SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2007; **12**: 165–189
Published online 27 December 2006 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.313

Research Section

Certification Process Artifacts Defined as Measurable Units for Software Assurance

Seok-Won Lee*,†, Robin A. Gandhi and Gail-Joon Ahn
Knowledge-Intensive Software Engineering Research Group, Department of Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, NC 28223, USA

Certification and Accreditation (C&A) process artifacts for software-intensive systems are characterized by the metrics and measures required to be produced from their units of analysis for assessing system behaviour. Software-intensive systems are complex clusters of closely interdependent system of systems that include underlying software, systems, people, processes, and operational environments. Naturally, such systems require carefully designed C&A artifacts that consider metrics and measures from multiple dimensions at different levels of abstraction in the Universe of Discourse (UoD) in order to understand, predict, and control their emergent behaviour. Hence, C&A artifacts defined as measurable units for software assurance should be the result of an aggregated reasoning of evidences from various dimensions, while maintaining traceability and alignment to real world goals/objectives in all stages of the system lifecycle. To address these research objectives, we present a novel integration framework that promotes cohesion and traceability among metrics and measures from multiple dimensions in the problem domain on the basis of the definition of a common language. By applying our framework to automate the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP), we also motivate the design principles and modelling techniques necessary to generalize a course of action to conduct C&A processes with appropriate tool support for software-intensive systems. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: software-intensive systems; requirements engineering; certification and accreditation; metrics and measures; ontological engineering; risk assessment

1. INTRODUCTION

Software-intensive systems provide various critical services related to computing, communications, and information processing in the government, private, and defense sectors. For such systems, Certification and Accreditation (C&A) processes play an important role in promoting trust in their acquisition and subsequent operation in the organization to achieve their real world goals/objectives. C&A processes provide a management infrastructure for carefully designing the engineering activities that affect all phases of the software-intensive system lifecycles. Depending on their focus, C&A processes evaluate

* Correspondence to: Seok-Won Lee, Knowledge-Intensive Software Engineering Research Group, Department of Software and Information Systems, The University of North Carolina at Charlotte, Charlotte, NC 28223, USA
† E-mail: seoklee@uncc.edu

their target entities (Organization, Software Process, Software Product or Practices) on the basis of qualities that satisfy the metrics and measures for the procurement of certification status. Essentially, the artifacts produced from C&A process activities and their units of analysis are characterized by the metrics and measures required for assessing target system behaviour. However, the increasing complexity of software-intensive systems with a large amount of software, several constituent systems, a high degree of connectivity, and diversity of their socio-technical operational environments pose significant challenges to the current practices and techniques for understanding and evaluating the related C&A process artifacts. As a result, despite enormous efforts and resources being currently spent on C&A processes (FISMA 2005), their effectiveness in the real world is only limited (Davis 2005).

Software-intensive systems are inherently complex clusters of closely interdependent *systems of systems* operating in diverse socio-technical environments. Their complexity arises from the interdependencies among themselves as well as with their operational environment to satisfy the required behaviour. Diverse socio-technical environments contribute to multiple viewpoints that introduce different semantics and levels of abstraction in specifying the services required from these systems. These characteristics make it hard to understand, predict, and control their emergent behaviour that can provide unanticipated benefits or deviate from the required capabilities, which is potentially the biggest risk factor eminent throughout their lifecycle. Software-intensive systems encompass the underlying software, systems, people, processes and operational environments, which introduce metrics and measures from multiple dimensions at different levels of abstraction in the Universe of Discourse (UoD) to understand and explain their behaviour. Naturally the related C&A process artifacts should be based on metrics and measures that are objective, transparent, and traceable across various dimensions in the problem domain to assure reliable software behaviour. Therefore, C&A artifacts, defined as measurable units for software assurance, should be the result of an aggregated reasoning of evidences from various dimensions while maintaining traceability and alignment to real world goals/objectives in all stages of the system lifecycle. To address these research objectives, we present a novel integration framework to elicit, represent, model and analyze the diversity of metrics and measures associated with software-intensive systems in socio-technical environments. Within our framework, we combine the strengths of multiple complementary requirements engineering (RE) modelling techniques in a unifying ontological knowledge engineering process to provide the definition of a *common language* (Lee and Gandhi 2005b). On the basis of a uniform representation format, the definition of a common language promotes cohesion and traceability among metrics and measures from multiple dimensions in the problem domain through shared evidences generated from the synergy between the application domain concepts, properties and their interdependencies.

In our research, we focus on software-intensive systems included in the Defense Information Infrastructure (DII). The DII connects the Department of Defense (DoD) mission support, command and control, and intelligence computers and users through voice, data, imagery, video, and multimedia services, and provides information processing and value-added services. For such a critical infrastructure, the DoD requires all systems in the DII to be certified and accredited following the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) (Department of Defense Instruction DoDI 5200.40 1997). The DITSCAP application manual (DoD 8510.1-M 2000) outlines well-defined tasks and activities that identify the units of analysis and the criteria to assess the security of software-intensive systems throughout their lifecycle from various perspectives. DITSCAP provides a management infrastructure for gathering metrics and measures from multiple dimensions, which can be used to guide as well as assess secure software engineering activities. However, the task of collectively understanding and analyzing these metrics and measures can be quite overwhelming owing to the long and exhaustive process of information gathering, documentation, and analysis as suggested by a multitude of DITSCAP-enforced directives applicable to software-intensive systems prevalent within the DoD. To cope with these issues within the DITSCAP problem domain, we apply the models and methods of our generic framework to provide the necessary language, methods, models, and tools to support its automation (Lee, Gandhi and

DOI: 10.1002/spip

Ahn 2005a). We outline a stepwise methodology to capture, model and analyze DITSCAP-enforced security requirements, related domain knowledge, user/system criteria, and their interdependencies across several dimensions and levels of abstractions in the DITSCAP domain. The resulting DITSCAP Problem Domain Ontology (PDO) from this effort offers the opportunity for the analysis and assurance of a comprehensive coverage of the problem domain metrics and measures by actively assisting the process of discovering missing, conflicting, and interdependent pieces of information. Throughout the article, examples derived from our case study motivate the feasibility and appropriateness of our approach in achieving the objectives of DITSCAP automation.

This article is organized as follows. In Section 2, we provide the background information necessary to understand DITSCAP, with the motivations and objectives behind its automation. We also elaborate on the theoretical foundations that guide our efforts for DITSCAP automation. Section 3 outlines a stepwise methodology to capture, model, and analyze DITSCAP-enforced security requirements, related domain knowledge, user/system criteria, and their interdependencies in order to understand and organize DITSCAP problem domain concepts. Section 4 elaborates on various models developed in the DITSCAP problem domain, followed by a discussion on the elicitation and representation of metrics and measures identified from DITSCAP-related guidance documents and available expertise in the subject matter in Section 5. In Section 6, we introduce the concept of multi-dimensional link analysis (MDLA), which motivates the feasibility and appropriateness of our approach in achieving the objectives of DITSCAP automation through examples derived from our case study. In Section 7 we discuss related work and summarize our contributions and future work in Section 8.

2. BACKGROUND

2.1. DITSCAP

The DITSCAP is defined as the standard DoD process for identifying information security requirements, providing security solutions, and managing security activities of information systems (DoDI 5200.40 1997). The DITSCAP achieves its goals by prescribing a standard DoD-wide process of establishing a management infrastructure that leads to the acquisition and maintenance of C&A for secure operations of information systems. In order to establish the extent to which a particular design and implementation meets a set of specified security requirements (DoDI 5200.40 1997) in the context of information systems, DITSCAP defines certification as a comprehensive evaluation of the technical and non-technical security features of an information system and other safeguards made in support of the accreditation process. Following the certification activities, the accreditation statement is an approval by a Designated Approving Authority (DAA) to operate the information system in a particular security mode using a prescribed set of safeguards at an acceptable level of risk. It should be noted that the relationship of the C&A process with the information systems is not something that is established once and got over with; rather, it is a life-time commitment (Kimbell and Walrath 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 rigour adopted for the C&A process depends on the certification level chosen for the information system among the four levels available, which are (i) Minimal security checklist, (ii) minimum analysis, (iii) detailed analysis, and (iv) extensive analysis. The DITSCAP application manual describes these certification levels and their phases with associated activities in detail.

The key roles of the DITSCAP are that of the Program Manager, DAA, Certifier and User Representatives who 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' should be defined and agreed upon as per the key roles, which is documented as a System Security Authorization Agreement (SSAA). DITSCAP follows a single-document approach and records all artifacts produced through C&A activities into the SSAA. The SSAA is especially important because it is used throughout the DITSCAP to guide actions, document decisions, specify Information Assurance (IA) requirements, document certification tailoring and level-of-effort, identify potential solutions, and maintain operational systems security (DoDI 5200.40 1997). The SSAA forms the baseline security configuration document for the

DOI: 10.1002/spip

target system. It records the outcome of tasks and activities in each phase of the DITSCAP, which produce several measures by inspecting and analyzing their units of analysis and assessing them on the basis of metrics considered for the procurement of certification status. Sections of the SSAA are representative of the artifacts that are generated from the DITSCAP. Figure 1 depicts the DITSCAP process components, their units of analysis, and related metrics and measures documented in the SSAA. These C&A process artifacts act as measurable units to assess and guide secure software engineering activities throughout the system lifecycle.

2.1.1. Motivation for DITSCAP Automation and Its Objectives

DITSCAP supports a rich environment for the development of metrics and measures from several dimensions, such as security requirements, process, organization, cost, time, data sensitivity, user clearance, system capabilities, development, deployment, operation, maintenance, architecture, inventory, impact, and several others that can be used to guide as well as assess secure software engineering activities. However, the task of understanding and analyzing such metrics and measures in practice is complicated because of the global consequences of non-functional DITSCAP-enforced security requirements with abstract specifications for maintaining applicability over a variety of target systems, operational environments, and situations. In addition, the DITSCAP itself is quite overwhelming owing to the long and exhaustive process of information gathering, documentation, and analysis, as suggested by a multitude of DITSCAP-enforced guidance documents. These issues are further complicated by the complexity and diversity of software-intensive systems and socio-technical operational environments prevalent within the DoD. All these factors together bring about a strong and urgent need for a well-defined and comprehensive framework for DITSCAP automation in order to gain a high level of trustworthiness expected from the services of the software-intensive information systems within the DII.

The objective of DITSCAP automation is to provide a framework within which DITSCAP artifacts are produced on the basis of well-defined metrics and measures gathered from multiple dimensions that are closely associated with the way we understand and interpret them in the real world. This approach is essential because a single-dimensional metric (for example, just the technical attributes of the system) cannot

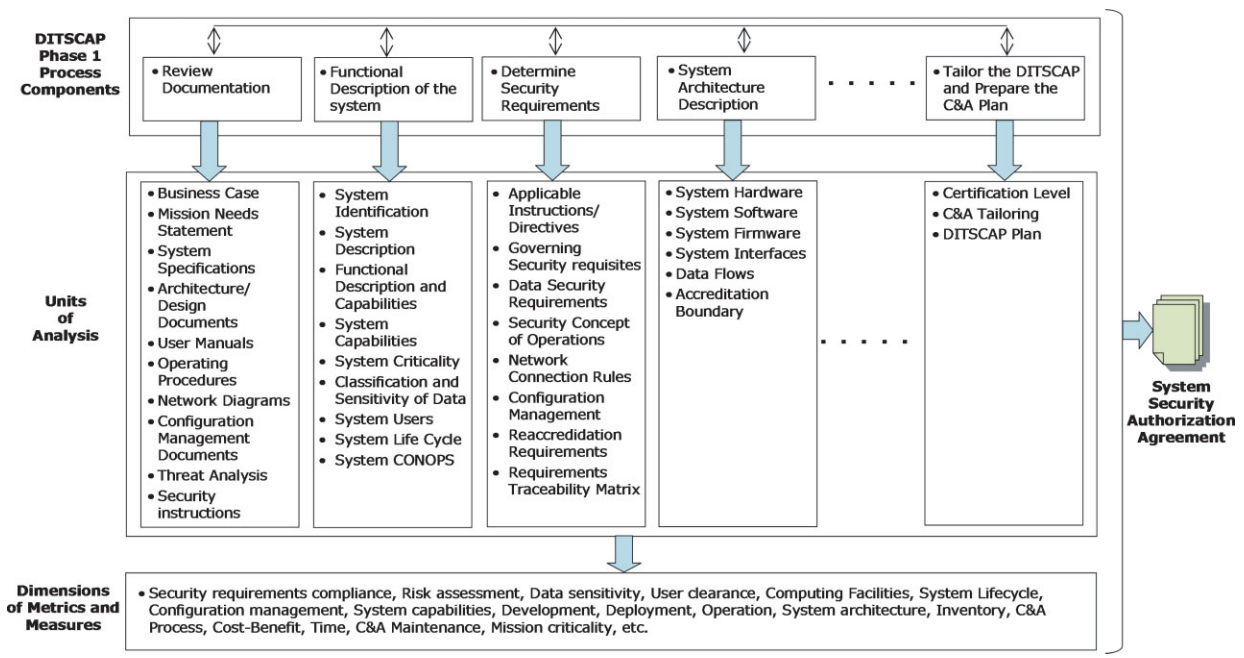


Figure 1. The C&A process components as measurable units to guide secure software engineering activities

DOI: 10.1002/spip

possibly capture the range of properties that need to be assessed for predicting the multifaceted behaviour of software-intensive systems. Thus, metrics and measures collected throughout the DITSCAP should be processed on the basis of an understanding reflected from multiple dimensions of the problem domain along with support for their interactions. To focus efforts in this direction, in the following subsection we provide a brief overview of a novel integration framework to elicit, represent, model, and analyze the diversity of metrics and measures associated with software-intensive systems.

2.2. The Ontology-based Active Requirements Engineering (Onto-ActRE) Framework

Traditionally, software engineering practices related to the procurement, development, maintenance, and usage of software-intensive systems have focussed only on the technical attributes of the software system, but the software system itself is embedded within an environment that caters to the real world goals of the associated users, businesses, and organizations. The need to understand this domain, the interface between the 'machine' and 'environment', and the interdependencies between them have been well documented in the RE literature (Jackson 1997) and realized by the community (Offen 2002). This concept is even more relevant for software-intensive systems because their capabilities rely heavily on the emergent behaviour resulting from the collective influences of individual systems on each other as well as their interdependencies with the operational environment. Therefore, an integrated and comprehensive framework that adopts a system's perspective by encompassing multiple dimensions of the problem domain is inevitable in order to practice software engineering for software-intensive systems. Figure 2 provides a conceptual overview of the Ontology-based Active Requirements Engineering (Onto-ActRE) framework (Lee and Gandhi 2005a) that takes a step in this direction. The Onto-ActRE framework provides the means to understand and evaluate the effects of system functions and constraints in the light of the concepts, properties and their relationships that exist in the UoD from the perspectives of the real world goals of the users, organization, operational environment, and business/mission requirements. Furthermore, to organize the diversity of factors associated with a software system, it is necessary to consider different perspectives and viewpoints from different stakeholders.

The Onto-ActRE framework, through its theoretical foundations as a mixed-initiative approach, offers flexibility to gather metrics and measures

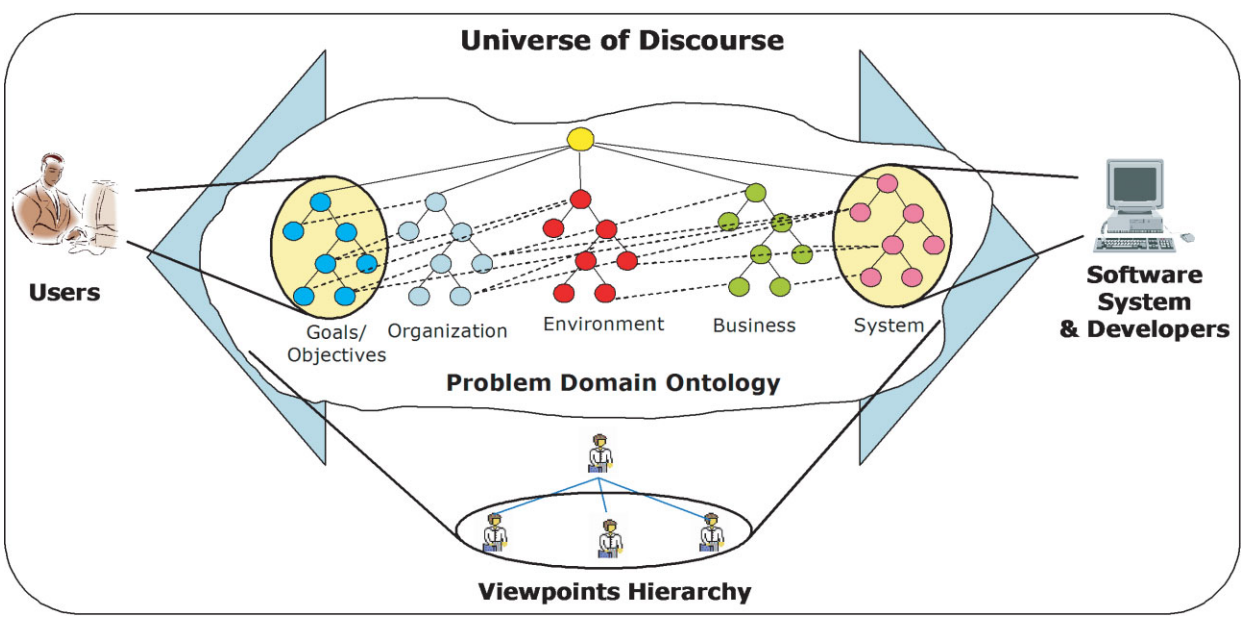


Figure 2. The Onto-ActRE framework conceptual overview

DOI: 10.1002/spip

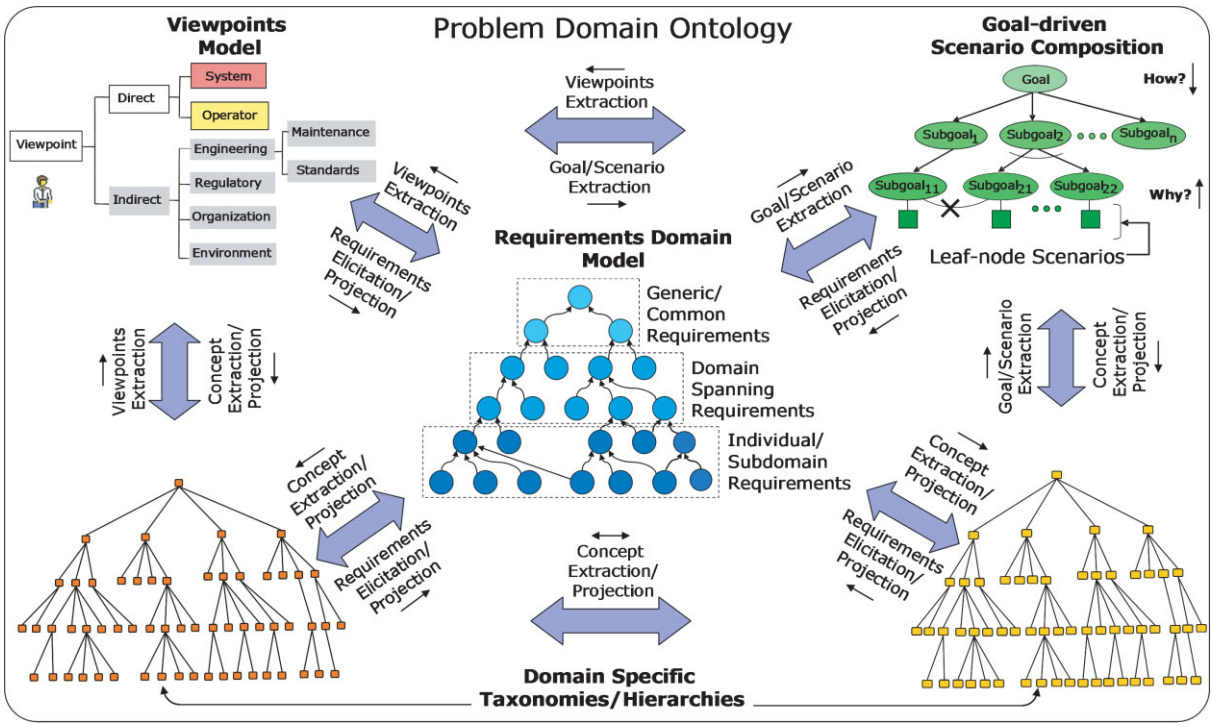


Figure 3. The Onto-ActRE problem domain ontology

from multiple dimensions necessary to explain the emergent behaviour of software-intensive systems. Within the framework, ontological engineering processes are the primary method of representing and analyzing metrics and measures gathered from the problem domain based on complementary modelling techniques with different semantics and levels of abstraction. The Onto-ActRE framework includes contributions from popular and well-studied RE modelling techniques based on the notions of goals, viewpoints, scenarios, and their combinations. More specifically, the Onto-ActRE framework includes models and methods for (i) Goal-driven scenario composition, (ii) requirements domain model, (iii) viewpoints hierarchy, and (iv) other domain-specific taxonomies to hierarchically organize the application domain concepts, properties, and their relationships. Figure 3 depicts several possible models and methods of the Onto-ActRE framework and the synergistic interactions between them. We elaborate on the Onto-ActRE models and methods in the context of DITSCAP automation in Section 4.

The Onto-ActRE framework provides the definition of a common language through the creation of PDO from the UoD in which building a software system is the problem frame. The PDO is a machine understandable and hierarchical representation that is engineered using object-oriented ontological domain modelling techniques. The inherent benefits of such a PDO lie in the uniformity of its representation for capturing metrics and measures based on different philosophies and semantics and its traceable rationales to promote cohesion among them. The models within the PDO, based on the notions of goals, scenarios, viewpoints and other domain-specific considerations, provide well-defined metrics and measures to understand and align the behaviour of software-intensive systems from the perspectives of their real world objectives.

3. THE DITSCAP AUTOMATION

While practicing DITSCAP, one has to refer several guidance documents, such as the DITSCAP application manual, Federal Laws, DoD Policies and Implementations, Department of Navy (DoN) site/agency specific guidance, National Institute of Standards and Technology (NIST) best practices,

Softw. Process Improve. Pract., 2007; **12**: 165–189
DOI: 10.1002/spip

and several other reference directives and security requisites, to identify the applicable security requirements. Each document usually ranges from 25 to 200 pages, making it extremely difficult to comprehend their contents and the interdependencies among them, thus challenging the objectivity and repeatability of the criteria adopted for producing the corresponding C&A artifacts. The lack of traceability to the real world goals/objectives from the specific assessment criteria creates gaps between the DITSCAP standards and their interpretation and enforcement in the real world practice. These issues are further complicated by the non-functional nature of DITSCAP security requirements that impose global consequences and thus lack the convenience of a localized assessment to comprehend system behaviour. In addition, to maintain flexibility in applying the C&A process to a variety of systems, environments, and situations, DITSCAP expresses its guidance and security requirements at an abstract level. All these issues contribute to subjective interpretations, non-standard implementations, and breakdowns in a common understanding of the criteria among the collaborating stakeholders. To address these issues, the very first step should be to provide the definition of a common language and understanding between the various stakeholders in the DITSCAP domain. To promote such a common understanding, a stepwise methodology for the creation of a DITSCAP PDO is discussed in the next subsection.

3.1. A Stepwise Methodology for Creating a DITSCAP Problem Domain Ontology

As an important step towards achieving the objectives for DITSCAP automation, we define a systematic methodology for extracting and organizing concepts in the DITSCAP problem domain on the basis of the Onto-ActRE framework. The resulting DITSCAP problem domain ontology creates a common understanding among various stakeholders to promote the development of objective, traceable, justifiable, and repeatable metrics and measures from multiple dimensions. On the basis of the theoretical foundations of the Onto-ActRE framework, the metrics and measures established through this approach are not readily available as the by-product of applying the C&A process or derived from system operation but are well-designed to provide close alignment with how we understand and interpret them to accomplish our real world goals/objectives.

A stepwise methodology for creating such a DITSCAP PDO following the models and methods of the Onto-ActRE framework is shown in Figure 4. Each step in the methodology is characterized by the available inputs, tasks to be performed, techniques to be used, and their outcomes. Although the process appears to be sequential, a lot of synergistic interactions exist between its steps.

In the development phase of the DITSCAP PDO, the first step is to gather information from the DITSCAP subject matter experts as well as DITSCAP guidance documents in order to identify the problem domain concepts, properties, and their relationships using the models and methods of the Onto-ActRE framework. In the DITSCAP domain we specifically identify concepts related to security requirements, C&A process components, risk factors, stakeholders and their responsibilities, IA services, and security controls specified at different levels of abstraction in various guidance documents as well as through knowledge elicitation from subject matter experts. Currently, we have processed well over 800 pages of various DITSCAP-related regulatory documents that are a good representative set of the breadth and depth of DITSCAP. As an example for eliciting problem domain concepts, properties, and their relationships from natural-language documents, consider the security requirements excerpts shown in Figure 5. The documents in Figure 5 are organized hierarchically on the basis of the organizational structure of the DoD. From the security requirement labeled as '**1**', we identify the security requirement concept of '**Screen Individuals**' as a subconcept of '**Personnel Security**'. In addition, each concept is annotated with several properties that help to characterize the associated security requirements, such as source of the document, type of agency that enforces the requirement, stakeholders, etc. In Figure 5 we also identify the interdependencies that exist between various security requirements, for example, the '**realized_by**' relationship conveys the meaning that the security requirement labeled as '**3**' depends on the security requirement labeled as '**4**' to realize itself. Similarly, several other relationships are identified and explained in Figure 5.

The concepts identified in the first step act as input to the second step and it involves their classification

DOI: 10.1002/spip

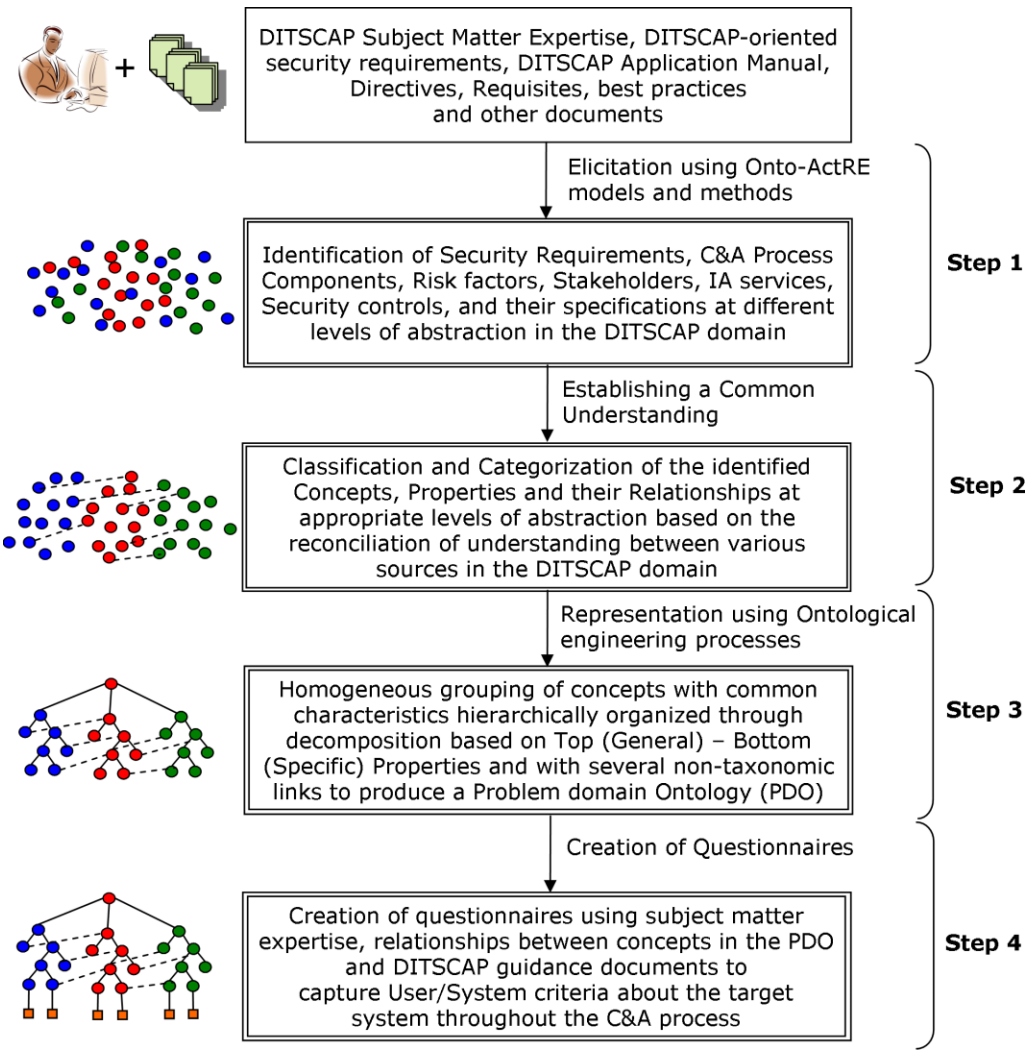
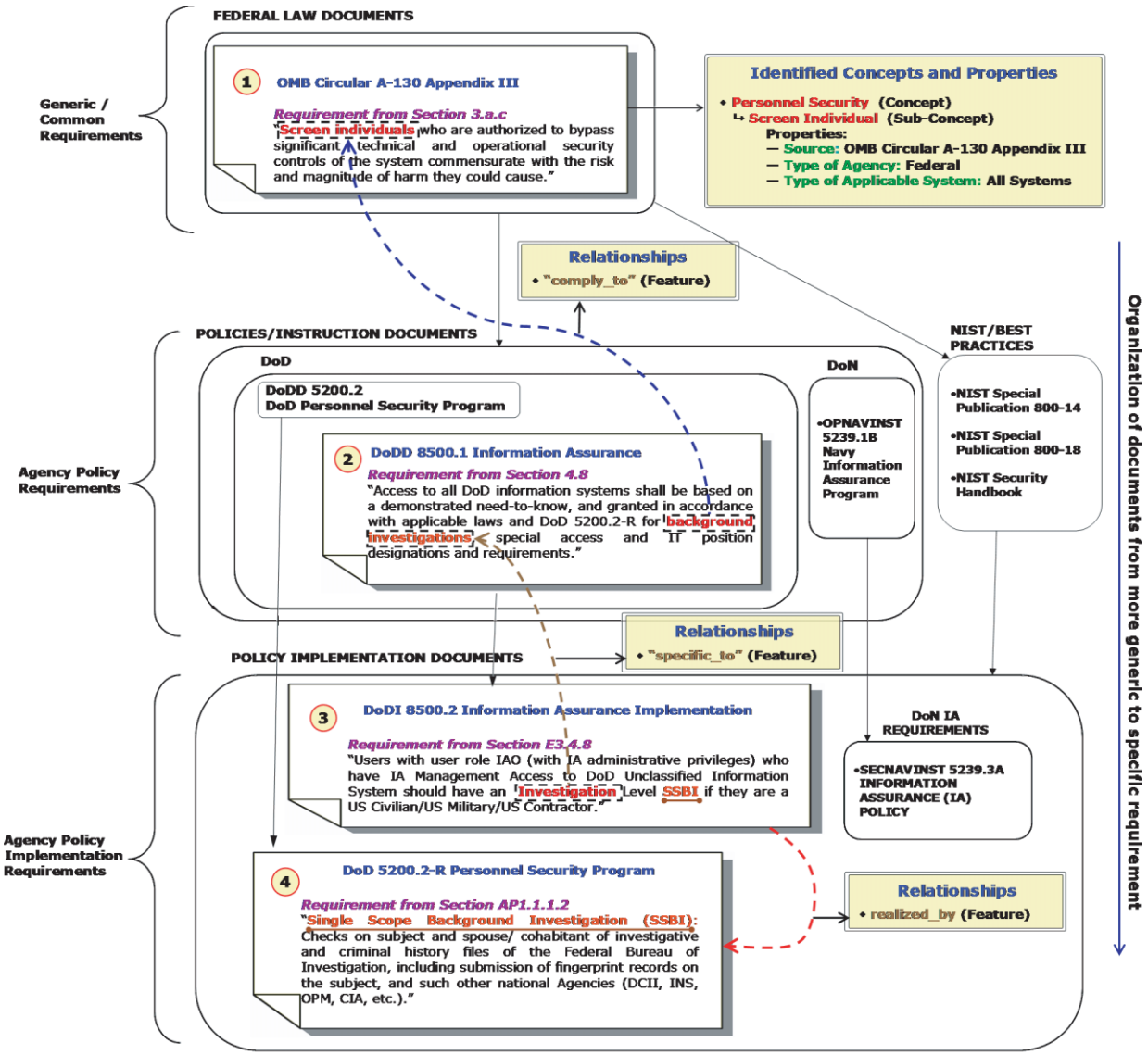


Figure 4. Process of creating the DITSCAP problem domain ontology on the basis of the Onto-ActRE framework

and categorization at appropriate levels of abstraction by establishing a common understanding on the basis of linking of information from multiple sources. Such links are identified on the basis of several factors, for example, through relationships between concepts identified from documents or subject matter experts at various levels in the organizational structure (abstract to more specific). Such relationships are also indicated in Figure 5 (through the '**comply_to**' and '**specific_to**' relationships), which depict the relationships between various security requirement concepts (generic to more specific) from various documents in the DoD organizational hierarchy. A common understanding in the problem domain is achieved by explicitly identifying the relationships between various concepts through their usage across multiple sources of information. Such relationships expose the cross-cutting nature of various concepts within as well as across information sources and promote a shared understanding of the criteria used to produce the metrics and measures required for DITSCAP artifacts through traceability to their real world objectives.

The third step involves creating homogeneous groupings of concepts with shared properties and hierarchically organizing them through decomposition from the most generic concepts to the

DOI: 10.1002/spip



Symbol	Meaning
	Refers to key concepts that are identified from Security Requirements
	"comply_to": Identifies relationships between a security requirement and other requirements it needs to comply with at different levels of abstraction in the organizational hierarchy
	"specific_to": Identifies relationships between policies enforced through requirements and their specific realization guidance in the organizational hierarchy
	"realized_by": Identifies relationships between a security requirement and other requirements that it depends on to realize itself
	Refers to keywords in requirements descriptions which helps to identify related requirements

Figure 5. Identification of concepts, properties, and their interdependencies from security requirements in the DITSCAP domain

most specific ones. Non-taxonomic relationships identified between various concepts are also represented in this step. To support the representation of such rich knowledge structures in the DITSCAP PDO, various ontological engineering processes are provided by the GENeric Object Model (GenOM)

DOI: 10.1002/spip

(Lee and Yavagal 2005) toolkit. GenOM is an integrated development environment for ontological engineering processes with functionalities to create, browse, access, query, and visualize associated knowledge-bases. It inherits the theoretical foundation of the frame representation and is compatible with the Open Knowledge Base Connectivity (OKBC) specification (Chaudhri *et al*. 1998) as well as the Web Ontology Language (OWL) representation (McGuinness and van Harmelen 2004) format. The conceptual architecture of GenOM is shown in Figure 6. The GenOM meta-language consists of *Objects, Properties*, and *Features* with semantics that effectively support knowledge acquisition and representation. GenOM *Objects* with support for single or multiple inheritances are used to model hierarchical structures that describe the concepts in a domain. GenOM *Properties* are used to describe the characteristics or attributes of *Objects* and *Features*. Finally, GenOM *Features* are used to describe the relationship or dependencies that exist between *Objects*. Once the *Objects, Properties*, and *Features* are defined, they are instantiated to represent specific *Instances* that exist in a problem domain. GenOM is also associated with an inference engine (Carroll *et al*. 2004) that supports reasoning based on the *Objects, Properties*, and *Features* and *Instances* defined in its knowledge-bases. In summary, GenOM supports object modelling in its representation, usage of objects in its application model, and ability to aggregate evidence that supports the analysis of objects' behaviours (through the associated properties and relationships between objects). GenOM's rich modelling constructs coupled with easily understandable semantics make it a good choice for the creation of a common language with participation from diverse stakeholders and experts in the UoD. Currently, the DITSCAP PDO in GenOM houses close to 20,000 modelling artifacts, which include objects, properties, features, object instances, feature instances, and rules in the knowledge base. From a DITSCAP perspective, measures such as the number of modelling artifacts used by a process component support the creation of

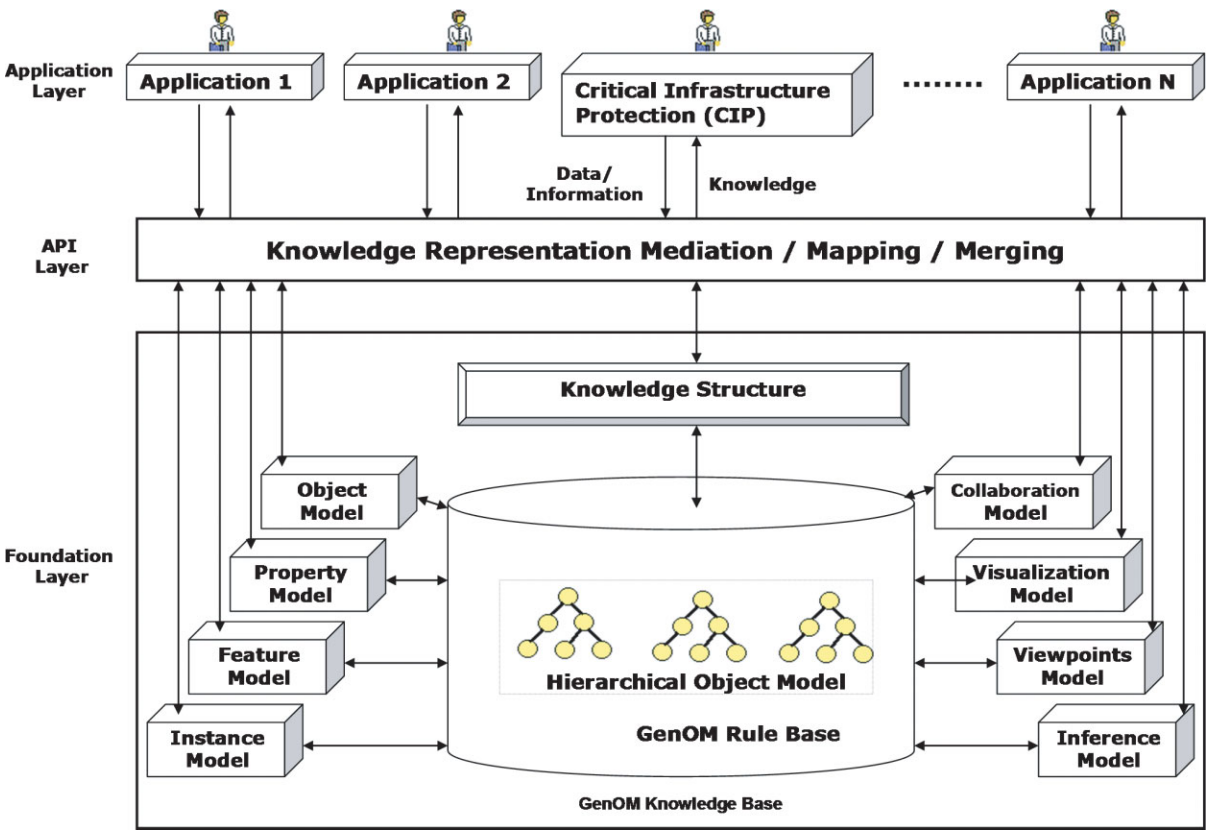


Figure 6. GenOM conceptual architecture

DOI: 10.1002/spip

metrics for complexity and coverage of the problem domain.

DITSCAP is all about carefully collecting evidence regarding the target software-intensive system on the basis of the execution of its tasks and activities for assessment purposes. To systematically assimilate such information, the fourth and final step involves the creation of questionnaires related to various concepts within the DITSCAP PDO. On the basis of the hierarchical structure of the PDO, non-leaf-node concepts can be analyzed with respect to its child nodes; however, to analyze leaf-node concepts we introduce questionnaires that provide guidance to collect appropriate user/system criteria related to DITSCAP tasks and activities. In addition, the individual questions within each questionnaire have predefined answer options that act as measures collected for the concepts they are related to. The criteria addressed by the questionnaires are established on the basis of DITSCAP-related documentation, best practices, as well as views of subject matter experts. An example of a leaf-node questionnaire for a DITSCAP security requirement '**Enclave Boundary Defense**' is shown in Figure 7. Such questionnaires can support qualitative as well as quantitative assessments on the basis of weights associated with their answer option in the application domain. The questionnaires, their interdependencies (the sequence in which questions are presented), and their relationships with concepts in the DITSCAP PDO are also modeled using ontological engineering processes in GenOM.

4. MODELS WITHIN THE DITSCAP PDO

On the basis of the steps defined in the previous subsection, the DITSCAP PDO captures various dimensions of the problem domain through hierarchical representations suggested by the Onto-ActRE framework. Specifically, the DITSCAP PDO contains (i) the overall DITSCAP process aspect knowledge captured using goal-driven scenario

Security Requirement: Enclave Boundary Defense

Description: Boundary defense mechanisms to include firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary to the wide area network, at layered or internal enclave boundaries and at key points in the network, as required. All Internet access is proxied through Internet access points that are under the management and control of the enclave and are isolated from other DoD information systems by physical or technical means.

Questionnaire:

1. Are adequate boundary defense mechanisms in place?

   Answer Options (Choose one):

   1. Firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary and at internal key points.
   2. Firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary.
   3. Boundary defense mechanisms are not in place.

2. Are adequate internet proxies in place?

   Answer Options (Choose one):

   1. All Internet access is proxied through Internet access points that are under the management and control of the enclave and are isolated from other DoD information systems. Demilitarized zones are implemented.
   2. Internet access is proxied but not necessarily isolated from other DOD systems. Demilitarized zones are implemented.
   3. Internet access is not proxied.

3. Are the software used for Firewall and Intrusion Detection System approved by the NSA approved processes like Common Criteria or FIPS ?

   Answer Options (Choose one):

   1. Yes
   2. No

Question/risk information Sources: Questions 1-2 are from the elaborations of the current security requirement based on subject-matter expertise. Question 3 is from the related requirement of Acquisition standards (DCAS-1) and Specified robustness (DCSR-1) identified based on requirements descriptions

Figure 7. Example leaf-node questionnaire for a security requirement concept in the DITSCAP PDO

DOI: 10.1002/spip

composition, (ii) a requirements domain model that hierarchically organizes requirement categories, (iii) viewpoints hierarchy of various stakeholders and IA services required by the DITSCAP, (iv) a domain-specific risk assessment taxonomy that gathers risk factors from a broad spectrum of perceived risk sources in the DITSCAP domain, (v) a network information discovery taxonomy that provides meta-knowledge about information learned from network discovery/monitoring tools, and (vi) interdependencies between the entities in the PDO. We now further elaborate on each of these models in the following subsections.

4.1. DITSCAP Goal Hierarchy

A DITSCAP goal hierarchy is created following the goal-driven scenario composition method of the Onto-ActRE framework. The method leverages the effectiveness of existing well-defined techniques for goal- (Lamsweerde 2001) and scenario-based (Sutcliffe 1998) approaches in an integrated fashion. Through this method we capture the real world goals of the C&A process, which are expressed as the tasks and activities that need to be followed for satisfying C&A objectives. The possible realization criteria for such goals in the hierarchy are captured using scenarios. Through a systematic derivation of scenarios from the goals or vice versa, the coverage of scenarios over the application domain can be established, or in the other case the goals selected for composing scenarios provide constraints to restrict their scope.

In the context of DITSCAP automation, the goal-driven scenario composition method provides a process-driven workflow that systematically captures C&A tasks and activities throughout the lifecycle of a software-intensive system. A partial C&A goal hierarchy is shown in Figure 8, which is created from the goals extracted from a homogeneous grouping of C&A task and activities outlined in the DITSCAP application manual (DoD 8510.1-M 2000). Such goals are extracted from DITSCAP process components at various levels of abstraction and then represented hierarchically by decomposing generic goals to specific ones and modelling them using ontological engineering processes. The questionnaires that capture user/system criteria in the leaf nodes of such a hierarchy are the representatives of various scenarios that satisfy their parent C&A goals. Several questionnaires are logically grouped into the process components to systematically guide the C&A process through related tasks as well as identify the dependencies between them. The user/system criteria gathered through these questionnaires also bring into focus the applicable security requirements enforced by DITSCAP. Several metrics can be established through the DITSCAP goal hierarchy, such as (i) process complexity, (ii) certification progress, (iii) task/activity requirements coverage, (iv) task/activity interdependencies/proximity, and (v) level of abstraction/inheritance, etc., to assist the C&A process. In addition, the model also helps align and trace specific C&A activities with their real world goals/objectives.

4.2. DITSCAP Requirements Domain Model (RDM)

Within the Onto-ActRE framework, a Requirements Domain Model (RDM) organizes the problem domain requirements through a hierarchical representation that includes top-level generic requirements, mid-level domain spanning requirements, and leaf-node sub-domain requirements. Such an organization of requirements allows for their exploration to be conservative in nature, i.e. to be more inclusive rather than exclusive. The scope of the RDM spans over the requirements of the system (functional and non-functional) and its related entities in the environment, such as organization, business/mission requirements, and other domain-specific considerations. In the context of the DITSCAP problem domain, a partial RDM related to '**security plan for information systems**' is shown in Figure 9, which elaborates on the '**Physical and Environmental Security Controls**' and '**Personnel Controls**' categories of security requirements. On the basis of these generic RDM categories, the security requirements enforced through Federal laws, DoD policies, and site/agency specific requirements and implementations are systematically extracted from their corresponding documents and organized under relevant categories at appropriate levels of abstraction. Figure 10 demonstrates the requirements extracted and organized from each of the Federal laws, DoD policies, and DoD policy implementation documents. More specifically, the categories in the RDM of Figure 9 are used to extract and structure requirements extracted from generic Federal laws, DoD policy/instructions, and DoD

DOI: 10.1002/spip

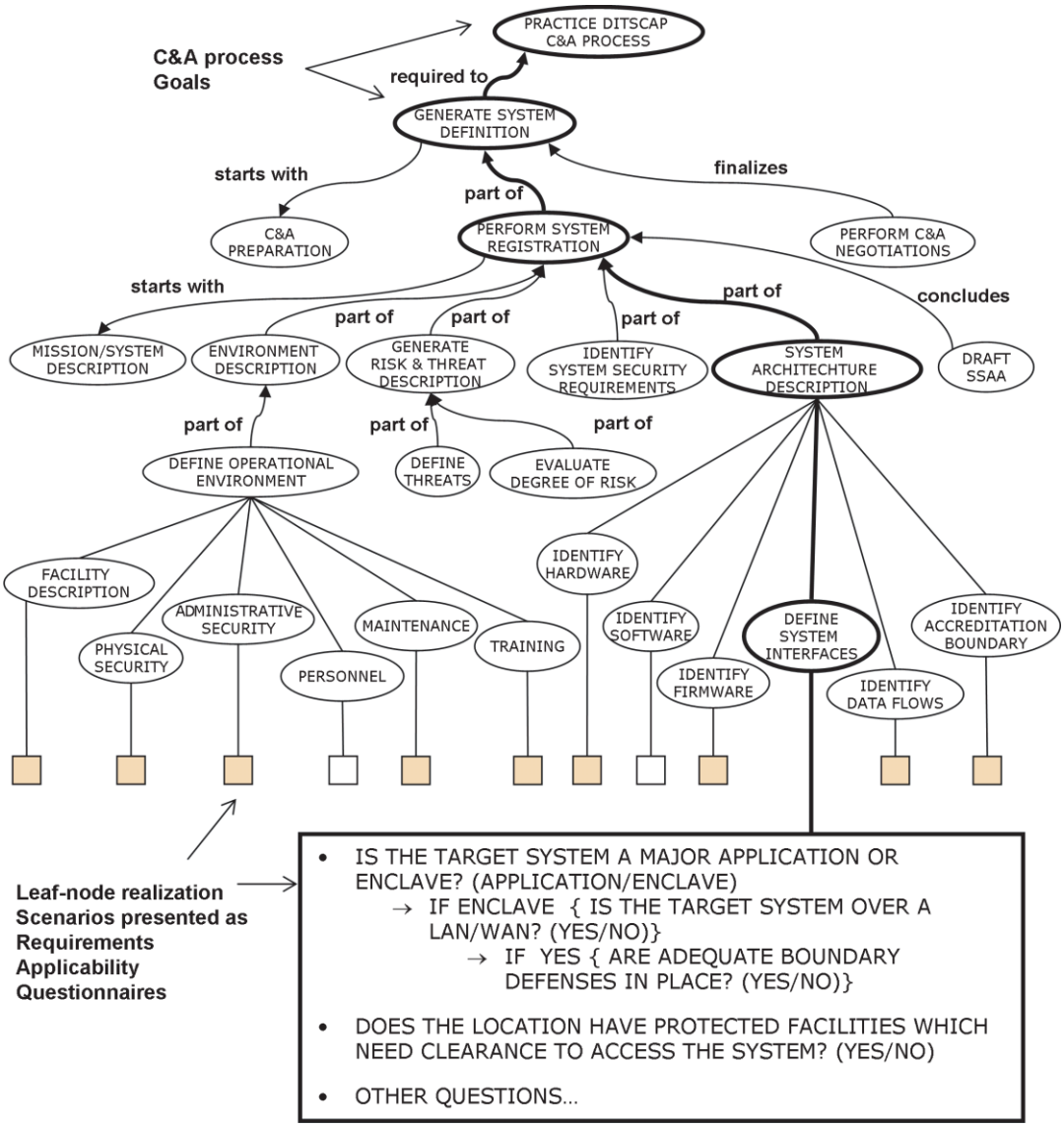


Figure 8. A partial DITSCAP goal hierarchy

policy implementation documents to produce hierarchical structures shown in Figure 10. Figure 10 can also be understood in the context of various documents and their corresponding security requirements shown in Figure 5, i.e. when the RDM of Figure 9 is applied to each level of documents shown in Figure 5, the extracted requirements are structured as shown in Figure 10.

For example the RDM of Figure 9 when applied to the Federal Laws document in Figure 5, the requirement marked as '**1**' is organized under the '**Federal Personnel Security**' category shown in Figure 10. Similarly, the RDM of Figure 9 when applied to the DoD policy/instructions level document in Figure 5, the requirement marked as '**2**' is organized under the '**DoD Personnel Security**' category shown in Figure 10. The numbers marked against the security requirements shown in Figure 10 correspond to the markers for security requirements shown in Figure 5. In addition, the relationships that exist between requirements at various levels in the organizational hierarchy as well as the interdependencies between them are captured and modeled using '**realized_by**',

DOI: 10.1002/spip

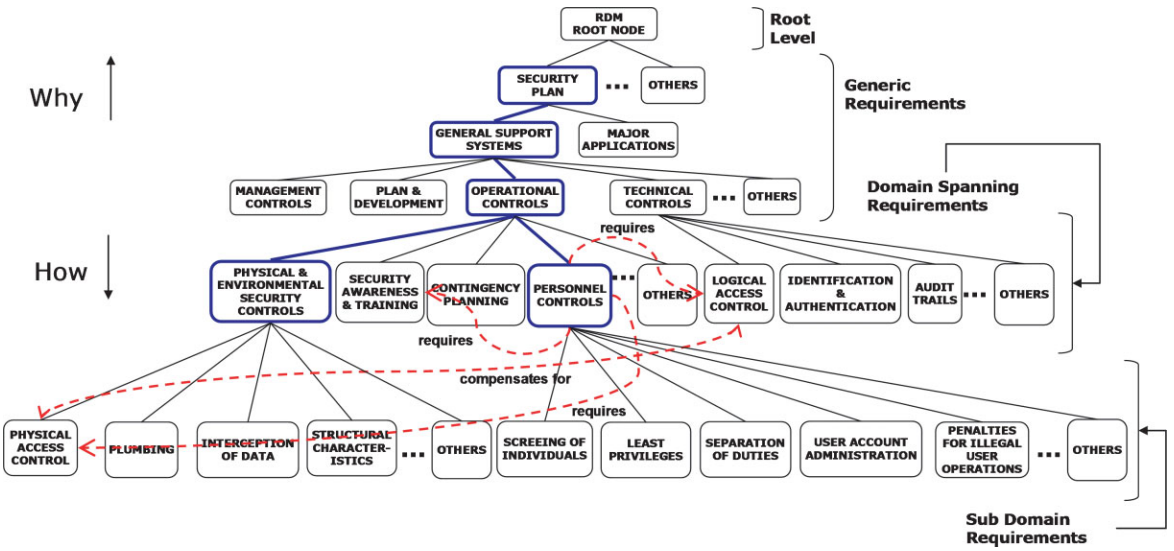


Figure 9. A partial requirements domain model in the DITSCAP domain

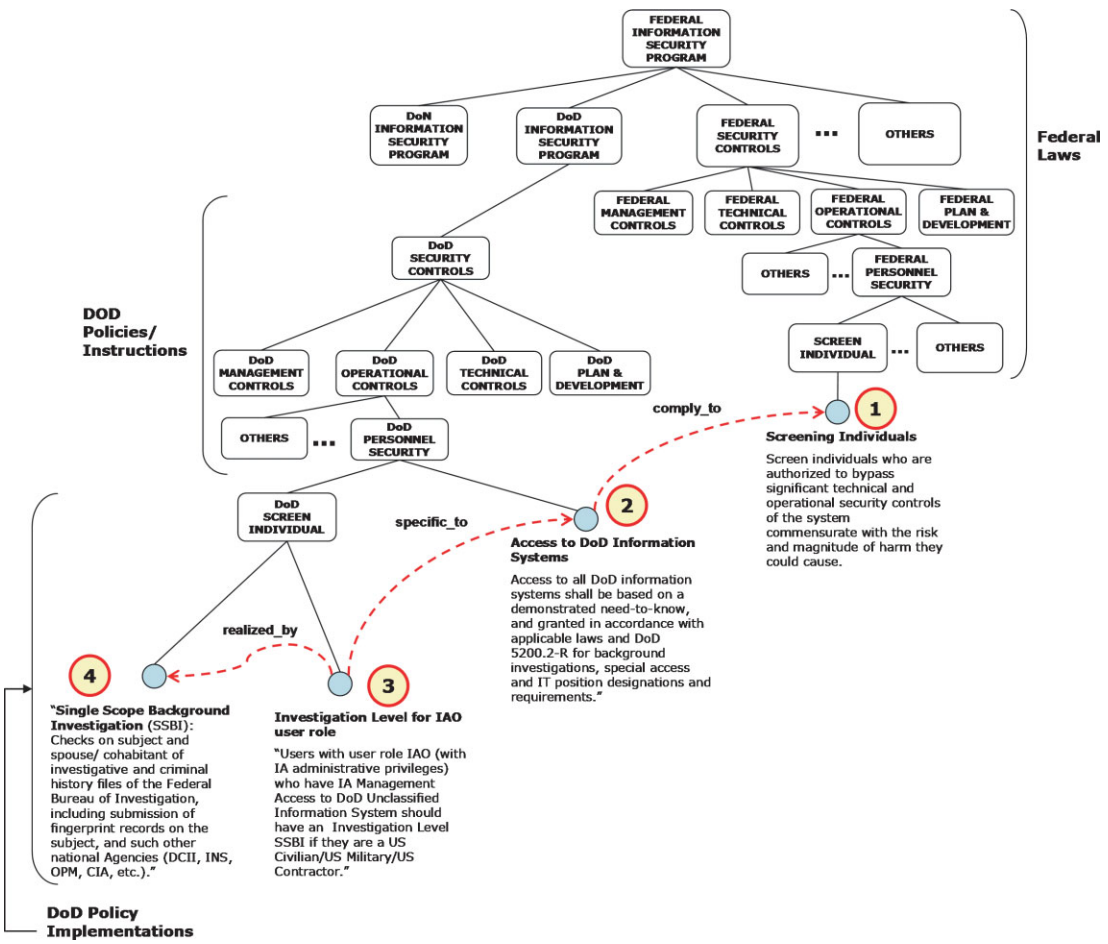


Figure 10. A partial hierarchical organization of DITSCAP-enforced security requirements in the DITSCAP PDO

DOI: 10.1002/spip

'**specific_to**', and '**comply_to**' GenOM Features as shown in Figure 10, which have semantics as explained in Figure 5.

Such a requirements hierarchy allows the determination of applicable security requirements by successively decomposing the high-level generic requirements into a set of specific applicable requirements in the leaf nodes on the basis of user criteria elicited using leaf-node questionnaires in the C&A goal hierarchy. Furthermore, non-taxonomic links can be utilized to effectively interpret and enforce requirements by identifying the requirements in related categories as well as relationships with other concepts in the PDO. The RDM supports the creation of several metrics such as (i) Domain complexity, (ii) requirements applicability, (iii) requirements compliance, (iv) requirements interdependencies/proximity, and (v) level of abstraction/inheritance, etc., to assist the C&A process. The RDM also helps to align and trace specific technical implementation and interpretation with their high-level policies and laws that enforce them in the organizational hierarchy.

4.3. DITSCAP Viewpoints hierarchy

Requirements usually capture ideas, perspectives, and relationships at various levels of detail and they are interpreted differently from different viewpoints (Kotonya and Sommerville 1998). To provide a systematic and controlled approach for identifying such viewpoints, we advocate the use of the Viewpoint-Oriented Requirements Definition (VORD) (Kotonya and Sommerville 1998) viewpoints class template. On the basis of this template, we create a viewpoints hierarchy with higher level nodes consisting of viewpoints, such as the DoD Components (refers to all organizational entities in the DoD) that map to generic requirements in the RDM, and leaf nodes representing viewpoints, such as those of specific system stakeholders (for example, a system administrator), and IA services or security controls that relate to more specific requirements in the RDM through non-taxonomic links and properties. A partial viewpoints hierarchy in the DITSCAP PDO is shown in Figure 11. Through the viewpoints hierarchy we can establish several metrics on the basis of their relationships with requirements in the RDM. We identify metrics such as (i) viewpoint coverage, (ii) viewpoint intersections/overlaps, and (iii) responsibility satisfaction level of a viewpoint, etc., to assist the C&A process. Such metrics help in understanding the DITSCAP from a particular viewpoint as well as support negotiations between conflicting viewpoints.

4.4. Domain-Specific Risk Assessment Taxonomy

The DITSCAP PDO also includes a domain-specific risk assessment taxonomy, which aggregates a

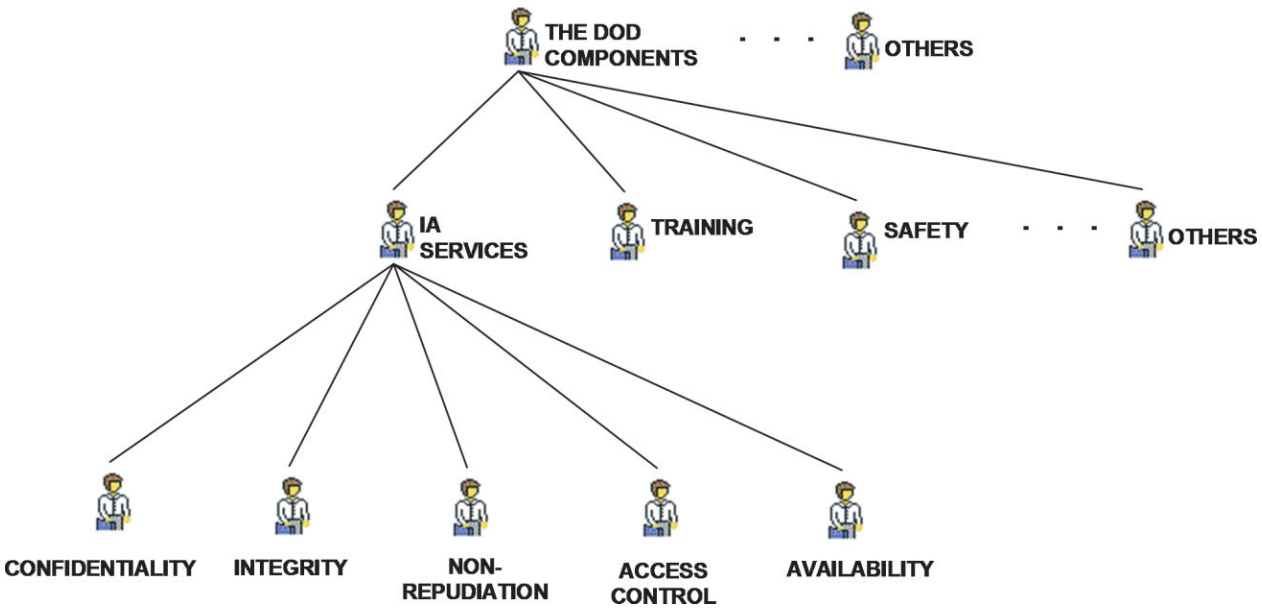


Figure 11. A partial viewpoints hierarchy in the DITSCAP PDO

DOI: 10.1002/spip

broad spectrum of possible categories and classification of risk related information in the DITSCAP domain. The risk assessment goals expressed in the higher level non-leaf nodes of this taxonomy are achieved using specific criteria addressed in its leaf nodes. The upper level non-leaf nodes in the taxonomy consist of threat, vulnerabilities, countermeasures, mission criticality, assets, and other categories related to risk assessment. Each non-leaf node is then decomposed into more specific categories. Furthermore, the non-taxonomic links that exist between the categories of the taxonomy are critical to understand the relationships/dependencies between various risk factors. A partial decomposition along the threat dimension is shown in Figure 12. In addition, the threat categories in Figure 12 also have properties that further characterize them as natural/man-made, intentional/unintentional, insider/outsider, and physical/cyber. Each threat category is also associated with other dimensions in the risk assessment taxonomy as well as requirements in the RDM.

The categorization and classification of concepts in the risk assessment taxonomy is based on the information sources available in the DITSCAP domain. We currently restrict its scope to the DITSCAP Application Manual, the DITSCAP Minimal Security Checklists (DoD 8510.1-M 2000), and DITSCAP-oriented directives and security requisites. The risk assessment taxonomy based on its relationships with the requirements (Lee, Gandi and Ahn 2005b) in the RDM supports the development of several metrics, such as (i) necessity and sufficiency conditions between risk factors (for example, if a set of countermeasures related to a vulnerability is 'necessary', then the risk associated with that vulnerability is not mitigated unless all the necessary countermeasures are satisfied), (ii) requirements coverage, (iii) risk mitigation levels through requirements compliance, (iv) level of interdependency of risk factors, and (v) asset criticality, etc., to assist the C&A process. Such metrics help in performing cost-benefit analysis for establishing adequate security as well as prioritizing requirements.

4.5. Meta-knowledge About Information Learned from Network Discovery/Monitoring Tools

One of the objectives of DITSCAP automation is to allow a comparison between the intended and actual operational environments. To achieve such goals, we introduce a domain-specific model within the DITSCAP PDO, called the network information discovery taxonomy (NIDT), to gather metrics and measures on the basis of concepts that reflect the

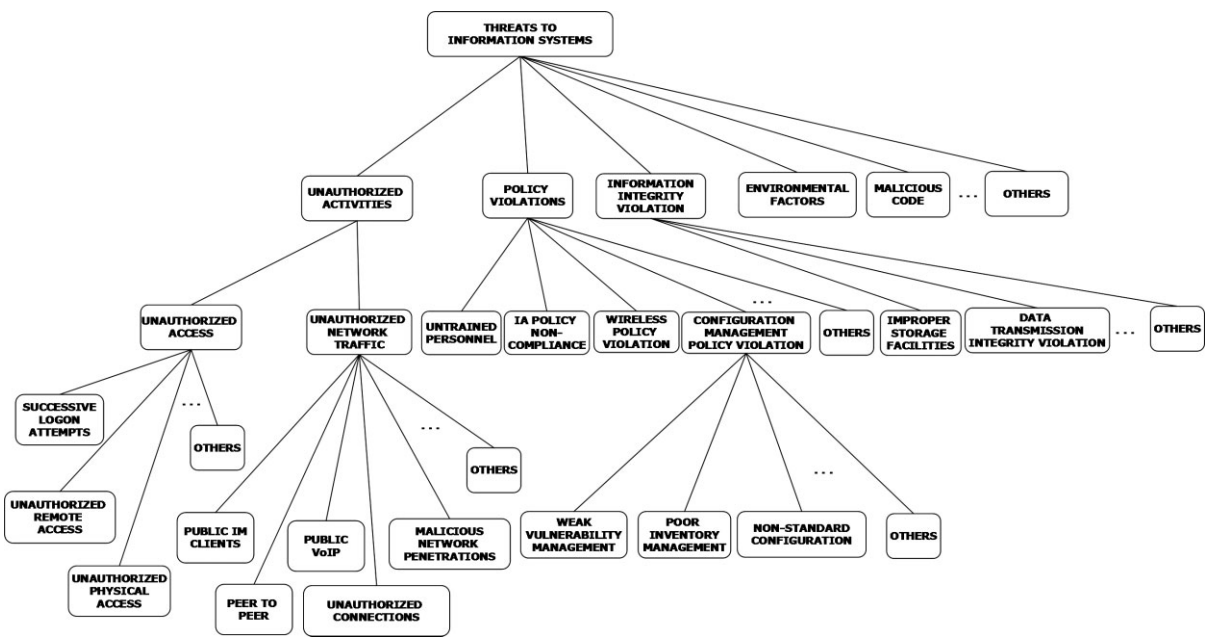


Figure 12. A partial decomposition along the threat dimension of the risk assessment taxonomy in the DITSCAP PDO

DOI: 10.1002/spip

actual operational environment. The NIDT aggregates information discovered by a set of network tools and automated scripts selected on the basis of the information required for DITSCAP, such as (i) hardware, software, and firmware inventories, (ii) configuration information of network devices and services, and (iii) vulnerability assessment using penetration testing. The taxonomy that depicts the current scope of our network discovery capabilities is shown in Figure 13. Each leaf-node concept in the taxonomy is associated with various information gathering criteria, based on which the automated tools and scripts gather information from the operational environment.

In the following section, we discuss how various models available within the DITSCAP PDO help to systematically elicit and organize the evidences available throughout the DITSCAP.

5. COLLECTING EVIDENCES THROUGH DITSCAP SECURITY REQUIREMENTS

DITSCAP artifacts are defined by the metrics and measures gathered on the basis of the execution of its tasks and activities for assessment purposes. However, identifying, eliciting, representing, and organizing the diverse range of metrics and measures required for DITSCAP artifacts is inherently difficult using manual documentation approaches. To address these issues, the traceable rationales of the DITSCAP PDO and its supporting infrastructure help to systematically identify, capture, and organize such metrics and measures throughout the DITSCAP. We identify that DITSCAP security requirements and guidance documents are rich sources of information, which include diverse pieces of information related to DITSCAP artifacts. As a result, DITSCAP security requirements and their associated questionnaires modeled through the RDM can be utilized to systematically assimilate evidence regarding the target software-intensive system. In addition, the evidences collected through each requirement are also related to other concepts within the DITSCAP PDO. To demonstrate this idea, consider the example requirement and its description shown in Figure 14, for which we identify various related concepts within the DITSCAP PDO by visualizing its interdependencies through the GenOM tool support in Figure 15.

The GenOM instance visualization shown in Figure 15 depicts the relationships of DITSCAP-enforced security requirements concepts with C&A goals, related/dependent requirements, associated viewpoints in the domain, questionnaires, requirements source, and related risk factors. The boxes and their interconnections in Figure 15 represent instances of various domain objects and features modeled in GenOM for the DITSCAP PDO. The '**Enclave Boundary Defenses**' security requirement under consideration is an instance

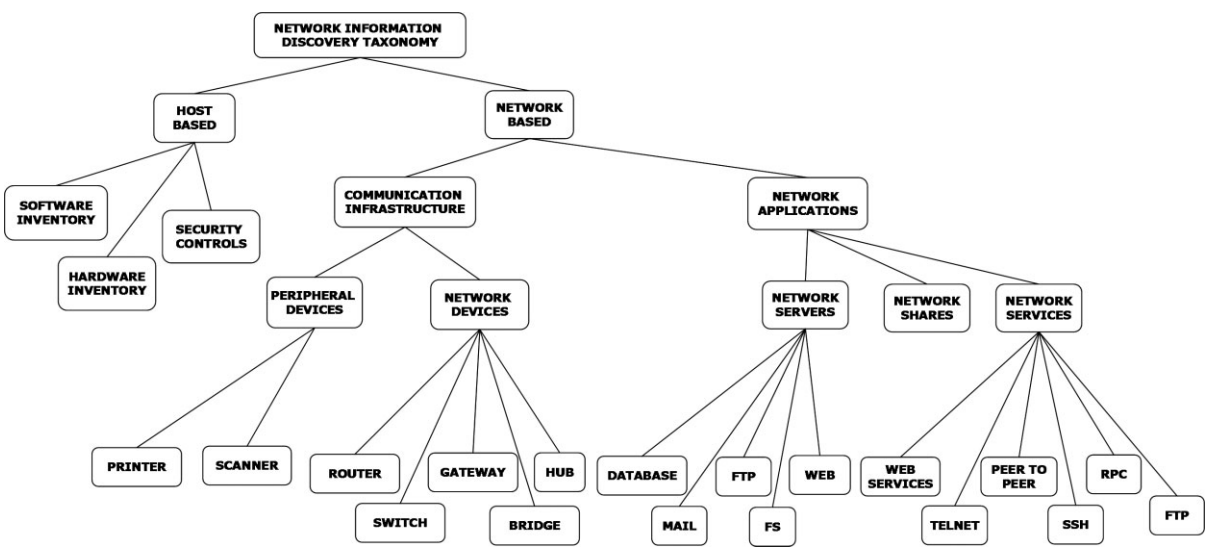


Figure 13. Network information discovery taxonomy in the DITSCAP PDO

DOI: 10.1002/spip

Enclave Boundary Defense

EBBD-2 Boundary Defense

Boundary defense mechanisms to include firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary to the wide area network, at layered or internal enclave boundaries and at key points in the network, as required. All Internet access is proxied through Internet access points that are under the management and control of the enclave and are isolated from other DoD information systems by physical or technical means.

Figure 14. Example requirement from DITSCAP-related guidance document

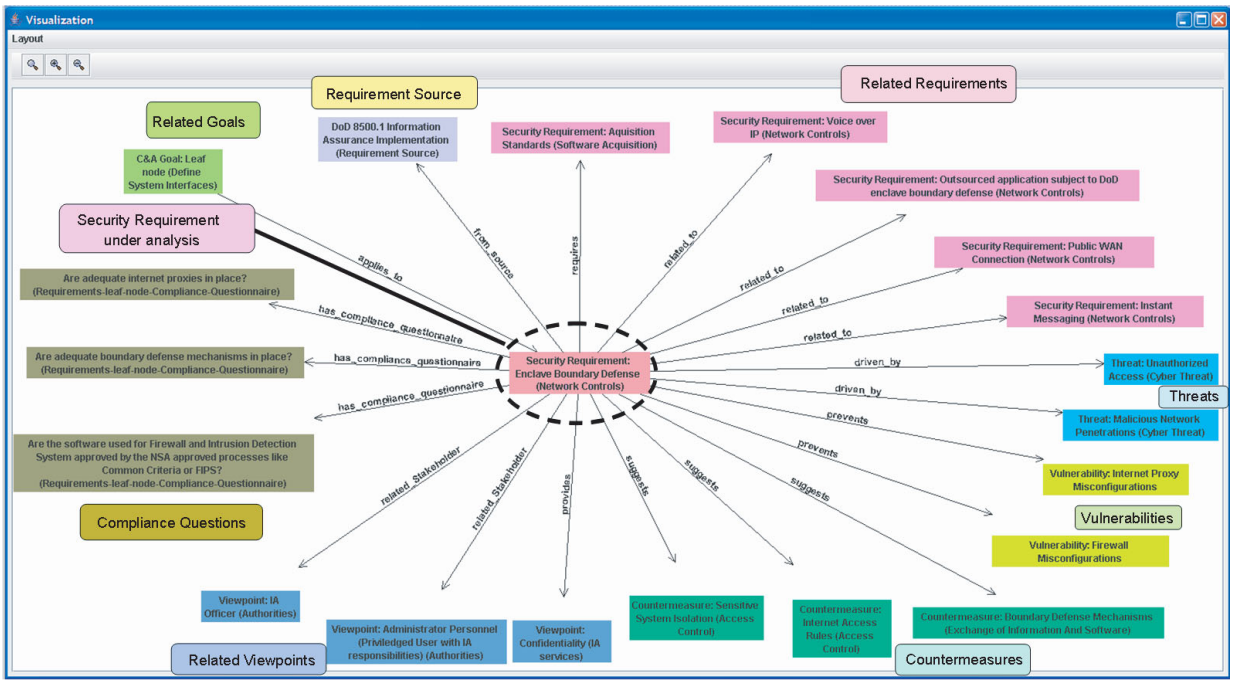


Figure 15. GenOM visualization of a security requirement in the DITSCAP PDO and its relationships with other concepts

of the '**Network Controls**' category object of the requirements domain model in the DITSCAP PDO. This security requirement instance is related to the compliance criteria question instances through the '**has-compliance-questionnaire**' feature. The compliance criteria questions are modeled as instances of the '**Requirements-leaf-node-Compliance-Questionnaire**' object.

The '**Enclave Boundary Defenses**' requirement instance also relates to instances of other models within the DITSCAP PDO through several features that represent non-taxonomic relationships among them. In Figure 15, the '**Enclave Boundary Defenses**' instance relate to instances of the DITSCAP C&A goal hierarchy as well as instances of the viewpoints hierarchy through the '**applies-to**' and '**related-stakeholder**' features, respectively. Other features such as '**from-source**' relate the requirement to the source from which it was extracted. Also, the relationships of the '**Enclave Boundary Defenses**' requirement with other requirements instances are captured through the '**related_to**' and '**requires**' features. The ontological characteristics of the DITSCAP PDO also

DOI: 10.1002/spip

help in utilizing the relationships that exist between security requirements and the various factors considered for risk assessment (Lee, Gandi and Ahn 2005b). By taking advantage of the synergy between these models, we can systematically identify the threats, vulnerabilities, and countermeasures associated with the target system from the compliance information gathered for the security requirements. From a requirements perspective, the relationships between risk factors can help to identify and elaborate on the interdependencies between requirements, which may not be readily apparent. Figure 15 depicts the relationships between '**Enclave Boundary Defenses**' requirement and risk factors with the features '**suggest**', '**prevents**', and '**driven_by**' for countermeasures, vulnerabilities, and threats, respectively. The information available through such relationships when combined with asset value and mission criticality provides the basis to perform cost-benefit analysis and requirements prioritization necessary to establish adequate security.

The relationships in Figure 15 have been identified through keywords in requirements descriptions, domain knowledge of subject matter experts, as well as through the synergy that exists between various models in the DITSCAP PDO based on their interdependencies. Once identified, such relationships help systematically interpret the evidences gathered through questionnaires in the context of the related concepts within the DITSCAP PDO. The individual questions within each questionnaire with predefined answer options act as measures collected for the concepts they are related to. For the security requirement in Figure 14, such questions and their answer options with related concepts from the risk assessment taxonomy are identified in Table 1. As an example, the answers chosen for the questions in Table 1 act as evidences, among many others, collected for the target software-intensive system in the context of the '**Cyber Threat**' of '**Malicious Network Penetrations**.' In turn, each requirement can be analyzed for its level of compliance along with the impact on its effectiveness on the basis of concepts from single or multiple dimensions of the problem domain through such questionnaires. Questionnaires for each requirement in the RDM, and the corresponding answer options for each question provide a comprehensive collection of information gathered from various sources such as user input, automated network information discovery (from NIDT), system documentation, etc., which can be used to systematically understand, analyze, and produce DITSCAP artifacts from various dimensions in the DITSCAP PDO.

6. MULTI-DIMENSIONAL LINK ANALYSIS

We introduce the concept of Multi-Dimensional Link Analysis (MDLA) to promote cohesion between diverse metrics and measures, which are necessary to collectively explain, predict, and control the emergent behaviour of software-intensive systems. These metric and measures are often expressed in different ways or obtained from different sources. The theoretical foundations behind the Onto-ActRE framework allow for such analysis to be initiated from various dimensions in the problem domain. In the context of DITSCAP automation, MDLA provides an 'active' environment in which the evidence gathered through the questionnaires as well as the metrics and measures made available through various models in the DITSCAP PDO (discussed in Section 4) collectively help to produce DITSCAP artifacts that have strong alignment and traceability with real world goals/objectives. We believe that these 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 (Lee and Rine 2004a). The evidences gathered through questionnaires, as shown in Table 1, when linked from multiple dimensions such as requirements from the RDM, DITSCAP process goals, risk factors in the risk assessment taxonomy, or viewpoints in the viewpoints hierarchy, act as shared evidences for the metrics and measures from these dimensions. To motivate the feasibility of this approach, consider an example in which the goal is to assess the threat of '**Unauthorized Access**' for the target software-intensive system based on the criteria collected through the DITSCAP. Such an analysis based on a manual approach to DITSCAP would not be possible as it lacks the traceability and understanding of the complex interdependencies that exist among information gathered from several sources throughout the DITSCAP.

In contrast to a manual approach, MDLA supports the aggregation of various pieces of information from multiple models in the DITSCAP PDO

DOI: 10.1002/spip

Table 1. Question and Answer sets for DITSCAP security requirements and their related Concepts in the DITSCAP PDO

Questions	Answers			Related Concepts in the DITSCAP PDO			
	Option 1	Option 2	Option 3	Assets	Threats	Countermeasures	Vulnerabilities
Are adequate boundary defense mechanisms in place?	Firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary and at internal key points.	Firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary.	Boundary defense mechanisms are not in place.	• Enclave → Firewall → IDS	• Cyber Threats → Unauthorized Access → Malicious Network Penetrations	• Exchange of Information and Software → Boundary Defense Mechanisms	• Misconfigurations → Firewall misconfiguration
Are adequate internet proxies in place?	All Internet access is proxied through Internet access points that are under the management and control of the enclave and are isolated from other DoD information systems. Demilitarized zones are implemented.	Internet access is proxied but not necessarily isolated from other DOD systems. Demilitarized zones are implemented.	Internet access is not proxied.	• Enclave → Internet Access Points	• Cyber Threats → Unauthorized Access → Malicious Network Penetrations	• Access Control → Internet Access Rules → Sensitive System Isolation	• Misconfigurations → Internet Proxy misconfiguration

DOI: 10.1002/spip

Table 2. A subset of security requirements that provide shared evidences for well-defined metric categories from the RDM, goal hierarchy, viewpoints hierarchy, and asset taxonomy to understand and assess the threat of '**Unauthorized Access**'

Requirements that contribute to provide evidences for the concept of "Unauthorized Access" Threat	Requirements Metric Categories	C&A Goals Metric Categories	IA Services Viewpoints Metric Categories	Assets Metric Categories
IAIA-1 Individual Identification and Authentication	Authentication	Define Maintenance Procedures	Confidentiality	DoD Information Systems
ECLO – Logon				
ECAN-1 Access for Need-to-Know		Define User Clearance		
PESL-1 Screen Lock	Physical Access Control	Define Operational Environment	Integrity	Display Devices
PECF-1 Access to Computing Facilities			Confidentiality	Enclave Computing Facilities
PEPS-1 Physical Security Testing				
PEVC-1 Visitor Control to Computing Facilities				
EBRP-1 Remote Access for Privileged Functions	Network Access Control	Define System Interfaces and Data flows		DoD Information Systems
EBRU-1 Remote Access for User Functions				
ECND-1 Network Device Controls			Integrity	
ECIM-1 Instant Messaging				
DCMC-1 Mobile Code				
ECVI-1 Voice over IP			Availability	
EBVC-1 VPN Controls				
Outsourced application subject to DoD enclave boundary defense			Confidentiality	Enclave
EBBD-2 Boundary Defense				
Interconnection between DoD and foreign nations information systems	System Interconnection			
Interconnections between different security domains				
Interconnection with external networks				
Formal Authorization of interconnection				
ECAR-2 Audit Record Content	Audit	Define Data Security Requirements		Audit Records
ECTP-1 Audit Trail Protection			Integrity	
–	–	–	–	–
–	–	–	–	–

whose entities satisfy the necessary and sufficient conditions for being members of the concept of the threat of '**Unauthorized Access**'. Such membership is decided on the basis of the properties and features of the concepts modeled in the DITSCAP PDO. Table 2 enumerates a subset of the DITSCAP-enforced security requirements modeled in the PDO, which provide the evidences to assess the threat of '**Unauthorized Access**'. Table 2 also enumerates the categories of metrics from the dimension of the RDM, goal hierarchy, viewpoints hierarchy, and assets for which the information gathered through requirements act as shared evidences for the threat of '**Unauthorized Access**'. The table clearly demonstrates the interaction between metrics and measures from these dimensions in the context of their shared evidences. Furthermore, the evidences gathered from these requirements can also be interpreted from other related dimensions such as Vulnerabilities, Countermeasures, Network discovered information, and associated stakeholders through MDLA. In addition, the information from different dimensions can compensate for each other and actively assist in the process of discovering missing, conflicting, and interdependent pieces of information throughout the DITSCAP. The inference engine (Carroll *et al*. 2004) associated with GenOM along with MDLA's integrated framework for analysis helps to establish metrics and measures on the basis of a common understanding and the reflected language from multiple dimensions.

7. RELATED WORK

Security maturity models, such as the Systems Security Engineering Capability Maturity Model (SSE-CMM 2003), provide a reference model for evaluating the maturity in practices of a given engineering discipline. Similarly, other models, such as the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC 15504 1998) for process assessment, use their reference models and capability level scales as a framework for assessment. Such process-based assurance methods rely on the premise that improved processes can be used to augment current IA approaches and practices. On the other hand,

DOI: 10.1002/spip

the ISO/IEC 15408 (Common Criteria 1999) provides a complementary product-based assessment that evaluates an information security product or system, which it calls a target of evaluation. Several other processes (ISO/IEC 15288 2000, ISO/IEC 12207 1995), frameworks (ISO/IEC 15443 2001, BS 7799 1999), and guidelines (ISO/IEC 13335 1996, Swanson 2001, Swanson *et al*. 2003, Ross *et al*. 2004) exist to identify, evaluate, or manage IA metrics and measures from various perspectives. We also observe that the general principles across these standards/processes can be applied synergistically. However, despite their existence, the criteria to establish software assurance levels is often confined and restricted to the experts in the domain or trained professionals who are familiar with specific standards, operating systems, programming languages, and communication protocols. In addition, the complex interdependencies that exist among information from such diverse sources significantly restrict human ability to effectively engineer secure systems and identify, evaluate, and report their assurance levels. Furthermore, commercial tool support and services (Xacta 2004), which exist to help achieve C&A, use proprietary methods and procedures to assess compliance, which are usually not available to the research community for evaluation. To further aggravate the situation, C&A processes are often reduced to a mere bureaucratic necessity for obtaining approval to begin functioning by generating the required documentation without specifically focussing on assessing and managing the operational risks of the site and system (Davis 2005).

In the domain of information security, a popular approach for risk assessment is the Operationally Critical Threat, Asset, and Vulnerability Evaluation^{SM} (OCTAVE^{SM}) (Alberts and Dorofee 2001). This criteria provides the definition of a general approach for evaluating and managing information security risks. However, OCTAVE^{SM} relies on the organization to develop their own methods and tools to satisfy its criteria. The CORAS (Aagedal *et al*. 2002) project advocates a UML-based approach for risk assessment, but their focus is to combine several methods and standards of risk assessment (Freeman *et al*. 1997), while proposing a risk assessment methodology for large heterogeneous systems, outlined the following essential characteristics for an effective risk assessment: (i) provide the best available results in a timely manner; (ii) the effort should be commensurate with the value of results; (iii) the process should be comprehensive; (iv) evaluation and reporting of threats, vulnerabilities, and risks should be consistent; and (v) the results should be understandable, with as solid a technical basis as possible, and be communicated at the appropriate level of abstraction with preferably direct traceability to technical rationales. We believe that our efforts for DITSCAP automation provide a good setting for practicing a methodology that successfully satisfies these characteristics.

Vaughn *et al*. (2003) explored the work that has been done for the development of IA metrics and measures and expressed faith in the security mechanisms and countermeasures. They have identified that these metrics and measures usually are specific to an organization and depend on their technical, organizational, and operational needs and the resources they can make available. They quote from their findings of Information-Security-System Rating and Ranking (ISSRR) Workshop 2001 that IA metrics should be developed as a cross product of what needs to be measured, why it needs to be measured, and for whom it is to be measured. We believe that the achievement of such an alignment of metrics and measures with their real world objectives is a challenge without an integrated framework and tool support for inherently complex software-intensive systems, in which software, systems, processes, practice, and environment contribute to gain trust and assurance.

The Goal Question Metric (GQM) (Basili and Rombach 1988) and balanced scorecard framework (Kaplan and Norton 1996) are metrics development approaches that have been applied frequently for supporting goal-oriented software process improvement. Their influences can be seen in several approaches for defining metrics and measures for IA (Swanson *et al*. 2003) (Lekkas and Spinellis 2005). The Tailoring A Measurement Environment (TAME) project (Basili and Rombach 1988) outlines several important characteristics and principles of measurements with the implementation of a GQM-based approach. The PROduct Focused improvement for Embedded Software processes (PROFES) (Järvinen *et al*. 1999) approach combines the GQM approach with a software process assessment framework to achieve continuous process assessment. Taxonomy-based questionnaire approaches have also been explored for risk assessment (Carr *et al*. 1993) as well

DOI: 10.1002/spip

as to support decision making related to enterprise information security (Johansson and Johnson 2005). However, we find that such approaches are more focussed on the development of metrics and measures but do not support traceability and communication between them from dimensions that are necessary to understand their impact on the emergent behaviour of software-intensive systems.

8. CONTRIBUTIONS AND FUTURE WORK

C&A process artifacts and their corresponding metrics and measures gathered throughout the software process lifecycle are important entities that foster confidence in the assurance provided by software-intensive systems. However, to produce metrics and measures that are faithful indicators of the emergent properties of complex software-intensive systems within multifaceted socio-technical environments is to date a challenging research issue. In addition, metrics and measures that provide a close alignment with the real world goals/objectives are not readily available as the by-product of applying the C&A process or derived from technical attributes of system operation. To address these issues in this article, we focus on producing C&A process artifacts for software-intensive systems within a novel integration framework that promotes synergistic interactions between well-defined metrics and measures from multiple dimensions with complementary semantics and different levels of abstractions on the basis of shared evidences gathered throughout the software-intensive system's lifecycle. From this perspective, we identify the following contributions. Firstly, we provide a comprehensive overview of a stepwise methodology for eliciting, representing, and modelling problem domain concepts on the basis of well-defined semantics and ontological engineering techniques supported by the Onto-ActRE framework and related GenOM tool support. We also present real examples from our case study on DITSCAP automation, which illustrate several heuristics for extracting and organizing problem domain concepts, properties, and their interdependencies from documents available at various organizational levels. Secondly, we elaborate on models from different dimensions produced within the DITSCAP problem domain ontology and the corresponding metrics and measures available from them, which have strong traceability and alignment with real world goals/objectives, interpretations, and practices. Thirdly, the systematic identification of relationships between various concepts in the DITSCAP PDO through keywords in requirements descriptions, subject matter experts, as well as through the synergy that exists between various models in the DITSCAP PDO lead to the creation of exhaustive questionnaires that gather evidence related to the target system. On the basis of such relationships, coverage of the problem domain concepts by the questionnaires can also be established. Fourthly, we introduce MDLA for analytical analysis, which promotes cohesion between the metrics and measures expressed in different ways or obtained from different sources in order to collectively explain, predict, and control the emergent behaviour of software-intensive systems. We also present motivational examples from our case study to rationalize the applicability and feasibility of MDLA in DITSCAP automation.

As part of future work, we would like to address the following on-going research objectives. Firstly, the various concepts in the Onto-ActRE PDO, in terms of the properties they possess and the relationships (biases) that hold, should be formalized in order to identify appropriate problem solving dimensions that foster systematic analysis using MDLA. Secondly, we are also focussing our efforts on outlining a case study designed research methodology (CSM) (Lee and Rine 2004b) for evaluating the effectiveness of our methodology. We chose CSM because of the characteristics of our methodology that require interventions from the domain subject matter expert in order to perform each appropriate step in the automated DITSCAP on demand, and also because the characteristics of its validation procedure cannot favour alternatives, either because of its novelty and uniqueness or because of the relative difference in the level of understanding of the domain and analytical skill of subject matter experts in the 'experimental' conditions of the actual case study. A validation exercise based on the case study designed methodology will produce several metrics and measures for the units of analysis of our methodology and provide the opportunity for various scientific explanations to be generated through analytical generalization.

DOI: 10.1002/spip

ACKNOWLEDGEMENT

This work is partially supported by the grant (Contract# N65236-05-P-3672) from the Critical Infrastructure Protection Center, Space and Naval Warfare (SPAWAR) Systems Center, US Department of Navy, Charleston, SC, and by the National Science Foundation (NSF) grant DUE: Federal Cyber Service: SFS #0416042.

REFERENCES

Aagedal JO, den Braber F, Dimitrakos T, Gran BA, Raptis D, Stolen K. 2002. Model-based risk assessment to improve enterprise security. In *Proceedings of the 6th International Enterprise Distributed Object Computing Conference*, Lausanne, Switzerland, Sept 17, 51–62.

Alberts C, Dorofee A. 2001. *OCTAVE*SM *Criteria, Version 2.0*. CMU/SEI-2001-TR-016 ESC-TR-2001-016.

Basili VR, Rombach HD. 1988. The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* **14**(6): 758–773.

BS 7799. 1999. *Information Security Management*.

Carr MJ, Konda SL, Monarch I, Ulrich FC, Walker CF. 1993. Taxonomy-based risk identification. Technical Report CMU/SEI-93-TR-6 ESC-TR-93-183. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Carroll JJ, Dickinson I, Dollin C, Reynolds D, Seaborne A, Wilkinson K. 2004. Jena: implementing the semantic web recommendations. In *Proceedings of the 13th International World Wide Web Conference*, New York, USA, May 17–22, 74–83.

Chaudhri VK, Farquhar A, Fikes R, Karp PD, Rice JP. 1998. OKBC: a programmatic foundation for knowledge base interoperability. In *Proceedings of the 15th National Conference on Artificial Intelligence, AAAI*, Menlo Park, CA, 600–607.

Common Criteria. 1999. *Common Criteria for Information Technology Security Evaluation*. Parts 1, 2 & 3. Version 2.1. ISO/IEC 15408-1 : 1999(E).

Davis T. 2005. Federal Computer Security Report Card Grades of 2004. *Press Release*, http://reform.house.gov/UploadedFiles/021605FISMAstatement.pdf.

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

DoDI 5200.40. 1997. *Department of Defense Information Technology Security Certification and Accreditation (DITSCAP)*.

FISMA. 2005. Federal Information Security Management Act (FISMA) 2004 Report to Congress, http://www.whitehouse.gov/omb/inforeg/2004_fisma_report.pdf.

Freeman JW, Darr TC, Neely RB. 1997. Risk assessment for large heterogeneous systems. In *Proceedings of the 13th Annual Computer Security Applications Conference*, San Diego, CA, 44.

ISO/IEC 12207. 1995. *Information Technology – Software Life Cycle Processes*.

ISO/IEC 13335. 1996. Guidelines for the management of IT security. Technical Report BS ISO/IEC TR 13335-1:1996, ISBN: 0580303918.

ISO/IEC 15504. 1998. Software Process Assessment (SPICE). Technical Report, http://www.isospice.typepad.com/isospice_is15504/.

ISO/IEC 15288 CD2. 2000. *Life Cycle Management – System Life Cycle Processes*.

ISO/IEC 15443. 2001. *Information Technology – Security Techniques – A Framework for IT Security Assurance*.

ISSRR. 2001. Proceedings of Workshop on Information-Security-System Rating and Ranking (ISSRR) held in Williamsburg, Williamsburg, VA.

Jackson M. 1997. The meaning of requirements. *Annals of Software Engineering*, Vol. 3. Baltzer Science Publishers: 5–21.

Järvinen J, Hamann D, Van Solingen R. 1999. On integrating assessment and measurement: towards continuous assessment of software engineering processes. In *Proceedings of the 6th International Symposium on Software Metrics METRICS*. IEEE Computer Society: Boca Raton, Florida, USA, Nov 4–6, 22.

Johansson E, Johnson P. 2005. Assessment of enterprise information security – estimating the credibility of the results. *Proceedings of the Symposium on RE for Information Security (SREIS 05), Requirements Engineering (RE'05)*. IEEE CS Press: Paris, France.

Kaplan RS, Norton DP. 1996. *The Balanced Scorecard: Translating Strategy into Action*. Harvard Business School Press: Boston, MA, USA.

Kimbell J, Walrath M. 2001. Life cycle security and DITSCAP. *IANewsletter*. **4**(2): 16–26. http://iac.dtic.mil/iatac.

DOI: 10.1002/spip

Kotonya G, Sommerville I. 1998. *Requirements Engineering – Processes and Techniques*. John Wiley: New York.

Lamsweerde A. 2001. Goal-oriented requirements engineering: a guided tour. In *Proceedings of the 5th International Symposium on Requirements Engineering*, Toronto, Canada, August, 249–262.

Lee SW, Rine DC. 2004a. Missing requirements and relationship discovery through proxy viewpoints model. *Studia Informatica Universalis: International Journal on Informatics* **3**(3): 315–342.

Lee SW, Rine DC. 2004b. Case study methodology designed research in software engineering methodology validation. In *Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering*. Banff, Alberta, Canada, 117–122.

Lee SW, Gandhi RA. 2005a. Ontology-based active requirements engineering frame-work. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05)*. IEEE Computer Society Press: Taipei, Taiwan.

Lee SW, Gandhi RA. 2005b. Engineering dependability requirements for software-intensive systems through the definition of a common language. In *Proceedings of the 13th IEEE International RE Conference, Workshop on Requirements Engineering for High-Availability Systems (RHAS)*. Software Engineering Institute (SEI), Carnegie Mellon University & IEEE Press: Paris, France, 40–48.

Lee SW, Gandhi RA, Ahn G. 2005a. Establishing trustworthiness in services of the critical infrastructure: automating the DITSCAP. In *Proceedings of the 27th IEEE International Conference on Software Engineering (ICSE '05), Workshop on Software Engineering for Secure Systems (SESS)*., Vol. 30(4). Also appeared in ACM SIGSOFT Software Engineering Notes, ACM Press: St. Louis, MO, 43–49.

Lee SW, Gandhi RA, Ahn G. 2005b. Security requirements driven risk assessment for critical infrastructure information systems. In *Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS 05), Requirements Engineering (RE'05)*. IEEE Computer Society: Paris, France, August.

Lee SW, Yavagal D. 2005. GenOM User's Guide V2.0. Technical Report TR-NiSE-05-05, Knowledge Intensive Software Engineering Research Group, Department of Software and Information Systems, UNC, Charlotte.

Lekkas D, Spinellis D. 2005. Handling and reporting security advisories: a scorecard approach. *IEEE Security and Privacy* **3**(4): 32–41.

McGuinness D, van Harmelen F (eds). 2004. *OWL Web Ontology Language Overview*. W3C Recommendation: http://www.w3.org/TR/owl-features/.

Offen R. 2002. Domain understanding is key to successful to system development. *Requirements Engineering Journal*, Vol. **7**(3). Springer-Verlag: London, UK, 172–175.

Ross R, Swanson M, Stoneburner G, Katzke S, Johnson A. 2004. *Guide for the Security Certification and Accreditation of Federal Information Systems*. NIST Special Publication #800-37, Gaithersburg, MD, USA.

SSE-CMM. 2003. System Security Engineering Capability Maturity Model (SSE CMM) Model Description Document, Version 3.0. http://www.sse-cmm.org.

Sutcliffe A. 1998. Scenario-based requirements analysis. *Requirements Engineering Journal*, Vol. 3(1). Springer-Verlag: New York, USA, 48–65.

Swanson M. 2001. *Security Self-assessment Guide for Information Technology Systems*. NIST Special Publication #800-26, Gaithersburg, MD, USA.

Swanson M, Bartol N, Sabato J, Hash J, Graffo L. 2003. *Security Metrics Guide for Information Technology Systems*. NIST Special Publication #800-55, Gaithersburg, MD, USA.

Vaughn RB, Henning R, Siraj A. 2003. Information assurance measures and Metrics–state of practice and proposed taxonomy. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Hawaii, USA, 331–340.

Xacta. 2004. Web C&A™ Reference Manual, Version 4.0. Service Pack 2. http://www.xacta.com/.

DOI: 10.1002/spip