International Journal of Software Engineering
and Knowledge Engineering
Vol. 16, No. 6 (2006) 851–884
© World Scientific Publishing Company

World Scientific
www.worldscientific.com

BUILDING DECISION SUPPORT PROBLEM DOMAIN ONTOLOGY FROM NATURAL LANGUAGE REQUIREMENTS FOR SOFTWARE ASSURANCE

SEOK-WON LEE*, DIVYA MUTHURAJAN†, ROBIN A. GANDHI‡,
DEEPAK YAVAGAL§ 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 28223-0001, USA
* *seoklee@uncc.edu*
† *dmuthura@uncc.edu*
‡ *rgandhi@uncc.edu*
§ *dsyavaga@uncc.edu*
¶ *gahn@uncc.edu*
http://nise.sis.uncc.edu

The process of engineering software-intensive systems that comply with their Certification and Accreditation (C&A) requirements involves many critical decision-making activities for the related stakeholders. Considering the exhaustive nature of C&A activities together with the complexity of software-intensive systems, effective decision making relies heavily on the ways to understand and structure the problem domain concepts concerning decision points for interpretation, applicability, scope, evaluation, and impact of the enforced C&A requirements. These decision points are further complicated by natural language specifications of inherently non-functional C&A requirements scattered across multiple regulatory documents with complex interdependencies at different levels of abstractions in the organizational hierarchy, which often result in subjective interpretations and non-standard implementations of the C&A process. To address these issues, we define a systematic methodology using novel techniques from software Requirements Engineering (RE) and knowledge engineering for understanding and structuring the problem domain concepts based on a uniform representation format that promotes common understanding among stakeholders. Specifically, we use advanced ontological engineering techniques driven by theoretical RE foundations to systematically elicit, model, understand, and analyze problem domain concepts concerning significant and difficult decision points throughout the C&A process. We demonstrate the appropriateness of our methodology in creating decision support problem domain ontology using several examples derived from our experiences on automating the Department of Defense Information Technology Security C&A Process (DITSCAP).

Keywords: Software-intensive systems; requirements engineering; certification and accreditation; critical infrastructure protection; ontological engineering; decision making.

1. Introduction

Software-intensive systems are increasingly supporting various critical functions of computing, communications, and information processing. Consequently they are subject to additional dependability requirements for availability, continuity, performance, security, and safety to promote the high level of trustworthiness in their behavior as required by stringent regulatory C&A standards. The goal of C&A processes is to determine that the target system meets the established set of accreditation requirements and will continue to maintain the accredited posture throughout its lifecycle. However, achieving such goals is not straightforward as we take into account the complexity of software-intensive systems together with a long and exhaustive process of documentation and analysis based on C&A activities.

Software-intensive systems are clusters of closely interdependent *systems of systems* with interdependencies among themselves as well as with their operational environment to satisfy the required behavior. In addition, diverse socio-technical operational environments contribute to multiple viewpoints that introduce different semantics and levels of abstraction in specifying the functions of and constraints on these systems. Therefore, to understand, predict, and control the global consequences of numerous inherently non-functional C&A requirements on emergent software behavior, the related decision-making activities frequently engage in difficult decision points for their interpretation, applicability, scope, evaluation, and impact. We define decision support as the process of systematically understanding and structuring problem related domain concepts, properties, and their interdependencies in objective, traceable, repeatable, and justifiable ways to provide valuable insights into these decision points.

Infrastructure-centric standard C&A processes are often enacted through multiple regulatory documents, with each document partially expressing concerns from different levels in the organizational hierarchy. These documents comprise natural language specifications of C&A requirements with heavy cross-referencing to other regulatory and guidance documents at different levels in the organizational hierarchy. In addition, natural language non-functional C&A requirements have varying levels of abstractions in their specification that address artifacts from multiple dimensions related to a socio-technical environment. These factors make it difficult to ensure objectivity, repeatability, and justifiability of the criteria adopted at critical decision points throughout the C&A process. As a result, despite enormous efforts and resources currently spent on C&A processes, their effectiveness in the real world is limited [51, 50] and their results under-utilized. From our experience on automating the DITSCAP [44, 45], enforcing C&A requirements involves critical decision points, as shown in Table 1.

These are significant decision points throughout DITSCAP, however the related problem domain concepts that provide a context to understand, implement, and evaluate them are dispersed across multiple documents in the organizational hierarchy. These issues together with the non-functional nature of security requirements

Table 1. Critical DITSCAP decision points.

Decision Point Categories	**Decision Points**
Interpretation Applicability Scope	**DP1.** Which regulatory documents should be used to identify C&A requirements? **DP2.** What level in the organizational hierarchy are the requirements identified? **DP3.** What are the types of the systems (For example, a major application or general support system) addressed by security requirements? **DP4.** Is the identified set of applicable requirements complete?
Interpretation Applicability	**DP5.** What interdependencies exist between the applicable set of requirements and how to identify them? **DP6.** What redundancies exist among the requirements and how to discover them? **DP7.** Who is responsible for or affected by (stakeholders) the requirements?
Interpretation Evaluation Impact	**DP8.** What are the criteria to assess requirements compliance for the target system?
Evaluation Impact	**DP9.** Do the compliance criteria provide a complete coverage of the different dimensions addressed by a given requirement? **DP10.** What are the risks associated with the system at a particular compliance level? **DP11.** Is the system operating at an acceptable level of risk?

often lead to breakdowns in communications between stakeholders, subjective interpretations, and non-standard implementations in the real world. To address these issues, we focus our efforts on effectively supporting critical decision-making activities throughout the DITSCAP by eliciting, representing, and modeling the relevant problem domain concepts that provide the definition of a *common language* and promote a *common understanding* among the stakeholders involved. Specifically, we use advanced ontological engineering techniques driven by theoretical RE foundations [49] to systematically elicit, model, understand, and analyze problem domain concepts concerning significant and difficult decision points throughout the C&A process. To demonstrate the appropriateness of our approach, in this paper we outline a step-wise methodology to systematically build decision support Problem Domain Ontology (PDO) from multiple DITSCAP-oriented security requirements documents. The PDO captures diverse aspects of natural language requirements along with relevant domain knowledge based on a uniform representation format offered by rich ontological engineering processes. In this paper although we use several domain-specific examples from DITSCAP to depict a detailed application of our methodology, the examples also motivate several domain-independent heuristics for systematically eliciting and modeling the problem domain concepts related to requirements that are dispersed across several documents/sources.

Organization of the paper is as follows. In Sec. 2, we discuss the related work in conceptual domain modeling that has been widely used in the software requirements engineering field. In Sec. 3, we provide the background information necessary to understand DITSCAP and the objectives for its automation. Section 4, along with the rationales behind our approach, outlines a step-wise methodology to capture, model and analyze DITSCAP-oriented security requirements, related domain knowledge, user/system criteria and their interdependencies, to understand and organize DITSCAP problem domain concepts that form a decision support PDO.

Section 4 also demonstrates the effectiveness of our approach at key decision points throughout the DITSCAP as compared to the existing practices and methods. In Sec. 5, we summarize our contributions and future work.

2. Related Work

The need to understand the problem domain concepts, the interface between the "machine" and the "environment" and the nexus of causal chains that exist between them [32, 38, 7], are critical to gain assurance of predictable and trustworthy software system behavior. We believe that such knowledge should be propagated and maintained throughout the lifecycle of a software-intensive system. This is even more relevant for C&A processes such as DITSCAP having a lifecycle approach and specific focus on requirements, which are "relationships" to be maintained or established in the problem domain [32]. From a RE perspective, to capture the real-world goals for the functions of constraints on a software system and to reason about them, popular modeling methods of goal-driven approaches [3], viewpoints-oriented approaches [17, 18], scenario-based approaches [8, 2, 24] and other techniques that are a combination of them [28, 4, 9] have been developed and experimented with. However, the selection of any single method often restricts the task of understanding and structuring the problem domain based on a limited set of modeling constructs and tools that may not be appropriate for the diverse range of characteristics/constraints required for decision making activities related to software-intensive systems.

The need to understand and model the problem domain has also been realized by research initiatives for integrating functional and non-functional aspects of the system. The Language Extended Lexicon (LEL) approach [31] supports the elicitation and representation of concepts, based on natural language processing. However, the LEL uses simple hypertext links to represent relationships between its concepts, which lack the rich semantics required to understand and structure the problem domain. The use of LEL to construct machine understandable ontologies from the requirements engineering process has been pointed out in [26]. Mylopoulos *et al.* [23] rationalize the design process based on the contribution of various design decisions to the corresponding non-functional requirements; however, they do not identify a systematic methodology to represent, and analyze non-functional requirements with appropriate tool support for large and complex systems. Liu *et al.* [29, 30] analyze security and privacy requirements based on social relationships between problem domain actors using the i^* modeling language. They also explore techniques that assist in attacker, vulnerability, and countermeasure analysis. Giorgini *et al.* [36] further extends the i^* modeling language while focusing on modeling the entire organization including social relationships between the actors for analyzing security requirements. However, the i^* modeling language necessitates a goal and agent-based representation of the domain, which may not be appropriate for all decision-making activities. In the early and late requirements engineering stages, other conceptual modeling approaches also exist to identify illicit usage or threat scenarios using misuse/abuse cases [16, 25], abuse frames [27], or intruder anti-goals [5], but they only uncover a limited set of threats based on the current context of analysis rather than considering a holistic view of the system. In [6] Anton *et al.*, propose general privacy-goal taxonomies for understanding and analyzing website privacy requirements using a goal-driven analysis. In [52] they produce restricted natural language statements from document listing privacy policies to facilitate their modeling and analysis; however, they do not address the identification and representation of diverse characteristics associated with natural language requirements.

For critical software-intensive systems, formal methods have often been used to provide *a priori* evidence that the overall system behavior will be dependable [1]. Apart from being costly, formal approaches are not very effective to gain a common understanding between stakeholders. Conceptual modeling techniques based on the Unified Modeling Language (UML) [39], are more focused on understanding software behavior rather than understanding the problem domain. Other object-oriented domain modeling techniques [19], knowledge-based techniques [22, 40] and enterprise modeling [37, 35] also exist that have been used and experimented with. We identify that, while each conceptual modeling technique has its own advantages, the choice of a particular technique(s) must be carefully evaluated for its capability to provide the required results.

We believe that understanding, structuring, and reasoning in the problem domain should not be restricted to specific methods, which may result in the decision-making criteria becoming too narrow focused or stove-piped that it may fail to capture the key aspects required to engineer quality software-intensive systems. To address these shortcomings, we advocate the use of ontological modeling processes within our methodology to capture a diverse range of characteristics/constraints based on the needs of the problem domain through multiple RE modeling philosophies and their synergies [49]. A uniform representation format provides flexibility in the choice of a modeling technique while facilitating a common understanding and wider participation from stakeholders with diverse skill sets.

3. Background

3.1. *The DITSCAP overview*

DITSCAP is a standard security C&A process for systems operational within the Defense Information Infrastructure (DII), which supports local as well as worldwide information needs of the Department of Defense (DoD). DITSCAP defines certification in the context of information systems 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, to establish the extent to which a particular design and implementation meets a set of specified security requirements [13]. 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 lifetime commitment [21]. 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 DITSCAP application manual [12] describes these phases with associated activities in detail.

The key roles of the DITSCAP are the Program Manager, DAA, Certifier, and the User Representative 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" should be defined and agreed upon by the key roles, which is documented in the System Security Authorization Agreement (SSAA). The SSAA forms the baseline security configuration for the target system. It records the outcome of tasks and activities in each phase of the DITSCAP, which produce several C&A metrics and measures by inspecting and analyzing their units of analysis pertaining to the target system.

3.2. *Objectives for DITSCAP automation*

Practicing DITSCAP requires familiarity with several guidance documents from different levels in the organizational hierarchy such as the DITSCAP application manual, Federal laws, DoD and Department of Navy (DoN) policies and implementations, National Institute of Standards and Technology (NIST) best practices, and other directives and security requisites to identify the applicable set of security requirements. Each document usually ranges from 25 to 200 pages making it extremely difficult to comprehend their contents as well as the interdependencies between them, challenging the objectivity and repeatability of the criteria adopted for various decision-making activities. In addition, the Federal, DoD, DoN, and other organizational concerns for secure software assurance throughout the DII are scattered across multiple documents and cross-cut several requirements. These issues, together with abstract natural language specifications of non-functional security requirements often complicate the C&A process and reduce the communicability of its results. Therefore, a key objective of DITSCAP automation is to provide the definition of a common language and understanding between various stakeholders in the DITSCAP domain. Through the definition of a common language we seek to provide a framework within which various pieces of information from multiple dimensions and levels of abstraction are systematically elicited, structured, modeled, and analyzed to satisfy the primary goals and objectives of DITSCAP for software assurance.

4. A Methodology for Building Decision Support Problem Domain Ontology from Security Requirements

Security requirements enforced by the DITSCAP are rich sources of information but expressed in natural language with little or no structural regularity in their specifications. Based on the seven facets of a "complete" requirement: *Who, Where, What, When, Why, Which* and *How*, the specification of a security requirement typically requires to identify problem domain concepts related to

(1) the assets that it protects;
(2) the threats that it is driven by;
(3) the vulnerabilities that it prevents;
(4) the countermeasures that it suggests;
(5) the mission criticality that it is subject to;
(6) its source;
(7) the goal of the security requirement;
(8) the related stakeholders; and
(9) other domain-specific concepts that need to be considered [45] for creating a context that facilitates their uniform interpretation.

However, most security requirements available from DITSCAP-oriented documents do not explicitly identify or consider these concepts. Furthermore, these concepts are either missing or dispersed in multiple documents, which make it difficult for stakeholders to make effective decisions regarding their interpretation, applicability, and implementation effectiveness. Therefore, as an important step towards achieving the objectives for DITSCAP automation, we develop a systematic methodology for extracting and organizing concepts in the DITSCAP problem domain in the form of decision support PDO that provides the definition of a common language. This effort helps to produce a hierarchical organization of ontological concepts to capture several key dimensions of the problem domain with related properties and non-taxonomic dependencies among them. The ontological concepts are elicited from various sources such as users, documents, laws and regulations, domain-specific taxonomies, organizational policies, environmental constraints, etc. The resulting PDO is a machine understandable, hierarchical representation, engineered using object-oriented ontological domain modeling techniques. The inherent benefits of the PDO lie in the uniformity of its representation and its traceable rationales to promote cohesiveness between problem domain concepts from multiple dimensions at different levels of abstraction.

An overview of a step-wise methodology for creating decision support PDO from natural language documents and related domain knowledge is depicted in Fig. 1. Although the process appears to be sequential, many synergistic interactions exist between its steps. Based on this methodology, we systematically model various facets of a security requirement in the DITSCAP domain, by including structured and well defined representations of:

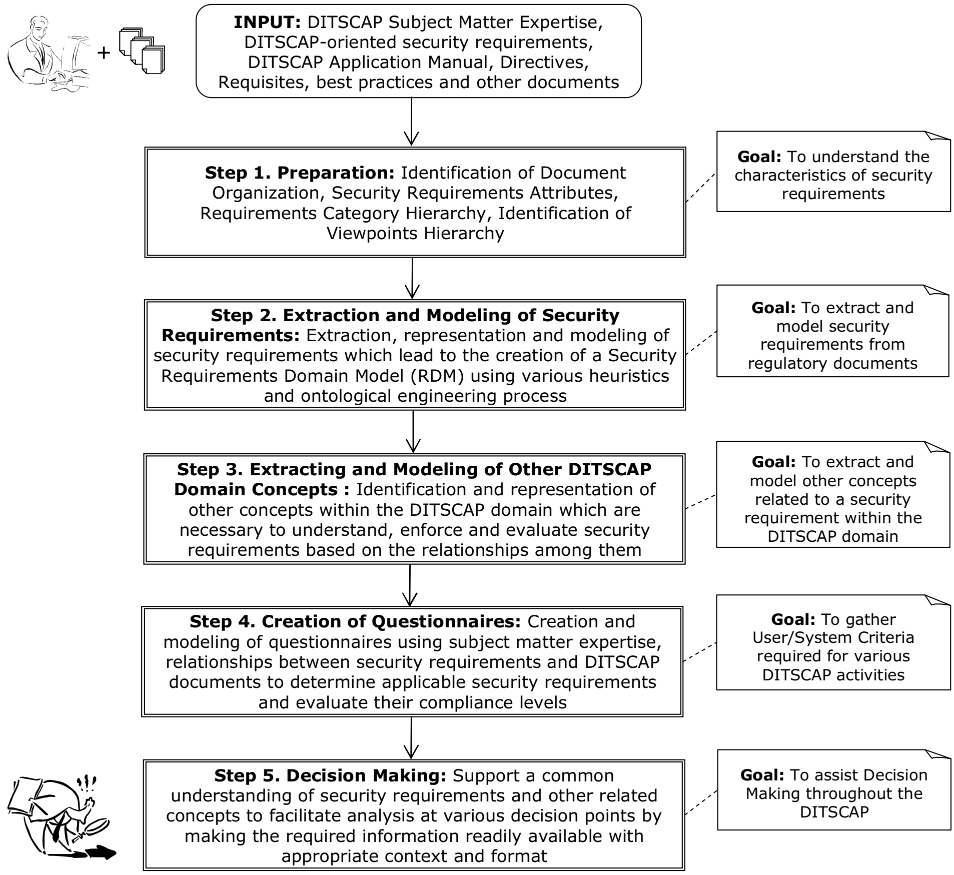

INPUT: DITSCAP Subject Matter Expertise, DITSCAP-oriented security requirements, DITSCAP Application Manual, Directives, Requisites, best practices and other documents
Step 1. Preparation: Identification of Document Organization, Security Requirements Attributes, Requirements Category Hierarchy, Identification of Viewpoints Hierarchy
Goal: To understand the characteristics of security requirements
Step 2. Extraction and Modeling of Security Requirements: Extraction, representation and modeling of security requirements which lead to the creation of a Security Requirements Domain Model (RDM) using various heuristics and ontological engineering process
Goal: To extract and model security requirements from regulatory documents
Step 3. Extracting and Modeling of Other DITSCAP Domain Concepts : Identification and representation of other concepts within the DITSCAP domain which are necessary to understand, enforce and evaluate security requirements based on the relationships among them
Goal: To extract and model other concepts related to a security requirement within the DITSCAP domain
Step 4. Creation of Questionnaires: Creation and modeling of questionnaires using subject matter expertise, relationships between security requirements and DITSCAP documents to determine applicable security requirements and evaluate their compliance levels
Goal: To gather User/System Criteria required for various DITSCAP activities
Step 5. Decision Making: Support a common understanding of security requirements and other related concepts to facilitate analysis at various decision points by making the required information readily available with appropriate context and format
Goal: To assist Decision Making throughout the DITSCAP

Fig. 1. Step-wise methodology to prepare decision support PDO from DITSCAP-oriented regulatory documents.

(1) A Requirements Domain Model (RDM) that hierarchically organizes requirements categories with leaf-node security requirements extracted from DITSCAP-oriented regulatory documents;
(2) A viewpoints hierarchy that captures different perspectives and related stakeholders of a security requirement;
(3) A risk assessment taxonomy that gathers risk factors from a broad spectrum of perceived risk sources in the DITSCAP domain;
(4) Overall DITSCAP process aspect knowledge captured as a hierarchy of goals with leaf-node questionnaires to gather user/system criteria;
(5) Meta-knowledge about information learned from network discovery/monitoring tools; and
(6) Interdependencies between various concepts, in the DITSCAP PDO.

In the following sub-sections we now elaborate on each of the steps shown in Fig. 1. For the scope of this paper, during each step we focus on the modeling techniques and heuristics involved in the creation of a RDM based on DITSCAP-oriented regulatory documents. However, we briefly discuss other models in Sec. 4.3, and further details about them can be found in [43].

4.1. *STEP 1: The preparation step*

For effectively supporting various decision points throughout the C&A process, it is important to consider the needs and characteristics of the problem domain. To perceive such needs, the goal of the preparation step is to understand the organization and contents of documents relevant to the scope of the PDO. Specifically, we identify and model document interdependencies, characterize the types of requirements with corresponding attributes in the problem domain, and identify various viewpoints to systematically understand and organize the diversity associated with natural language requirements.

4.1.1. *Understanding document organization*

The relationships that exist between various DITSCAP-oriented documents are used to systematically guide subsequent extraction and categorization of requirements from them. Identifying these relationships provide an initial structure to the collection of documents selected to be within the scope of the PDO as well as uncover the interdependencies between requirements dispersed across multiple documents. To facilitate their discovery, we identify several steps/heuristics for organizing a collection of documents. To identify generic document categories, firstly, gain a high-level comprehension of the documents through their content and usage analysis. It usually involves analyzing the scope of the document and its applicability. Secondly, group the documents into generic categories based on their purpose. Document purpose can be identified by asking the question, for example: "*Why was the document created?*" Such a generic document categorization also serves as a basis to identify specific document hierarchies as required by the needs of the problem domain. As the documents categories become apparent, relationships between them are identified based on the cross-referential structure of their member documents. Our experience with DITSCAP-oriented regulatory documents suggests that sets of documents within a category usually share a significant set of relationships with corresponding sets of documents in another category.

Analyzing the organization of DITSCAP-oriented documents within our scope, we determine a hierarchical relationship between the generic Federal-level documents, domain-spanning requirements from DoD and DoN policy/instruction documents, and site/agency specific DoD and DoN implementation guidance documents. We capture and model the interdependencies among these documents that are from different levels in the DoD organizational hierarchy, using relationships with well-defined semantics as shown in Fig. 2. For example, the "comply_to" relationship in

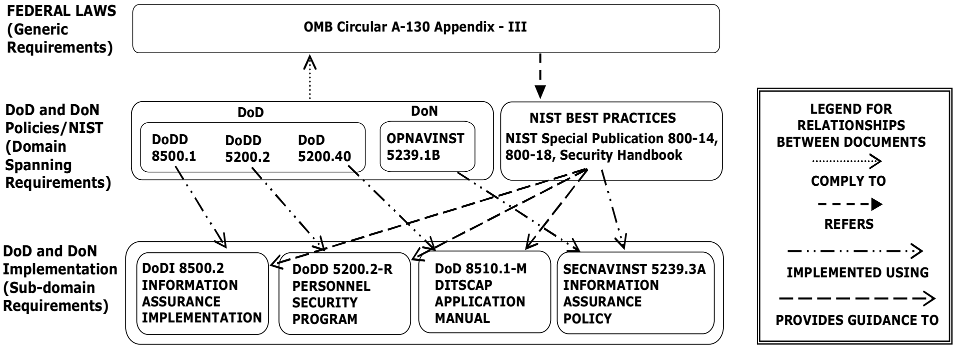

FEDERAL LAWS (Generic Requirements)
OMB Circular A-130 Appendix - III
DoD and DoN Policies/NIST (Domain Spanning Requirements)
DoD
DoN
DoDD 8500.1
DoDD 5200.2
DoD 5200.40
OPNAVINST 5239.1B
NIST BEST PRACTICES
NIST Special Publication 800-14, 800-18, Security Handbook
DoD and DoN Implementation (Sub-domain Requirements)
DoDI 8500.2 INFORMATION ASSURANCE IMPLEMENTATION
DoDD 5200.2-R PERSONNEL SECURITY PROGRAM
DoD 8510.1-M DITSCAP APPLICATION MANUAL
SECNAVINST 5239.3A INFORMATION ASSURANCE POLICY
LEGEND FOR RELATIONSHIPS BETWEEN DOCUMENTS
COMPLY TO
REFERS
IMPLEMENTED USING
PROVIDES GUIDANCE TO

Fig. 2. A document organization diagram in the DITSCAP domain.

Fig. 2 indicates that the DoD and DoN policy/instruction documents have requirements which fulfill the policies of the Federal-level documents. Similarly, the "refers" relationship indicates that the NIST special publication documents are referred to by Federal-level policy documents.

4.1.2. *Identification of requirements categories*

Within the PDO, a RDM characterizes the types of 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 their exploration to be conservative in nature i.e. to be more inclusive rather than exclusive. The hierarchical representation of the RDM provides a way to establish the extent to which the higher-level requirements are satisfied through specific policies, procedures, or technical rationales in the actual environment, thus avoiding subjective interpretations of requirements. The RDM also helps to systematically aggregate security requirements and reason about them at different levels of abstractions from multiple dimensions, while providing a comprehensive coverage and traceability of requirements expressed across multiple regulatory documents.

The creation of a RDM is iterative and based on top-down, middle-out, as well as bottom-up approaches to identify requirements categories as they are elicited from multiple documents/sources. However, in the preparation step, when only a limited overview of the problem domain is available, we adopt a top-down goal decomposition approach to produce an initial requirements category hierarchy that facilitates the creation of a RDM in the later stages of the methodology. The requirements category hierarchy also utilizes the relationships identified in the document organization diagram, as shown in Fig. 2, to identify security requirement categories at different abstraction levels.

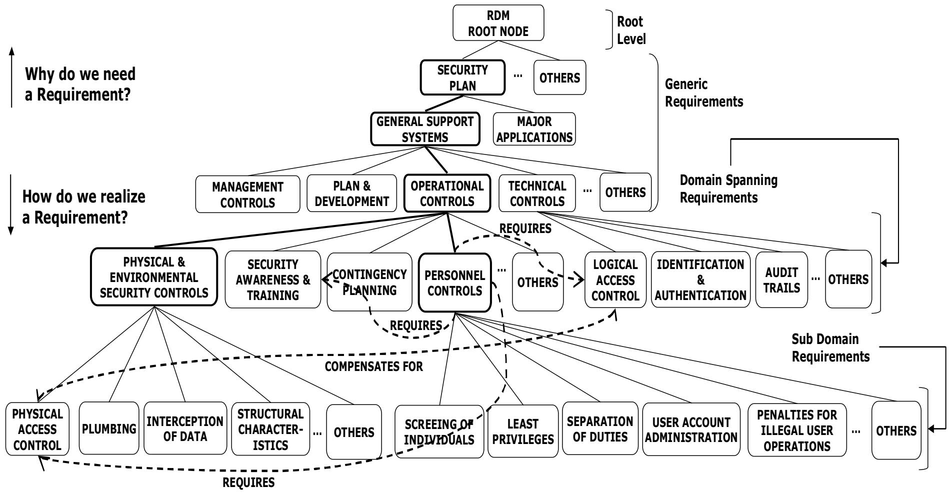
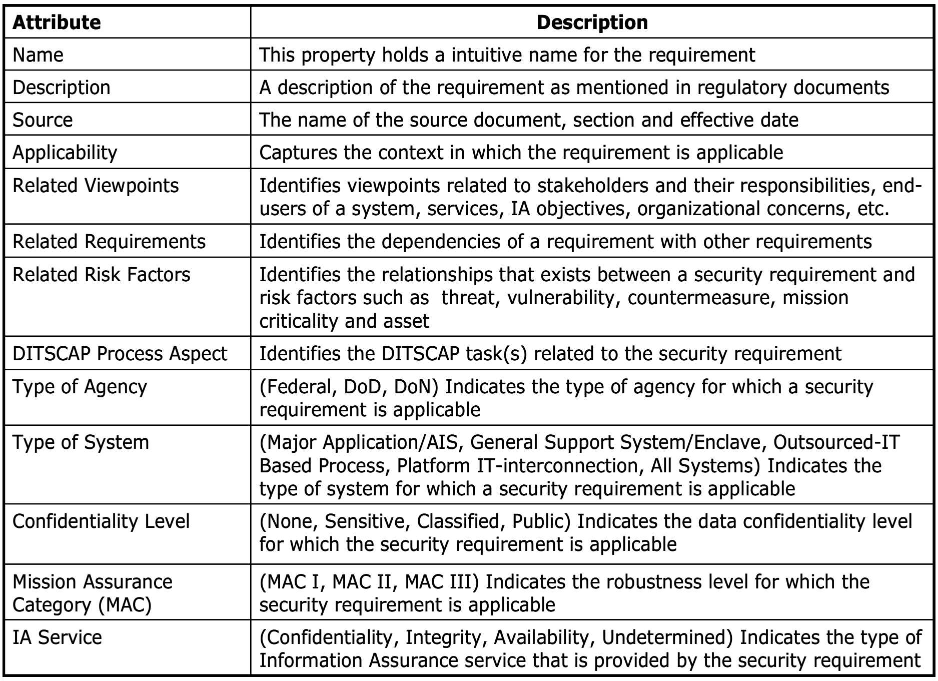


Fig. 3. Partial security requirements category hierarchy.

During the preparation step, creating a security requirements category hierarchy can be difficult without knowing *why* we need to build it and *what* to start with. Therefore, as a first step, we identify the high level goals for the types of requirements to be extracted from documents; otherwise, the whole process quickly turns into an ad hoc approach. Following a goal-driven top-down approach [3], we select a theme for requirements extraction that starts with high-level goals expressed in high-level documents in Fig. 2 to identify the generic types of requirements sought after in the documents. A decomposition of higher-level goals into specific goals by asking the *How* questions (goal-operationalization) identifies corresponding lower-level requirements categories. Following this approach, construction of the requirements category hierarchy as shown in Fig. 3 is based on the selection of "Security Plan" as a theme/goal. The theme for a "Security Plan" is identified from requirements enforced by the Office of Management and Budget (OMB) Circular Appendix-III [30], a Federal-level document. The goals of having a "Security Plan" are operationalized by having an overview of the security requirements applicable to the target system, describing the controls in place or planned for meeting these requirements, and delineating responsibilities and expected behavior of all individuals who access the system. Based on the analysis of these requirements categories suggested by documents at different levels of abstraction, as shown in Fig. 2, we form the requirements category hierarchy as shown in Fig. 3. For example, the requirements categories of "Physical and Environmental Security Controls" and "Personnel Controls" of Fig. 3 are obtained from specific agency documents (DoDD 8500.1 [10] and DoDI 8500.2 [11]) in the document organization diagram of Fig. 2.

The initial requirements category hierarchy facilitates the extraction and organization of natural language security requirements from DITSCAP-oriented



Attribute	Description
Name	This property holds a intuitive name for the requirement
Description	A description of the requirement as mentioned in regulatory documents
Source	The name of the source document, section and effective date
Applicability	Captures the context in which the requirement is applicable
Related Viewpoints	Identifies viewpoints related to stakeholders and their responsibilities, end-users of a system, services, IA objectives, organizational concerns, etc.
Related Requirements	Identifies the dependencies of a requirement with other requirements
Related Risk Factors	Identifies the relationships that exists between a security requirement and risk factors such as threat, vulnerability, countermeasure, mission criticality and asset
DITSCAP Process Aspect	Identifies the DITSCAP task(s) related to the security requirement
Type of Agency	(Federal, DoD, DoN) Indicates the type of agency for which a security requirement is applicable
Type of System	(Major Application/AIS, General Support System/Enclave, Outsourced-IT Based Process, Platform IT-interconnection, All Systems) Indicates the type of system for which a security requirement is applicable
Confidentiality Level	(None, Sensitive, Classified, Public) Indicates the data confidentiality level for which the security requirement is applicable
Mission Assurance Category (MAC)	(MAC I, MAC II, MAC III) Indicates the robustness level for which the security requirement is applicable
IA Service	(Confidentiality, Integrity, Availability, Undetermined) Indicates the type of Information Assurance service that is provided by the security requirement

Fig. 4. Requirement attributes in the DITSCAP domain.

documents under relevant requirements categories at appropriate levels of abstraction later on in the methodology. However, during the preparation step, emphasis is on creating the requirements categories rather than extracting requirements.

4.1.3. *Identifying requirements attributes*

Well-designed attributes provide clear, concise, and structured information about requirements as compared to natural language descriptions. Therefore, the third task in the preparation step involves the creation of a suitable representation template for systematically representing security requirements and related concepts which are extracted from DITSCAP-oriented regulatory documents. Attributes for such a requirements representation template are chosen by analyzing their importance in supporting various decision-making activities. Based on the perceived decision-making needs of the DITSCAP problem domain, Fig. 4 is an initial list of attributes that keeps growing iteratively as other intuitive attributes are discovered. The given attribute list includes generic as well as DITSCAP domain-specific attributes that have been identified through our analysis. A complex conjunction of such attributes captures the diverse characteristics and constraints associated with security requirements that facilitate reasoning and analysis during decision-making activities.

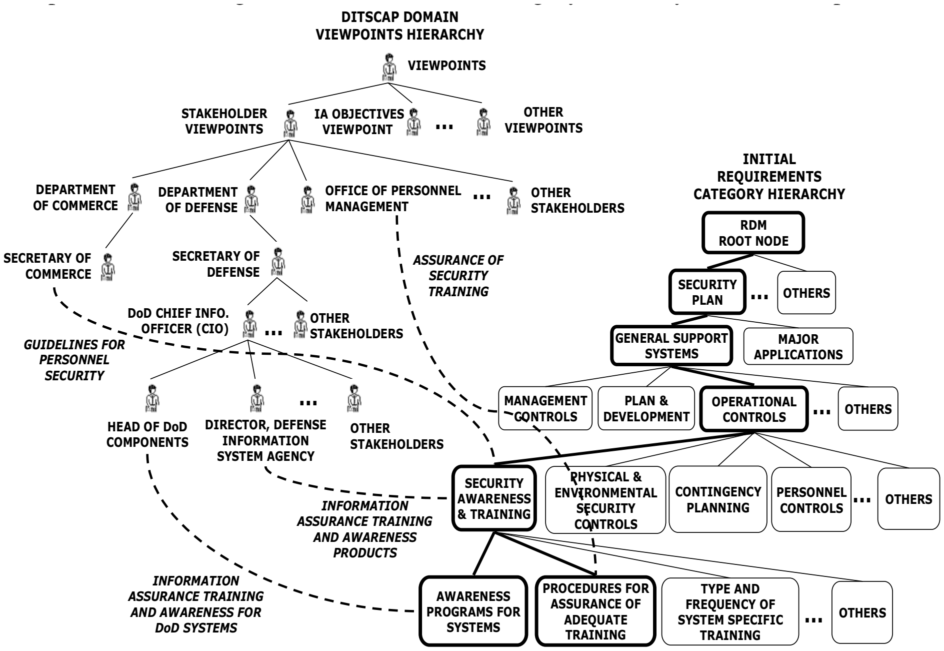

DITSCAP DOMAIN
VIEWPOINTS HIERARCHY
VIEWPOINTS
STAKEHOLDER VIEWPOINTS
IA OBJECTIVES VIEWPOINT
...
OTHER VIEWPOINTS
DEPARTMENT OF COMMERCE
DEPARTMENT OF DEFENSE
OFFICE OF PERSONNEL MANAGEMENT
...
OTHER STAKEHOLDERS
SECRETARY OF COMMERCE
SECRETARY OF DEFENSE
ASSURANCE OF SECURITY TRAINING
DoD CHIEF INFO. OFFICER (CIO)
...
OTHER STAKEHOLDERS
GUIDELINES FOR PERSONNEL SECURITY
HEAD OF DoD COMPONENTS
DIRECTOR, DEFENSE INFORMATION SYSTEM AGENCY
...
OTHER STAKEHOLDERS
INFORMATION ASSURANCE TRAINING AND AWARENESS PRODUCTS
INFORMATION ASSURANCE TRAINING AND AWARENESS FOR DoD SYSTEMS
INITIAL REQUIREMENTS CATEGORY HIERARCHY
RDM ROOT NODE
SECURITY PLAN
...
OTHERS
GENERAL SUPPORT SYSTEMS
MAJOR APPLICATIONS
MANAGEMENT CONTROLS
PLAN & DEVELOPMENT
OPERATIONAL CONTROLS
...
OTHERS
SECURITY AWARENESS & TRAINING
PHYSICAL & ENVIRONMENTAL SECURITY CONTROLS
CONTINGENCY PLANNING
PERSONNEL CONTROLS
...
OTHERS
AWARENESS PROGRAMS FOR SYSTEMS
PROCEDURES FOR ASSURANCE OF ADEQUATE TRAINING
TYPE AND FREQUENCY OF SYSTEM SPECIFIC TRAINING
...
OTHERS

Fig. 5. A partial viewpoints hierarchy in the DITSCAP domain and their relationships to security requirements categories.

4.1.4. *Identification of viewpoints hierarchy*

Requirements usually capture ideas, perspectives, and relationships at various levels of detail and they are interpreted differently from different viewpoints [14]. In the DITSCAP domain we identify several viewpoints which map to classes of end-users of a system, services, DITSCAP stakeholders, information assurance objectives, and other organizational concerns such as training and awareness to organize the diverse requirements types specified in DITSCAP-oriented regulatory documents. To provide a systematic and controlled approach for identifying viewpoints, we advocate the use of the VORD [15] viewpoints class template. Based on our analysis, a partial viewpoints hierarchy in the DITSCAP domain is shown in Fig. 5. The dotted lines in Fig. 5 capture well-defined relationships between the stakeholder viewpoints and the categories of the requirements category hierarchy. For example, the high-level organizational stakeholder viewpoints of the Department of Commerce (DoC), Department of Defense (DoD), and Office of Personnel Management (OPM) as shown in Fig. 5, are related to the generic-level requirements categories identified from the OMB Circular Appendix-III [30] and DODD 8500.1 [10] documents. During the preparation step, we use a viewpoint representation template to capture various characteristics of a viewpoint with attributes:

(1) *Viewpoint Name*: This field briefly identifies the role of a viewpoint;

(2) *Viewpoint Description*: This field captures a detailed description for the viewpoint; and

(3) *Dependant Requirement or Viewpoint*: This field captures the dependencies between a viewpoint and the categories of the requirements category hierarchy or other viewpoints.

4.2. *STEP 2: Security requirements extraction and modeling step*

The next step in the methodology outlines the heuristics and techniques involved in building a RDM. This step includes extracting and modeling natural language security requirements specified in regulatory documents, related viewpoints, and identifying interdependencies between the extracted requirements. The modeling activities are performed in parallel with requirements extraction activities using object-oriented ontological domain modeling techniques. The synergy between extraction and modeling activities also helps the ontology author and/or Subject Matter Experts (SMEs) to effectively manage large amounts of information in natural language documents.

4.2.1. *RDM categorization and requirements extraction*

The requirements category hierarchy and attributes available from the preparation step are applied as a template for extracting security requirements from each DITSCAP-oriented regulatory document. However, before extracting security requirements it is necessary to tailor the initial requirements category hierarchy according to the types of categories available from each regulatory document. Several iterations are required to form the RDM categories as shown in Fig. 6 following an incremental approach that iterates with the initial requirements category hierarchy being applied to each document.

The generic set of categories provided by the RDM also promotes consistency between requirements extracted from multiple documents. For example in Fig. 6, the sub-categorization of "Security Controls" category is consistent across the Federal, DoD, and DoN categorizations to provide consistency and traceability between requirements extracted from respective agency documents. The RDM categories along with the attributes identified in Sec. 4.1.3 provide appropriate placeholders for representing security requirements extracted from various regulatory documents. As an example of extracting security requirements and their characteristics/constraints from natural language documents, consider the security requirements excerpts shown in Fig. 7. Documents in Fig. 7 are organized hierarchically based on the document organization diagram as shown in Fig. 2. From the security requirement description labeled as "1", we identify the security requirements category of "Screen Individuals" as a sub-category of the "Personnel Security" category in the RDM. In addition, the extracted security requirements are also annotated with attributes that are available by analyzing their natural language descriptions

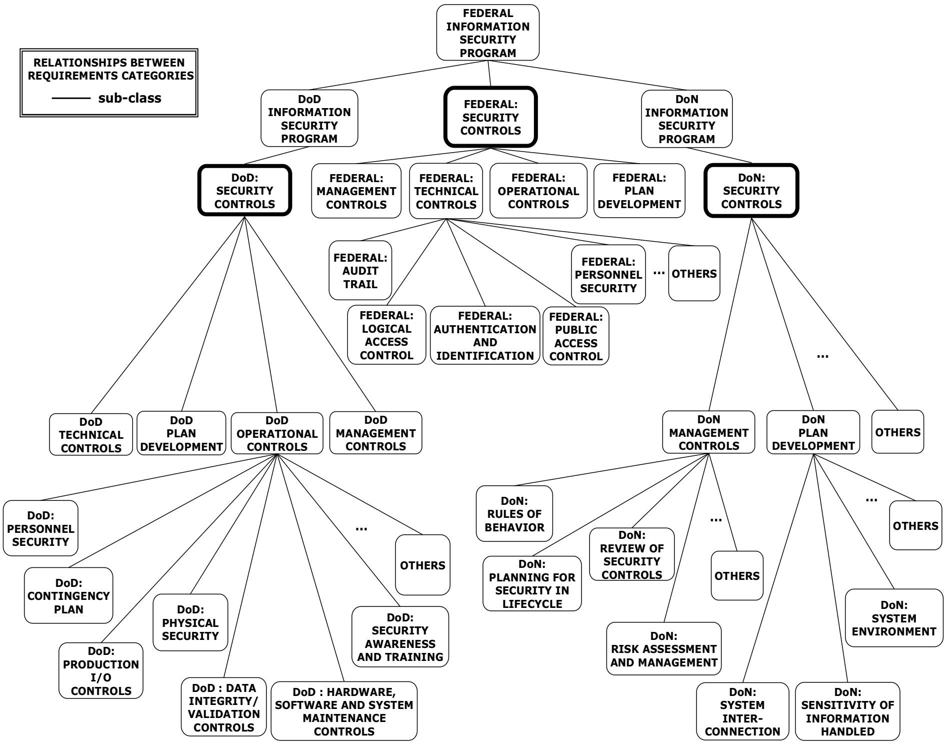

RELATIONSHIPS BETWEEN REQUIREMENTS CATEGORIES
sub-class
FEDERAL INFORMATION SECURITY PROGRAM
DoD INFORMATION SECURITY PROGRAM
FEDERAL: SECURITY CONTROLS
DoN INFORMATION SECURITY PROGRAM
DoD: SECURITY CONTROLS
FEDERAL: MANAGEMENT CONTROLS
FEDERAL: TECHNICAL CONTROLS
FEDERAL: OPERATIONAL CONTROLS
FEDERAL: PLAN DEVELOPMENT
DoN: SECURITY CONTROLS
FEDERAL: AUDIT TRAIL
FEDERAL: PERSONNEL SECURITY
...
OTHERS
FEDERAL: LOGICAL ACCESS CONTROL
FEDERAL: AUTHENTICATION AND IDENTIFICATION
FEDERAL: PUBLIC ACCESS CONTROL
...
DoD TECHNICAL CONTROLS
DoD PLAN DEVELOPMENT
DoD OPERATIONAL CONTROLS
DoD MANAGEMENT CONTROLS
DoN MANAGEMENT CONTROLS
DoN PLAN DEVELOPMENT
OTHERS
DoD: PERSONNEL SECURITY
...
OTHERS
DoD: CONTINGENCY PLAN
DoD: PHYSICAL SECURITY
DoD: SECURITY AWARENESS AND TRAINING
DoD: PRODUCTION I/O CONTROLS
DoD : DATA INTEGRITY/ VALIDATION CONTROLS
DoD : HARDWARE, SOFTWARE AND SYSTEM MAINTENANCE CONTROLS
DoN: RULES OF BEHAVIOR
DoN: REVIEW OF SECURITY CONTROLS
...
OTHERS
DoN: PLANNING FOR SECURITY IN LIFECYCLE
DoN: RISK ASSESSMENT AND MANAGEMENT
...
OTHERS
DoN: SYSTEM ENVIRONMENT
DoN: SYSTEM INTER- CONNECTION
DoN: SENSITIVITY OF INFORMATION HANDLED

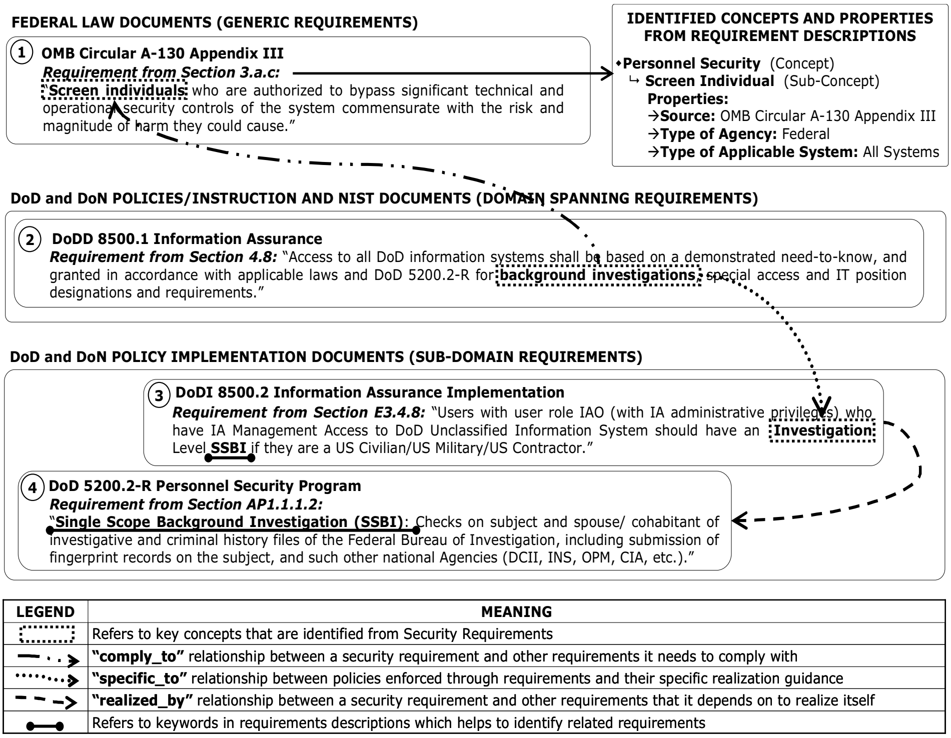
Fig. 6. The DITSCAP requirements domain model.

and the related domain knowledge of SMEs. Attributes for the security requirement labeled as "1" in Fig. 7, identify its name, source, and its applicability to the Federal agency for all types of systems. In addition, other *missing* attributes such as related risk factors, viewpoints etc. act as triggers for identifying the missing information from related documents or through SMEs.

During the requirements extraction activities we faced several problems related to consistency, completeness, redundancy, etc., which are typical defects for natural language requirements documents. We now discuss some frequently encountered defects during our requirements extraction activities and present heuristics to overcome them.

(i) Requirements descriptions are often long and verbose. If such descriptions address more than one security requirement category then decompose the description into separate requirements. The decompositions provide focused attention for the involved stakeholders and offer ease of evaluation for requirement compliance. However, decompositions that tend to change the meaning/context of the requirement as a whole should be avoided.

(ii) Requirement descriptions have varying levels of abstraction. Such requirements



FEDERAL LAW DOCUMENTS (GENERIC REQUIREMENTS)

1 OMB Circular A-130 Appendix III
Requirement from Section 3.a.c:
"Screen individuals: who are authorized to bypass significant technical and operational security controls of the system commensurate with the risk and magnitude of harm they could cause."

IDENTIFIED CONCEPTS AND PROPERTIES FROM REQUIREMENT DESCRIPTIONS
- **Personnel Security** (Concept)
  - ↳ **Screen Individual** (Sub-Concept)
  - **Properties:**
  - →**Source:** OMB Circular A-130 Appendix III
  - →**Type of Agency:** Federal
  - →**Type of Applicable System:** All Systems

DoD and DoN POLICIES/INSTRUCTION AND NIST DOCUMENTS (DOMAIN SPANNING REQUIREMENTS)

2 DoDD 8500.1 Information Assurance
Requirement from Section 4.8: "Access to all DoD information systems shall be based on a demonstrated need-to-know, and granted in accordance with applicable laws and DoD 5200.2-R for background investigations, special access and IT position designations and requirements."

DoD and DoN POLICY IMPLEMENTATION DOCUMENTS (SUB-DOMAIN REQUIREMENTS)

3 DoDI 8500.2 Information Assurance Implementation
Requirement from Section E3.4.8: "Users with user role IAO (with IA administrative privileges) who have IA Management Access to DoD Unclassified Information System should have an Investigation Level SSBI if they are a US Civilian/US Military/US Contractor."

4 DoD 5200.2-R Personnel Security Program
Requirement from Section AP1.1.1.2:
"Single Scope Background Investigation (SSBI): Checks on subject and spouse/ cohabitant of investigative and criminal history files of the Federal Bureau of Investigation, including submission of fingerprint records on the subject, and such other national Agencies (DCII, INS, OPM, CIA, etc.)."

LEGEND	MEANING
⋯	Refers to key concepts that are identified from Security Requirements
—·→	**"comply_to"** relationship between a security requirement and other requirements it needs to comply with
···→	**"specific_to"** relationship between policies enforced through requirements and their specific realization guidance
– –→	**"realized_by"** relationship 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

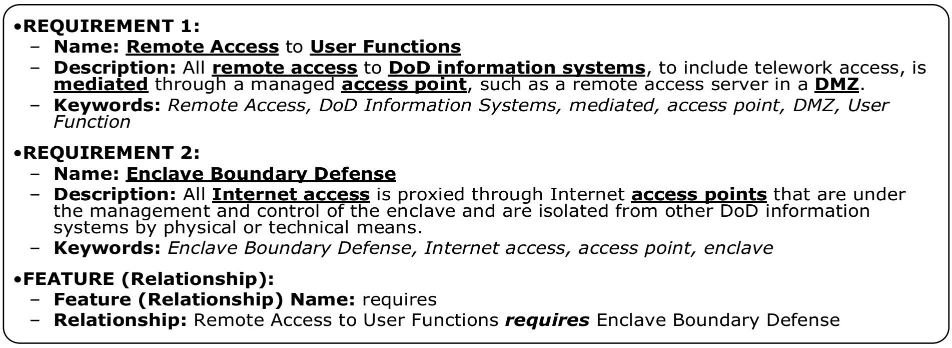
Fig. 7. Extraction of security requirements, categories, properties, and their interdependencies from DITSCAP-oriented regulatory documents.

are appropriately decomposed and placed at proper level of abstraction in the RDM.

(iii) Requirements are applicable to more than one requirement category. In such cases, the requirement is placed in a category that is most applicable based on the domain knowledge of the SMEs.

(iv) Multiple requirements represent the same requirement but using different terminologies. Such redundancies have been observed between requirements extracted from a single document, different documents of the same agency or different agencies. For systematically identifying them, we discover mappings between the terminologies used in high-level documents and the corresponding terminologies used in lower-level documents based on the document organization diagram of Fig. 2.

(v) The same requirement may be expressed with different connotations. In such cases, the relationships of the requirement with other domain concepts within the DITSCAP PDO (for example, association with different viewpoints) should be captured in a way that suggests the dimensions along which the given requirement can be interpreted and analyzed.



•**REQUIREMENT 1:**
- **Name: Remote Access** to **User Functions**
- **Description:** All **remote access** to **DoD information systems**, to include telework access, is **mediated** through a managed **access point**, such as a remote access server in a **DMZ**.
- **Keywords:** *Remote Access, DoD Information Systems, mediated, access point, DMZ, User Function*

•**REQUIREMENT 2:**
- **Name: Enclave Boundary Defense**
- **Description:** 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.
- **Keywords:** *Enclave Boundary Defense, Internet access, access point, enclave*

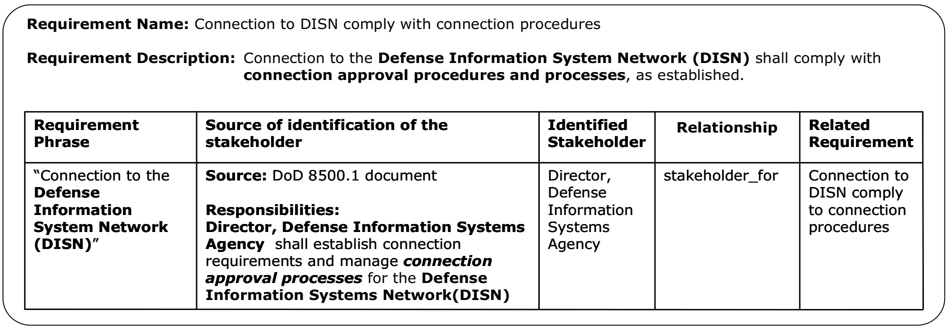
•**FEATURE (Relationship):**
- **Feature (Relationship) Name:** requires
- **Relationship:** Remote Access to User Functions ***requires*** Enclave Boundary Defense

Fig. 8. Identifying dependencies using keywords.

4.2.2. *Identifying interdependencies among requirements*

Identifying the relationships that exist among security requirements extracted from multiple sources exposes their crosscutting nature and promotes a shared understanding of the decision-making criteria used to interpret and evaluate them. Figure 7 identifies several such interdependencies based on well-defined semantics, 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. The document organization diagram in Fig. 2 also provides guidance for identifying related security requirements across documents. Such interdependencies are shown in Fig. 7 through the "comply_to" and "specific_to" relationships between security requirements.

The interdependencies between security requirements are discovered either from their specifications or through the domain knowledge of SMEs. In the former case, interdependencies are systematically discovered by a thorough keyword analysis of natural language security requirements specifications. Keywords for each requirement are identified by analyzing their names, descriptions, or parent categories in the RDM. Once the keywords have been identified, security requirements or requirements categories with a similar set of keywords are analyzed for interdependencies. To understand this process, consider the requirements shown in Fig. 8. Interdependencies between the requirements for "Remote Access to User Functions" and "Enclave Boundary Defense" are discovered based on the keyword "access point" which is common to both requirements. Based on the semantics of this interdependency, a "requires" relationship is determined such that "Remote Access to User Functions" requirement requires the "Enclave Boundary Defense" requirement to realize itself. Interdependencies can be also identified between a requirement and a set of requirements under a particular category of the RDM.



Requirement Name: Connection to DISN comply with connection procedures

Requirement Description: Connection to the **Defense Information System Network (DISN)** shall comply with **connection approval procedures and processes**, as established.

Requirement Phrase	Source of identification of the stakeholder	Identified Stakeholder	Relationship	Related Requirement
"Connection to the **Defense Information System Network (DISN)**"	**Source:** DoD 8500.1 document **Responsibilities:** **Director, Defense Information Systems Agency** shall establish connection requirements and manage ***connection approval processes*** for the **Defense Information Systems Network(DISN)**	Director, Defense Information Systems Agency	stakeholder_for	Connection to DISN comply to connection procedures

Fig. 9. Identifying stakeholder viewpoints from requirements specifications.

4.2.3. *Extracting viewpoints*

Identifying viewpoints during requirements extraction and modeling, provides additional information that helps to understand requirements from different perspectives. Figure 9 provides an example of identifying stakeholder viewpoints from DITSCAP-oriented documents. Such viewpoints are usually discovered by associating requirement descriptions to the responsibilities identified for various stakeholders. Figure 9 identifies the stakeholder viewpoint of "Director of the Defense Information Systems Agency" to manage the connection approval processes for the Defense Information System Network (DISN), based on the requirement and responsibilities descriptions given in DoDD 8500.1 [29] document.

4.2.4. *Modeling the PDO using ontological engineering processes*

To support the representation and modeling of rich knowledge structures required by the PDO, various ontological engineering processes are provided by the GENeric Object Model (GenOM) [47] 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 [53] as well as the Web Ontology Language (OWL) representation [34] format. 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 associated with an inference engine [20], which

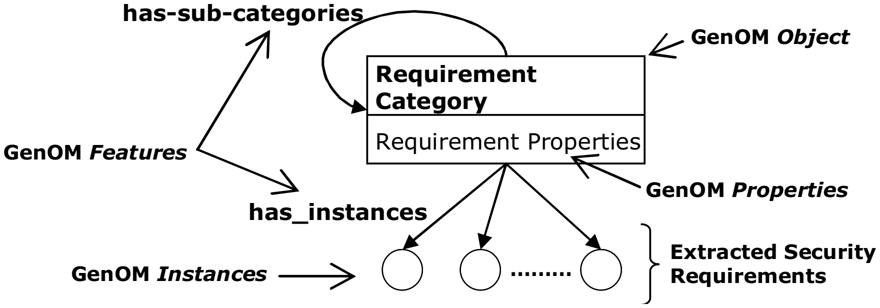


Fig. 10. GenOM representation format for the RDM.

supports reasoning based on the *Objects, Properties, Features* and *Instances* defined in its knowledge-bases. In summary, GenOM supports object modeling in its representation, usage of objects in its application model, and ability to aggregate evidence that supports the analysis of objects' behaviors (through the associated properties and relationships between objects). GenOM's rich modeling 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 problem domain.

Using various GenOM modeling constructs, Fig. 10 shows the representation format for modeling non-leaf node categories of the RDM as an *Object* hierarchy and leaf-node security requirements extracted from regulatory documents as their *Instances*. Each "Requirement Category" *Object* is also associated with various *Properties* that model the characteristics/constraints captured through requirements attributes identified in Sec. 4.1.3. *Properties* in GenOM are of type String, Set, *Object*, Boolean, Integer, or Real and are single-valued or multi-valued depending on their cardinality. The hierarchy of "Requirement Category" *Objects* is modeled using the "has-sub-categories" *Feature* that relates a parent-node in the RDM to its child-nodes. The "Requirement Category" *Objects* are related to their *Instances* using the "has-instances" *Feature*. The security requirements extracted from documents and modeled as *Instances* of "Requirement Category" *Objects*, also inherit the *Properties* associated with their parent requirement categories. The interdependencies among security requirements identified in Sec. 4.2.2 as well as with other concepts in the PDO are represented using various *Features* with well-defined semantics.

To facilitate the modeling of viewpoints related to a requirement, Fig. 11 shows the GenOM representation format to represent and model stakeholder viewpoints in the decision support PDO. The "Stakeholder Viewpoints" *Object* modeled in GenOM has three child objects, which are "Authorities," "Office," and "Department". The "Stakeholder Viewpoints" *Object* is also related to the "Requirement Category" *Object* through the "stakeholder_for" *Feature* that helps to represent the relationships between viewpoints and a set of requirements. The "stakeholder_for" *Feature* is

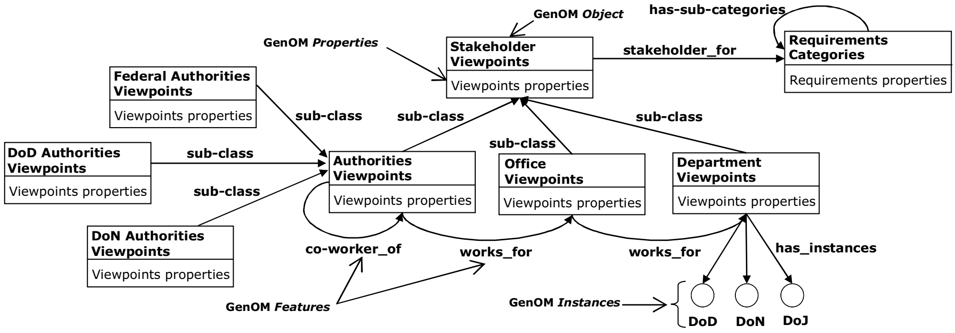


Fig. 11. GenOM representation format for stakeholder viewpoints.

also associated with the "Responsibility" *Property* to hold responsibility information for the related stakeholder viewpoint.

4.3. *STEP 3: Capturing and modeling other concepts in the DITSCAP domain*

Traditionally, software engineering practices related to the procurement, development, maintenance, and usage of software-intensive systems have focused only on the software system technical attributes, but the software system itself is embedded within an environment that caters to the real world goals of the associated users, business, and organization [41]. This concept is even more relevant for software-intensive systems as their capabilities rely heavily on the emergent behavior 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 encompassing multiple dimensions of the problem domain is inevitable to practice software engineering for software-intensive systems [49].

To focus efforts in this direction in the DITSCAP domain, we identify well-defined aspects of DITSCAP problem domain that are close to the real world goals and objectives of a software-intensive system, and carefully extract concepts related to them from DITSCAP-oriented regulatory documents, other guidance documents, and SMEs. Therefore, in addition to the requirements in the RDM and the associated viewpoints hierarchy, the decision support PDO also includes the DITSCAP process aspect as a C&A goal hierarchy, risk assessment taxonomy, network-based information discovery taxonomy and the interdependencies among these models as mentioned in Sec. 4. However, due to the scope of this paper, we briefly discuss these models and their contributions to various decision-making activities in the DITSCAP domain.

To capture the DITSCAP process aspect, C&A goals are extracted from the homogenous groupings of well-defined tasks and activities outlined in the DITSCAP Application Manual [12]. The resulting hierarchical representation of the overall C&A process systematically guides stakeholders through the DITSCAP as well as identifies the dependencies between various tasks, activities, and phases of the DITSCAP that need to be considered during decision-making activities. In addition, the traceability between C&A goals and security requirements introduces real world objectives and context into the decision-making activities. The DITSCAP decision support PDO also includes a risk assessment taxonomy, which aggregates a broad spectrum of possible categories and classification of risk related information from the DITSCAP domain. The upper level nodes of the risk assessment taxonomy consist of threat, vulnerabilities, countermeasures, asset properties, and mission criticality concepts related to risk assessment. Each high-level node is then further decomposed into specific risk categories. In addition, several non-taxonomic relationships between security requirements and risk factors are discovered from security requirements descriptions, research literatures, or SMEs. The relationships between risk factors and security requirements in the PDO support a requirements-driven risk assessment in the DITSCAP domain [45] as well as establish the necessity and sufficiency of applicable security requirements in addressing the risk factors perceived in the operational environment. Such assessments may also uncover additional sets of security requirements that were previously missing. The decision support PDO also allows comparisons between the intended and actual operational environments through the Network-based Information Discovery Taxonomy (NIDT). The NIDT aggregates information from a set of network tools and automated scripts selected to assess the compliance levels of security requirements in the actual environment where possible. The NIDT includes tool and scripts for gathering:

(1) hardware, software and firmware inventory;
(2) configurations of network devices and services; and
(3) system vulnerabilities using penetration testing.

When these concepts come together (semantic neighborhood) in the context of a security requirement, driven by goals of analysis, they help in effectively interpreting, implementing and evaluating the requirement in the real world. For example, consider the DITSCAP security requirement: "*Enclave Boundary Defense (EBBD-2): 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.*" For this requirement, we identify various related concepts within the PDO and visualize using GenOM as shown in Fig. 12.

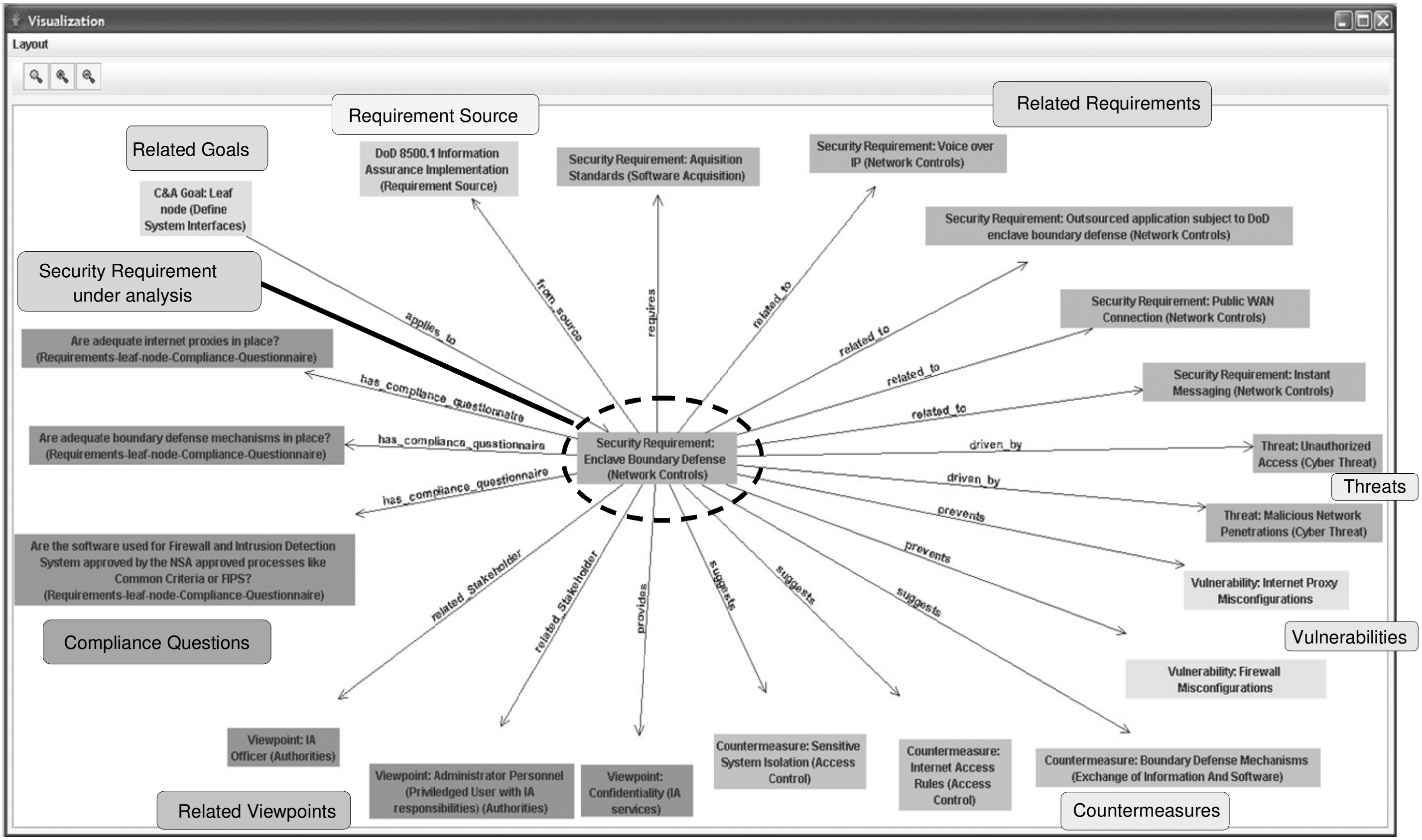

Visualization
Layout
Requirement Source
Related Requirements
Related Goals
DoD 8500.1 Information Assurance Implementation (Requirement Source)
Security Requirement: Aquisition Standards (Software Acquisition)
Security Requirement: Voice over IP (Network Controls)
C&A Goal: Leaf node (Define System Interfaces)
Security Requirement: Outsourced application subject to DoD enclave boundary defense (Network Controls)
Security Requirement under analysis
Security Requirement: Public WAN Connection (Network Controls)
Are adequate internet proxies in place? (Requirements-leaf-node-Compliance-Questionnaire)
Security Requirement: Instant Messaging (Network Controls)
Are adequate boundary defense mechanisms in place? (Requirements-leaf-node-Compliance-Questionnaire)
Security Requirement: Enclave Boundary Defense (Network Controls)
Threat: Unauthorized Access (Cyber Threat)
Threats
Threat: Malicious Network Penetrations (Cyber Threat)
Are the software used for Firewall and Intrusion Detection System approved by the NSA approved processes like Common Criteria or FIPS? (Requirements-leaf-node-Compliance-Questionnaire)
Vulnerability: Internet Proxy Misconfigurations
Compliance Questions
Vulnerabilities
Vulnerability: Firewall Misconfigurations
Viewpoint: IA Officer (Authorities)
Countermeasure: Sensitive System Isolation (Access Control)
Countermeasure: Internet Access Rules (Access Control)
Countermeasure: Boundary Defense Mechanisms (Exchange of Information And Software)
Viewpoint: Administrator Personnel (Priviledged User with IA responsibilities) (Authorities)
Viewpoint: Confidentiality (IA services)
Related Viewpoints
Countermeasures
applies_to
from_source
requires
related_to
related_to
related_to
related_to
has_compliance_questionnaire
has_compliance_questionnaire
has_compliance_questionnaire
driven_by
driven_by
prevents
prevents
related_Stakeholder
related_Stakeholder
provides
suggests
suggests
suggests

Fig. 12. GenOM visualization of a security requirement in the DITSCAP PDO and its relationships with other concepts.

4.4. *STEP 4: The questionnaire creation step*

DITSCAP is all about carefully collecting evidences regarding the target software-intensive system based on the execution of its tasks and activities to evaluate the extent to which the target system meets a set of applicable security requirements. To achieve these objectives, the DITSCAP involves critical decision points for:

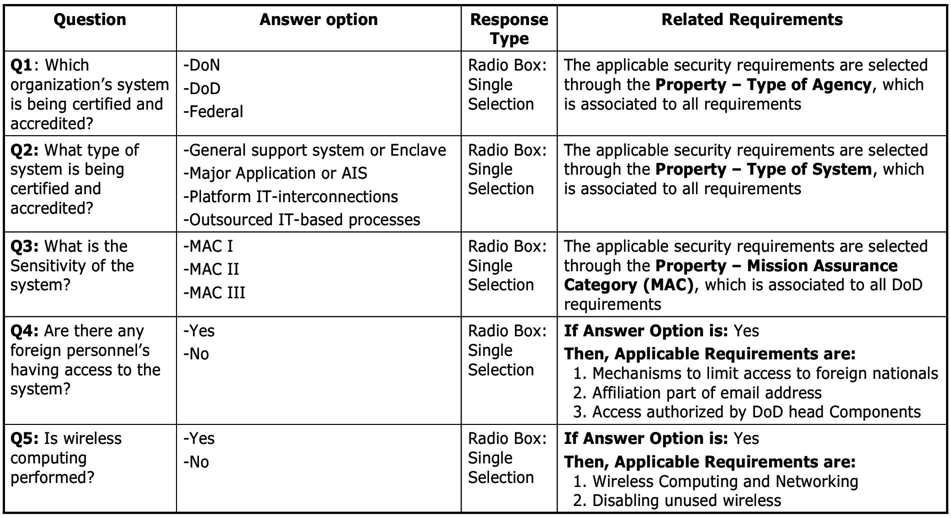
(1) Determining a complete and justifiable set of applicable security requirements for the target system; and
(2) establishing the extent to which the target system satisfies the identified set of applicable security requirements.

To address these decision points, the fourth step in our methodology involves the creation of two types of questionnaires for systematically capturing evidences that justify decision-making activities based on objective and repeatable criteria. The first questionnaire set, called the *requirements applicability questionnaire*, captures the characteristics and constraints relevant to the target system in its operational environment and maps them to the characteristics/constraints of the security requirements categories in the DITSCAP RDM to determine their applicability. The second questionnaire set, called the *requirements compliance questionnaire*, establishes well-defined metrics and measures related to the compliance levels of each security requirement for systematically evaluating the extent to which they are satisfied in the context of the target system and environment. The program manager, DAA, certifier, and the user representative are responsible for answering both questionnaires and ensuring the accuracy of the selected answer options. Although the gathered evidences may seem to be subjective, their subjectivity is avoided to an extent based on objective criteria outlined by well-defined answer options. In addition, the manually gathered evidences can also be cross-checked with the evidences gathered from the actual environment through automated means where possible.

4.4.1. *The requirements applicability questionnaire*

Following the DITSCAP, identification of an applicable set of security requirements for the target system requires sifting through multitude of DITSCAP-oriented documents, carefully scrutinizing their interdependencies, applicability, and scope. To systematically conduct this process, the requirements applicability questionnaires prune the security requirements space based on the mappings of their member questions and corresponding answer options to attributes of security requirements in the DITSCAP RDM and establish criteria for requirements applicability. Table 2 provides examples of such mappings. Following a laddering structure, requirements applicability questions are organized in a hierarchical fashion, with high-level questions (Q1, Q2 and Q3 in Table 2) selecting large sets of requirements, which are successively pruned using specific questions (Q4 and Q5 in Table 2) that are related to fewer requirements. The interdependencies between requirements in the RDM are also used to expand the set of applicable requirements by making suggestions

Table 2. Requirements applicability questionnaire examples.



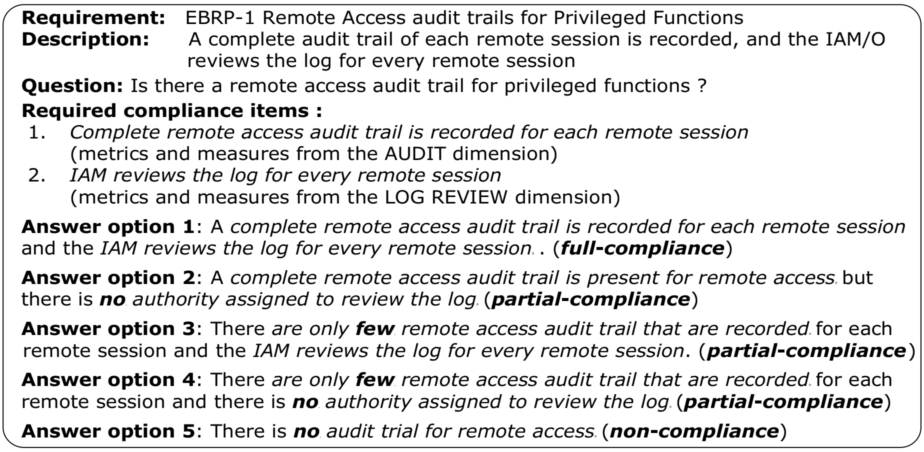
Question	Answer option	Response Type	Related Requirements
Q1: Which organization's system is being certified and accredited?	-DoN -DoD -Federal	Radio Box: Single Selection	The applicable security requirements are selected through the **Property – Type of Agency**, which is associated to all requirements
Q2: What type of system is being certified and accredited?	-General support system or Enclave -Major Application or AIS -Platform IT-interconnections -Outsourced IT-based processes	Radio Box: Single Selection	The applicable security requirements are selected through the **Property – Type of System**, which is associated to all requirements
Q3: What is the Sensitivity of the system?	-MAC I -MAC II -MAC III	Radio Box: Single Selection	The applicable security requirements are selected through the **Property – Mission Assurance Category (MAC)**, which is associated to all DoD requirements
Q4: Are there any foreign personnel's having access to the system?	-Yes -No	Radio Box: Single Selection	**If Answer Option is:** Yes **Then, Applicable Requirements are:** 1. Mechanisms to limit access to foreign nationals 2. Affiliation part of email address 3. Access authorized by DoD head Components
Q5: Is wireless computing performed?	-Yes -No	Radio Box: Single Selection	**If Answer Option is:** Yes **Then, Applicable Requirements are:** 1. Wireless Computing and Networking 2. Disabling unused wireless

to the involved stakeholders. Finally, the answers gathered for applicability questionnaires can also be designed to perform complex inferences on the requirements space.

4.4.2. *The requirements compliance questionnaire*

The hierarchical structure of the RDM facilitates the evaluation of non-leaf node requirements categories at different levels of abstraction based on the compliance levels of their leaf-node security requirements. For each leaf-node requirement, the compliance questions have pre-defined answer options that represent various levels of compliance. These levels are systematically prepared from the conjunction of metrics and measures from multiple dimensions necessary to evaluate a security requirement. The selected answer options can provide qualitative (requirements that cannot be evaluated based on a numerical scale are assigned to three qualitative compliance levels of full-compliance, partial-compliance or non-compliance, for example consider the requirement shown in Fig. 13) or quantitative (typically values are assigned based on a numerical scale or Boolean values) values; however, they are normalized based on appropriate weights to support uniform interpretation and evaluation of compliance levels in the application domain. The selection of weights is usually specific to an organization or agency. The answers options collected through these questionnaires are also used to augment the applicable set of security requirements based on the mappings between the characteristics of the chosen answer option and the related security requirements in the RDM.

In addition to determining the need for qualitative or quantitative compliance information for a requirement, the task of producing compliance questions and



Requirement: EBRP-1 Remote Access audit trails for Privileged Functions
Description: A complete audit trail of each remote session is recorded, and the IAM/O reviews the log for every remote session

Question: Is there a remote access audit trail for privileged functions ?

Required compliance items :

1. *Complete remote access audit trail is recorded for each remote session*
   (metrics and measures from the AUDIT dimension)
2. *IAM reviews the log for every remote session*
   (metrics and measures from the LOG REVIEW dimension)

Answer option 1: A *complete remote access audit trail is recorded for each remote session* and the *IAM reviews the log for every remote session*. . (***full-compliance***)

Answer option 2: A *complete remote access audit trail is present for remote access*. but there is ***no*** *authority assigned to review the log*. (***partial-compliance***)

Answer option 3: There *are only* ***few*** *remote access audit trail that are recorded*. for each remote session and the *IAM reviews the log for every remote session*. (***partial-compliance***)

Answer option 4: There *are only* ***few*** *remote access audit trail that are recorded*. for each remote session and there is ***no*** *authority assigned to review the log*. (***partial-compliance***)

Answer option 5: There is ***no*** *audit trial for remote access*. (***non-compliance***)

Fig. 13. Example for single-selection type compliance question.

associated answer options involves several other important design choices, which are as follows:

- For each requirement, identify a set of *compliance items* that represent the evidences related to metrics from multiple dimensions. Each answer option must contain one or more compliance items to provide valid conjunctions of metrics and measures that represent different compliance levels, as shown in Fig. 13.
- The number of compliance questions for a security requirement depends on: (1) the diversity of metrics and measures that need to be gathered for a requirement; and (2) the criticality of the requirement. For example, a compliance question can be designed to capture compliance information for one or more requirements however, for a critical security requirement several questions may be necessary.
- The answer options are normalized into three categories: full-compliance, partial-compliance, and non-compliance.
- Multiple-selection type compliance questions (check boxes) that allow multiple answer options to be selected are used when the number of requirements compliance items is large. Single-selection type compliance questions (radio buttons) that allow only a single answer option to be selected are used when the number of requirements compliance items is relatively small as shown in Fig. 13.

Responses for requirements compliance questionnaires are gathered from various sources such as users, operating manuals, plans, architecture diagrams, or through automated network-based information discovery toolkits. The gathered responses also help in perceiving the operational risks based on level of compliance with security requirements and identifying the coverage of the gathered compliance criteria at different levels of abstraction in the RDM [45].

Table 3. GenOM representation format for requirements applicability questionnaire.

GenOM *Object*	**Description**		
Applicable Questionnaire	The object holds instances of all requirements applicability questions		
Answer Option	The object holds instances of all answer options for each question		
Requirements Categories	The object holds instances of security requirements in the requirements category hierarchy		
GenOM *Property*	**Type**	**Cardinality**	**Values**
Answer Response Type	String	Single	- Radio Box: Single Selection - Combo Box: Multiple Selection - Text Input **Description:** This property is used to dynamically configure the user interface based on the answer response type
Answer Value	String	Multiple	**Description:** This property holds the answer option selected by the user
Applicable	Boolean	Single	**Description:** This property is associated with all requirements in the requirement hierarchy. It is set to True/False based on the answer option selected by the user for a question. It helps to identify if a requirement is applicable or not
GenOM *Feature*	**Description**		
have_answer_option	Relates each question to its answer options		
next_question	Links an *Applicable Questionnaire* or *Answer Option* to the *Applicable Questionnaire Object,* which determines the next question that should be asked based on the answer option selected for the current question		
related_requirement	Links the *Applicable Questionnaire* or *Answer Option* to the *Requirements Category Object* in the RDM		

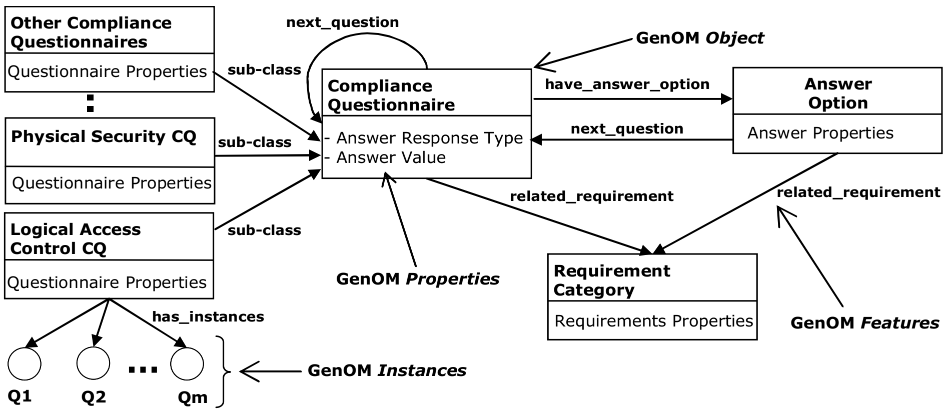


Fig. 14. GenOM Representation format for requirements compliance questionnaire.

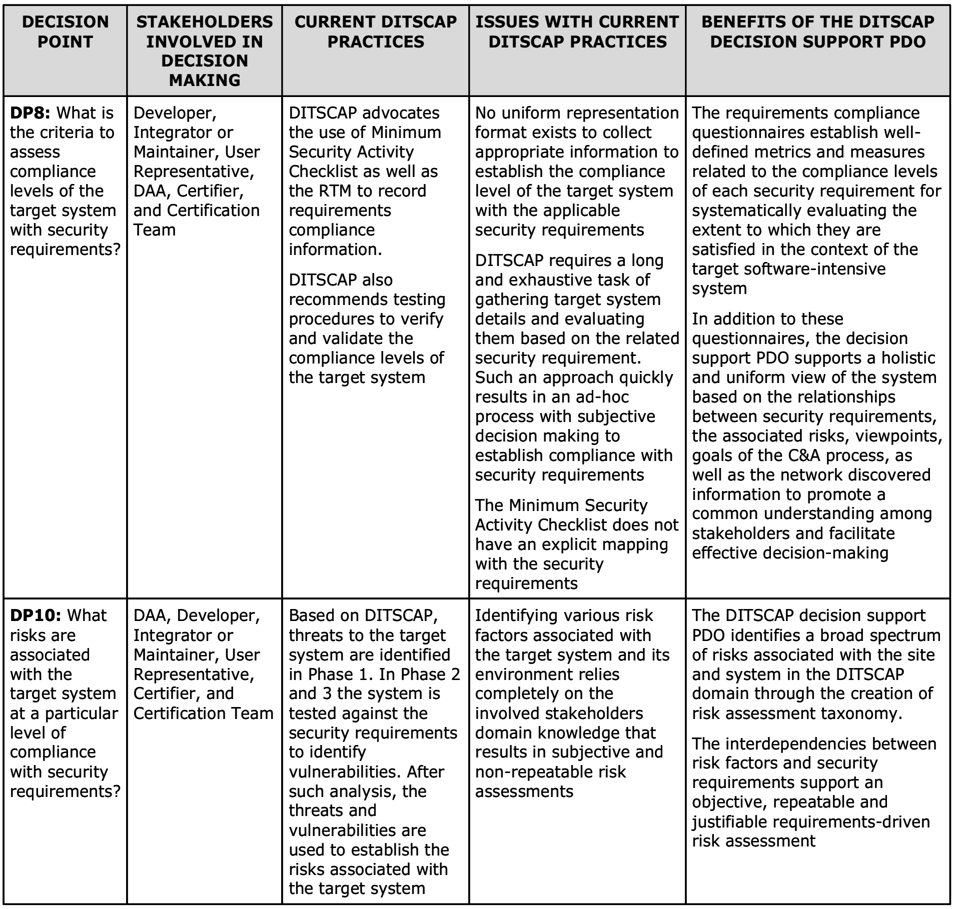
4.4.3. *Modeling the questionnaires*

Considering the rich set of characteristics and constraints relevant to security requirements and the target system, the related questionnaires require flexible and well-defined formats for their representation. To address these needs, the *Objects*, *Properties*, and *Features* of a GenOM representation model for the requirements applicability questionnaires are summarized in Table 3. While being mostly similar to requirements applicability questionnaires, the GenOM representation formats for requirements compliance questionnaires, as shown in Fig. 14, include a few additional modeling constructs. For the requirements compliance questionnaires representation format, the "Compliance Questionnaire" *Object* is further decomposed

Table 4. Key decision points supported by the DITSCAP decision support PDO.

DECISION POINT	STAKEHOLDERS INVOLVED IN DECISION MAKING	CURRENT DITSCAP PRACTICES	ISSUES WITH CURRENT DITSCAP PRACTICES	BENEFITS OF THE DITSCAP DECISION SUPPORT PDO
DP4: Is the identified set of security requirements complete?	DAA, Certifier, Program Manager, and User Representative	Based on their domain knowledge stakeholders manually extract security requirements from various security documents. Once these security requirements are identified, stakeholders generate the Requirement Traceability Matrix (RTM) and organize requirements based on their characteristics The RTM forms the baseline to determine the completeness of the security requirements identified	Security by its nature requires a broad spectrum of knowledge and system information, but we often have to rely on the domain knowledge and experience of subject matter experts to make decisions regarding the completeness of identified security requirements leading to subjective decision making criteria Generating the RTM is a long and tedious process prone to error as it requires sifting through a multitude of DITSCAP related documents and comprehending their interdependencies	The DITSCAP PDO provides a requirements applicability questionnaire that captures the characteristics and constraints relevant to the target system and environment to retrieve an applicable set of security requirements based on their properties A structured and machine understandable format of the PDO helps in overcoming the drawbacks of long-exhaustive manual process of documentation and analysis for generating the RTM The traceable rationales of the DITSCAP PDO and GenOM tool support help to systematically identify missing security requirements based on the interdependencies between various problem domain concepts
DP5: What are the interdependencies between the applicable set of security requirements and how are they identified?	DAA, Program Manager, Developer, Integrator or Maintainer, User Representative, Certifier, and Certification Team	Security requirements are grouped into high-level categories while preparing the RTM. Interdependencies between security requirements are usually identified based on explicit description in documents or from the domain knowledge of SMEs	The process of identifying interdependencies between security requirements is complicated by their non-functional nature coupled with the emergent properties of a software-intensive system. Also interdependencies between requirements become apparent only when considered from a certain perspective at an appropriate abstraction level. However, SMEs rely only on their domain knowledge to identify interdependencies among the applicable set of security requirements. The RTM characterizes requirements based on only a few attributes which are not sufficient to identify their interdependencies	The DITSCAP PDO has a well-structured representation format with various attributes associated with each security requirements that facilitate uniform interpretation and analysis at various levels of abstraction. We also introduce within our methodology, a keyword based technique which makes it relatively easy to identify interdependencies between security requirements that have not been explicitly identified in their specifications. Each requirement in the DITSCAP PDO can also be understood and analyzed based on its interdependencies with various concepts in the DITSCAP domain to identify related or missing security requirements
DP7: Which stakeholders are responsible for or affected by the security requirements?	DAA, Program Manager, Developer, Integrator or Maintainer, User Representative, Certifier, and Certification Team	Separate sections exist in DITSCAP-oriented documents, which identify stakeholder responsibilities. The IA objectives associated with security controls are also explicitly identified.	Stakeholder responsibilities are identified in DITSCAP-oriented document but no mappings exist between the stakeholders and the security requirements they are responsible for. Several different perspectives are intermingled within natural-language specifications of security requirements	The DITSCAP PDO includes a Viewpoint Hierarchy with viewpoints related to end-users of the system, services, stakeholders, IA objectives, organizational concerns, etc. to analyze security requirements from these perspectives The Stakeholder Viewpoints are explicitly mapped to security requirements based on their responsibility and security requirement descriptions

Table 4. (*Continued*)

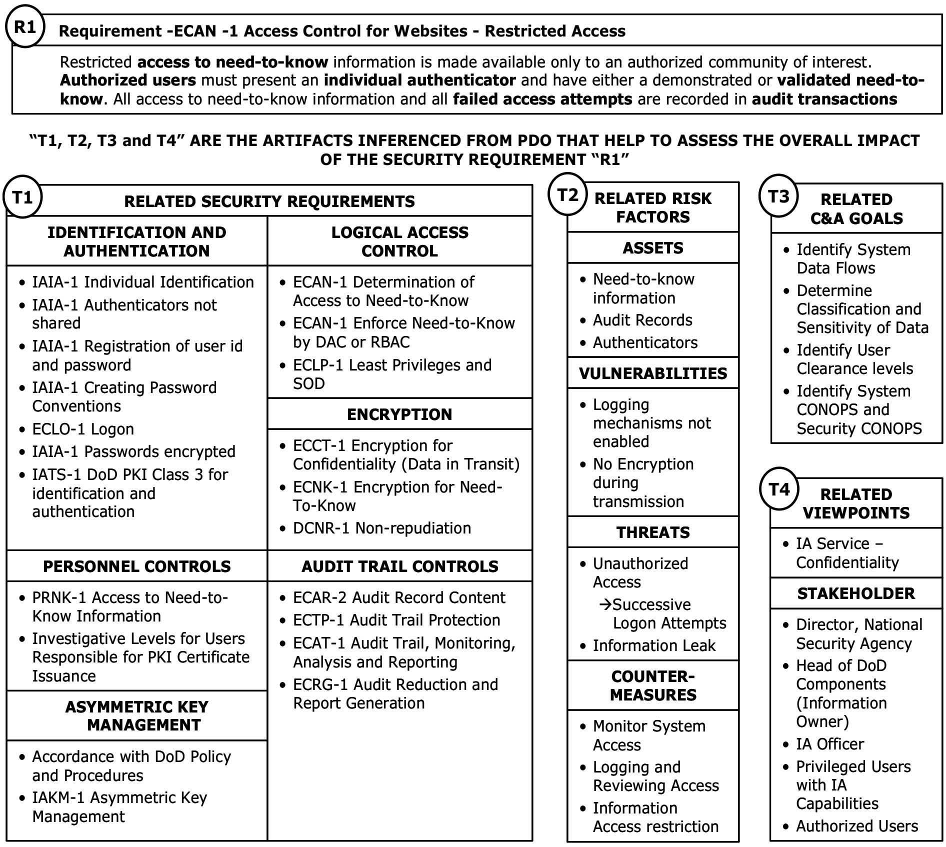


DECISION POINT	STAKEHOLDERS INVOLVED IN DECISION MAKING	CURRENT DITSCAP PRACTICES	ISSUES WITH CURRENT DITSCAP PRACTICES	BENEFITS OF THE DITSCAP DECISION SUPPORT PDO
DP8: What is the criteria to assess compliance levels of the target system with security requirements?	Developer, Integrator or Maintainer, User Representative, DAA, Certifier, and Certification Team	DITSCAP advocates the use of Minimum Security Activity Checklist as well as the RTM to record requirements compliance information. DITSCAP also recommends testing procedures to verify and validate the compliance levels of the target system	No uniform representation format exists to collect appropriate information to establish the compliance level of the target system with the applicable security requirements DITSCAP requires a long and exhaustive task of gathering target system details and evaluating them based on the related security requirement. Such an approach quickly results in an ad-hoc process with subjective decision making to establish compliance with security requirements The Minimum Security Activity Checklist does not have an explicit mapping with the security requirements	The requirements compliance questionnaires establish well-defined metrics and measures related to the compliance levels of each security requirement for systematically evaluating the extent to which they are satisfied in the context of the target software-intensive system In addition to these questionnaires, the decision support PDO supports a holistic and uniform view of the system based on the relationships between security requirements, the associated risks, viewpoints, goals of the C&A process, as well as the network discovered information to promote a common understanding among stakeholders and facilitate effective decision-making
DP10: What risks are associated with the target system at a particular level of compliance with security requirements?	DAA, Developer, Integrator or Maintainer, User Representative, Certifier, and Certification Team	Based on DITSCAP, threats to the target system are identified in Phase 1. In Phase 2 and 3 the system is tested against the security requirements to identify vulnerabilities. After such analysis, the threats and vulnerabilities are used to establish the risks associated with the target system	Identifying various risk factors associated with the target system and its environment relies completely on the involved stakeholders domain knowledge that results in subjective and non-repeatable risk assessments	The DITSCAP decision support PDO identifies a broad spectrum of risks associated with the site and system in the DITSCAP domain through the creation of risk assessment taxonomy. The interdependencies between risk factors and security requirements support an objective, repeatable and justifiable requirements-driven risk assessment

into specific sub-categories that correspond to the security requirements categories in the RDM.

4.5. *STEP 5: Decision support in the DITSCAP domain*

In the previous steps of the methodology, we extracted security requirements from multiple documents, annotated them with appropriate attributes, provided modeling techniques for their structural representations, identified interdependencies between requirements as well as with other domain concepts, and established well-defined techniques for assessing requirements applicability and compliance. To identify the impact of these techniques and the resulting decision support PDO at critical decision points throughout the C&A process, Table 4 outlines the existing DITSCAP practices, and their shortcomings and compares them to the advantages



R1 **Requirement -ECAN -1 Access Control for Websites - Restricted Access**

Restricted **access to need-to-know** information is made available only to an authorized community of interest. **Authorized users** must present an **individual authenticator** and have either a demonstrated or **validated need-to-know**. All access to need-to-know information and all **failed access attempts** are recorded in **audit transactions**

"T1, T2, T3 and T4" ARE THE ARTIFACTS INFERENCED FROM PDO THAT HELP TO ASSESS THE OVERALL IMPACT OF THE SECURITY REQUIREMENT "R1"

T1 **RELATED SECURITY REQUIREMENTS**

IDENTIFICATION AND AUTHENTICATION
- IAIA-1 Individual Identification
- IAIA-1 Authenticators not shared
- IAIA-1 Registration of user id and password
- IAIA-1 Creating Password Conventions
- ECLO-1 Logon
- IAIA-1 Passwords encrypted
- IATS-1 DoD PKI Class 3 for identification and authentication

LOGICAL ACCESS CONTROL
- ECAN-1 Determination of Access to Need-to-Know
- ECAN-1 Enforce Need-to-Know by DAC or RBAC
- ECLP-1 Least Privileges and SOD

ENCRYPTION
- ECCT-1 Encryption for Confidentiality (Data in Transit)
- ECNK-1 Encryption for Need-To-Know
- DCNR-1 Non-repudiation

PERSONNEL CONTROLS
- PRNK-1 Access to Need-to-Know Information
- Investigative Levels for Users Responsible for PKI Certificate Issuance

ASYMMETRIC KEY MANAGEMENT
- Accordance with DoD Policy and Procedures
- IAKM-1 Asymmetric Key Management

AUDIT TRAIL CONTROLS
- ECAR-2 Audit Record Content
- ECTP-1 Audit Trail Protection
- ECAT-1 Audit Trail, Monitoring, Analysis and Reporting
- ECRG-1 Audit Reduction and Report Generation

T2 **RELATED RISK FACTORS**

ASSETS
- Need-to-know information
- Audit Records
- Authenticators

VULNERABILITIES
- Logging mechanisms not enabled
- No Encryption during transmission

THREATS
- Unauthorized Access →Successive Logon Attempts
- Information Leak

COUNTER-MEASURES
- Monitor System Access
- Logging and Reviewing Access
- Information Access restriction

T3 **RELATED C&A GOALS**
- Identify System Data Flows
- Determine Classification and Sensitivity of Data
- Identify User Clearance levels
- Identify System CONOPS and Security CONOPS

T4 **RELATED VIEWPOINTS**
- IA Service – Confidentiality

STAKEHOLDER
- Director, National Security Agency
- Head of DoD Components (Information Owner)
- IA Officer
- Privileged Users with IA Capabilities
- Authorized Users

Fig. 15. Artifacts inferenced from the PDO that help to assess the overall impact of a given requirement.

gained through the availability of the PDO. Each decision point in Table 4, involves complex negotiations between various system stakeholders to establish a precise understanding of the system and determine a course of action at decision points for interpretation, applicability, scope, evaluation, and impact of the enforced C&A requirements.

In essence, the PDO supports various decision points by proactively revealing the relationships of security requirements with other domain concepts through the nexus of causal chains that exist in the problem domain. The PDO makes these concepts readily available through various inference mechanisms based on its ontological structure and semantics. As an example, in Fig. 15, consider the security requirement marked as "R1" and the concepts "T1, T2, T3, and T4" obtained from the PDO which serve as metrics and measures from various dimensions to assess the overall impact of the security requirements on the target system and environment.

5. Contributions and Future Work

Evaluating and establishing secure system assurance for software-intensive systems in a socio-technical environment requires gathering knowledge artifacts from multiple dimensions, abstractions, and sources (that include organizational policies, across organizational policies, cross-border policies, people, interconnection between different levels of sensitivity of systems and the environment as a whole, etc.). Therefore, in view of decision support as the process of systematically understanding and structuring diverse knowledge artifacts in ways that promote valuable insight, the main contribution of this research is a detailed and step-wise methodology for systematically constructing a decision support PDO from diverse natural language security requirements scattered across multiple natural language documents from various levels in the organizational hierarchy. In addition, the decision support PDO combines functional as well as non-functional aspects of system requirements along with relevant domain knowledge based on a uniform representation format offered by rich ontological engineering processes, to support objective, repeatable and justifiable decision making activities. For each step in our methodology, we outlined the modeling techniques and heuristics necessary to elicit, model and analyze security requirements and associated characteristics/constraints from their natural language specifications. Finally, we also identified the impact of these techniques on the critical decision points perceived throughout the C&A process.

In general, the methodology presented in this paper provides heuristics that help in capturing the characteristics of information present sparsely in documents and the way these characteristics can be represented using ontological modeling processes to infer valuable knowledge that assists decision-making activities. Hence, we contend that our methodology can be extended to any domain where the decision making activities require sifting through large volumes of information. Due to the nature of the ontological engineering, currently the PDO has been constructed using frequent feedback and refinement by SMEs using GenOM tool support as a workbench. Although the problem domain concepts are extracted and modeled manually, it is possible to incorporate techniques for automatically processing natural language documents and identify problem domain concepts that SMEs can refine further.

Our methodology has been applied in the effort for automating the DITSCAP and a prototype implementation has been generated which is intended to serve as a vehicle for identifying the strengths and weaknesses of our approach through the experience gained from interactions with real users performing real tasks. That said, we are currently in the process of outlining a case study designed research methodology (CSDM) [48] for evaluating the effectiveness of practicing DITSCAP using the PDO against the completely manual approach followed in DITSCAP and the results obtained will become part of our future publications. Table 4 also serves as an initial baseline to evaluate the manual and automated approaches at key decision points. The DITSCAP being a knowledge-intensive process, it requires several interventions from the SMEs and its results depend on the level of understanding of the domain and skills of SMEs that participates in the actual studies. From this perspective, the CSDM is a suitable evaluation mechanism because, it requires designing a case study that is specific to the characteristics of the methodology that requires interventions from the domain SMEs in order to perform on demand each appropriate step in the methodology; and to the characteristics of the validation procedure that cannot favor the methodology over alternatives because of the uniqueness of the methodology or the relative different level of understanding of the domain and analytical skill of SMEs in the actual case study "experimental" conditions [48]. Based on the designed case study, we plan to evaluate the effectiveness of our methodology by having a group of SMEs perform DITSCAP with and without using the decision support PDO and related tool support. The results of case study will be recorded using appropriate metrics and measures that are faithful indicators of successfully performing each DITSCAP component, and these metrics will eventually serve as a basis for comparison between a manual and automated approach.

We also continue to refine the methodology by applying it to various other problem domains in order not to limit its applicability to the security requirements domain. Currently, the models within the decision support PDO capture the functional, non-functional and the related domain aspects based on the characteristics of the security requirements. We plan to study how these models can be used and extended to capture other dependability requirements like reliability, safety etc. and also how PDO supported decisions can be made for handling incompleteness, inconsistencies, and other types of conflicts between various dependability requirements present in the domain applications.

Acknowledgments

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

1. A. de Groot, J. Hooman, M. Lemoine, G. Gaudiere, V. L. Winter, and D. Kapur, A survey: Applying formal methods to a software intensive system, in *Proc. 6th IEEE Int. Symp. on High Assurance Systems Engineering* (2001), pp. 55–64.
2. A. Sutcliffe, Scenario-based requirements analysis, *Requirements Engineering Journal* **3**(1) (1998) 48–65.
3. A. van Lamsweerde, Goal-oriented requirements engineering: A guided tour, in *Proc. 5th Int. Symp. on Requirements Engineering*, Toronto, Canada, 2001, pp. 249–262.
4. A. van Lamsweerde, R. Darimont, and E. Letier, Managing conflicts in goal-driven requirements engineering, in *IEEE Trans. on Software Engineering* **24**(11) (1998) 908–926.

5. A. van Lamsweerde, S. Brohez, R. De Landtsheer, and D. Janssens, From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering, in *Proc. Requirements for High Assurance Systems Workshop* (*RHAS'03*), *11th Int. Conf. on Requirements Engineering* (2003).
6. A. I. Anton, J. B. Earp, and A. Reese, Analyzing website privacy requirements using a privacy goal taxonomy, in *Proc. IEEE Joint Int. Conf. on Requirements Engineering* (2002), pp. 23–31.
7. B. Nuseibeh and S. Easterbrook, Requirements engineering: A roadmap, in *Proc. Conf. on the Future of Software Engineering*, Limerick, Ireland (ACM Press, 2000), pp. 35–46.
8. C. Potts, Using schematic scenarios to understand user needs, in *Proc. Conf. on Designing Interactive Systems*, Ann Arbor, Michigan (ACM Press, 1995), pp. 247–256.
9. C. Rolland, C. Souveyet, and C. B. Achour, Guiding goal modeling using scenarios in *IEEE Trans. on Software Engineering* **24**(12) (1998) 1055–1071.
10. DoD 8500.1. Information Assurance (Oct. 2002).
11. DoD 8500.2. Information Assurance Implementation (Feb. 2003).
12. DoD 8510.1-M, DITSCAP Application Manual (2000).
13. DoD Instruction 5200.40, DITSCAP (1997).
14. G. Kotonya and I. Sommerville, *Requirements Engineering — Processes and Techniques* (John Wiley, New York, 1998).
15. G. Kotonya and I. Sommerville, Requirements engineering with viewpoints, in *BCS/IEE Software Engineering Journal* **11**(1) (1996) 5–18.
16. G. Sindre and A. L. Opdahl, Eliciting security requirements by misuse cases, in *Proc. 37th Int. Conf. on Technology of Object-Oriented Languages and Systems* (2000), pp. 120–131.
17. I. Sommerville and P. Sawyer, Viewpoints: Principles, problems and a practical approach to requirements engineering, *Annals of Software Engineering* **3** (1997) 101–130.
18. J. Andrade, J. Ares, R. Garcia, J. Pazos, S. Rodriguez, and A. Silva, A methodological framework for viewpoint-oriented conceptual modeling, *IEEE Trans. on Software Engineering* **30**(5) (2004) 282–294.
19. J. Evermann and Y. Wand, Ontology based object-oriented domain modeling: Fundamental concepts, *Requirements Engineering Journal* (2005).
20. J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, Jena: Implementing the semantic web recommendations, in *Proc. 13th Int. World Wide Web Conference*, New York, USA, 2004, pp. 74–83.
21. J. Kimbell and M. Walrath, Life cycle security and DITSCAP, *IANewsletter* **4**(2) (2001) http://iac.dtic.mil/iatac.
22. J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis, TELOS: Representing knowledge about information systems, *ACM Trans. on Information Systems* **8**(4) (1990) 325–362.
23. J. Mylopoulos, L. Chung, and B. Nixon, Representing and using nonfunctional requirements: A process-oriented approach, *IEEE Trans. on Software Engineering* **18**(6) (1992) 483–497.
24. J. C. S. P. Leite, G. D. S. Hadad, J. H. Doorn, and G. N. Kaplan, A scenario construction process, *Requirements Engineering Journal* **5** (2000) 38–61.
25. K. Allenby and K. Tim, Deriving safety requirements using scenarios, in *Proc. 5th Int. Symp. on Requirements Engineering* (2001), pp. 228–235.
26. K. K. Breitman and J. Leite, Ontology as a requirements engineering product, in

Proc. 11th IEEE Int. Conf. on Requirements Engineering, Mini-tutorial on Ontology Development (2003).

27. L. Lin, B. Nuseibeh, D. Ince, and M. Jackson, Using abuse frames to bound the scope of security problems, in *Proc. 12th IEEE Int. Conf. on Requirements Engineering* (2004), pp. 354–355.
28. L. Liu and E. Yu, Designing information systems in social context: A goal and scenario modeling approach, *Information Systems Journal* **29**(2) (2003).
29. L. Liu, E. Yu, and J. Mylopoulos, Analyzing security requirements as relationships among strategic actors, in *Proc. 2nd Symp. on Requirements Engineering for Information Security* (*SREIS '02*), Raleigh, NC, 2002.
30. L. Liu, E. Yu, and J. Mylopoulos, Security and privacy requirements analysis within a social setting, in *Proc. 11th IEEE Int. Conf. on Requirements Engineering* (2003), pp. 151–161.
31. L. M. Cysneiros and J. C. S. P. Leite, Nonfunctional requirements: From elicitation to conceptual models, *IEEE Trans. on Software Engineering* **30**(5) (2004).
32. M. Jackson, The meaning of requirements, *Annals of Software Engineering* **3** (1997) 5–21.
33. OMB Circular No. A-130, Management of Federal Information Resources (1996).
34. *OWL Web Ontology Language Overview*, eds. D. McGuinness and F. van Harmelen, W3C Recommendation (2004) http://www.w3.org/TR/owl-features/.
35. P. Donzelli, A goal-driven and agent-based requirements engineering framework, *Requirements Engineering Journal* **9**(1) (2004) 16–39.
36. P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, Modeling security requirements through ownership, permission and delegation, in *Proc. 13th IEEE Int. Conf. on Requirements Engineering* (2005), pp. 167–176.
37. P. Loucopoulos and E. Kavakli, Enterprise modeling and the teleological approach to requirements engineering, *Int. J. Intelligent and Cooperative Information Systems* **4**(1) (1995) 45–79.
38. R. Offen, Domain Understanding is key to successful system development, *Requirements Engineering Journal* **7**(3) (2002) 172–175.
39. Rational Corporation, *UML: Unified Modeling Language Method* (1999).
40. S. Greenspan, J. Mylopoulos, and A. Borgida, On formal requirements modeling languages: RML revisited, in *Proc. 16th Int. Conf. on Software Engineering*, Sorrento, Italy (IEEE Computer Society Press, 1994), pp. 135–147.
41. S. W. Lee and R. A. Gandhi, Engineering dependability requirements for software-intensive systems through the definition of a common language, in *Proc. 13th IEEE Int. Requirements Engineering Conference* (*RE '05*)*, Workshop on Requirements Engineering for High-Availability Systems* (*RHAS*), Software Engineering Institute (SEI), Carnegie Mellon University & IEEE Press (Paris, France, 2005), pp. 40–48.
42. S. W. Lee, R. A. Gandhi, G. Ahn, and D. Yavagal, Active automation of the DITSCAP, in *Proc. IEEE Int. Conf. on Intelligence and Security Informatics*, Lecture Notes in Computer Science, Vol. 3495, Springer, 2005, pp. 479–485.
43. S. W. Lee, R. A. Gandhi, and G. Ahn, Certification process artifacts defined as measurable units for software-intensive systems lifecycle, in *Software Process: Improvement and Practice Journal* (Wiley, 2006).
44. S. W. Lee, R. A. Gandhi, and G. Ahn, Establishing trustworthiness in services of the critical infrastructure: Automating the DITSCAP, in *Workshop on Software Engineering for Secure Systems* (*SESS05*)*, 27th Int. Conf. on Software Engineering* (2005), pp. 43–49.
45. S. W. Lee, R. A. Gandhi, and G. Ahn, Security requirements driven risk assess-

ment for critical infrastructure information systems, in *Proc. Symp. on Requirements Engineering for Information Security* (*SREIS 05*), *13th Int. Conf. on Requirements Engineering* (IEEE Computer Society, Paris, France, 2005).

46. S. W. Lee, R. A. Gandhi, and G. Ahn, Engineering information assurance for critical infrastructures: The DITSCAP automation study, in *Proc. 15th Annual Int. Symp. Int. Council on Systems Engineering* (*INCOSE*), Rochester, NY, 2005.
47. S. W. Lee and D. Yavagal, 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, 2005.
48. S. W. Lee and D. C. Rine, Case study methodology designed research in software engineering methodology validation, in *Proc. 16th Int. Conf. on Software Engineering and Knowledge Engineering*, Banff, Alberta, Canada, 2004, pp. 117–122.
49. S. W. Lee and R. A. Gandhi, Ontology-based active requirements engineering framework, in *Proc. 12th Asia-Pacific Software Engg. Conference* (*APSEC '05*), IEEE CS Press, Taiwan, 2005.
50. T. Davis, Chairman Government Reform Committee, Davis Statement on 2004 Federal Computer Security Report Card Grades, Press Release, http://reform.house.gov/UploadedFiles/021605FISMAstatement.pdf.
51. T. Davis, Chairman Government Reform Committee, *No Computer System Left Behind: A Review of the 2005 Federal Computer*, Press Release (2005) http://reform.house.gov/UploadedFiles/TMD FISMA 06 Opener.pdf.
52. T. D. Breaux and A. I. Antón, Analyzing goal semantics for rights, obligations, and permissions, in *Proc. 13th Int. Conf. on Requirements Engineering*, Paris, France, 2005.
53. V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice, OKBC: A programmatic foundation for knowledge base interoperability, in *Proc. 15th National/10th Conf. on Artificial Intelligence/Innovative Applications of Artificial intelligence*, AAAI, Menlo Park, CA, 1998, pp. 600–607.