To appear in *Proceedings of the Fifteenth Annual International Symposium of the International Council on Systems Engineering (INCOSE '05), - Systems Engineering: Bridging Industry, Government, and Academia,* Rochester, NY, July 10-15. 2005.

Engineering Information Assurance for Critical Infrastructures: The DITSCAP Automation Study

Seok Won Lee, Gail-Joon Ahn and Robin A. Gandhi
Dept. of Software and Information Systems, The University of North Carolina at Charlotte
Charlotte, NC 28223-0001, USA. {seoklee, gahn, rgandhi}@uncc.edu

Abstract. Recent advances in information technology have transformed the way in which mission-critical services get delivered and are evaluated today. These services are heavily and increasingly relying on an interdependent crossed network of critical information infrastructures, spanning from private to government sectors. In order to enable such infrastructures to efficiently mitigate risks, optimize their security posture and evaluate their information assurance (IA) practices, we identify the need for a structured and comprehensive methodology for IA-aware critical infrastructure protection. In this paper, we focus on the automation study of the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) that is a standard for certifying and accrediting the information networks that comprise of the Defense Information Infrastructure (DII). We attempt to generalize a course of actions in DITSCAP that motivate our design principles and modeling techniques, supported by their theoretical backgrounds and demonstrable prototype interfaces to establish their appropriateness.

Keywords Information Security Requirements Engineering, Information Systems Certification and Accreditation, Critical Infrastructure Protection, Risk Assessment, Ontological Engineering

INTRODUCTION

Critical Infrastructure Protection (CIP) (Bush 2002) is essentially dependent on the quality of underlying software, systems, practice and environment to ensure high quality of service and trust, as which the information infrastructures are increasingly a major component of business, industry, government and defense. Especially, ensuring high quality of IA for software systems demands careful and thorough awareness of policies and goals for computation, access, service, and trust, as well as continual monitoring and assessment of operations, and requires taking corrective actions based on well-defined metrics and measures. (CNSS-4009 2003) defines IA as: "*Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. These measures include providing for restoration of information systems by incorporating protection, detection, and reaction capabilities.*"

In order to achieve such goals for both CIP and quality of IA, it is essential to develop a well-defined (theoretically justifiable), systematic (repeatable to validate), and practical (deployable to organizations) IA engineering methodology. In this paper, we address these research concerns from the aspect of *Department of Defense (DoD) Information Technology Security Certification and Accreditation Process* (DITSCAP) (DoD 5200.40 1997, DoD 8510.1-M 2000), that is a standard certification and accreditation (C&A) process for information systems that comprise of the DII. We propose to develop the *DITSCAP Automation Tool* (DITSCAP-AT) which will automate major portions of the DITSCAP by applying and integrating novel techniques from

software systems requirements engineering, knowledge engineering, and information security. We describe our design principles, modeling techniques, and their theoretical backgrounds with demonstrable prototype interfaces. In the next section, we provide a brief overview of the DITSCAP followed by the motivation and objectives of the DITSCAP automation.

DITSCAP OVERVIEW

The Role of DITSCAP. DITSCAP Application Manual (DoD 8510.1-M 2000) defines the DITSCAP as "The standard DoD process for identifying information security requirements, providing security solutions, and managing information systems security activities". DITSCAP achieves the goals outlined in this definition, by prescribing a standard DoD wide process and establishing a management infrastructure leading to the acquisition and maintenance of C&A for secure operations of information systems. DITSCAP focuses on the system mission, environment, architecture, and life cycle while assessing the impact of operation of that information network on the DII.

DITSCAP Motives, Roles and Artifacts. DITSCAP defines *Certification* in the context of information systems as the comprehensive evaluation of the technical and non-technical security features of an information system and other safeguards made in support of the accreditation process, to establish the extent to which a particular design and implementation meets a set of specified security requirements. Following the certification activities, the accreditation statement is an approval to operate the information system in a particular security mode using a prescribed set of safeguards at an acceptable level of risk by a Designated Approving Authority (DAA). It should be noted that, the relationship of the C&A process with information systems is not something that is established once to get over with, but it should be a life time commitment (Kimbell et al. 2001). DITSCAP tries to fulfill this commitment by distributing its activities over four phases that range from the initiation of the C&A activities to its maintenance and reaccreditations. The level of rigor adopted for the C&A process depends on the certification level chosen for the system among the four levels available which are 1) Minimal Security Checklist; 2) Minimum Analysis; 3) Detailed Analysis; and 4) Extensive Analysis. (DoD 8510.1-M 2000) describes these levels, their selection criteria and the associated activities in each phase of the DITSCAP in further detail.

The Program Manager, DAA, Certifier and User Representative are the key roles of DITSCAP that tailor and scope the C&A efforts to the particular mission, environment, system architecture, threats, funding and schedule of the system through negotiations. The DITSCAP requires that a "system" (traditionally referred to as a "security domain") be defined. Once the system definition has been agreed upon by the key roles of DITSCAP, it becomes the Software Security Authorization Agreement (SSAA). The SSAA is especially important because it is used throughout the entire DITSCAP process to guide actions, document decisions, specify IA requirements, document certification tailoring and level-of-effort, identify potential solutions, and maintain operational systems security (DoD 8510.1-M 2000).

THE DITSCAP AUTOMATION TOOL

Motivation for DITSCAP Automation Tool (DITSCAP-AT). DITSCAP has been carefully engineered to conduct a comprehensive C&A of information systems from organizational, business, technical and human perspectives, however, we identify that DITSCAP in its current approach has certain limitations and missing components. DITSCAP itself can be quite

overwhelming due to its long and exhaustive process of documentation and the analysis that follows it. Although the DITSCAP Application Manual (DoD 8510.1-M 2000) lays out a standard template outlining a list of tasks to be carried out along with the roles and responsibilities of the associated personnel, it is expressed at a very abstract level to maintain general applicability. This inherent abstractness makes it hard to ensure objectivity, predictability and repeatability in the interpretation and enforcement of the high level policies and directives of DITSCAP in real world settings. Lack of automated means to conduct cross checks through long and exhaustive documentation resulting from the C&A process further aggravates this problem. The multitude and diversity of DoD directives and security requisites required to be referenced and comprehended to determine the applicable security requirements based on specific user criteria, also restricts human ability to engineer systems that are compliant with DITSCAP. A well-defined risk assessment methodology to organize the collection of threat and vulnerability information from a broad spectrum of risk sources, technical and non-technical, is also missing in the current approach.

We believe that the automated DITSCAP will result in certification costs savings due to the need of fewer resources to conduct, manage and maintain the C&A process, by providing an integrated environment to articulate the C&A efforts. Also, such an integrated environment is inevitable to maintain efficiency of C&A activities and thus, rapidly reducing development time of information systems. We believe that all these factors advocate a strong and urgent need for a well-defined and comprehensive IA engineering methodology to gain a high level of assurance from the systems that are subject to the DITSCAP. Currently we limit the scope of DITSCAP-AT to level one DITSCAP certification as applied to networked systems only.

DITSCAP-AT Objectives. From the perspective of providing tool support for DITSCAP, the first and foremost objective of DITSCAP-AT is to guide the personnel involved in conducting the C&A process throughout DITSCAP by a well-organized flow of tasks in different activities. To reduce the amount of long and exhaustive documentation, user friendly interfaces need to be employed and leverage well-formed questions and criteria with predefined answers to them, which become the basis for building well-defined metrics and measures. DITSCAP-AT should also map to and reflect the language of the existing DoD directives, instructions and other requisites in all the artifacts (i.e. the SSAA) produced by it. DITSCAP-AT also requires network self-discovery capabilities that allow the comparison of the intended environment with the actual environment and compliance assessment of network security practices in the operational environment with the enforced requirements.

Furthermore, to satisfy the C&A goals of DITSCAP, the automation framework should be able to establish the extent to which an information system meets the specified security requirements by supporting the process of identifying and interpreting the applicable requirements. DITSCAP-AT also demands structured, justifiable and repeatable methods to have for a comprehensive risk assessment methodology providing a firm basis to establish cost versus risk measures. In the following section, we present the DITSCAP-AT conceptual architecture conceived through our analysis to realize these objectives.

THE DITSCAP-AT CONCEPTUAL ARCHITECTURE

The DITSCAP-AT conceptual architecture consists of three modules; they are the DITSCAP Process-driven Workflow, Requirements Repository and Multi-strategy Machine Discovery, as shown in Figure 1.

Process-driven Workflow. This module consists of a set of process components derived from the activities and tasks outlined in the DITSCAP Application Manual (DoD 8510.1-M 2000), organized in a way that systematically guides the user through DITSCAP as well as generates an SSAA. The process-driven workflow is logically partitioned into various process components (P_1, P_2...P_n), as shown in Figure 1, based on the homogeneous grouping of activities derived from a comprehensive analyses of DITSCAP using activity diagrams. Depending on the tasks contained in each process component, wizard-based interfaces provide well-designed questionnaires/forms for DITSCAP users to fill out necessary parts of the SSAA, employing checkboxes, radio buttons and drop-down menus, in order to gather and establish well-defined metrics and measures that are amenable to automated analysis.

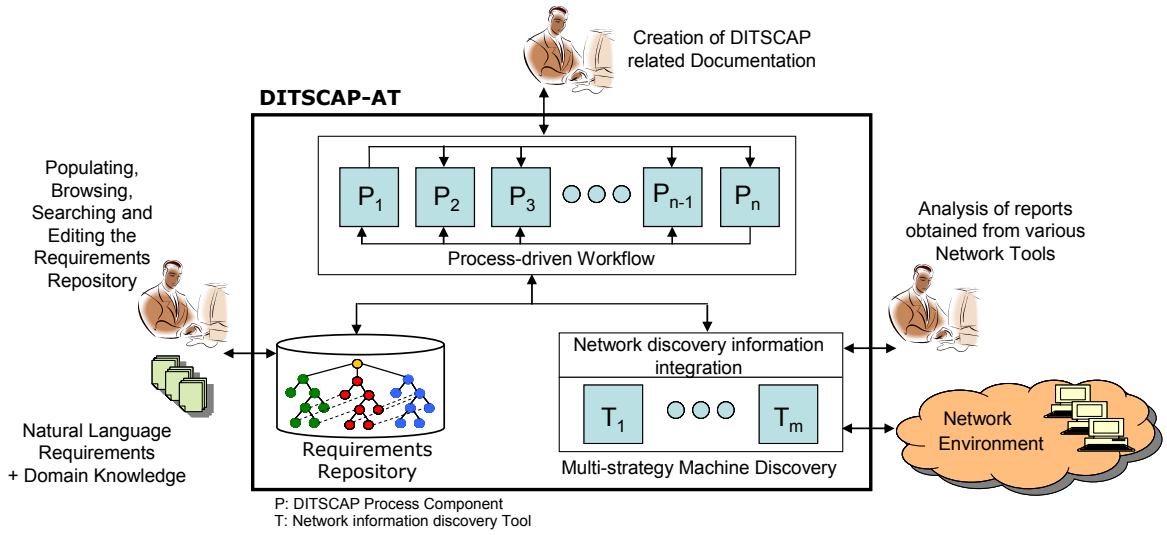

Figure 1. DITSCAP-AT Conceptual Architecture

Requirements Repository. It is a specialized module built on top of the tool support for Ontology-based Active Requirements Engineering. The theoretical background along with the methods and features of this novel framework for requirements engineering are discussed in more detail in the next section. The requirements repository will provide utilities to support representation of requirements, meta-knowledge creation, and ability to query pre-classified and categorized information structures with other browsing and inferencing functionalities. More specifically, the requirements repository is built upon the GENeric Object Model (GenOM) toolkit (Lee et al. 2004b), an integrated development environment for ontological engineering (Swartout et al. 1999) processes with functionalities to access and visualize associated knowledge-bases. A self explanatory conceptual architecture of GenOM is illustrated in Figure 2.

Multi-Strategy Machine Discovery. This module consists of a set of network monitoring & detection tools. We conducted a survey of network discovery & monitoring tools based on the following dimensions: 1) ability to detect hardware, software and firmware installed on network nodes; 2) analyze configuration of popular servers, detect various services provided by the network; and 3) analyze configuration of popular network security services and perform vulnerability assessment. Based on this survey we establish that such network related information can be discovered by leveraging features of several different tools and scripting languages. Therefore, we employ multi-strategy machine discovery techniques using a collection

of network discovery tools and network scripting languages, and then 'fuse' the discovered results using a network meta-knowledge representation in the requirements repository.

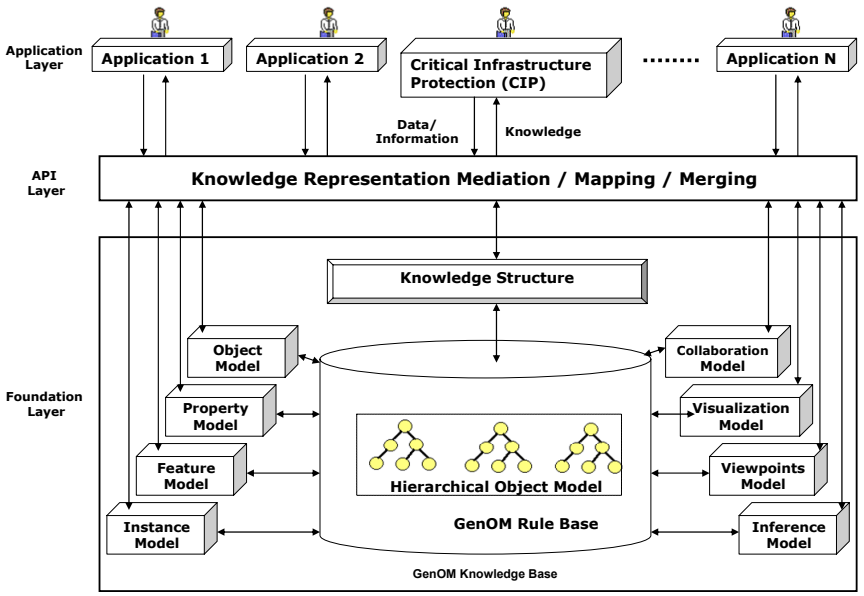

Figure 2. GenOM Conceptual Architecture

In the following sections we introduce the Ontology-based Active Requirements Engineering framework and other related research topics discussing their role in providing a structured and comprehensive IA engineering methodology for IA-aware critical infrastructure protection.

Ontology-based Active Requirements Engineering (Onto-ActRE)

To assist autonomous agent interactions, use of machine understandable ontologies created as a result of the requirements engineering process has been pointed out in (Brietman et al. 2003). We take a step further, and propose the Ontology-based Active Requirements Engineering (Onto-ActRE [onto-æktər]) framework which uses ontological engineering processes as its primary method of modeling system requirements, its operational environment and their interdependencies at various levels of abstractions from several dimensions. The GenOM toolkit, introduced in the previous section, is aimed at providing a complete tool support for the Onto-ActRE framework.

Traditionally the software developers have restricted their focus only to the software system attributes, but the software system itself is embedded within an environment that caters to the real world goals of the associated business and organization. Therefore, it is inevitable for the success of a software system to capture the relationships of system attributes with the environment and the nexus of causal chains that exist to satisfy the real world goals of the associated business and organization and then utilize this information for driving the software engineering process. This concept is even more relevant for secure systems engineering processes as security is not something that can be operationalized as a single module but rather it is the "emergent feature" of the software system, working as a whole, under a certain configuration with various technical and non-technical factors and their relationships in the given environment. Therefore, we contend that an integrated and comprehensive requirements

engineering framework that adopts a system's perspective for enabling secure systems engineering practices is inevitable to successfully exercise the DITSCAP.

To address these needs of DITSCAP, the product of Onto-ActRE is a problem domain ontology that provides the definition of a common language and understanding through the application domain concepts, properties and relationships between them in the universe of discourse. It provides the necessary means to understand and evaluate the effects of system functions and constraints in light of the concepts, properties and relationships that exist in the application domain. Onto-ActRE is an active requirement engineering framework as it transforms static record keeping requirements repositories to active ones that link to each other from different perspectives. Such active repositories are useful to attain automation as well as provide necessary means to better understand and enforce the applicable requirements in a particular application domain. An inherent benefit of having a problem domain ontology is the structured & comprehensive view that helps to understand and enforce secure software engineering practices. It also provides a systematic way to predict, plan and justify various aspects of secure software design and development. Furthermore, to address the specific needs of an organization in an ever evolving environment; Onto-ActRE can support evolutionary changes in requirements and environmental conditions by dynamically driving software configurations via the concepts, properties and relationships represented in the active requirements repositories.

Following Onto-ActRE, the DITSCAP problem domain ontology will specifically include a requirements hierarchy constructed by carefully extracting requirements from DITSCAP-oriented directives, security requisites and policies, a risk assessment taxonomy including leaf node questionnaires that have pre-defined answers with associated weights and priorities, meta-knowledge about information learned from network discovery/monitoring tools, overall DITSCAP process aspect knowledge that includes C&A goals/objectives, and representations of interdependencies between elements in the ontology.

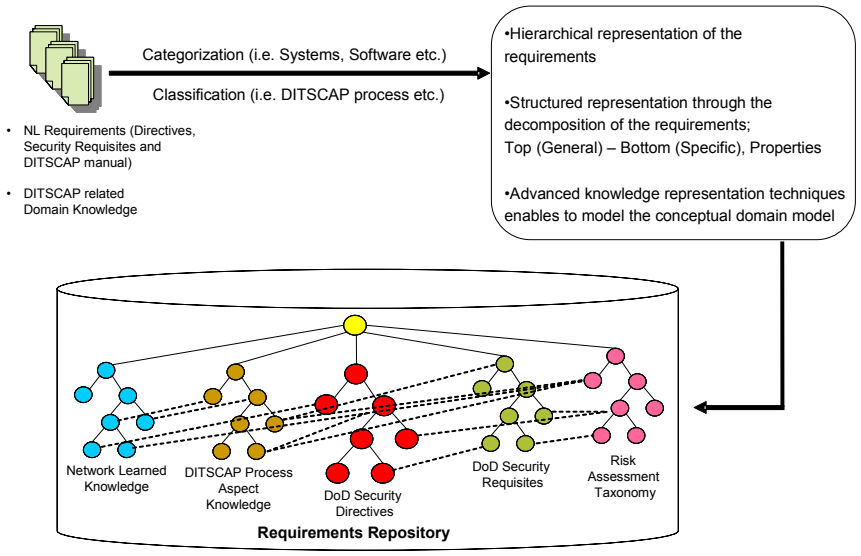

Figure 3. Onto-ActRE Methods and Features for DITSCAP-AT

The Onto-ActRE methods and features for deriving the above mentioned problem domain ontology for DITSCAP-AT is described in Figure 3. It specifically involves categorization and classification of natural language requirements expressed in DoD directives/security requisites

and DITSCAP related domain knowledge. This step creates hierarchical representation of requirements with top level nodes representing very general categories which are successively decomposed into more specific requirements in the leaf nodes. In this step, advanced knowledge representation techniques supported by GenOM enable modeling of DITSCAP related domain knowledge as well. Domain models obtained through this process reside in the requirements repository which provides necessary functionalities to browse, edit and query them.

Risk and Threat Assessment through Risk Assessment Taxonomy

Following the concepts put forth in Onto-ActRE and to satisfy the goals of the DITSCAP, we create a risk assessment taxonomy which aggregates a broad spectrum of all possible categories and classification of risk related information. The goals of risk assessment expressed in the higher non-leaf nodes of this taxonomy can be achieved using specific questionnaires criteria addressed in the leaf nodes. An example risk assessment taxonomy is shown in Figure 4. Such a risk assessment taxonomy provides a structured and comprehensive view of various risk categories associated with the system. Based on the canonical definition of risk assessment, the risk assessment taxonomy consists of concepts such as threats, vulnerabilities, countermeasures and other categories related to mission impact assessment in its higher level non-leaf nodes. Each non-leaf node is then further decomposed into more specific categories as shown in Figure 4. We identify various risk categorizations for DITSCAP-AT based on the National IA Glossary (CNSS-4009 2003) as well as other sources such as the DITSCAP Application Manual (DoD 8510.1-M 2000) and Minimal Security Checklists (DoD 8510.1-M 2000).

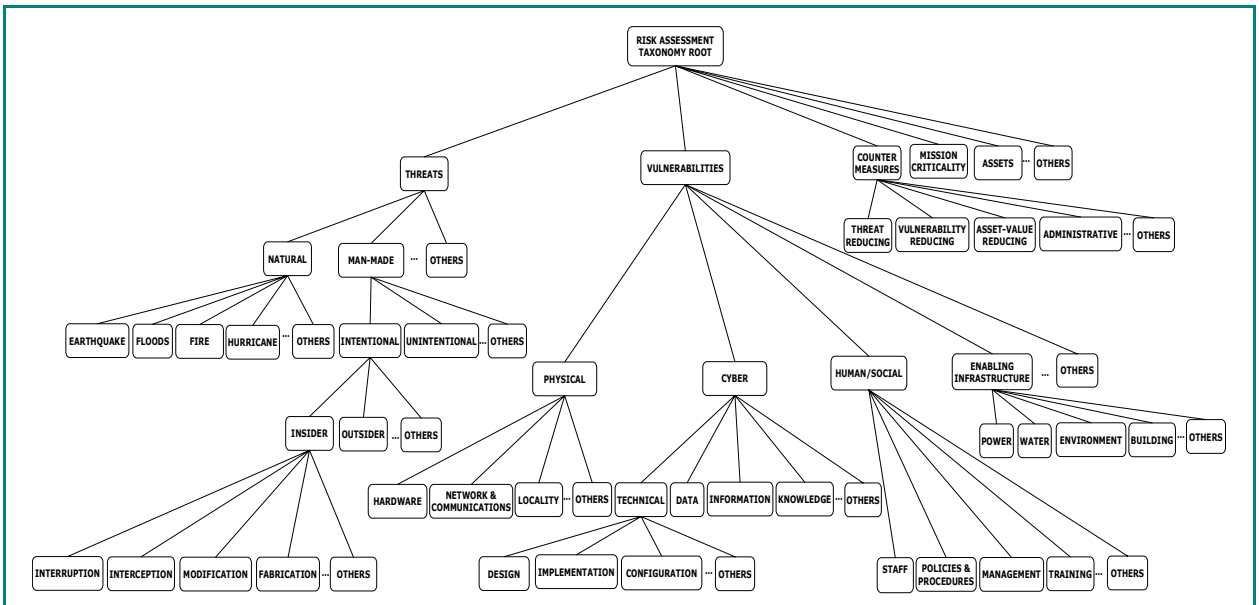

Figure 4. Example Risk Assessment Taxonomy

A predictable and quantitative risk analysis can be carried out using weights and priorities assigned to pre-classified answers for specific questions/criteria in the leaf nodes that are aimed at satisfying the goals of the higher non-leaf nodes. Such leaf node questionnaires including predefined answers with weights are shown in Figure 5. The answers can be elicited from a variety of sources such as DITSCAP-AT users, machine discovery information, or other sources. Once the leaf nodes are populated with answers for the leaf node questionnaires, complex

heuristics and formulas can be applied to calculate the final risk score. The risk calculation algorithms can take advantage of the advanced object oriented ontology representations supported by the requirements repository to represent the risk taxonomy and the relationships between its categories and other models that reside in the DITSCAP problem domain ontology as discussed in the previous section. This enables further articulation of other critical weakest points in the critical infrastructures through the analysis of links and dependencies.

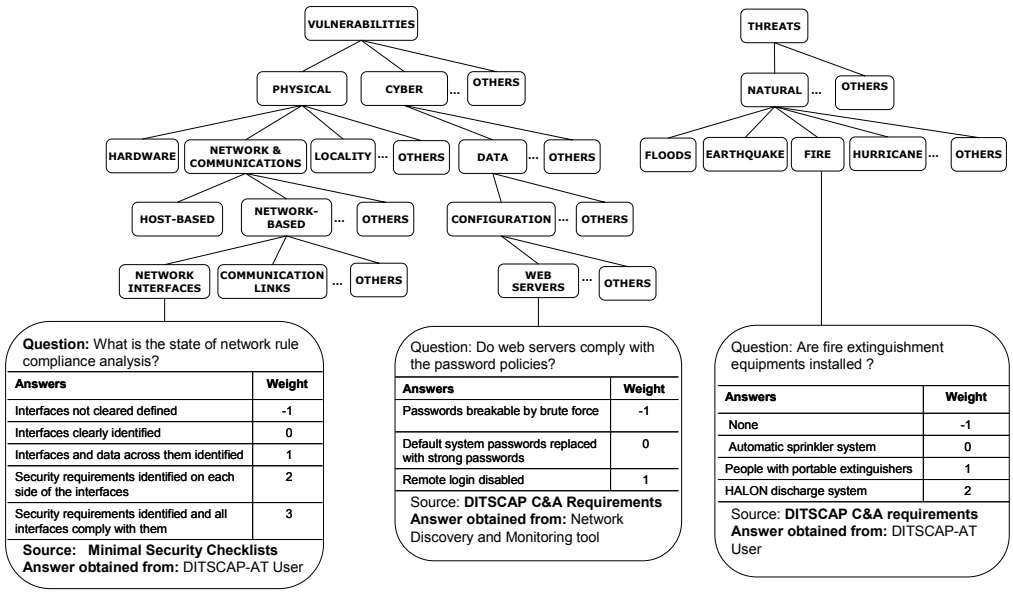

Figure 5. Example Leaf-node Questions in the Risk Assessment Taxonomy

Multi-dimensional Link Analysis

The root of Multi-Dimensional Link Analysis (MDLA) lies in the concept of the viewpoints model introduced by the PVRD methodology (Lee et al. 2004a). Lee suggests that "*Individual pieces of information finally become valuable knowledge when they establish 'links' with each other from various aspects/dimensions based on a certain set of goals*". Following this paradigm, MDLA can be carried out from different dimensions such as user criteria, viewpoints, system goals, business/mission requirements, regulatory requirements, specific operational concepts, and risk categories based on the DITSCAP C&A goals which can help understand various interdependencies between DITSCAP-oriented requirements, facilitating their interpretation and enforcement. The Onto-ActRE framework fosters such analysis due to the inherent properties of an active repository that links various representations that reside in the requirements repository from different perspectives and dimensions. Such analysis can also reveal the "emergent" or "missing" requirements that weren't easy to be identified at the onset of the C&A process. Figure 6 provides a conceptual overview of Multi-dimensional Link Analysis from different dimensions such as the requirements hierarchy, the viewpoints hierarchy and the risk assessment taxonomy based on the user criteria and network learned information captured in the leaf nodes of these hierarchies.

A conceptual overview of MDLA is shown in Figure 6. The categorized requirements hierarchy, as shown in Figure 6, is constructed by carefully extracting requirements from DITSCAP-oriented directives, security requisites and policies by following the methods and

features of the Onto-ActRE framework. Furthermore, requirements engineering involves the capture and analysis of ideas, perspectives and relationships at varying levels of detail and they are interpreted differently from different viewpoints. Such factors need to be accounted for to analyze the effect of imposing the DITSCAP-oriented requirements in a particular application domain. Considering this and following the concepts put forth in (Sommerville et al. 1997) (Lee et al. 2004a), we use the viewpoints hierarchy as a natural way to organize and structure the diversity of factors associated with requirements. The higher level non-leaf nodes in the viewpoints hierarchy, as shown in Figure 6, specifically consists of viewpoints, such as organizational viewpoints, which map to very general requirements in the requirements hierarchy. The lower level leaf nodes representing viewpoints such as those of system stakeholders are related to specific requirements in the leaf nodes of the requirements hierarchy. In the DITSCAP domain, the viewpoints are extracted from the security requirements by identifying the associated stakeholders and services, implicit or explicit. The user criteria related to security requirements can be further analyzed based on the dependent risk categories from the risk assessment taxonomy that express their testability in the form of specific criteria in which the level of compliance can be measured.

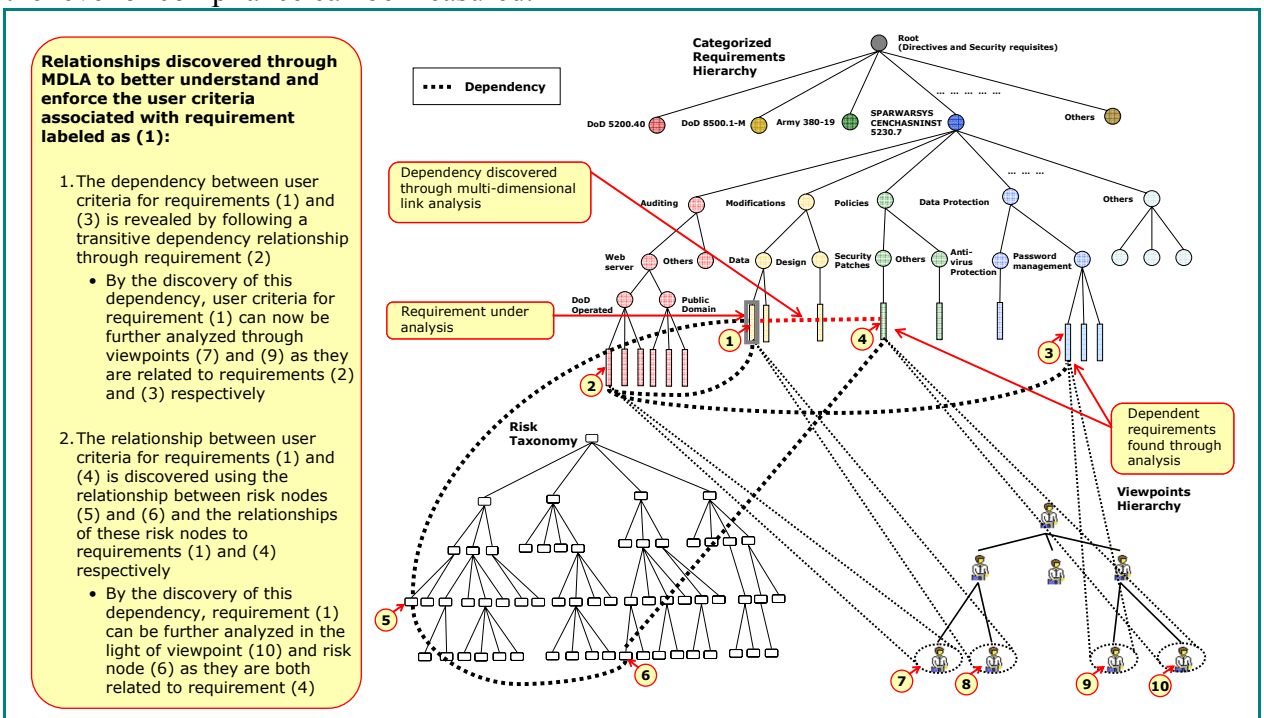

Figure 6. Conceptual overview of Multi-dimensional Link Analysis

To populate the several models that are a part of the DITSCAP automation framework with user criteria, we have designed several core mock interfaces for DITSCAP-AT to realize a complete course of actions for gathering and analyzing the required information. Such mock interfaces provide a thorough understanding of the important aspects of DITSCAP-AT user interaction and offer valuable insight and assurance in realizing the theoretical aspects of Onto-ActRE models and methods used for DITSCAP automation. In the next section, we present a few key mock interfaces of DITSCAP-AT created for elicitation of its design requirements.

Mock Interface Prototypes: A Proof-of-Concept System

We adopt an exploratory prototyping approach (Lichter et al. 1993) to elicit the design requirements for DITSCAP-AT. Figure 7 describes our prototyping approach. Following this approach, we iteratively produced mock interfaces that assist the user to perform all the major tasks and activities of the DITSCAP. We present two well-annotated self-explanatory mock interfaces, as shown in Figure 8 and Figure 9, to demonstrate how the research ideas presented in this paper contribute towards realizing the practical aspects of DITSCAP automation.

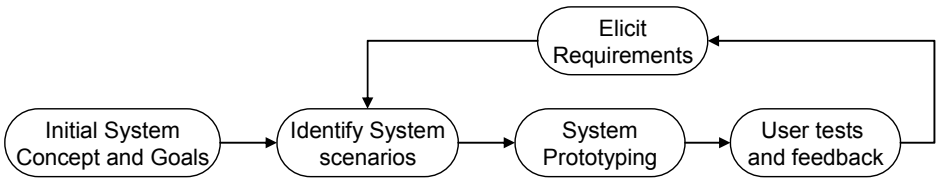

Figure 7. The Prototyping Approach

Figure 8 shows the *System Security Requirements* process component tab active in the *Select Directives* task that displays a categorized list of applicable security requirements derived from the requirements repository based on elicited user criteria and network self-discovered information. The interface provides a categorized list of applicable security requirements for the user to browse through. For each requirement selected, it can be further interpreted based on dimensions such as associated stakeholders and dependencies through which it is related to other requirements.

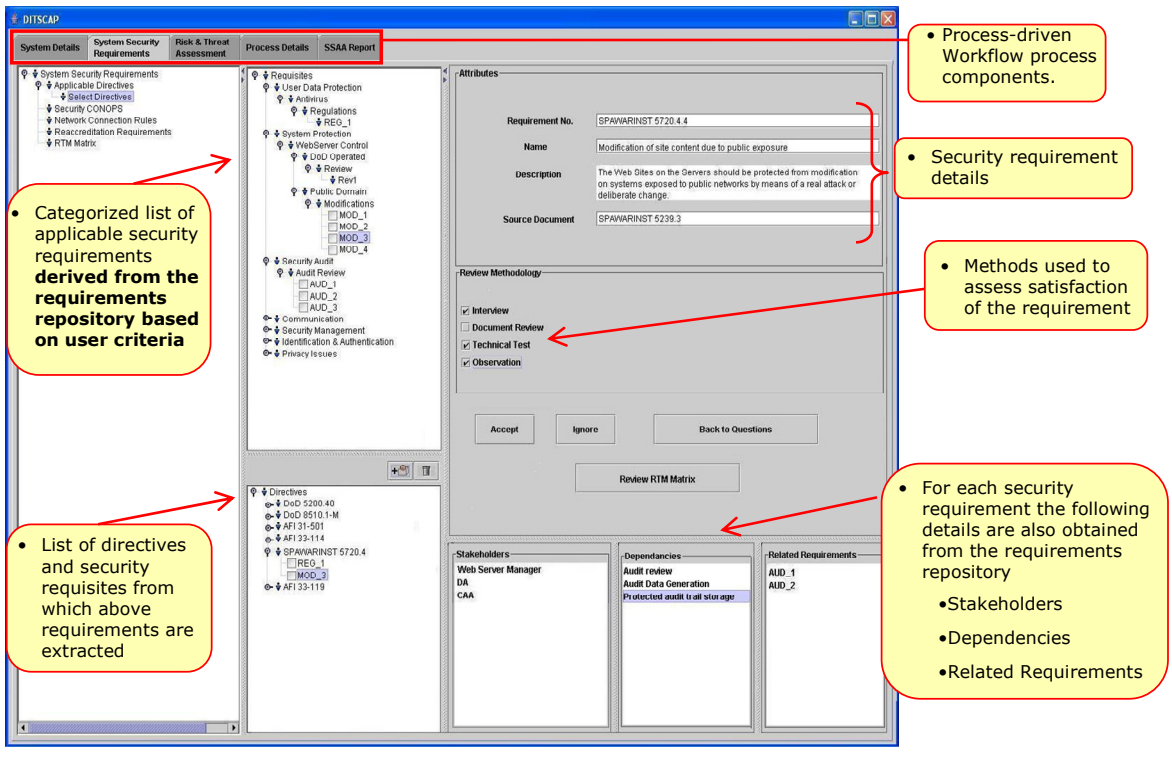

Figure 8. Applicable Security Requirements Presented in DITSCAP-AT

Figure 9 shows the *Risk and Threat Assessment* process component that presents the user with a categorized questionnaire based on the risk assessment taxonomy. The final risk score associated with the system is computed based on the weights and priorities of the answers gathered through this questionnaire as well as user criteria and network self-discovered information gathered in other stages of the DITSCAP process, to provide quantitative as well as qualitative measures for the overall operational risk of the system.

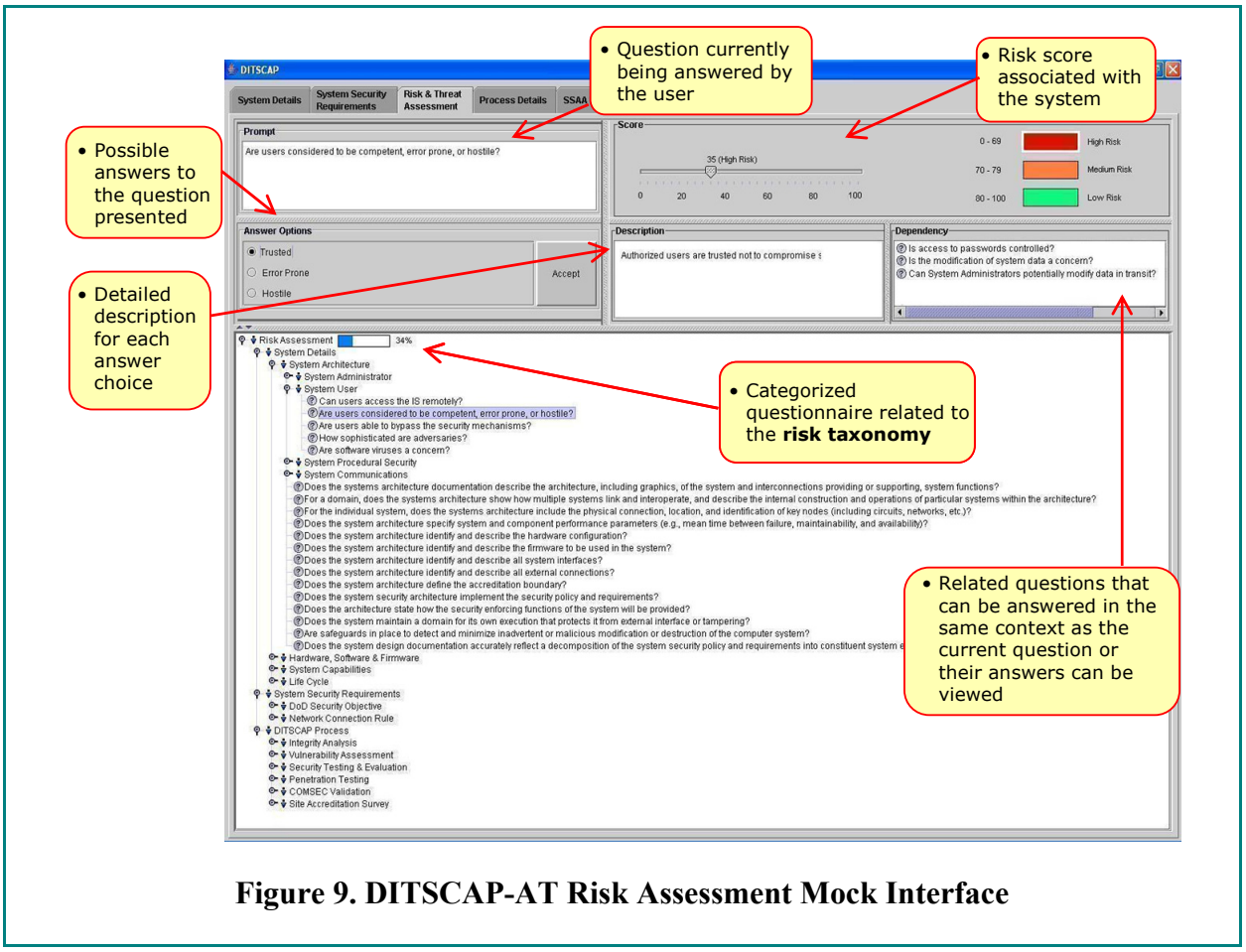

Figure 9. DITSCAP-AT Risk Assessment Mock Interface

CONCLUSION AND FUTURE WORK

In this paper, we have presented the automation study of DITSCAP, which addresses the need for a structured and comprehensive IA engineering methodology for critical information infrastructure protection. We discussed the design principles for DITSCAP-AT, the modeling techniques adopted and its supporting theoretical foundations along with demonstrable prototype interfaces. Our contributions in this effort can be outlined as follows: Firstly, we present an active approach to automating the DITSCAP, through a structured and comprehensive framework to aggregate and analyze C&A related information, using a uniform representation scheme, allowing for its reuse and evolution through all stages of the system C&A lifecycle. Secondly, we provide a conceptual overview of the DITSCAP automation tool support that assists in the process of identifying, interpreting and enforcing the multitude of DITSCAP polices and requirements along with a structured and comprehensive approach to gather and analyze risk assessment information from a broad spectrum of categories contributing to risk.

Finally, we show the ability to perform multi-dimensional link analysis that may reveal "emergent" or "missing" information pieces and further provides the assurance of a comprehensive coverage of the certification compliance space.

In our future work, we expect to use our IA engineering methodology to advance our understanding of security requirements in critical infrastructures in diverse forms at a rapid pace. It will include the identification of categories and classifications required to populate the requirements repository, create a comprehensive risk assessment taxonomy and multi-dimensional link analysis from different viewpoints, and goals using the requirements repository, and develop techniques to analyze information obtained from network discovery tools in a more rigorous manner.

Acknowledgements. This work is partially supported by the grant from the Critical Infrastructure Protection Center (CIPC), Space and Naval Warfare (SPAWAR) Systems Center, Charleston, SC. USA. (Contract # N65236-04-P7779). The authors acknowledge the support and encouragement from Scott West, John Linden, Bill Bolick, and Bill Chu. Finally, the authors thank Deepak Yavagal, Divya Muthurajan and Vikram Parekh for their contributions to this research.

REFERENCES

Breitman, K.K. and Leite, J., "Ontology as a Requirements Engineering Product." In Proceedings of the IEEE Int'l Requirements Engineering Conference, 2003.

DoD 8510.1-M, "Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) Application Manual," July 2000.

DoD Instruction 5200.40, "DoD Information Technology Security Certification and Accreditation Process (DITSCAP)", December 1997.

Bush, G.W., "Executive order on critical infrastructure protection", In Proceedings of the 12th annual conference on Computers, freedom and privacy, pp. 1-10, San Francisco, California, ACM Press, 2002.

Kimbell, J. and Walrath, M., "Life Cycle Security and DITSCAP." IANewsletter, Vol. 4(2), Spring 2001, http://iac.dtic.mil/iatac

Lee, S.W. and Rine, D.C., "Missing Requirements and Relationship Discovery through Proxy Viewpoints Model". In Studia Informatica Universalis: International Journal on Informatics, December 2004a.

Lee, S.W. and Yavagal, D., "GenOM User's Guide." Technical Report, Dept. of Software and Information Systems, UNC Charlotte, 2004b.

Lichter, H., Schneider-Hufschmidt, M., and Zullighoven, H., "Prototyping in industrial software projects-bridging the gap between theory and practice", In Proceedings of the 15th International Conference on Software Engineering, Baltimore, MD, pp. 221-229,1993.

National Information Assurance Glossary, CNSS Instruction No. 4009, National Security Agency, 2003, `http://www.nstissc.gov/Assets/pdf/4009.pdf`

Sommerville, I. and Sawyer, P., "Viewpoints: Principles, Problems and a Practical Approach to Requirements Engineering" In Annals of Software Engineering, Vol. 3, pp. 101-130, 1997.

Swartout, W. and Tate, A., "Ontologies" In Intelligent Systems, IEEE, 14 (1), pp. 18-19, 1999.

BIOGRAPHY

Dr. Seok-Won Lee is an Assistant Professor of Software and Information Systems (SIS) at the University of North Carolina at Charlotte (UNCC). His areas of specialization include software engineering with specific expertise in ontological requirements engineering, domain modeling, science of design and software evaluation research, and knowledge engineering with specific expertise in knowledge acquisition, machine learning and knowledge-based systems. Prior to joining UNC Charlotte, Dr. Lee was affiliated with Science Applications International Corporation (SAIC) and IBM TJ Watson Research Center. He is a member of the ACM, IEEE, IEEE Computer Society and AAAI.

Dr. Gail-Joon Ahn is an Assistant Professor at UNCC and a key representative of a National Center of Academic Excellence in Information Assurance Education designated by National Security Agency and Department of Homeland Security. His principal research and teaching interests are in information and systems security and his research has been supported by NSF, NSA, DoD, DoE, Bank of America, Hewlett Packard, and RWJ Foundation. Dr. Ahn is currently an information director of ACM Special Interest Group on Security, Audit and Control and he is a recipient of Department of Energy Early Career Principal Investigator Award.

Robin Gandhi is currently pursuing a Ph.D. in Information Technology and is a research assistant with the Department of Software and Information Systems at UNC Charlotte since Spring 2003. He received his undergraduate degree in Electronics Engineering from Sardar Patel University, Gujarat, India in July 2000 and his Master of Science in Computer Science from UNC Charlotte in December 2001. He also worked as a Software Engineer in the area of GIS, 3D modeling and simulation. His research interests include requirements engineering, science of design, knowledge intensive software engineering, ontology-based object-oriented domain modeling, and computer supported cooperative work.