

# Support tool development for real-time risk prediction in interdependent critical infrastructures

Thomas Schaberreiter, Cédric Bonhomme, Jocelyn Aubert, Christophe Incoul and Djamel Khadraoui  
Service Science & Innovation (SSI)  
Centre de Recherche Public Henri Tudor  
29, avenue John F. Kennedy, L-1855 Luxembourg, Luxembourg  
{firstname.lastname}@tudor.lu

**Abstract**—Critical infrastructure (CI) services are consumed by the society constantly and we expect them to be available 24 hours a day. A common definition is that CIs are so vital to our society that a disruption or destruction would have a severe impact on the social well-being and the economy on a national and an international level. CI sectors include, amongst others, the electricity, telecommunication, air traffic and transport sectors. CIs can be mutually dependent on each other and a failure in one infrastructure can cascade to another interdependent infrastructure to cause service disruptions. Methods to better assess and monitor CIs and their interdependencies in order to predict possible risks have to be developed. In this work we present implementation details of RESCI-MONITOR (Real-time Evaluation of Security - MONITOR), a support tool enabling to simulate and evaluate previous work on CI security modelling. The CI security model enables to monitor CI services and it's associated dependencies in real-time by evaluating the current risk in CI services. The multi-agent based support tool is able to receive real-time measurements from the infrastructure, transform them into risk parameters and evaluate them in combination with the current risk in dependent infrastructure services.

**Keywords**-Critical infrastructures, risk, security modelling, multi-agent system.

## I. INTRODUCTION

CI security modelling was presented in [1], [2]. As illustrated in Figure 1, the aim of the approach is to transform real-world infrastructure information into common abstract risk related information (in our case confidentiality, integrity and availability- CIA), to use this information to monitor the state of the infrastructure and to share it with interdependent infrastructures in order to be able to evaluate the current infrastructure risk by taking into account the interdependencies.

Our methodology, as illustrated in Figure 2, is composed of three steps: an off-line risk assessment, a measurement aggregation and an on-line monitoring step.

The off-line risk assessment step deals with an analysis of the infrastructure. We see CIs as service providers which provide services to customers. Those customers can in turn be other dependent or interdependent CIs or CI services which need the service in order to provide their own service(s). In order to be able to observe the CI state,

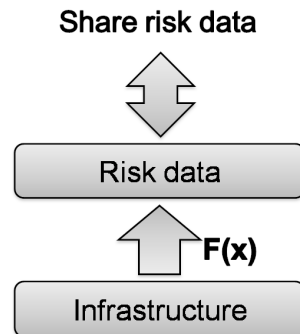


Figure 1. Security modelling approach

we need to identify observable entities in the infrastructure (base measurements). Each base measurement is associated to one or more CI service(s). Furthermore, to quantify the contribution of a base measurement to the confidentiality, integrity and availability of a service, a weight is associated to each base measurement. The same applies to each identified dependency. It is weighted to reflect how much a dependent service contributes to the confidentiality, integrity and availability of a service. In the model, each service can be composed of sub-services (0..n), base measurements (0..n) and dependencies (0..n).

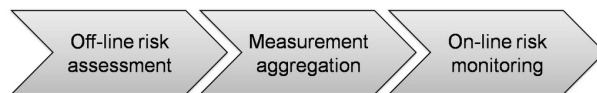


Figure 2. Steps of security modelling

In the measurement aggregation step, each identified base measurement is normalized to a five step scale to be able to compare base measurements. For discrete decimal values (e.g. current system load) this normalization is done by calculating the deviation from an expected value and categorizing this deviation into five classes. For Boolean data (e.g. switch is on/off) this normalization is easier, as it is either reached or not reached. After the normalization, the

base measurements assigned to each service are aggregated into risk data by calculating an averaged weighted sum of the normalized base measurements multiplied by their weights. This will produce three risk indicators (CIA) for each service, each representing a risk level between [1..5]. Sub-service risk levels are also taken into account in the service risk aggregation. Risk in this context can be seen as CI behaviour different from normal behaviour. This can be applied to virtually any situation where a CI service behaves different from normal operation. In our approach this can be expressed numerically in CIA indicators. The reduction to five levels of risk was chosen as a trade-off between granularity of risk representation and the interpretability of risk information by an operator in a stress situation.

In the on-line risk monitoring step, each CI service receives risk indicators from interdependent CI services and can, after applying the associated interdependency weight for confidentiality, integrity and availability, use this information to adjust the overall service risk in real-time.

The security modelling approach tries to address the challenge of on-line monitoring of the state of CI services and their interdependent services. To our knowledge, no approach was presented that does not only aim at monitoring the availability of a service, but also other important system parameters like confidentiality and integrity. Furthermore, an other advantage of our approach is the reduction of the complexity of a service through abstraction to a common (risk related) set of parameters. This enables to compare CIs designed to serve a very different purpose (electricity, telecommunication, air traffic,...) and that are composed of very different infrastructure components. Usually information about the state of CIs is confidential and providers hesitate to share the information that would enhance security of their infrastructure or the quality of their services. Information sharing between CIs is seen as a key feature to enhance CI protection [3] and we think that the abstraction to a small set of common parameters will encourage service providers to share them with interdependent providers.

In this work we present the designed implementation of RESCI-MONITOR, the support tool associated to the security model. More specifically, the tool implements the risk aggregation and real-time monitoring step of the security model based on a multi-agent system. The tool uses an infrastructure configuration that contains information provided by the infrastructure analysis in the off-line risk assessment step. The definition of a XML based configuration format was part of this work, but will not be covered in detail.

The rest of the paper is organized as follows: Section II introduces related work and Section III will cover aspects related to the risk assessment of the security model. In Section IV the architecture of the support tool is covered and Section V summarizes the paper and discusses future work.

## II. RELATED WORK

CI interdependencies are complex and not easy to understand. In [4] Rinaldi et al. provide an excellent overview on the dimensions in which interdependencies can occur. In [5] CIs and their interdependencies are analyzed and different suitable modelling techniques are discussed. Conceptual modelling is used in [6] by Sokolowski et al. to represent an abstract, simplified view of CIs. In [7] Panzerini et al. utilize the complex adaptive systems (CAS) approach. The model is derived by modelling the mutually dependent sub-systems of the infrastructure. In [8] Adar et al. discuss challenges in CI risk management and outline methods as well as best practice guidelines to address risk management in CIs. In [9] Tan et al. achieve real-time risk management in three phases: risk analysis, risk evaluation and risk prediction. The continuous time hidden Markov model is used for risk evaluation. In [10] Haslum et al. use continuous-time hidden Markov models for real-time risk calculation and estimation. In [11] Newman et al. assess risk in complex interacting infrastructures based on the coupling related to the risk of component failure. In [12] Baiardi et al. propose a risk management strategy based on a hyper-graph model to detect complex attacks as well as to support risk mitigation.

Multi-agent platforms have shown their relevance for implementation in several publications. In [13] a computer Weather forecast Alert Broadcasting System (WABS) based on Multi-Agent System (MAS) is introduced, the MAS architecture offers the advantage to be distributed and autonomous. In [14] Utopia, an institution-oriented and institution-based programming framework, is presented. Utopia permits to easily and automatically set up a MAS thanks to a XML specification file. Multi-agent technology is used in [15] as an innovative approach that focuses on business goals for defining access rights. In [16] security related issues in multi-agent implementations are discussed and concepts to solve those issues are presented.

## III. RISK ASSESSMENT DETAILS - INFRASTRUCTURE DECOMPOSITION

At this point and before describing the actual implementation, one aspect of the risk assessment step of the security model will be described. In our point of view this will greatly improve the understanding of the structure and functionality of the support tool. As mentioned before, the risk assessment step deals with identifying relevant critical services and interdependencies as well as infrastructure measurements. In order to have a meaningful representation of complex infrastructure risk, we think it is necessary to decompose infrastructure into smaller parts, each calculating the risk separately on lower levels before aggregating it into combined infrastructure risk levels on the higher level. A hierarchical, tree-like representation of CIs was chosen. To stay with the concept of services, each infrastructure is decomposed into services and its sub-services, a service can



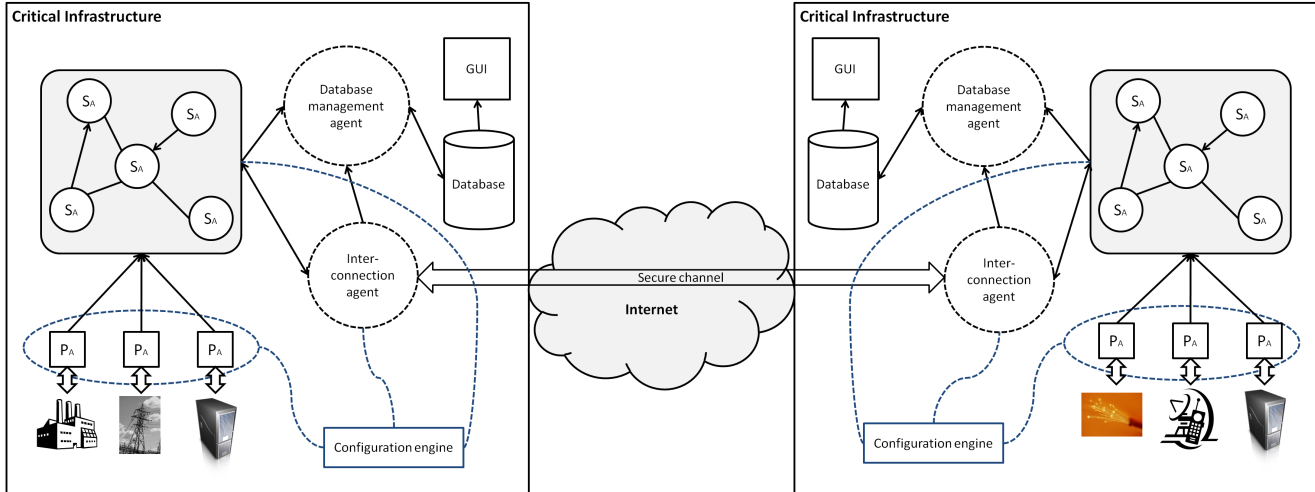


Figure 5. Global architecture of support tool

infrastructure interface they need to connect and which base measurement they need to monitor. Furthermore, the configuration contains connection information to interdependent CIs (or actually to their interconnection agents). This information is used to configure the interconnection agents and allows them to connect to an interdependent infrastructure if required. The last class of information in the XML configuration is information about identified CI services. Each entry contains information about the associated base measurements (which is used to configure a service to register to the appropriate probe agents), the associated sub-services (which is used to configure a service to register for security states updates of those services) and it contains information about dependencies (which is used to configure a service to register to security states updates of internal or external dependent services). Each base measurement, sub-service and dependency assigned to a service has a specific weight assigned by an expert indicating how much this entity contributes to the confidentiality, integrity and availability of the service. This information is used during risk aggregation and is assigned to the service agents in the configuration phase.

### B. Probe agents

Probe agents are actually organized in two parts: a generalized part that is used for base measurement normalization and a individual part that acts as an interface to the CI equipment that can provide base measurements. It has to be implemented separately for each type of CI equipment. The interface part is located at the CI equipment and communicates to the measurement agent via an IP interface. A base measurement is characterized by a unique name and the current base measurement value. A probe agent can register for a base measurement and the infrastructure interface will

send updated values to all registered probe agents. In our test set-up, CIs are currently simulated via python scripts that generate random base measurement values and distribute them to registered probe agents.

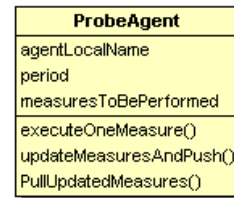


Figure 6. UML class diagram of probe agents

An overview of the generalized part of a probe agent can be seen in the UML diagram in Figure 6. When an agent receives an updated value, it transforms this raw value into a normalized value in the [1..5] range that can be compared and processed. This transformation involves computing the deviation of the current value towards an expected value, and comparing this deviation with four threshold values previously defined. This last step enables producing discrete natural values between [1..5]. The expected measurement value and the allowed deviations are obtained during off-line risk assessment and assigned to the probe agents at creation time via the configuration engine. By comparing this current normalized value to the previously obtained value, the agent is able to know whether or not it should push the value to the service agent which depends on this base measurement. In addition, the agent is also able to receive measurement update requests from a specific service agent, to provide a response in pull mode.

### C. Service agents

This generic agent is used to represent any level of service in the infrastructure decomposition tree: from the CI root itself to the lowest level service. Such an agent, presented in Figure 7 is in charge of retrieving risk indicators from any relevant information source to compute CIA indicators representing the current service risk. Information sources can be of three types:

- Base measurements provided by probe agents [0..n]
- Risk indicators provided by its sub-services [0..n]
- Risk indicators provided by services on which the service depends [0..n]

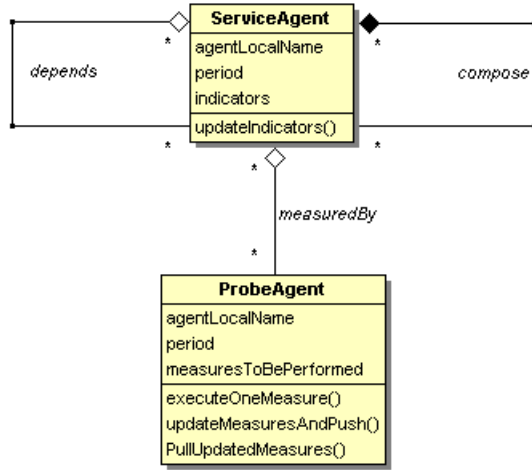


Figure 7. UML class diagram of service agents

The agent aggregates either base measurements into risk indicators (measurements aggregation) using the weights defined for each measure, or directly risk indicators (risk level aggregation) also using weights representing the dependency level between services. Sub-services can be seen as dependent services, each having a weight representing the influence a sub-service has to its super service. Once all data sources are integrated, the agent is able to aggregate its own risk level indicators.

This level of risk, if different from previous one, is then forwarded to potential super services and dependent services either directly (internal services) or through the interconnection agent (external services). In parallel, data is stored in a database through the database management agent. In doing so, each service level of an infrastructure is able to precisely determine its own level of risk. In the case of the root service (i.e. the infrastructure itself), these values provide an overall indication of the current state of the infrastructure. Similar to probe agents, a service agent can also operate in pull mode, responding to higher level service requests.

### D. Database and database management agent

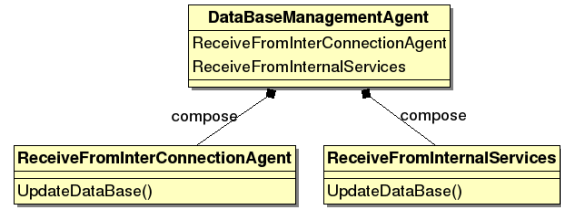


Figure 8. UML behaviour diagram of Database management agent

As depicted in Figure 5, each CI has his own database management agent and database to memorize the history of calculated and received risk level indicators.

The database management agent is used as the only point of connection to the relational database from within the multi-agent platform to operate CRUD (create, read, update and delete) operations on the data. This architecture was chosen to be able to host the database and database manager on a different, possibly more secure place since it is assumed that the database is a valuable point of attack. We utilize the fact that a distributed agent model permits to host the database management agent in a different location. The database has three main roles:

- It stores the tree representation of the infrastructure and the interdependencies to other infrastructures. This structure is derived from the XML configuration created in the off-line risk assessment step.
- It stores the risk information (CIA) of each service agent and interconnection agent tagged with a timestamp.
- It stores all the normalized measures calculated by the probe agents tagged with a timestamp.

The database management agent handles messages coming from the interconnection agent described in Section IV-E and the internal services. This is achieved using the two behaviours shown in Figure 8. Every time a risk value is received, the database is updated by performing CRUD operations.

### E. Interconnection agent

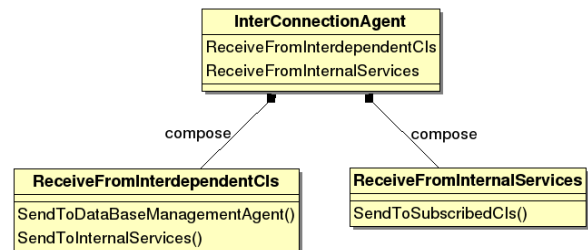


Figure 9. UML behaviour diagram of interconnection agent

As illustrated by the UML behaviour diagram in Figure 9, the interconnection agent is responsible for sending updated risk information of internal services to interdependent CIs and receiving updated risk values from interdependent CIs. Risk information received from interdependent CIs is distributed to subscribed internal services and additionally stored in the database via the database manager. Connection to interdependent CIs is achieved by maintaining a secure connection via a web service. A detailed description about set-up and maintenance of the secure connections can be found in Section IV-F. Configuration of the interconnection agent (e.g. what interdependent CI to connect to, what internal services need to be updated by a received external service risk, ...) is done at creation time by the configuration engine using the XML configuration.

#### *F. Security considerations of implementation*

Our main security concern in this implementation is communication between agents. Therefore we have to deal with security in two different areas, the security of communication between agents inside a CI (within a JADE framework instance) and security for the communication between CIs over the internet (communication between two JADE framework instances). Utilizing security features of the JADE framework, we assume that information security can be provided without additional attention for the communication of agents inside a JADE framework instance. However, the link between interdependent CIs implies a transfer of data over untrusted networks, such as the internet. It is important to ensure information security principles (confidentiality, integrity, authenticity and non-reputation) for data exchanged between linked interconnection agents. Our considerations for both communication links will be detailed in this Section.

1) *Communication between agents inside CIs:* The JADE framework provides a plug-in dedicated to security, JADE-S. More detailed information about the security features provided by JADE-S can be found for example in [16]. JADE-S implements different kinds of security features for the communication between agents and we utilize them in our implementation. More specifically, we ensure:

- user authentication
- authorization of actions performed by an agent
- message signature (data integrity, non repudiation)
- encryption (confidentiality)

JADE-S introduces the concept of a multi-user system where all components (agents, containers) in a platform are owned by authenticated users (with permissions). Authentication is carried out by the security service, the base of JADE-S. Standard login methods are supported by JADE like Unix, Kerberos and local plain text. In our implementation we use plain text authentication, credentials are stored in the XML configuration structure. Authentication of agents is done at creation time of a JADE platform instance.

An agent container and its agents can join the platform only after a successful login. Authorization is done via an ACL (Access Control List) which defines the permissions of authenticated users. For each agent, every permitted action has to be defined in this file. We consider a push approach, each agent (probe agent, service agent, interconnection agent) decides when to send updated information to subscribed agents. No access to internal information of an agent needs to be granted. Since our service agents and probe agents support a pull approach as well, we can ensure authorization with JADE-S.

To ensure message encryption (confidentiality) and message signatures (integrity and non repudiation) for data exchanged between agents, a public/private key pair is created by each agent at creation time and the public key is distributed to communication partners after authentication. Each agent encrypts the data with the public key of the communication partner and signs it with its own private key.

2) *Communication between interconnection agents of CIs:* Interconnection agents communicate over the internet through a secured channel. Data passing over the internet is handled by a web service based on Apache Axis and WS-Security. Axis is a web service based on the SOAP (Simple Object Access Protocol). SOAP is a specification for exchanging structured information and relies on XML (Extensible Markup Language) for its message format. SOAP message can be signed and verified using WS-Security (via Apache XML-Security). WS-Security is a member of the WS-\* family and provided by OASIS (Organization for the Advancement of Structured Information Standards). An implementation of WS-Security is WSS4J, which can be used with Apache Axis. WSS4J can generate the following SOAP bindings: XML Security (XML Signature, XML Encryption) and Tokens (Username Tokens, Timestamps).

Thus WSS4J addresses security by leveraging existing standards and specifications (X.509 certificates, kerberos tickets, userid/password credentials, etc.). Consequently security issues between CIs are handled by standards and well defined protocols.

In our prototype we choose the ElGamal asymmetric key encryption algorithm for data encryption. The keys of each agent are generated at configuration time using the keytool command. Authentication in our tool is done via the username/password method. After authentication the public part can be shared between agents. From this point it is possible to sign and encrypt messages (which are in SOAP XML format) with Apache XML-Security (as mentioned above).

#### *G. Infrastructure monitoring: A graphical user interface*

The graphical user interface presented in Figure 10 is designed to provide an easily understandable overview of the security status of a CI by displaying the real-time risk of each CI service. The major space on the left side of the



Figure 10. Graphical user interface for CI risk monitoring

interface (1) shows the risk of the service an operator is currently interested in. The displayed service risk includes the risk of all sub-services and dependencies of the service in question. The three lines in the risk diagram represent the evolution of the confidentiality, integrity and availability service risk parameters over time in a scale [1..5]. On the right side of the service risk (2), a list of all sub-service risks is shown to allow a fast evaluation of causes in case the service risk increases. To support this, the sub-service list is constantly rearranged to show the sub-services with the highest current risk on top. A click on one of the sub-services will cause the interface to display the risk centred around this sub-service and shows it in the area on the left side (1) as main service. Sub-service and dependency risk will be displayed according to the configuration of this service. This allows to browse through the tree-like structure of the infrastructure described in Section III and will help an operator to rapidly determine the root cause of increased risk in higher level services. The traffic light inspired indicators (4) show the assurance level of the currently displayed risk and therefore the confidence we have in the correctness of this values (The concept of assurance in the security model was introduced in [1], [2] but was not covered in this publication). Below the area of the main service, dependent service risks are displayed (3). The current risk of dependent services is indicated in different colouring, ranging from green (low risk) to red (high risk). The different visualization scheme (compared to the sub-service risk) was chosen since it is assumed that, if the service risk comes from another CI, no further investigation revealing the root cause of the failure

can be done. Only the change in service risk is received from the interdependent CI, no details about the internal structure causing the failure are revealed. The GUI is supposed to be hosted on the same server as the database and can directly access the stored risk information. To enable remote access to the GUI, a secure HTTP connection can be established. The interface is implemented in HTML/Javascript and can be displayed in any standard browser. PHP is used to access the database containing aggregated risk information to update the GUI contents.

#### H. Used tools

For the implementation we used a number of tools. Those tools include Eclipse 3.5.1 as development environment, the multi-agent platform JADE 4.0.1 and the JADE-S plug-in version 3.7. JavaScript InvoVis Toolkit 2.0.0 was used to generate dynamic graphs, the SQLite 3.0 library was utilized to provide a database management system using SQL. For the graphical user interface we used xHTML version 1.1 and PHP version 5.3 for database access. Apache HTTP server version 2.2.16 is used to enable remote access to the GUI. The tool was designed to be operating system independent.

#### V. SUMMARY AND FUTURE WORK

In this paper we presented the design and implementation of RESCI-MONITOR, an agent-based support tool that enables simulation of a previously presented risk based model enabling real-time monitoring of CI services. The nature of this security model perfectly supports the use of a multi-agent system for implementation. After introducing relevant parts of the security model, the global architecture

of the tool was presented and the individual parts of the architecture were detailed.

Current and future work focuses on further enhancements of the security model. More specifically, it will be extended with a risk prediction component based on probabilistic networks, allowing to estimate the most probable future risk given a state change in the current risk of either the service or of an interdependent service. This approach will be integrated in the service agent component of the support tool to be able to predict, in real-time, the most probable evolution of a change in service risk. The business based reference scenario provided by an industrial partner in the EU-FP7 project MICIE will be mapped to this tool. This allows validate the security model as well as the performance of the RESCI-MONITOR tool in a close to reality set-up. An evaluation of the GUI by CI operators will be possible. Additional work is done in developing an even more intuitive graphical monitoring interface allowing CI operators to quickly detect risks and react to them. Furthermore, the operator interface will be extended with a dynamical configuration option to add or remove CI services or dependencies in real-time to be able to react faster to structural changes.

#### ACKNOWLEDGEMENTS

This work has been carried out in the framework of the MICIE project, partially funded by the EU with the contract FP7-ICT-225353/2008 and by the Luxembourgish Ministry of Culture, Higher Education and Research (MCESR). The authors thank all project partners for many interesting discussions which greatly helped to formulate the security modelling approach which lead to the implementation of the presented tool. One of the authors would like to thank the Luxembourgish National Research Fund (FNR) for funding his PhD research under AFR grant number PHD-09-103.

#### REFERENCES

- [1] J. Aubert, T. Schaberreiter, C. Incoul, D. Khadraoui, and B. Gateau, "Risk-based methodology for real-time security monitoring of interdependent services in critical infrastructures," in *Availability, Reliability, and Security, 2010. ARES '10 International Conference on*, feb. 2010, pp. 262–267.
- [2] J. Aubert, T. Schaberreiter, C. Incoul, and D. Khadraoui, "Real-time security monitoring of interdependent services in critical infrastructures. case study of a risk-based approach," in *21th European Safety and Reliability Conference (ESREL 2010) (To be published.)*, September 2010.
- [3] ENISA, "Good practice guide network security information exchanges," ENISA - European Network and Information Security Agency, Tech. Rep., 2009.
- [4] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Systems Magazine*, vol. 21, pp. 11–25, 2001.
- [5] S. Rinaldi, "Modeling and simulating critical infrastructures and their interdependencies," in *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, jan. 2004, p. 8 pp.
- [6] J. Sokolowski, C. Turnitsa, and S. Diallo, "A conceptual modeling method for critical infrastructure modeling," in *Simulation Symposium, 2008. ANSS 2008. 41st Annual*, April 2008, pp. 203–211.
- [7] S. Panzieri, R. Setola, and G. Ulivi, "An approach to model complex interdependent infrastructures," in *16th IFAC World Congress, 2005, cISIA, Critical Infrastructures*.
- [8] E. Adar and A. Wuchner, "Risk management for critical infrastructure protection (CIP) challenges, best practices tools," in *Critical Infrastructure Protection, First IEEE International Workshop on*, 3-4 2005, p. 8 pp.
- [9] X. Tan, Y. Zhang, X. Cui, and H. Xi, "Using hidden markov models to evaluate the real-time risks of network," in *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, 21-22 2008, pp. 490–493.
- [10] K. Haslum and A. Arnes, "Multisensor real-time risk assessment using continuous-time hidden markov models," in *Computational Intelligence and Security, 2006 International Conference on*, vol. 2, 3-6 2006, pp. 1536–1540.
- [11] D. Newman, B. Nkei, B. Carreras, I. Dobson, V. Lynch, and P. Gradney, "Risk assessment in complex interacting infrastructure systems," in *System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on*, jan. 2005, pp. 63c–63c.
- [12] F. Baiardi, C. Telmon, and D. Sgandurra, "Hierarchical, Model-based Risk Management of Critical Infrastructures," *Reliability Engineering & System Safety*, vol. 94, no. 9, pp. 1403–1415, 2009, ESREL 2007, the 18th European Safety and Reliability Conference.
- [13] C. Feltus, D. Khadraoui, and C. Bonhomme, "Electric black-out prevention: Toward a computer-mediated weather alert broadcasting solution," in *International Conference on Society and Information Technologies (ICSIT 2010)*, 2010, pp. 45–50.
- [14] P. Schmitt, C. Bonhomme, J. Aubert, and B. Gâteau, "Programming electronic institutions with utopia," in *Demo Session of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10)*, 2010.
- [15] B. Gateau, C. Feltus, J. Aubert, and C. Incoul, "An agent-based framework for identity management: The unsuspected relation with ISO/IEC 15504," in *RCIS. IEEE*, 2008, pp. 35–44.
- [16] X. A. V. Sobrino, A. Schuster, and A. Riera, "Security for a multi-agent system based on jade," *Computers & Security*, vol. 26, no. 5, pp. 391–400, 2007.