many attempts to model and simulate their behavior. Most of the current research does not follow a complementary pattern. A few reasons can be namely mentioned as the explanation of such a situation. One of the most apparent reasons is that there has not been any clear attempt to define a common formal specification for infrastructures. The lack of such definition has caused a conceptual sloppiness (Dunn 2005) that disallows any collaborative research. However, there have been many attempts to model and simulate infrastructure systems behavior through pure mathematical models using differential and algebraic-differential equations. A common understanding exists amongst researchers that the exploitation of solitary mathematical models is not sufficient for modeling the complex and in many cases concealed infrastructure interdependencies (Amin 2000). These models represent only an approximation of the collective behavior of the infrastructure system components and hence lack the required details, and scalability features.

In this paper, we propose an extension to the Unified Modeling Language (UML) in order to clearly define the different aspects of an infrastructure organization and behavior. The rest of the paper is organized as follows. In Sections 2 and 3, the current state of the art in critical infrastructure modeling and simulation research is briefly reviewed. Section 4 gives an introduction to model driven development and architecture. Section 5 goes on to explain the main motivations that have led to the design of our proposed critical infrastructure reference model. The structure of a city network investigated as a case study is introduced in Section 6 that will also serve as a running example throughout the paper. Section 7 explains the features and structure of the UML-CI reference model in a detailed fashion using examples from the information gathered in the case study. The paper continues with some discussions in Section 8 and is then concluded in Section 9.

2 Critical infrastructure systems

Criticality as opposed to other objective measures such as reliability, which is the measure of the frequency and length of a disruption, is very much subjective. Different countries classify dissimilar infrastructures as critical from their own standpoint and based on their individual criteria. Criticality is related to the consequences of a disruption of an infrastructure's operation and its measurement is hence arguable (Thissen and Herder 2003), since it would be an overwhelming task to identify all of the outcomes of a failure and list all possible consequences. For this reason, modeling and specifying all aspects of a critical infrastructure has not been a great success and only high level descriptions of a critical infrastructure have been explored. In Amin (2000), Amin specifies the main characteristics of a critical infrastructure system as: 1) multi-scale, multi-component, heterogeneous and distributed in nature; 2) vulnerable to attack or disturbance that can spread instantaneously; 3) characterized by many points of interaction; 4) dramatic increase in the number of interactions with the increase in the number of participants; 5) too complex to model with conventional mathematical and control theories.

Although these types of characteristics can give us better insight into the operation and the extent of importance of such systems, they are still ineffective in creating a sound basis for the classification of infrastructure systems. We believe that any infrastructure can be regarded as critical under certain circumstances. For this reason, it would always be controversial to come up with certain exclusive attributes for critical infrastructure categorization. The first feature introduced by Amin shows that infrastructures can span a diverse geographical space and also be made up of dissimilar components. Their geographical distribution complicates the issue of security. These systems are hence very susceptible to malicious or unintentional interruptions. Terrorist attacks on the electric transmission grid in California caused the catastrophic blackout in the year 2000 (Barton et al. 2000). Although the attacks were only aimed at the transmission grid, many other infrastructures were also damaged due to infrastructure interdependencies.

The interdependencies of infrastructures are to a great degree veiled until an excessive interruption occurs in one of their systems or components. The failure will then ripple through different systems and disturbs the normal course of system action. The disturbance can even cause a system halt, which will itself have disastrous cascading or even snowball effects. The disruption level of each infrastructure in the case of a cascading effect is very much reliant on the degree of that infrastructure's interdependencies.

There have been different approaches in defining the common behavior and features of infrastructures. Infrastructures are complex adaptive systems from Rinaldi's point of view (Rinaldi et al. 2001). He believes that every infrastructure is a complicated set of components that interacts and changes as a result of a learning process. It would be certainly irrational to think that physical systems such as mechanical devices or technical tools have adaptive reactions in different situations. By adaptive, Rinaldi implies that every aspect of these systems undergo serious changes. Since human resource can be regarded as a constituent part of an infrastructure, they can learn based on their experiences and apply suitable control. The aggregate behavior of all these essential elements results in convergent behavior.

Based on the definitions provided by different authors and the mentioned characteristics of every infrastructure, we believe that infrastructures are *'complex networks of adaptive socio-technical systems'* (Bagheri and Ghorbani 2006a, b). They are complex in the sense that they are built from heterogeneous and distributed components. They can be regarded as adaptive systems due to the emergent behavior from a collective non-uniform performance of their components. The socio-technical aspect of the infrastructure stems from the fact that they are not only technical functioning systems, but other dimensions such as economics, social, environmental, security, or even political decisions can affect their behavior.

One of the most appealing facets of critical infrastructures for researchers and at the same time distressing characteristic for infrastructure owners and managers is the cascading effect of failure. The high interdependency of infrastructures is at many times neglected, since their consequences are not clearly understood. It would not be unrealistic to call infrastructures *'intertwined'* from a functionality point of view. The ignorance towards the high interdependency of infrastructures stems from the fact that the majority of failure types that cause serious cascading failure effects are amongst the *'low probability- high impact'* events (Dunn 2005). The severity of the damages that this kind of failure causes is so high that it does not form a rational trade-off with its low probability of occurrence.

2.1 Infrastructures' interdependencies

According to the classification given in Rinaldi et al. (2001) (see Table 1), interdependencies can have four distinct types. We introduce these categories in more detail in this section, since we intend to employ them in the forthcoming sections:

(1) *Physical:* The first type of interdependency between two infrastructures is based upon the physical services that each receives from the other. Two infrastructures are physically related if the input of one infrastructure is directly supplied by the output of the other. For instance, a transportation infrastructure is physically dependant on the energy industry for receiving fuel. If enough fuel is not provided, the transport infrastructure will fail to keep up with its satisfactory service level.

(2) *Cyber:* Cyber interdependency affects infrastructure systems through electronic and informational bridges. Infrastructure management has recently moved onto information based control. For this reason, a consistent operation of an infrastructure

Table 1 The different dimensions of a critical infrastructure specification

Dimensions of infrastructure interdependency					
Types of interdependency	State of operation	Coupling and response behavior	Type of failure	Infrastructure characteristics	Environment
Physical	Repair	Coupling order	Common cause	Organizational	Economic and business opportunities
Cyber	Disrupted	Coupling degree	Cascading	Operational	Public policies and legal concerns
Logical	Normal	Type of interaction	Escalating	Temporal	Security issues
Geographical				Spatial	Government decisions

is dependant on information. If two infrastructures are dependant on each other's information, they are considered to be cyber interdependent. This type of interdependency can be thought of as an extension to the previous class of interdependencies. Although information is only a subtype of the commodities that can be exchanged between infrastructures, the separation of these two types of interdependency depicts the high degree of importance of information compared with any other asset.

(3) *Geographical:* Geographical interdependency is the third type of interdependency. The side effects of an infrastructure operation can have consequences on other infrastructures in a close spatial proximity. For example, Although the water and sewage infrastructure has no clear interdependency with transport, a car colliding into a building can result in a pipe burst that causes a breakdown in the water and sewage infrastructure.

(4) *Logical:* The last type of infrastructure interdependency is a dependence that is neither physical, cyber nor geographical and is termed as logical interdependency. The fluctuations in the gold market have always caused an increase in the price of oil. Although these two assets are by no means interdependent through any of the previous three types of interdependency, they have logical relationships.

Infrastructures have dependencies on many factors other than their input, output or state of operation. The infrastructure operation environment has direct affect on the infrastructures strategic objectives. Economic and business opportunities, public policies, legal concerns, security issues and government decisions are amongst the factors that can affect the state of infrastructure operation.

The models of two different infrastructure pairing are known as the coupling and response behavior. Coupling can be characterized by three different factors: the degree of coupling, the coupling order and the type of interactions. Two infrastructures are tightly coupled, if they are highly interdependent and even a small failure in one infrastructure propagates to the other. Coupling order specifies whether two infrastructures are directly interdependent or are interdependent through some other infrastructure. This shows that infrastructure dependence and failure propagation are transitive. The interactions between infrastructures can further be classified as linear or complex. The types of failure that may occur in an infrastructure may either be common cause, cascading or even escalating. Common cause failures are simultaneous failures occurring in different infrastructures that originate from a common source. Failures can be cascading that show rippling fault propagation between different infrastructures. Escalating failures are synergetic, but independent failures that cause increased severity of a single problem. The final dimension of infrastructure operation focuses on the state of each infrastructure. Each infrastructure operation status can range from optimal functionality to total failure.

3 Critical infrastructure modeling: Related work

The attempts to model and simulate the behavior of critical infrastructure systems can be classified into two main groups. The first group of research aims at modeling the infrastructure behavior through pure mathematical models using differential and algebraic-differential equations. Some of these models have been based upon Leontief's I-O model (Leontief 1951, 1966). This model is capable of describing the interconnectedness degree of different economic sectors. The formulation of this I-O model is as follows:

$$x = Ax + c \Longleftrightarrow \forall i \left\{ x_i = \sum_{j=1}^{n} a_{ij} x_j + c_j \right\}$$

In this formulation x_i is the total operation output of Sector i. The coefficient a_{ij}, specifies the ratio of Sector i's input into Sector j with regard to the total requirements of Sector j. c_j refers to the remaining demand for Sector i that has not been expressed through sector to sector interconnections. This can be the portion of direct resource requests from the end users. Haimes et al. (2005a, b) create an inoperability input-output model for the interdependent infrastructure sectors that is based upon the Leontief theory. The model assumes that all the different types of infrastructure interdependencies can be modeled through financial interactions. In this approach, the total consumption percentage of each sector from the other sectors' productions shows the degree of their interdependency. Although commodity based sector to sector relationships that are employed in this form can reveal and represent physical and to some extent logical hidden interdependencies, they still lack the ability to represent cyber and geographical interdependencies. The other challenging aspect of such approach is the high complexity involved in creating a complete model. It can be anticipated that as the number of involved sectors increases and more minutiae's are added, the complexity of the models such as the one introduced in Haimes et al. (2005b) will dramatically rise and hence complicates the modeling and simulation process.

Infrastructures consist of many heterogeneous, but interrelated systems and components. This makes their high level functionality very sophisticated. A bottom-up design can alleviate the modeling process. In a bottom-up modeling process the building blocks of a system are modeled independently, but the aggregation of all these simple models provides an understanding of the collective operation of the system. Multiagent systems are bottom-up design models that can be used in critical infrastructure modeling and simulation. One of the very first attempts to incorporate the multiagent based modeling perspective into infrastructure design resulted in the design of ASPEN by the Sandia national laboratory (Basu et al. 1998). ASPEN is a parallel agent based Monte-Carlo simulation of the US economy. The US macroeconomics is modeled and simulated based on the aggregate microeconomics of the role players. The agents incorporated into the design of ASPEN are industries, banks, households, retail markets, and the government. Each agent aims to maximize its profit through an n-person zero-sum game. Agents can learn how to behave based on a learning classifier system. ASPEN-EE and N-ABLE were later developed based on the ASPEN framework to extend its features and make it more powerful (Barton et al. 2000; Schoenwald et al. 2004). A more recent approach to the simulation of the interdependencies of critical infrastructure systems has resulted in an agent based simulation tool called CISIA (Panzieri et al. 2005). CISIA aims to evaluate the short term effects of one or more faults on an infrastructure behavior through what-if analysis and by removing critical elements of the infrastructure.

From a process systems' engineering perspective, Thissen and Herder (2003) have devised a three layered structure for infrastructures internal representation. This three layered approach consists of *Physical, Operation and Management, and Products and Services* layers. Other attempts have been made to contribute to modeling and simulation of critical infrastructure systems and their interdependencies (Allenby 2004; Barton and Stamber 2000; Brown et al. 2004; Herder et al. 2000; Nozick et al. 2005; Rinaldi 2004). The main gap that can be seen in these models is that there is no standard definition for the organization of an infrastructure system. Different simulation or modeling tools have been created that provide a suitable basis for a certain task and lack sufficient capabilities to be mapped to other purposes. We attempt to form a solid structure of the internal and external organization of an infrastructure. The proposed reference model is inclusive enough for most kinds of infrastructures, but it is also carefully abstracted to exclude any domain specific features.

4 Model driven development

Models serve as the means to portray physical, abstract, cyber, or hypothetical beings. Their main feature is context dependent-ness. Models are used to abstract and classify realities into relevant groups. Abstraction removes redundant information while classification groups the modified abstract concepts into relevant groups. The formation of a collection of related models is the main property of Model Driven Development/Architecture (MDD/A) (Atkinson and Kuhne 2003; Selic 2003). UML, as a widely accepted standard for modeling and designing different types of systems, sits at the focal point in MDD/A. MDD/A models can be formally expressed using any kind of modeling language; however, UML has been the dominant choice in both academy and industry. The use of models allows us to create high level descriptions of a system. Although the employment of models seems to be desirable, raising the abstraction level even higher can allow modelers to create structure, semantics, and constraints for a family of models.

The models employed at higher layers are called metamodels. Having different abstraction layers subtly implies that our models can be structurally organized. The Meta Object Facility (MOF) provided by the Object Management Group offers even more abstraction. MOF provides a set of modeling constructs to describe and work with metamodels (Fuentes-Fernandez and Vallecillo-Moreno 2004). The levels of abstraction that range from the real world instances to the high level MOF classes form a four-tiered metamodeling architecture. At the highest level lay the very fine grained models, or the actual instances of the models. This layer is called M0. The next level of abstraction constitutes the application level instances of a metamodel. The realized classes in a UML model are placed in the M1 level. The M2 level constitutes the meta-information that capture the high level abstractions of a domain specific modeling language. The UML itself resides in M2. M3 is the most fundamental metamodeling layer. It describes the features that a standard metamodel in the other layers can have. An example four layered metamodeling architecture is shown in Fig. 1.

The devised models in MDD/A can have two forms (Uml infrastructure specification 2007). The first form of models are independent of the operating platform. These types of models are called Platform Independent Models (PIM). PIMs are abstract models that do not directly map to a specific environment. In order to instantiate the PIMs, Platform Specific Models (PSM) should be created. PSM specification relies on the definition of their particular target platform. For instance, the UML-CI reference model introduced in this paper is platform independent; however, in order for UML-CI models to be able to create a real simulation, an agent based tool can be selected. The PIM should then be transformed into a PSM suitable for that target platform for it to be executable.

5 Motivations and incentives

The formation of a collection of related models is one of the major tasks of many large scale industrial projects. This collection provides the base understanding of the intended future as well as the current state of the system being studied. From the wealth of technologies that can assist the process of modeling, the UML (Fowler 2004) is a widely accepted standard and has been the dominant choice over the past decade for several reasons:

- It offers a graphical notation, which is a result of many years of industrial experience as well as academic research and a fruitful mutual collaboration.
- The standard graphical notation eases the understanding and communication between both the team members and the stakeholders.
- The related standards to this notation are maintained by the object management group as open standards.
- The notation has been widely adopted in industry and extensively taught in many academic institutes throughout the world that alleviates the common problem of insufficient human resources.
- UML is extensively supported by a range of open source or proprietary CASE tools. The strong tool support behind UML that provides comprehensive

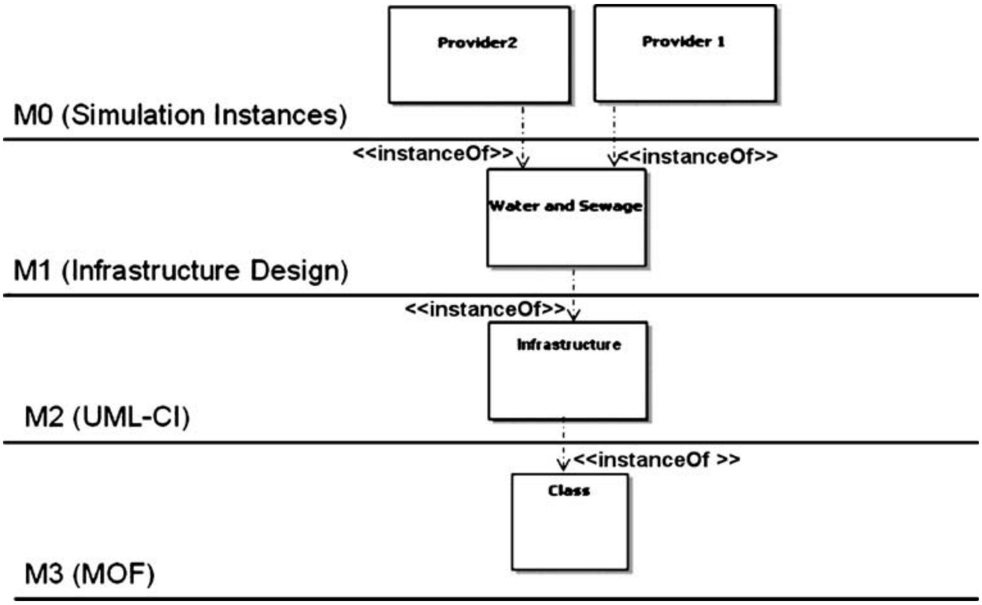

Fig. 1 A four-layered metamodeling architecture extending UML for modeling the water and sewage infrastructure

capabilities for model manipulation and maintenance makes it even more appropriate for large-scale projects.

- It provides a standard mechanism (profiles) for defining domain specific metamodels through extensions to its base models.

For these reasons, we have chosen to extend primary UML functionality through profiles. A 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 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 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 1 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 systems are multifold. In this section, 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 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:* Based on the same framework, different modeling teams 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 because it benefits from a graphical notation. The models created based on this reference model can be further re-used. Similar infrastructures can also be modeled through the refinement of an existing model without the need for starting from scratch.

6 A real-world case study

To demonstrate how our reference model actually addresses the process of modeling and profiling critical infrastructure systems, and how it maps to real world systems, we will employ a subset of the network of the city of Fredericton, which is one of the first North American cities to provide its citizens with free ubiquitous wireless access to the Internet (http://www.fred-ezone.ca/). This network consists of five main sub-networks namely: wireless community network, community network core, city network core, city hall segment, and the police network. We have chosen to demonstrate how two sub-networks of the city hall segment and the city network core can be modeled using our proposed reference model. It should be noted that some minutiae's of the model have been removed in order to conform to a non-disclosure agreement. The sub-networks used as a case study in this paper are shown in Fig. 2.

In the city network core also known as the server room, various public servers are connected to a DMZ VLAN, which is accessible to the public from the Internet. The server room also includes 42 internal servers connected to two 3Com Superstack II 3300 switches. Each switch has 24 Ethernet 100 Mb ports. The connection between fiber optic and switches is done through fiber optic transceivers. The city hall segment subnetwork is connected to the server room through a 1 Gb Multi Mode fiber. The central switch of the City Hall network is a 3Com SuperStack Fiber switch 9300 with 12 fiber ports connecting the networks situated on different levels of the City Hall building. There are about 400 users in the City Hall network that constantly use the services provided by the 42 servers located in the server room. Thus, any problem with the 3Com SuperStack Fiber switch 9300 will disconnect them from the whole network.

The network topology described in this section will be used as a reference infrastructure system throughout the rest of the paper and will aid in showing how different elements of the reference model will be employed for modeling infrastructure systems.

7 High-level critical infrastructure metamodels

The proposed critical infrastructure reference model consists of five main metamodels, each of which addresses a different issue. Each of the metaclasses within a metamodel is relevant to the concept that rules the metamodel. These five metamodels are briefly explained in the following lines:

(1) *Ownership and Management Metamodel:* The elements within this metamodel 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 Metamodel:* This metamodel provides the means for specifying the make up of an infrastructure system. It includes three major metaclasses, namely infrastructure, system, and task.

(3) *Resource Metamodel:* 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 metamodel provides the metaclasses for defining different types of

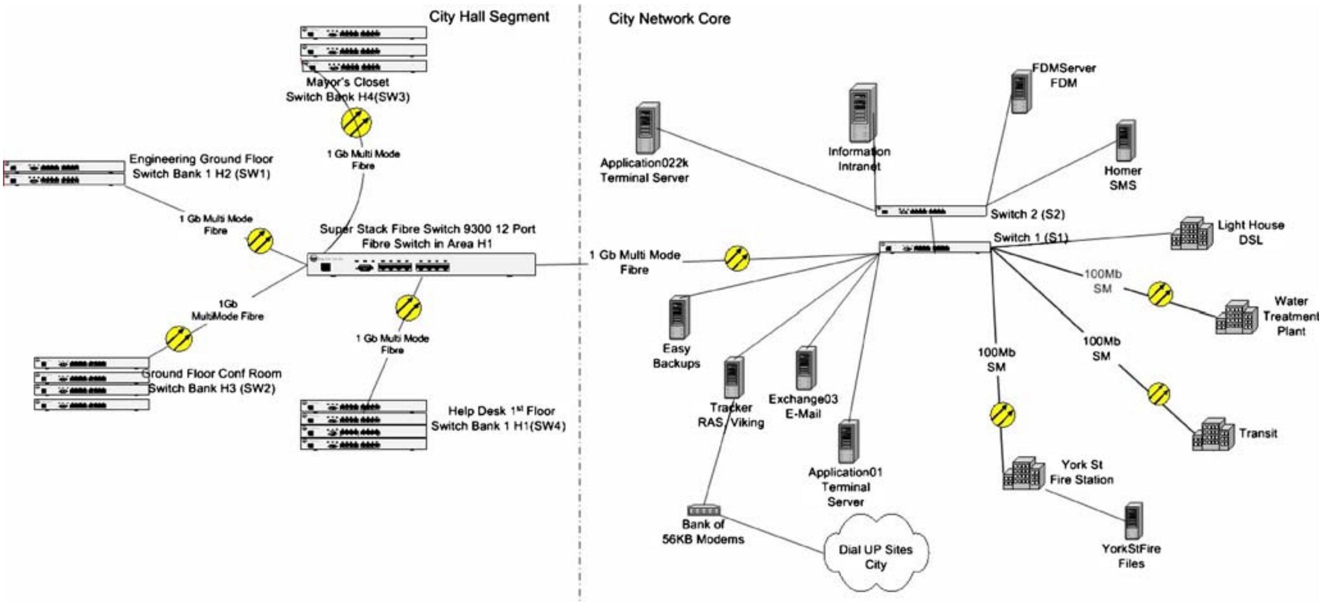

Fig. 2 The sample infrastructure topology for the network employed in the case study

resources available within a critical infrastructure environment.

(4) *Threat, Risk, Vulnerability (TRV) Metamodel:* One of the main reasons for modeling infrastructure systems is to identify the hazards that threaten its operation. The TRV metamodel, provides various metaclasses so that these hazards can be categorized and their causes, consequences and possible mitigation strategies be clearly specified.

(5) *Relationship Metamodel:* The relationship metamodel provides many different metaclasses (derived from KernelAssociation) for connecting and joining the concepts that have been identified in the previous metamodels. For example, although various hazards that threaten the operation of an infrastructure can be specified in the TRV metamodel, they are not specifically attached to the system, or task that they actually threatening. To address this concern, the relationship metamodel provides suitable metaclasses, so that all of the created metaclasses in the previous metamodels can be integrated into one unique representation.

In the rest of this section, we will discuss the minutiae of each high-level metamodel in more detail. Each subsection introduces and explains the metamodels in three parts, namely: Structure, Description, and Constraints. Structure depicts the overall form of the metamodel, description describes the role of the metamodel through an example usage, and the constraints define the restrictions that should be observed while instantiating that specific metamodel.

The models introduced in the structure and description subsections of the reference model can only show the abstract syntax of the overall design. By this we mean that if a modeler makes any mistakes in connecting the high level models together, there is no correctness checking mechanism to alert him/her of his/her error; therefore, detailed composition restrictions have to be devised and applied as audits to the abstract syntax to enable model correctness and consistency checking (which is based on semantics). Object constraint language is a language that can express additional and necessary information about the models and other artifacts used in the metamodeling procedure, and should be used in conjunction with graphical models (Warmer and Kleppe 2003).

These rules restrict the open association of the different metaclasses with each other, and therefore, result in semantically correct model instances. To see how these well-formedness rules control the composition of different models, consider the case where a modeler is trying to attach two instances of the Government territory metaclass to a single government class. As will be shown in future sections, based on ***R0***, this is not permitted and therefore, the modeler will receive a message stating that such composition is not allowed. In a similar way, if a modeler has defined a task that has two operational requirements, but no manufactured products, he/she will be notified that each task should at least have one input and one output according to ***R8***.

7.1 Ownership and management metamodel

7.1.1 Structure

The ownership and management metamodel addresses three major concerns: 1) Infrastructure supervision and possession; 2) policy making and regulation; and, 3) safety and protection (SP). The structure of the available metaclasses in this metamodel is depicted in Fig. 3.

7.1.2 Description

Infrastructure supervision and possession provides the means for identifying infrastructure stakeholders, their

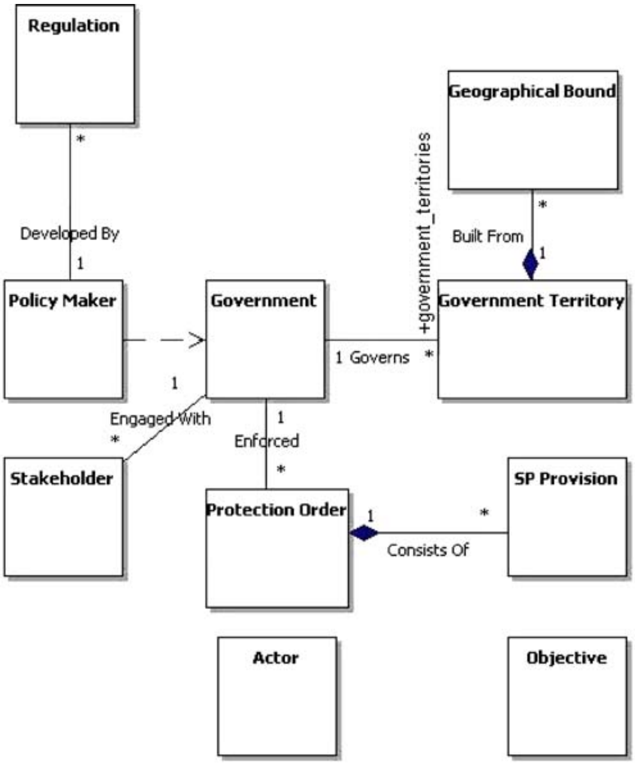

Fig. 3 The structure describing the ownership and management metamodel

objectives (e.g. high annual profit, high customer satisfaction, etc.) and their connection with various governments. These concepts are specified with the stakeholder, objective, and government metaclasses, respectively. The operations of the current infrastructure systems seem to be of international (or at least multi-national) scale. Therefore, each infrastructure may interact with several different countries (we assume that each country has a unique government) depending on its geographical presence. For example, a telecommunication company that is providing cell phone services within North America should comply with the regulations and policies of both Canada and USA depending on the specific geographical location that it is providing its services. There are also cases (e.g. where conflicting regulations exist) that the stakeholders of an infrastructure should make proper decisions about their operations, so that they can conform to the regulations of both countries.

The regulations of each government are made through respective policy making authorities that can be defined through the policy maker metaclass. Each policy maker can define and regulate a set of policies that should be observed by an infrastructure throughout its operation. For instance, in our case study the city network should conform at all times to the Information Technology Support Sub-Committee (ITSS) procedures enforced directly from the provincial government; therefore, the ITSS is a policy maker in our case study. The ITSS has developed many regulations among which we have shown three in the diagram: use of electronic services, customer inquiry and feedback, and employee discipline. In the use of electronic services regulation, the sub-committee has stated clearly that: ‘access to e-mail is provided to users to assist them to perform their work and their use of e-mail must not jeopardize operation of the system or the reputation and/or integrity of the city’. The set of such policies, laws, and regulations enforced by some governmental authority are defined through the instantiation of the regulation and policy maker metaclasses in UML-CI.

To specify each government’s borders of jurisdiction, we have defined the geographical territory and geographical bound metaclasses. Each government has a geographical territory in which his policies and regulations should be followed. The geographical territory of each government is defined through a collection of geographical bounds. Geographical bounds specify the locations that are within the authority of a government. The granularity of the geographical bound selection is very much dependent on the decisions made by the modelers. For the case study performed presented in this paper, we have decided to specify the locations were services are either provided or consumed as geographical bounds. For this reason, eight geographical bounds have been specified. Referring back to Fig. 2, we can see for example, that the Fire Station is a remote user of the services provided by the network server room, and so it has been selected as one of the geographical bound locations in the instantiated UML-CI model. In any case, the collection of the instantiated geographical bound metaclasses would form the infrastructure geographical territory.

The SP sub-metamodel offers two main metaclasses, protection order, and SP provision, for defining the rules and regulations that monitor the operation of an infrastructure for security and safety purposes. Each protection order that consists of various SP provisions can be both devised and enforced by the government or the infrastructure managerial authorities. In the case study, the Core Operating Procedure are the set of policies that need to be enforced for the safety of the city network. Core Operating Procedure is a Protection Order, that consists of further SP Provsions such as ‘all firewalls must be configured in proxy mode’, ‘discard all broadcast messages’, ‘no remote management of devices’, ‘network activity logs must be kept for at least one year’. Figure 4 depicts an instantiated sample of the ownership and management metamodel for our running example.

7.1.3 Constraints

The relevant constraints of this metamodel are described below:

R0: Each *Government* only *Governs* one *Government Territory*.
Context Government
inv: **self**. *government_territories*→ size() = 1
R1: Every *Regulation* is only *Developed by* one *Policy Maker*.

7.2 Structure and organization metamodel

7.2.1 Structure

The technical, constructional, and organizational aspects of an infrastructure are tackled in the structure and organization metamodel of the proposed reference model. This metamodel consists of three main metaclasses namely: infrastructure, system, and task. The organization of the metaclasses of this metamodel is shown in Fig. 5.

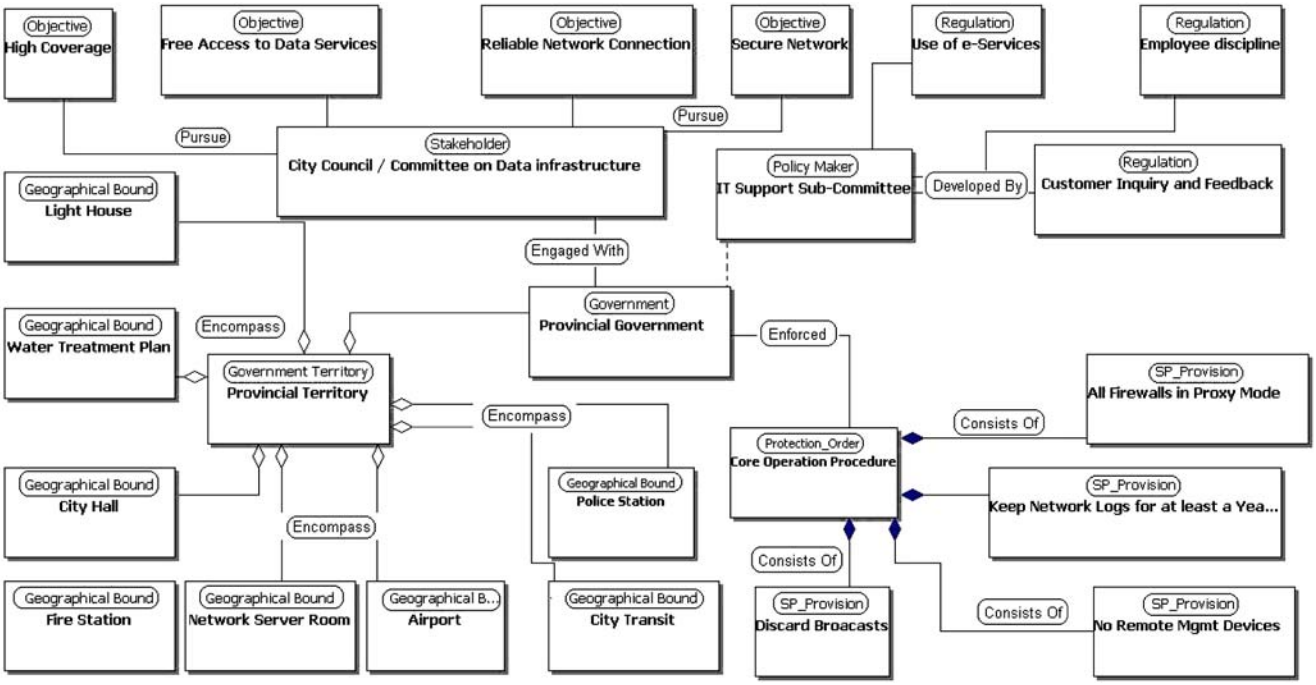

Fig. 4 An instantiated ownership and management metamodel for the introduced network

7.2.2 Description

The metaclasses in this metamodel provide the means to identify the systems that are available in an infrastructure. The operations related to each system can be modeled through the task metaclass. The systems of an infrastructure can be either of technical or organizational types. By this we mean that various types of systems can be modeled through the employment of this structure. For example, the business perspective of an infrastructure also consists of systems and tasks; however, these systems and tasks are more of a business process nature. Returning to the case study, the systems within the city network are the set of switch banks, bank of modems, application servers, fibre switches, and various other switches. Similar to the identification of the geographical bounds, systems of an infrastructure can themselves consist of other systems; therefore, they can be refined to an acceptable granularity. For example, a bank of modems can be refined and specified in a more detailed manner through its constituting modems and their types.

The operations performed by any of the identified systems of an infrastructure are modeled by the task metaclass. Each task can be either supportive, or consumptive/productive. A supportive task maintains the appropriate needs of other sibling tasks (the tasks that are performed by the same system that operates the supportive task); therefore, these tasks have no actual interaction with the outside world. Consumptive/productive tasks act like services. The collabora-

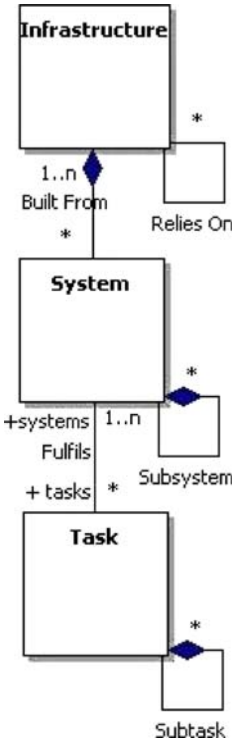

Fig. 5 The representation describing the structure and organization metamodel

tion of such tasks produces valuable outputs. Based on the metaclasses introduced in this section, our city network includes seventeen systems from a high level perspective, which are four switch banks (SW1 to SW4), three switches (a superstack fibre switch 9300, and two switches (S1&2)), a bank of modems and nine application servers.

A superstack fibre switch is a computer storage device that allows the creation of a fibre channel fabric. This fabric is a network of fibre channel devices, which provides services such as many-to-many communication, device name lookup, security, and redundancy. Therefore, the activities that a fibre switch should perform are many-to-many communication, device name lookup, provide security for the network, and create redundancy. These responsibilities of a fibre switch are classified under its tasks in the infrastructure model. The same procedure applies to other systems such as the bank of modems that is responsible for providing the possibility of dial-up connection to the network. Figure 6 shows a high level instantiation of a structure and organization metamodel.

The variety of systems within different infrastructure systems is so immense that no complete list of them can be enumerated; therefore, the UML-CI reference model does not contain any domain dependent metaclasses, but the modelers can create instances of widely used infrastructure systems (or tasks) and extend UML-CI by adding more domain specific components. This has been performed in our case study (See Fig. 6) where the system metaclass has been extended to create switch, bank of modems, application server, and switch bank metaclasses. This feature adds reusability to the currently available metamodels, in a

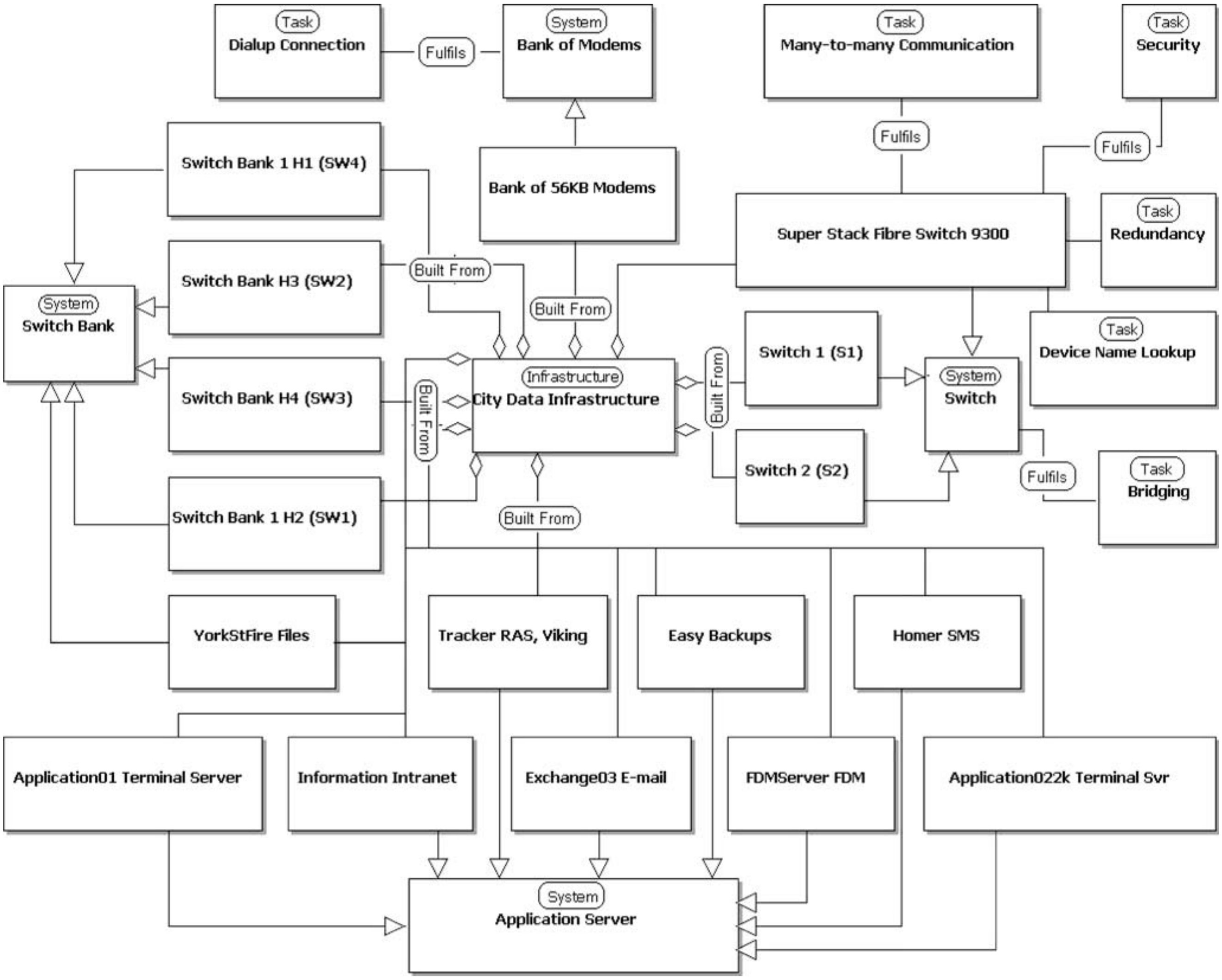

Fig. 6 A partial view of the structure and organization metamodel for the introduced network

sense that a metaclass that has been currently developed can be later extended for other purposes (or even parts of it can be used in other projects). Other than reusability, system instances that have been created based on an extended system metaclass can inherit the list of the tasks that are connected to their parent. For example, in Fig. 6 the Bank of 56KB Modems automatically inherits the tasks of the Bank of Modems metaclass, which is a dial-up connection. In this way, the modelers do not need to create similar classes from scratch each time they encounter a new system of the types that have already been modeled. This can also make models more readable and less cluttered.

7.2.3 Constraints

The relevant constraints of this metamodel are described below:

R2: Two different *Systems* cannot be the parent of the same *Task*.
Context Task
inv: **self**. *systems*→ size() = 1
R3: Each *System* is only used in the construction of one *Infrastructure*.
R4: Two *Infrastructures* only rely on each other if at least one of their *systems* has the *Depends on* relationship with a *system* in the other.

7.3 Resource metamodel

7.3.1 Structure

The resource metamodel is responsible for identifying the core material (either it be physical, non-physical or cyber) that are required for the correct operation of a system task. For example, fuel is a resource required for the operation of a car; whereas financial information are the necessary resources in an MIS system. The makeup of this metamodel is illustrated in Fig. 7.

7.3.2 Description

In the UML-CI reference model, resources are modeled through the asset metaclass. An asset can be physical, non-physical, or cyber (information). It is important to note that infrastructure services are organizational assets; therefore, they have been integrated into the resource metamodel via the non-physical asset metaclass. A physical asset can be itself generalized to being either transformable or non-transformable. A transformable asset is a resource that can be changed in the process of a task. Oil is one example of such assets that can be refined and transformed into gasoline or other products. On the other hand, non-transformable assets cannot be transformed into other types of assets. Clear examples of such assets are the machinery used in a critical infrastructure system.

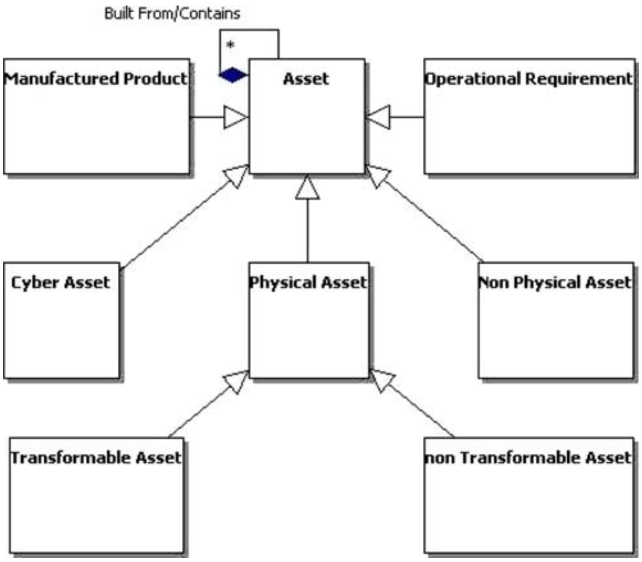

Fig. 7 The structure describing the resource metamodel

From the operation perspective, resources are inputs or outputs of a process performed in an infrastructure system/task. From this point of view, assets are either a requirement for or a production of a process. These concepts have been incorporated into the metamodel through the operational requirement, and manufactured product metaclasses.

Table 2 shows a subset of the different types of assets of the city network. Among these assets, the communication links in the network (i.e. server connection 100 MB FD, and 1 GB MM Fibre) are physical assets. Since these assets cannot be transformed into any other form and are used in their original form, they are classified as non-transformable physical assets. From the vast range of cyber assets in the city network, the routing information passed between the superstack fibre switch and switch S1, and also the email messages received on the Exchange03 email server have been shown as examples of such assets in Table 2. The city network also provides its users with various services such as backing up their data on the Easy backups server, and sending/receiving emails through the Exchange03 email server. These sort of services are the city network non-physical assets. Furthermore in Fig. 9, the many-to-many communication task is exploiting the 1 GB MM Fibre asset as its operational requirement (medium for performing the task), and produces the

Table 2 A short list of assets derived from the city network case study

Asset description	Asset type
Sever connection 100 MB FD	Non-transformable physical
1 GB MM Fibre	Non-transformable physical
Routing information from the superstack fibre switch to S1	Cyber
Email messages on Exachange03	Cyber
Data backup on easy backups server	Non-physical
Sending and receiving emails	Non-physical

Routing information and Messages as its manufactured products (results obtained from performing this task). These two assets, which form the inputs and outputs of the many-to-many communication task, are physical non-transformable, and cyber assets, respectively.

7.3.3 Constraints

There are no specific constraints in this metamodel. General constraints concerning the connection of the metaclasses of this metamodel with that of the other metamodels are given in the Relationship metamodel constraints.

7.4 Threat, risk, vulnerability (TRV) metamodel

7.4.1 Structure

One of the major motivations for the study of infrastructure behavior and organization is to identify the hazards that are threatening their proper operation. This is because any of these hazards may result in the transition of the infrastructure into an undesirable state, the worst of which is non-functional. We will not go into the details of hazard identification and management, since it is out of the scope of this paper and has been already extensively studied in the literature (Kletz 1999), but at the same time understand the necessity to provide suitable basis for profiling the results of these efforts; thus the TRV metamodel is incorporated into UML-CI to address this issue. The TRV model structure that we have incorporated into UML-CI is shown in Fig. 8. The structure addresses the causes, consequences and mitigation strategies related to a hazard threatening an infrastructure system.

7.4.2 Description

Any hazard that poses danger to the operations of an infrastructure system is either a risk, vulnerability, or threat. Each of these hazards has its own causes and consequences. It is of extreme importance that the causes of any hazard be identified, so that suitable preventive mechanisms can be devised. A study on the classification of infrastructure hazard causes has revealed that the origins of any hazard can be one of the following sources (Narich 2005):

- Natural disaster (e.g. flood, earthquake, etc);
- Human error (e.g. wrong data entry, careless inspection etc);
- Machine fault (e.g. imprecise measurement, component breakdown, etc);
- Malicious attacks (e.g. terrorist attempts).

The consequences of a hazard can also be classified into two categories. They can either be direct or indirect. A direct consequence of a hazard is its visible effects on the outside environment, whereas the indirect consequences are those veiled impacts that this hazard poses. Based on the occurrence probability of the causes of a hazard and the severity of its consequences, appropriate mitigation strategies can be devised. These mitigation strategies have three attributes: their implementation cost, time, and effectiveness. An

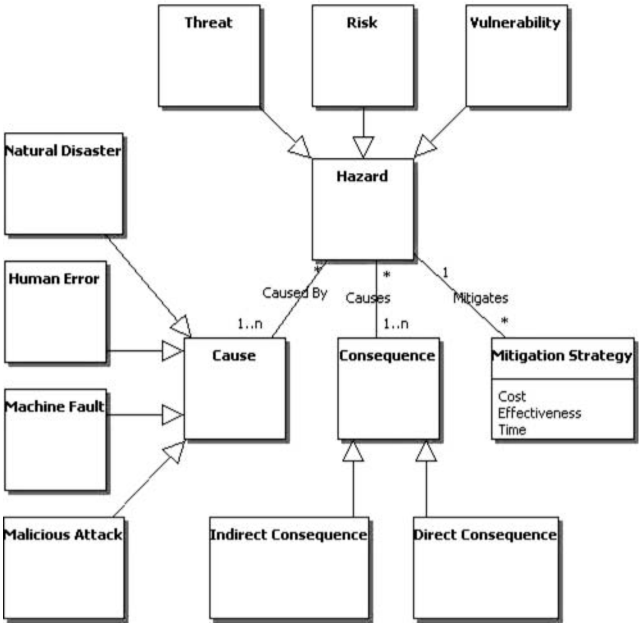

Fig. 8 The structure describing the threat, risk, vulnerability (TRV) metamodel

infrastructure manager can select the most suitable mitigation strategy according to these attributes.

To support the profiling of hazards that threaten an infrastructure system, our proposed reference model provides various metaclasses, namely: hazard (threat, vulnerability, risk), cause (natural disaster, human error, machine fault, malicious attack), consequence (direct consequence, indirect consequence), and mitigation strategy. Each hazard metaclass has a set of causes, consequences and mitigation strategy metaclasses attached to it that allows the modeling and profiling of the before-mentioned processes.

As was mentioned before, there are near 400 users in the city hall segment of the city network that are providing citizens with governmental services. The connection of the city hall segment with the outside world is only maintained through a single point of access, which is the superstack fiber switch 9300. It is easily conceivable that the collapse of this switch is a definite risk for the government. The failure of this switch may be due to various causes such as power outage, sudden breakdown of an internal circuit of the switch or even a mal-configuration of the switch. If the switch breaks down the direct consequence of this would be the disconnect of the city hall segment of the network from the city network, but the indirect consequences are much more severe. For example, until the switch is down the governmental services provided through the city hall (by near 400 government officials) would not be available. The mitigation plans that are currently been developed in the city hall segment of the city network are deploying a UPS, installing a second superstack fibre switch, and also planning on performing regular inspections of the superstack fibre switch. The operation of several tasks are also affected by this risk. These tasks are shown using the 'exposes to' metalink, which are many-to-many communication, device name lookup, redundancy, and security in this case. Figure 9 depicts an instance of the instantiated TRV metamodel for the case study.

7.4.3 Constraints

There are no specific constraints particularly related to the TRV metamodel. General constraints concerning the connection of the metaclasses of this metamodel with that of the other metamodels are given in the Relationship metamodel constraints.

7.5 Relationship metamodel

7.5.1 Structure

The metamodels introduced thus far provide the foundations for modeling many of the critical infrastructure's organizational aspects and behavior. The main

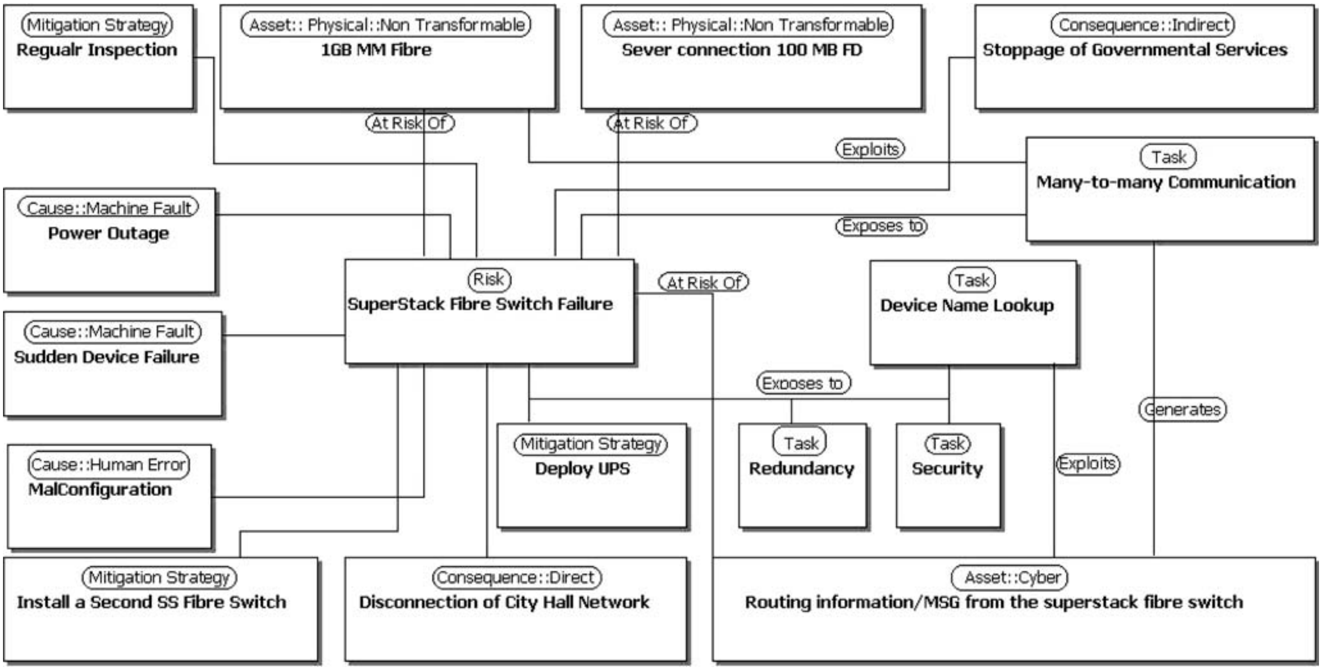

Fig. 9 The structure of the 'SuperStack Fibre Switch Failure' risk

point that remains before the completion of an infrastructure reference model is that appropriate means should be available so that these four different high-level metamodels can get integrated into a single representation. For this purpose, various association metaclasses (we call them metalinks in this paper) have been incorporated into UML-CI (Table 3). The role of the relationship metamodel is to provide the suitable metalinks so that models developed using the previous four metamodels can be unified into a single representation. The complete list of these metalinks along with their type, source and destination metaclasses (from other metamodels) can be found in Table 4. The description of all metaclasses in all other four metamodels are also given in Table 3. These two tables also explain the details of the stereotypes and structure of the UML-CI profile.

7.5.2 Description

To more specifically explain the role of this metamodel, the integration of the ownership and management metamodel and the structure and organization metamodel is studied in this section. The integration of the rest of the metamodels is quite similar.

There are various integration points within these two metamodels. First, an infrastructure needs to have a specific connection to its stakeholders. To make this connection explicit, the administers metalink should be employed. Therefore, an instance of an infrastructure should be connected to all of its stakeholders through the administers link. Indirectly, the connection between the infrastructure instance and its stakeholders specifies the infrastructure's borders of operation. This is because we have previously defined the connection

Table 3 UML-CI profile stereotype specification (Shared base class: Core::Class)

Stereotype	Semantics
Stakeholder	The stakeholders of an infrastructure
Government	The highest managing power controlling the infrastructure
Government_Territory	The geographical borders of the government
Infrastructure	An instance of an infrastructure
Actors	Any sort of role player
Policy_Makers	The decision making parties that effect the infrastructure operation
System	The constituent parts of an infrastructure
Regulation	The rules and policies on an infrastructure
Task	The actual operations performed by each system
Manufactured_Product	Outputs of each task
Operational_Requirement	Prerequisites for the operation of a task
Operational_Bound	Geographical operational points
Asset	An abstract class resembling the infrastructure resources
Non_Physical_Asset	Resources like a help desk
Physical asset	Physical resources
Cyber asset	Resources such as information
Transformable asset	Assets that can change their form e.g. oil
Non_Transformable_Asset	Assets like machinery
Risk	Risks facing an asset or task
Threat	Threats, e.g. terrorist attacks
Vulnerability	Incidents like a dam breakdown
Hazard	Abstract of TRV
Cause	Roots of TRV
Consequence	Outcomes of a TRV
Direct_Consequence	The direct outcomes of a hazardous incidence
Indirect_Consequence	The indirect results of a TRV
Malicious_Attack	Acts like terrorist attacks
Human_Error	Errors due to human negligence
Natural_Disaster	Flood, earthquake, etc.
Machine_Fault	Unforeseen faults in machinery
Protection_Order	Set of rules and regulations for infrastructure safety
SP_Provision	Instances of policies for protecting systems, assets, and tasks
Objective	Infrastructure goals and operation purpose
Mitigation strategy	Actions taken to overcome hazards

Table 4 UML-CI profile stereotype specification - metalink

Stereotype	Type	Source metaclass	Destination metaclass
Developed_By	Core::Association	Regulation	Policy_Maker
Fulfils	Core::Association	System	Task
Owns	Core::Association	System	Asset
Administers	Core::Association	Stakeholder	Infrastructure
Generates	Core::Association	Task	Manufactured_ Product
Governs	Core::Association	Government	Government_ Territory
Controlled_By	Core::Association	Task	Regulation
Performs_At	Core::Association	Task	Geographical_Bound
Accessible_At	Core::Association	Asset	Geographical_Bound
Exploits	Core::Association	Task	Operational_ Requirement
Encompass	Core::Aggregation	Government_Territory	Geographical_Bound
Built_From	Core::Aggregation	Infrastructure	System
Restricted By	Core::Association	Asset	Regulation
Engaged_With	Core::Association	Stakeholder	Government
Relies_On	Core::Association	Infrastructure	Infrastructure
Depends_On	Core::Association	System	System
Causes	Core::Association	Hazard	Consequence
Caused_By	Core::Association	Cause	Hazard
At_Risk_Of	Core::Association	Asset	Hazard
Exposes_To	Core::Association	Task	Hazard
Enforced	Core::Association	Government	Protection_Order
Pursues	Core::Association	Infrastructure/System	Objective
Consists_Of	Core::Aggregation	Protection_Order	SP_Provision
Protects	Core::Association	SP_Provision	System/Task/Asset
SubSystem	Core::Aggregation	System	System
Depends_On	Core::Association	System	System
SubTask	Core::Aggregation	Task	Task
Built_From/Contains	Core::Aggregation	Asset	Asset
Mitigates	Core::Association	Hazard	Mitigation_Strategy

between the stakeholders and various governments (each of which defines their own borders of jurisdiction through the geographical territory metaclass). On the other hand, the protection orders defined in the ownership and management metamodel should be connected to appropriate infrastructures and their related systems and task. The integration is required since not all protection orders and SP provisions are required for all infrastructure systems and tasks.

To show that an infrastructure needs to conform to a specific protection order, the enforced metalink can be employed. This high level link would allow the modelers to attach lower level SP provisions to the infrastructure systems and task through the employment of the protects metalink. The last integration point of these two metamodels is the integration of the regulations and the tasks of an infrastructure system.

Any task in an infrastructure system should obey the set of rules, regulations and policies that have been enforced by the government policy making authorities. To model this, the controlled by metalink should be utilized to connect the relevant regulations with the appropriate tasks. In our case study, the city data infrastructure needs to be integrated with its local board of directors that are city council/committee on data infrastructure, which is feasible through the use of the administers link. Figure 10 shows how these two models in the city network case study integrate into one single model. It can also be seen in this figure, that the Core Operation Procedure which is a Protection Order in the ownership and management metamodel has been attached to the City Data Infrastructure in the structure and organization metamodel through the Enforced link. This implies that the City Data Infrastructure should conform to the policies and regulations enforced by the Core Operation Procedure. In another case, the Employee Regulation class (from the ownership and management metamodel) has been attached to Dial-up Connection class (from the structure and organization metamodel) using the Controlled By link, which implies that there are certain regulations on how the employees can dial-up to the city network. The other metamodels can similarly join together and form a unique model.

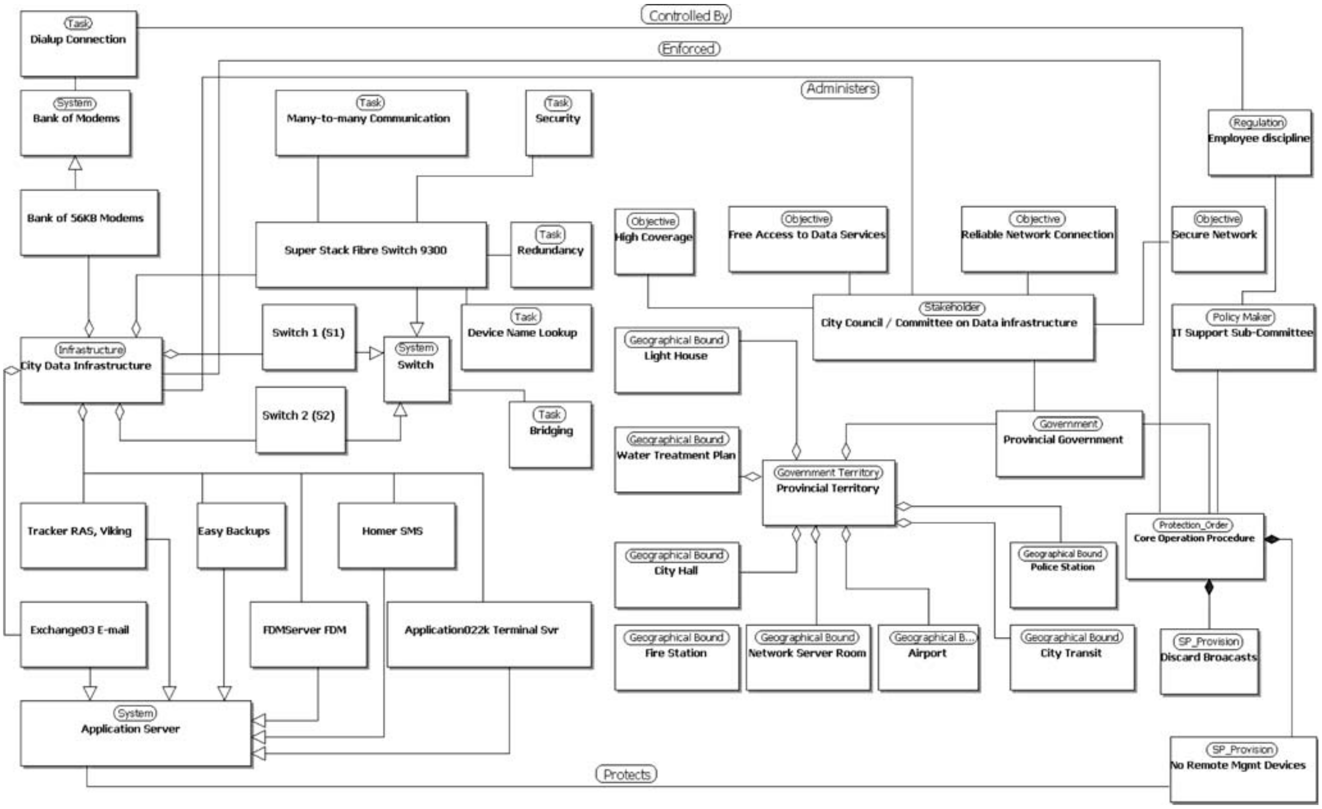

Fig. 10 The integration of two different instances of metamodels (the ownership and management metamodel and the structure and organization metamodel) into a unique representation

7.5.3 Constraints

The relevant constraints of this metamodel are described below:

R5: Every *Task* should at least have one input as *Operational Requirement* and one output as *Manufactured Product*.

Context Task

inv: **self**. *operational_requirements*→ size() > 0

and

self. *manufactured_products*→ size() > 0.

R6: If a certain *Asset* is *At Risk Of* a *Hazard*, the *Tasks* that rely on that *Asset* for input or output would also be *Exposed to* that *Hazard*.

R7: The defined association types should only be applied to their respective source and destination metaclasses. For example the Built From metalink can only be applied from the *Infrastructure* metaclass to the *System* metaclass.

R8: Each *Asset* should belong to a *System*'s parent *Infrastructure* in order to be *Exploited by* that *System's Tasks*.

R9: Each *Task* can only *Exploit Assets* that is *Accessible at* its *Geographical Bound*.

R10: Each *Task* can only *Exploit* or *Generate Assets* that are Owned by its parent *System*.

R11: If an *Asset* is *Restricted by* a *Regulation*, the corresponding *Task* should also be Controlled.

R12: All Association metaclasses are directed.

R13: The *Geographical Bounds* of a *Task* cannot go beyond the *Geographical Bounds* of its related *Governments*.

8 Discussions

In their seminal paper (Rinaldi et al. 2001); Rinaldi et al propose six dimensions that can be used to define the characteristics of infrastructure systems. These dimensions consist of various features for specifying infrastructures' interdependencies, their environment, their coupling behavior, types of failure, state of operation and characteristics. Although this classification scheme does not provide any means for actually profiling critical infrastructure characteristics, it offers good

Table 5 The contribution of UML-CI to the first three dimensions of critical infrastructure characteristics

High level metamodel	Types of interdependency			
	Physical	Cyber	Logical	Geographical
Ownership and management			Government, Strategic_Objective, Stakeholder, Actors, Policy_Maker, Regulation	Geographical_Bound, Operational_Bound, Operational_Bound
Structure and organization	Infrastructure, System, Task	Infrastructure, System, Task	Infrastructure, System	
TRV				
Resource	Asset, Physical_Asset, Operational_Requirement, Manufactured_Product	Asset, Cyber_Asset, Non_Physical_Asset, Operational_Requirement, Manufactured_Product	Operational_Requirement, Manufactured_Product	
Relationship	Consists_of, Subtask, Subsystem, Fulfils, Owns, Generates, Exploits, Built_From, Relies_On, Depends_On	Consists_of, Fulfils, Owns, Generates, Exploits, Relies_On, Depends_On	Relies_On, Depends_On	Performs_At, Accessible_At, Encompass

High level metamodel	Environment				State of operation		
	Economic and business opportunities	Public policies and legal concerns	Security issues	Government decisions	Repair	Disrupted	Normal
Ownership and Management	Stakeholder, Strategic_Objective, Actor	Policy_Maker, Regulation		Government, Policy_Maker, Regulation			
Structure and organization	Infrastructure			Infrastructure	System,Task	System,Task	System,Task
TRV			Protection_Order, SP_Provision, Mitigation				
Resource							
Relationship	Relies_On, Depends_On, Engaged_With, Pursues, Administers	Developed_By, Controlled_By, Restricted_By	Secures, Monitors	Governs			

Table 6 The contribution of UML-CI to the second three dimensions of critical infrastructure characteristics

High level metamodel	Type of failure			Infrastructure characteristics			Coupling and response behavior		
	Common cause	Cascading	Escalating	Organizational	Operational	Temporalspatial	Coupling order	Coupling degree	Type of interaction
Ownership and Management				Government, Strategic_ Objective, Stakeholder, Actors, Policy_Maker, Regulation					
Structure and Organization				Infrastructure	System, Task	Infrastructure, System, Task	Infrastructure, System, Task	Infrastructure, System, Task	Infrastructure, System, Task
TRV	Hazard, Threat Risk, Vulnerability Cause, Consequence	Hazard, Threat Risk, Vulnerability Cause, Consequence	Hazard, Threat Risk, Vulnerability Cause, Consequence	Mitigation, Protection_Order					
Resource						Asset and its Descendent	Operational_ Requirement, Manufactured_ Product	Operational_ Requirement, Manufactured_ Product	Operational_ Requirement, Manufactured_ Product
Relationship	Exposed_To, At_Risk_Of, Caused_By, Causes	Exposed_To, At_Risk_Of, Caused_By, Causes	Exposed_To, At_Risk_Of, Caused_By, Causes	Pursue, Controlled_By, Restricted_By	Generates, Fulfils, Performs_At, Exploits	Built_From, Encompass, Performs_At, Accessible_At, Consists_of, Subtask, Subsystem	Consists_Of, Subtask, Subsystem, Relies_On, Depends_On	Relies_On, Depends_On	Relies_On, Depends_On

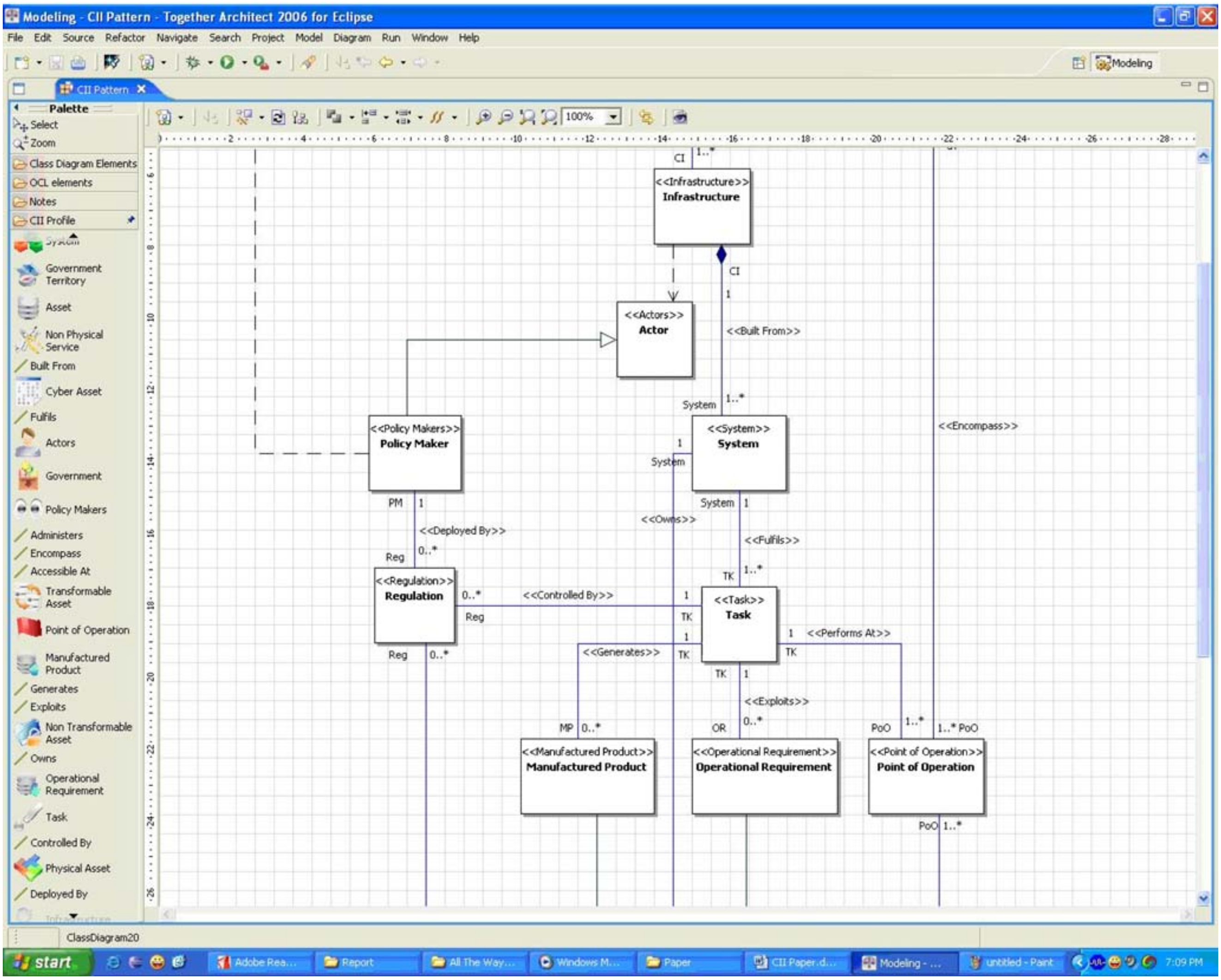

Fig. 11 A screenshot of the added UML-CI reference model to Borland Together Architect 2006

insight for understanding the features that should be sought while studying critical infrastructure systems and is currently the major touchstone for infrastructure characteristic and interdependency evaluation; therefore, it is required that its dimensions be covered in a reference model for such purpose. For this reason, we have employed these six dimensions as a way to evaluate the degree of contribution of our proposed UML-CI reference model to the correct perception and profiling of a typical critical infrastructure system organization and behavior.

To make a clear judgment on how well the reference model satisfies the requirements for profiling and modeling typical critical infrastructure characteristics, Tables 5 and 6 have been drawn. The columns of the tables represent various dimensions of Rinaldi's classification, while the rows specify one of the five metamodels of our proposed reference model. The confluence point of these two dimensions shows the metaclasses that are offered through UML-CI. As an example, the security issues within the environment dimension of Rinaldi's classification is satisfied by protection order, SP provision, and mitigation metaclasses in the TRV metamodel and by secures, and monitors metalinks in the relationship metamodel. It is visible from these tables that most of the Rinaldi's classification dimensions have been thoroughly addressed by at least one of the high level metamodels in the proposed UML-CI reference model. This implies that the proposed reference model can be used to model the type of infrastructure systems that are currently being addressed by the classification proposed by Rinaldi.

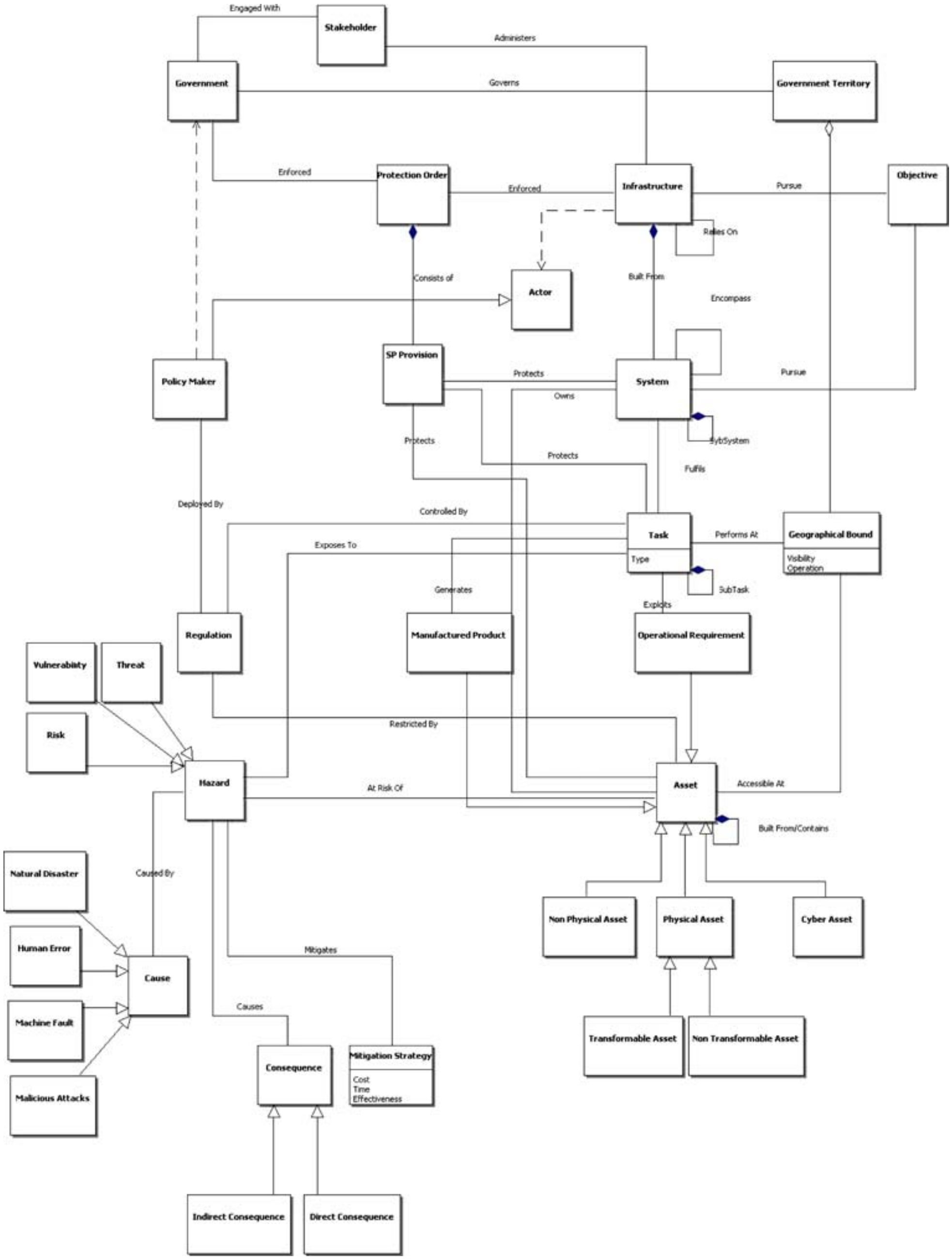

Fig. 12 A symbolic critical infrastructure pattern

The classification given in Tables 5 and 6 allow interested modelers to conveniently find and select the most appropriate metaclass that suites their requirements and needs. As it can be seen, each of the five metamodels in UML-CI captures several closely related issues of a single aspect; therefore, the selection of the most suitable metaclass for each circumstance requires close attention to the description of each metaclass, its metamodel, and its position in the devised tables, hence the tables can both serve as a means to show the modeling coverage of the UML-CI reference model and act as a tool for guiding the modelers that are employing UML-CI to choose the most suitable metaclasses for each of their specific needs.

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. A snapshot of the UML-CI profile palette in Together Architect 2006 is shown in Fig. 11.

To make the available metamodels in the UML-CI reference model more clear, and to show how the introduced high level metamodels can form a unique infrastructure organization, we have designed a sample instantiated model. This model does not intend to create a sample infrastructure representation and is only provided to show how different metaclasses that are available in UML-CI form a whole. As it can be seen in Fig. 12, the abstract metaclasses like Asset or Actor that will not be a part of the real modeling process (because they are only abstract metaclasses) have also been placed on the diagram for the sake of clarity and completeness. We call this diagram a *'Symbolic Critical Infrastructure Pattern'*. It can be used as a guideline for composing a typical critical infrastructure model, and understanding the proper composition of the UML-CI metaclasses.

9 Conclusions and future work

In this paper, we have proposed 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 metamodels 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.

As has been stated in the paper, 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. As opposed to other proposed models in the realm of critical infrastructures (Bagheri and Ghorbani 2008b), UML-CI takes a more fundamental approach to understanding critical infrastructure systems by providing means for modeling their structure and interdependencies without obliging the modelers to forward think about possible simulation scenarios. Such abstraction of modeling from simulation makes the models developed using UML-CI more suitable for different purposes and allows the possibility for porting such models into various simulation platforms.

The high-level metamodels of our proposed reference model have been compared with the most well-known classification of critical infrastructure system characteristics (Rinaldi et al. 2001). This comparison shows that UML-CI can fully cover the required dimensions for modeling infrastructure systems. It is worth noting that UML-CI moves beyond this classification by actually providing modelers with tangible means and tools to pursue their task.

Currently, we are extending the efforts made in this paper in two major directions:

(1) We are integrating the proposed reference model with a system analysis method (e.g. HHM (Haimes 1981), Astrolabe (Bagheri and Ghorbani 2008a)). System analysis is mostly involved with the identification of the structural and business aspects of the systems involved in an infrastructure. It attempts to depict a clear picture of the current infrastructure organization through the identification of its different components. The identification of system components facilitates the specification of their requirements and productions, which allows an initial recognition of the set of interdependencies. A system analysis method also assists us in the identification of various infrastructure components. Based on these findings the information can be incorporated into our proposed reference model. The integration of these two models can provide the modelers that intend to model a critical infrastructure system with a clear path to follow.

(2) We are creating a transformation mechanism for the UML-CI reference model, so that its models can be converted into executable simulation programs. This would allow a deeper understanding of the dynamic behavior of an infrastructure system based on its current setting and under various simulation scenarios. For the simulation aspect of this process, we have already developed a simulation suite, AIMS (Bagheri et al. 2007), that receives as input a structured critical infrastructure simulation specification document and creates a multiagent platform under which the behavior of an infrastructure system can be observed and analyzed. Our future work would be to convert the PIM developed using UML-CI into PSM that are suited for execution in AIMS. This would require a transformation engine to do the conversion.

References

Allenby, B. (2004). Infrastructure in the anthropocene: Example of information and communication technology. *Journal of Infrastructure Systems, 10*(3), 79–86.

Amin, M. (2000). *National infrastructures as complex interactive networks. Automation, control and complexity: An integrated approach* (pp. 263–286). New York: Wiley.

Atkinson, C., & Kuhne, T. (2003). Model-driven development: A metamodeling foundation. *Software, IEEE, 20*(5), 36–41.

Bagheri, E., & Ghorbani, A. A. (2006a). A service oriented approach to critical infrastructure modeling. In *Workshop on service oriented techniques*. National Research Council: Canada, August.

Bagheri, E., & Ghorbani, A. A. (2006b). Towards an mda-oriented uml profile for critical infrastructure modeling. In *International conference on privacy, security and trust*. ACM, October.

Bagheri, E., & Ghorbani, A. A. (2008a). Astrolabe: A collaborative multi-perspective goal-oriented risk analysis methodology. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, in press.

Bagheri, E., & Ghorbani, A. A. (2008b) The state of the art in critical infrastructure protection: A framework for convergence. *International Journal of Critical Infrastructures, 4*(3), 215–244.

Bagheri, E., Baghi, H., Ghorbani, A. A., & Yari, A. (2007). An agent-based service-oriented simulation suite for critical infrastructure behavior analysis. *International Journal of Business Process Integration and Management, 2*(4), 312–326.

Barton, D. C., Eidson, E. D., Schoenwald, D. A., Stamber, K. L., & Reinert, R. K. (2000). Aspen-ee: An agent-based model of infrastructure interdependency. Technical Report SAND2000-2925.

Barton, D. C., & Stamber, K. L. (2000). An agent-based microsimulation of critical infrastructure systems. Technical Report SAND2000-0808C.

Basu, N., Pryor, R., & Quint, T. (1998). Aspen: A microsimulation model of the economy. *Computational Economics, 12*(3), 223–41.

Brown, T., Beyeler, W., & Barton, D. (2004). Assessing infrastructure interdependencies: the challenge of risk analysis for complex adaptive systems. *International Journal of Critical Infrastructures, 1*(1), 108–117.

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.

Fowler, M. (2004). *Uml distilled: A brief guide to the standard object modeling language*. New York: Addison-Wesley Professional.

Fuentes-Fernandez, L., & Vallecillo-Moreno, A. (2004). An introduction to uml profiles. *UPGRADE, European Journal for the Informatics Professional, 5*(2), 5–13, April.

Haimes, Y. Y. (1981). Hierarchical holographic modeling. *Transaction on Systems, Man, and Cybernetics, 11*, 606–617.

Haimes, Y. Y., Horowitz, B. M., Lambert, J. H., Santos, J. R., Lian, C., & Crowther, K. G. (2005a). Inoperability input-output model for interdependent infrastructure sectors. I: Theory and methodology. *Journal of Infrastructure Systems, 11*(2), 67–79.

Haimes, Y.Y., Horowitz, B.M., Lambert, J.H., Santos, J.R., Lian, C., & Crowther, K.G. (2005b). Inoperability input-output model for interdependent infrastructure sectors. II: Case studies. *Journal of Infrastructure Systems, 11*(2), 80–92.

Herder, P. M., Turk, A. L., Subrahmanian, E., & Westerberg, A. W. (2000). Challenges for process systems engineering in infrastructure design. *Computers and Chemical Engineering, 24*, 1775–1780. M3: doi:10.1016/S0098-1354(00)00463-4.

Jonsson, D. K. (2005). The nature of infrasystem services. *Journal of Infrastructure Systems, 11*(1), 2–8.

Kletz, T. A. (1999). *Hazop and Hazan: Identifying and assessing process industry hazards*. London: Taylor Francis.

Leontief, W. (1951). *The structure of the American economy*. New York: Oxford University Press.

Leontief, W. (1966). *Input-output economics*. New York: Oxford University Press.

Narich, R. (2005). Critical infrastructure, continuity of services and international cooperation. *International Journal of Critical Infrastructures, 1*, 293–298.

Nozick, L. K., Turnquist, M. A., Jones, D. A., Davis, J. R., & Lawton, C. R. (2005). Assessing the performance of interdependent infrastructures and optimising investments. *International Journal of Critical Infrastructures, 1*(2/3), 144–154.

Panzieri, S., Setola, R., & Ulivi, G. (2005). An approach to model complex interdependent infrastructures. In *16th IFAC World Congress*.

Rinaldi, S. M. (2004). Modeling and simulating critical infrastructures and their interdependencies. In *System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on*, p. 8.

Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems Magazine, 21*(6), 11–25.

Schoenwald, D. A., Barton, D. C., & Ehlen, M. A. (2004). An agent-based simulation laboratory for economics and infrastructure interdependency. In *Proceedings of the 2004 American control conference* (vol. 2, pp. 1295–1300).

Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software, 20*(5), 19–25.

Thissen, W. A., & Herder, P. M. (2003). Critical infrastructures: challenges for systems engineering. In *IEEE International conference on systems, man and cybernetics, 2003* (vol. 2, pp. 2042–2047).

Uml infrastructure specification, v2.1.1. Technical report, Object Management Group, February 2007.

Warmer, J., & Kleppe, A. (2003). *The object constraint language: Getting your models ready for MDA*. New York: Addison Wesley.

Ebrahim Bagheri joined the Faculty of Computer Science at the University of New Brunswick, Canada as a PhD student in January 2006. Before going to UNB, he spent six years at the Ferdowsi University of Mashhad, completing his Bachelors in Computer Software Engineering and his Masters specializing in pervasive computing environments. At present, he is interested in the design of adaptive systems, collaborative modeling and design technologies, simulation of complex interconnected systems and also the application of data fusion and belief theory techniques to software engineering.

Ali A. Ghorbani is currently a Professor and Assistant Dean (Research & Outreach) at the University of New Brunswick (UNB), Canada. His current research focus is on web intelligence, network security, complex adaptive systems, critical infrastructure protection, and Trust & Security assurance. Dr. Ghorbani, the Director of Information Security Centre of Excellence, is also the coordinator of the Privacy, Security and Trust (PST) network annual conference. He holds UNB Research Scholar position and is co-Editor-in-Chief of the Computational Intelligence, an International Journal, and Associate Editor of International Journal of Information Technology and Web Engineering. Dr. Ghorbani is a member of ACM, IEEE, IEEE Computer Society, and CSCSI.