Building Problem Domain Ontology from Security Requirements in Regulatory Documents

Seok-Won Lee, Robin Gandhi, Divya Muthurajan, Deepak Yavagal and Gail-Joon Ahn
Department of Software and Information Systems
The University of North Carolina at Charlotte
9201 University City Blvd., Charlotte, NC 28223, USA
{seoklee, rgandhi, dmuthura, dsyavaga, gahn}@uncc.edu

ABSTRACT

Establishing secure systems assurance based on Certification and Accreditation (C&A) activities, requires effective ways to understand the enforced security requirements, gather relevant evidences, perceive related risks in the operational environment, and reveal their causal relationships with other domain concepts. However, C&A security requirements are expressed in multiple regulatory documents with complex interdependencies at different levels of abstractions that often result in subjective interpretations and non-standard implementations. Their non-functional nature imposes complex constraints on the emergent behavior of software-intensive systems, making them hard to understand, predict, and control. To address these issues, we present novel techniques from software requirements engineering and knowledge engineering for systematically extracting, modeling, and analyzing security requirements and related concepts from multiple C&A-enforced regulatory documents. We employ advanced ontological engineering processes as our primary modeling technique to represent complex and diverse characteristics of C&A security requirements and related domain knowledge. We apply our methodology to build problem domain ontology from regulatory documents enforced by the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP).

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications – *structured methodologies, tools.*

General Terms

Measurement, Reliability, Security and Standardization.

Keywords

Information Security Requirements Engineering, Information Systems Certification and Accreditation, Secure Software Assurance, Ontological Engineering.

1. INTRODUCTION

Software-intensive systems are complex clusters of closely interdependent *systems of systems* that are increasingly supporting critical information processing and other value-added services. Naturally they are subject to additional requirements for availability, continuity, performance, trustworthiness and other dependability dimensions. Security being a major dependability dimension, C&A processes which focus on assessing various secure system characteristics have gained wide-spread popularity to establish secure systems assurance. However, C&A security requirements are expressed at various levels of abstractions in the authoritative organizational hierarchy and documented in multiple regulatory documents with heavy cross-referencing to each other. Such regulatory documents achieve greater flexibility in their applicability through abstract specifications of enforced security requirements, however the requirements become hard to qualify or quantify based on clearly defined criteria as they allow for multiple subjective interpretations. In addition, the non-functional nature of security requirements imposes complex constraints on the emergent behavior of software-intensive systems, making them hard to understand, predict, and control based on current C&A practices and methods. As a result, despite enormous efforts and resources currently spent on C&A processes, their effectiveness in the real world is still limited [25].

We, therefore, believe that establishing secure systems assurance requires effectively understanding and representing security requirements. It includes systematically gathering evidences for establishing their applicability and compliance, perceiving related risks in the operational environment, and proactively revealing their proximity to other domain concepts through the nexus of causal chains that exist in the Universe of Discourse (UoD). Thus, to achieve an unambiguous and uniform understanding of security requirements, we provide the definition of a common language developed through a systematic methodology for extracting and organizing the problem domain concepts expressed in natural-language regulatory documents. The definition of a common language establishes a Problem Domain Ontology (PDO) based on well-defined dimensions that support a uniform and structured representation of security requirements, their attributes, and their interdependencies. We employ advanced ontological engineering processes as our primary modeling technique with related tool support for extracting, representing, and analyzing security requirements. From a theoretical perspective, the PDO combines functional and non-functional aspects of security requirements along with their related entities in the environment such as organization, business/mission requirements, and other domain-specific considerations.

Organization of the paper is as follows. Section 2, provides an introduction to DITSCAP and the motivations for developing its PDO. Section 3, outlines a stepwise methodology to capture, model and analyze security requirements using examples from DITSCAP regulatory documents. Section 4 compares our methodology with the existing practices and methods of DITSCAP using examples. Section 5, discusses related work and Section 6 summarizes our contributions and future work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SESS'06, May 20–21, 2006, Shanghai, China.
Copyright 2006 ACM 1-59593-085-X/06/0005...$5.00.

2. THE DITSCAP

The 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 [6]. 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 records the outcome of tasks and activities in each phase of the DITSCAP, which produce several measures by inspecting and analyzing their units of analysis and assessing them based on metrics considered for the procurement of certification status. Our earlier efforts on DITSCAP automation [14] provide a generic framework that systematically supports the C&A process. In [17], we outline theoretical foundations behind our approach based on a comprehensive Requirements Engineering (RE) framework for software-intensive systems [18].

2.1 The Need for a DITSCAP PDO

The DITSCAP requires multitude of directives, security requisites, and other regulatory documents, for a system to be compliant with. The security requirements specified in these documents are expressed in natural language with little or no structural regularity in their representations. 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 [15] for creating a context that facilitates their uniform interpretation and evaluation. However, for most security requirements available from DITSCAP-oriented regulatory documents these concepts are either missing or dispersed in multiple sources. Therefore, the need to systematically capture and organize DITSCAP problem domain concepts related to security requirements is apparent for effective decision-making activities regarding their interpretation, applicability, and implementation effectiveness. To address these needs, the DITSCAP PDO is a hierarchical organization of ontological concepts that capture well-defined dimensions of the problem domain with related properties and non-taxonomic dependencies among them. The inherent benefits of the PDO lie in the uniformity of its representation and traceable rationales to promote cohesiveness between concepts from various dimensions at different levels of abstraction necessary to understand and analyze security requirements.

To systematically capture various facets of a security requirement in the DITSCAP domain, the PDO includes structured and well defined representations of: 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 PDO. For the scope of this paper, we elaborate on the modeling techniques and heuristics involved in the creation of a RDM based on DITSCAP-oriented regulatory documents. Details about other models can be found in [14] [16].

3. BUILDING THE PDO

The process of building PDO is evolutionary with many synergistic interactions between the steps in our methodology. We now elaborate on the steps, modeling techniques and the design decisions involved in building the PDO.

3.1 The Preparation Step

3.1.1 Identifying Document Interdependencies

C&A related regulatory documents usually range from 25 to 200 pages with heavy cross-referencing to each other, which makes it extremely difficult to comprehend their contents. Therefore, as a first step it is necessary to understand the interdependencies between the documents as well as the generic types/categories of security requirements dispersed across multiple documents. We identify the following heuristics to organize regulatory documents: 1) Gain a high-level comprehension of the documents through their content and usage analysis. It usually involves reading their abstracts, title, headings, etc. 2) Group the documents into generic categories based on their purpose (for example, *Why the document was created?*); and 3) Identify interdependencies between the generic categories of documents by analyzing the abstract, references or heading sections of the documents in a particular category.

For DITSCAP-related documents within our scope, we determine a hierarchical relationship between the generic Federal-level documents, domain spanning DoD and DoN policy/NIST documents, and site/agency specific DoD and DoN implementation guidance documents as shown in Figure 1.

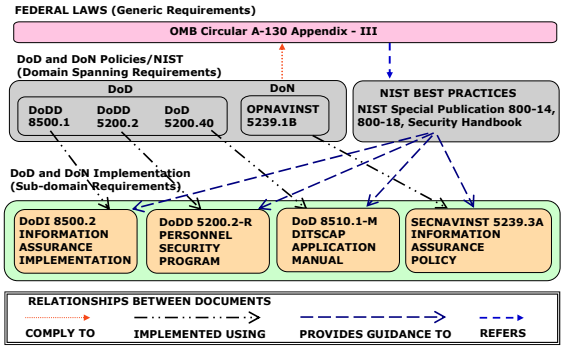


Figure 1: Document Organization Diagram

3.1.2 Creating a Requirements Category Hierarchy

To systematically aggregate security requirements and reason about them at different levels of abstractions from multiple dimensions, it is necessary to develop a security requirements category hierarchy that provides a comprehensive coverage of requirements expressed in regulatory documents. However, creating an initial security requirements category hierarchy can be

difficult without knowing *why* we need to build it and *what* to start with. To address this issue, we select a theme for requirements extraction by following a goal-driven requirements elicitation strategy [28] that starts with high-level goals of the problem domain to identify generic types of requirements sought after in the documents. A decomposition of higher-level goals into specific goals by asking the *How* questions, identify corresponding lower-level requirements categories. One can navigate up in the hierarchy by asking the *Why* questions. Following this approach, construction of the requirements category hierarchy, as shown in Figure 2, is based on the selection of "Security Plan" as a theme/high-level goal that is identified from Federal (high-level) documents [24]. Decomposition of this high-level goal and after thorough analysis of the requirements categories suggested by regulatory documents at different levels of abstraction, as shown in Figure 1, we identify specific categories in the requirements category hierarchy. For example, Figure 2 also elaborates on requirements categories of "Physical and Environmental Security Controls" and "Personnel Controls" that are obtained from lower-level agency documents (DoDD 8500.1 [7] and DoDI 8500.2 [8]) in the document organization diagram.

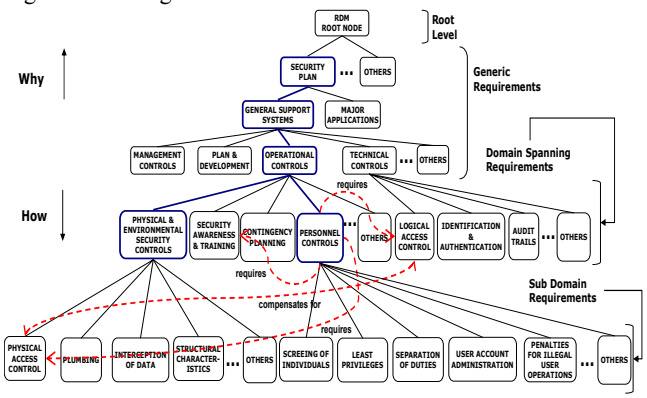


Figure 2: Partial Security Requirements Category Hierarchy

3.1.3 Identifying Security Requirements Attributes

Well-designed attributes provide clear, concise, and structured information about requirements as compared to natural-language descriptions. A complex conjunction of such attributes capture the diverse characteristics and constraints associated with security requirements that facilitate reasoning and analysis processes. Table 1 provides a list of generic as well as DITSCAP domain-specific attributes that have been identified.

Table 1: Requirements Attributes in the DITSCAP domain

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 the 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 a security requirement is applicable for
Type of System	(Major Application/AIS, General Support System/Enclave, Outsourced-IT Based Process, Platform IT-interconnection, All Systems) Indicates the type of system 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

3.2 The Requirements Extraction Step

3.2.1 RDM Categorization and Requirements Extraction

As the requirements category hierarchy and attributes become available, they 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 DITSCAP RDM categories as shown in Figure 3 following an incremental approach that iterates with the initial requirements category hierarchy being applied to each regulatory document. The RDM also promotes consistency in extracting requirements from multiple documents by providing a generic set of categories. For example in Figure 3, 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 different agency documents.

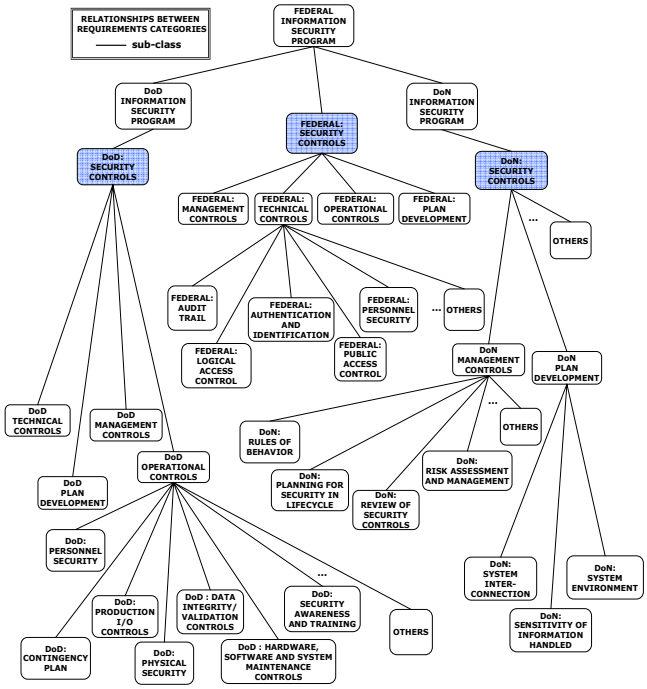
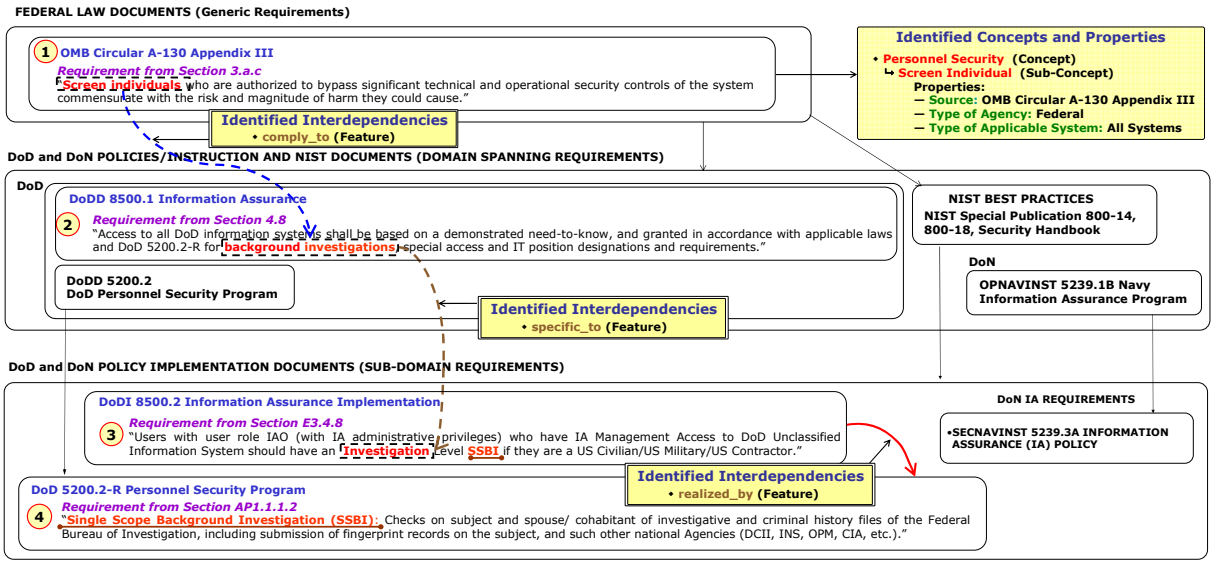


Figure 3: DITSCAP Requirements Domain Model Categories

The RDM categories along with attributes identified in Section 3.1.3 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 Figure 4. Documents in Figure 4 are organized hierarchically based on the document organization diagram as shown in Figure 1. 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 DITSCAP RDM. In addition, the extracted security requirements are also annotated with attributes that are available by analyzing their natural-language descriptions and the related domain knowledge of Subject Matter Experts (SME). Attributes for the security requirement labeled as "1" in Figure 4, identify its name, source, and its applicability to the Federal agency for all types of systems.



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

Figure 4: Extraction of Security Requirements, Categories, Properties, and their Interdependencies from DITSCAP-oriented regulatory documents

3.2.2 Dealing with Natural-Language Requirements

Requirements from natural-language documents, suffer from various problems related to consistency, completeness, redundancy, etc. To address such issues, we discuss typical cases encountered during requirements extraction and present heuristics to overcome them. 1) 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 offers ease of evaluation for requirement compliance. However, decompositions that tend to change the meaning/context of the requirement as a whole should be avoided; 2) Requirement descriptions have varying levels of abstraction. Such requirements should be appropriately decomposed and placed at proper level of abstraction in the RDM; 3) Requirements fit into more than one category. In such cases, the requirement is placed in a category that is most applicable based on the domain knowledge of the SME; 4) 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. To address them, create mappings between the terminologies used in high-level documents to the terminologies used in lower-level documents based on the document organization diagram of Figure 1.

3.2.3 Identifying Requirements Interdependencies

Identifying the relationships that exist between security requirements extracted from multiple sources exposes their crosscutting nature and promotes a shared understanding of the criteria used to interpret and evaluate them. Figure 4 identifies several such interdependencies, 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 Figure 1 also provides guidance for identifying related security requirements across documents. Such interdependencies are shown in Figure 4 through the "comply_to" and "specific_to" relationships between security requirements. 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 Figure 5. 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. Interdependencies are also identified between a requirement and a set of requirements under a particular category of the RDM.

•**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, mediated*

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

Figure 5: Identifying dependencies using Keywords

3.3 The Questionnaire Creation Step

The selection and evaluation of security requirements require aggregation of multiple evidences from target system and environment about their characteristics and constraints. Therefore, identifying clearly defined criteria/evidences that are objective, repeatable, and justifiable are critical 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 applicable security requirements. To address these needs, the PDO development involves the creation of questionnaires with well-defined answer options that systematically gather evidences from the target system and environment.

3.3.1 The Requirements Applicability Questionnaire

The requirements applicability questionnaire captures the characteristics and constraints relevant to the target system in its operational environment. Well-defined answer options prune the security requirements space based on their mappings to attributes of security requirements and categories in the DITSCAP RDM and establish criteria for requirements applicability. Table 2 provides examples of such mappings. Requirements applicability questions are organized in a hierarchical fashion, with high-level questions selecting a large set of requirements that is successively pruned using specific questions that are related to fewer requirements. To systematically support such selection, the characteristics and constraints of security requirements should be captured as attributes in the RDM based on related decision-making activities in the problem domain. The answer options can also be designed to perform complex inferences on the requirements space based on ontological representation of the RDM.

Table 2: Requirements Applicability Questions Examples

Question	Answer option	Related Requirements
Which organization's system is being certified and accredited?	-DoN -DoD -Federal	The applicable security requirements are selected through the **Property – Type of Agency** discussed in Section 3.1, which is associated to all requirements
What type of system is being certified and accredited?	-General support system/Enclave -Major Application/AIS -Platform IT-interconnections -Outsourced IT-based processes	The applicable security requirements are selected through the **Property – Type of System** discussed in Section 3.1, which is associated to all requirements
Are there any foreign personnel's having access to the system?	-Yes -No	**Answer Option:** Yes **Related Requirements:** 1. Mechanisms to limit access to foreign nations 2. Affiliation part of email address 3. Access authorized by DoD head Components

3.3.2 The Requirements Compliance Questionnaire

The second questionnaire set, called the requirements compliance questionnaire, establishes well-defined metrics and measures to establish the compliance levels for each leaf-node security requirement in the RDM. Based on the hierarchical structure of the RDM, non-leaf node requirements categories at different levels of abstraction are analyzed with respect to their leaf-node security requirements. For each compliance question, pre-defined answer options that represent various levels of compliance 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 Figure 6) or quantitative (typically values are assigned based on a numerical scale or Boolean values) values; however, they are normalized based on appropriate weights associated with them to support uniform interpretation and evaluation of compliance levels in the application domain. We make the following design choices for producing compliance questions and corresponding answer options from security requirements descriptions in regulatory documents and related domain knowledge:

- For each security requirements we identify a set of *compliance items* that represent the evidences related to metrics from multiple dimensions. The answer options must contain one or more compliance items to provide valid conjunctions of metrics and measures that represent different compliance levels, as shown in Figure 6.
- 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. 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 Figure 6.

Responses for requirements compliance questionnaires can be gathered from various sources such as users, operating manuals, plans, architecture diagrams, or through automated network-based information discovery toolkits. The gathered responses also helps 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.

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***)

Figure 6: Single-selection type Compliance Question

3.4 The Requirements Modeling Step

To support the representation of rich knowledge structures required by the PDO, various ontological engineering processes are provided by the GENeric Object Model (GenOM) [20] 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 OKBC specification [4] as well as the OWL representation [23] 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 [3], which supports reasoning based on the *Objects*, *Properties*, and *Features* and *Instances* defined in its knowledge-bases.

Using various GenOM modeling constructs, Figure 7 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 represent the characteristics/constraints captured through requirements attributes identified in Section 3.1.3. *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 modeled as *Instances* of "Requirement Category" *Objects,* also inherit the *Properties* associated with their parent requirement categories. The interdependencies among security requirements identified in Section 3.2.3 as well as with other concepts in the PDO are represented using various *Features* with well-defined semantics.

We also create GenOM representation formats for applicability and compliance questionnaires along with their answers options. Figure 8 shows the GenOM representation format for requirements compliance questionnaires.

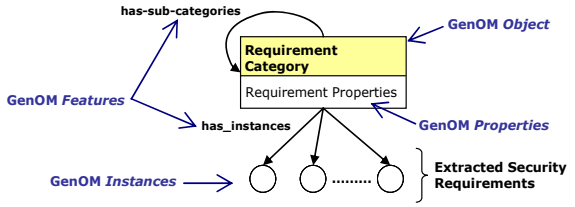


Figure 7: GenOM Representation Format for the RDM

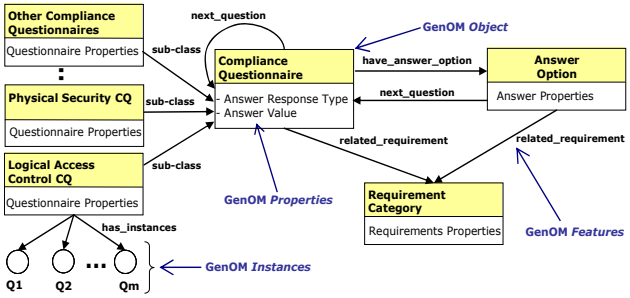


Figure 8: GenOM Representation Format for Requirements Compliance Questionnaire

DECISION POINT	CURRENT DITSCAP PRACTICES	ISSUES WITH CURRENT DITSCAP PRACTICES	BENEFITS OF THE DITSCAP DECISION SUPPORT PDO
What are the criteria to assess compliance levels of the target system with security requirements? **STAKEHOLDERS INVOLVED** Developer, Integrator, or Maintainer, User Representative, DAA, Certifier and Certification Team	• DITSCAP advocates the use of Minimum Security Activity Checklist (MSCL) as well as the Requirements Traceability Matrix (RTM) to record requirements compliance information. • DITSCAP recommends testing procedures to verify and validate the compliance levels of the target system	• *No uniform representation format* exists to collect objective, repeatable and justifiable evidences to establish the compliance level of the target system with the applicable security requirements • DITSCAP is a *long and exhaustive task of gathering target system details* related to security requirements without proper tool support. Such an approach quickly results in an *ad-hoc process* with subjective decision making to establish compliance with security requirements • The MSCL does not have an explicit mapping with the security requirements	• The requirements *compliance questionnaires* of the PDO establish *well-defined criteria to guide a systematic evaluation of compliance level* of the target system with the applicable security requirements. • The PDO supports a *holistic and uniform view of the security requirements* based on the interdependencies among them as well as with other problem domain concepts to promote a *common understanding among stakeholders.* • The PDO provides effective ways to systematically gather evidences for establishing *security requirements applicability and compliance*, *perceive related risks* in the operational environment, and *proactively reveal their relationships with other domain concepts* through the nexus of causal chains that exist in the UoD. The PDO makes these *artifacts* readily available through various inference mechanisms based on its ontological structure and semantics. As an example consider the security requirement marked as "**R1**" and the artifacts "**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 requirement* on the target system and environment.

EXAMPLE REQUIREMENT

R1 **Requirement - EBRP-1 Remote Access Audit Trails for Privileged Functions**

Remote access for privileged functions is discouraged, is permitted only for compelling operational needs, and is **strictly controlled**. In addition to **EBRU-1**, sessions employ security measures such as a **VPN with blocking mode enabled**. A **complete audit trail is recorded**, and the **IAM/O reviews the log** for every remote session.

ARTIFACTS INFERENCED FROM PDO THAT HELP TO ASSESS THE OVERALL IMPACT OF THE SECURITY REQUIREMENT "R1"

T1 **RELATED SECURITY REQUIREMENTS**

REMOTE ACCESS CONTROLS	ENCLAVE BOUNDARY CONTROLS	AUDIT TRAIL CONTROLS
• EBRU-1 Remote Access for User Functions • EBRU-1 Remote Access for User Functions use encryption • EBRU-1 Protection of remote access mechanisms for user functions	• EBPW-1 Public WAN Connection • EBBD-2 Boundary Defense • ECIM-1 Instant Messaging • ECVI-1 Voice over IP • Outsourced application subject to DoD enclave boundary defense.	• 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	THREATS
- DoD Information Systems - Audit Records	- Unauthorized Access - Information Leak
VULNERABILITIES	**COUNTERMEASURES**
- VPN Controls with blocking mode off - Logging mechanisms not enabled	- Monitory System Access - Logging and Reviewing Access - Information Access restriction

T3 **RELATED VIEWPOINTS**

- IA Service – Confidentiality
- Stakeholder Responsibility – IAM/O

T4 **RELATED DITSCAP C&A GOALS**

Identify the Network Connection Rules

Figure 9: Example of Support for Decision-making using the PDO

4. DECISION-MAKING USING THE PDO

To demonstrate the effectiveness of our approach, Figure 9 provides a self-explanatory comparison between the existing practices and the advantages gained through the PDO for critical decision-making activities related to the DITSCAP.

5. RELATED WORK

The need to understand the domain, the interface between the "machine" and the "environment" and the nexus of causal chains that exist between them is very apparent [10] for successful RE. Popular RE methods of goal-driven approaches [28], viewpoints-oriented approaches [12], scenario-based approaches [27] and other techniques that are a combination of them [26] have been developed and experimented with; however, their applicability and selection often restricts the requirements engineer to work with a limited set of modeling constructs and tools that lack interoperability and cross-model reasoning necessary for software-intensive systems.

The Language Extended Lexicon (LEL) approach [21] uses simple hypertext links to represent relationships between its concepts; however, they lack the rich semantics offered by ontological engineering processes. The use of LEL to construct machine understandable ontologies has also been pointed out in [2]. Other efforts for ontology based object-oriented domain modeling have been expressed in [9].

Liu et al [22] analyze security requirements based on the *i** modeling language but a goal and agent-based representation of the domain may not be appropriate for all decision-making activities. In [1] Breaux et al, produce restricted natural-language statements from privacy policies documents; however, their methodology lacks a systematic identification and representation of characteristics associated with requirements from their natural-language descriptions. In [13] Lau et al perform comparisons to identify government regulations with similar provisions. The PDO can help to augment the accuracy and effectiveness of such comparisons.

6. CONTRIBUTIONS & FUTURE WORK

We believe establishing secure systems assurance for software-intensive systems in a socio-technical environment requires knowledge from multiple dimensions, abstractions, and sources. From this perspective, the methodology presented in this paper provides modeling techniques and heuristics that help in capturing the characteristics and constraints of security requirements dispersed across multiple documents and the way these characteristics can be represented using ontological modeling processes to infer valuable knowledge that assists critical decision-making activities for establishing secure systems assurance. Also, the applicability of our methodology is not limited to security requirements domain and can be extended to any domain where the decision-making activities require understanding a large amount of information across various documents. As part of our ongoing and future work, we are in the process of deriving well-designed metrics and measures [16] from various models in the PDO that will help to evaluate the effectiveness of our approach in the real world applications using a case-study designed research methodology [19]. We also plan to study how the models within the PDO can be used and extended to capture the characteristics of other dependability requirements and their relationships in challenging domain applications.

ACKNOWLEDGMENTS: This work is partially supported by the grant (Contract: N65236-05-P-3672) from the Critical Infrastructure Protection Center (CIPC), Space and Naval Warfare (SPAWAR) Systems Center, Department of Navy, Charleston, SC. USA.

7. REFERENCES

[1] Breaux, T.D., Antón, A.I., "Analyzing Goal Semantics for Rights, Obligations, and Permissions." In Proc. of the 13th Int'l Conf. on Requirements Engineering (RE '05), Paris, France, 2005.

[2] Breitman, K.K., Leite, J., "Ontology as a Requirements Engineering Product", In Proc. of the IEEE Int'l Requirements Engineering Conf., 2003

[3] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K., "Jena: implementing the semantic web recommendations," In Proc. of the 13th Int'l World Wide Web Conf., USA, pp: 74-83, 2004

[4] Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., Rice, J. P., "OKBC: a programmatic foundation for knowledge base interoperability," In Proc. of the 15th National/10th Conf. on Artificial intelligence/innovative Applications of Artificial intelligence, AAAI, CA, USA, pp: 600-607, 1998.

[5] DoD 8510.1-M, "DITSCAP Application Manual," 2000.

[6] DoD Instruction 5200.40, "DITSCAP," 1997.

[7] DoDD 8500.1. Information Assurance. Oct. 2002.

[8] DoDI 8500.2. IA Implementation. Feb 2003

[9] Evermann, J., Wand, Y., "Ontology based object-oriented domain modeling: fundamental concepts," Requirements Engineering Journal, Springer 2005.

[10] Jackson, M., "The Meaning of Requirements," Annals of Software Engineering, Vol 3, Baltzer Science Publishers, pp. 5-21, 1997

[11] Kimbell, J., Walrath, M, "Life Cycle Security and DITSCAP". IANewsletter, Vol. 4, No. 2., Spring 2001

[12] Kotonya, G., Sommerville, I., "Requirements Engineering with Viewpoints," BCS/IEE Software Engineering Journal, pp. 5-18, Vol. 11, Issue: 1, Jan. 1996

[13] Lau, G.T., Kincho H. Law, Wiederhold, G., "Analyzing Government Regulations Using Structural and Domain Information," IEEE Computer, Vol. 38, Issue 12, pp.70 – 76, December 2005

[14] Lee, S. W., Gandhi, R. A., Ahn, G, "Establishing Trustworthiness in Services of the Critical Infrastructure: Automating the DITSCAP", In Proc. of the Workshop on Software Engineering for Secure Systems (SESS05), 27th Int'l Conf. on Software Engineering (ICSE '05), pp.43-49, May 2005

[15] Lee, S. W., Gandhi, R. A., Ahn, G., "Security Requirements Driven Risk Assessment for Critical Infrastructure Information Systems", In Proc. of the Symposium on Requirements Engineering for Information Security (SREIS 05), Requirements Engineering, France, IEEE Computer Society, 2005

[16] Lee, S. W., Gandhi, R. A., and Ahn, G., "Certification Process Artifacts Defined as Measurable Units for Software-intensive Systems Lifecycle" To Appear in Software Process Improvement and Practice Journal, Wiley, 2006

[17] Lee, S.W., Gandhi, R.A, "Engineering Dependability Requirements for Software-intensive Systems through the Definition of a Common Language", In Proc. of the 13th IEEE Int'l Requirements Engineering Conf. (RE '05), Workshop on Requirements Engineering for High-Availability Systems (RHAS), Paris, France, pp. 40-48, 2005

[18] Lee, S.W. and Gandhi, R. A., "Ontology-based Active Requirements Engineering Framework", In Proc. of 12th Asia-Pacific Software Engineering Conf. (APSEC '05), Taiwan, 2005. IEEE CS Press

[19] Lee, S.W., Rine, D.C., "Case Study Methodology Designed Research in Software Engineering Methodology Validation," In Proc. of the 16th Int'l Conf. on Software Eng. and Knowledge Eng., Canada, 2004, pp: 117-122

[20] Lee, S.W., Yavagal, D., "GenOM User's Guide V2.0," Technical Report TR-NiSE-05-05, Knowledge Intensive Software Engineering Research Group, Dept. of Software and Information Systems, UNC Charlotte, 2005

[21] Leite JCSP, Franco, APM. "A strategy for conceptual model acquisition," In proc. of the IEEE Int'l symposium on Requirements Engineering (RE '93). IEEE Computer Society Press, Los Alamitos, CA, pp 243–246, 1993

[22] Liu, L., Yu, E., "Designing Information Systems in Social Context: A Goal and Scenario Modeling Approach," Information Systems Journal. Vol. 29(2), Elsevier, 2003

[23] McGuinness, D., van Harmelen, F. (editors), "OWL Web Ontology Language Overview", W3C Recommendation, 10th February 2004

[24] OMB Circular No. A-130, "Management of Federal Information Resources," Feb 8 , 1996

[25] Press Release, "Tom Davis Statement on 2004 Federal Computer Security Report Card Grades", 2004

[26] Rolland, C., Souveyet, C., Achour, C. B., "Guiding Goal Modeling Using Scenarios," IEEE Trans. on Software Eng., Vol. 24:12, pp: 1055-1071, 1998

[27] Sutcliffe, A. "Scenario-based requirements analysis", Requirements Engineering Journal, Vol 3(1), Springer-Verlag, Inc., pp: 48-65, 1998.

[28] van Lamsweerde, A., "Goal-oriented requirements engineering: a guided tour," In Proc. of the 5th Int'l Symp. on RE, Canada, pp. 249-262, 2001