To appear in Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05)
Taipei, Taiwan, Dec. 15 - 17, 2005. IEEE Computer Society Press

# Ontology-based Active Requirements Engineering Framework

Seok Won Lee and Robin A. Gandhi
*Department of Software and Information Systems*
*The University of North Carolina at Charlotte, Charlotte, NC 28223, USA*
*{seoklee, rgandhi}@uncc.edu*

## Abstract

*Software-intensive systems are systems of systems that rely on complex interdependencies among themselves as well as with their operational environment to satisfy the required behavior. As we integrate such systems to create information infrastructures that are critical to the quality of our lives and the businesses they support, the need to effectively predict, control and evolve their behavior is ever increasing. To deal with their complexity, an important first step is to understand and model software-intensive systems, their environments and the interdependencies among them at different levels of abstractions from multiple dimensions. In this paper, we present an Ontology-based Active Requirements Engineering (Onto-ActRE [onto-ǽktər]) framework that adopts a mixed-initiative approach to elicit, represent and analyze the diversity of factors associated with software-intensive systems. The Onto-ActRE framework integrates various RE modeling techniques with complementary semantics in a unifying ontological engineering process. We also present examples from the practice of our framework with appropriate tool support that combines theoretical and practical aspects.*

## 1. Introduction

With advances in information technology, critical information infrastructures that integrate multiple software-intensive systems within and across institutional boundaries play an increasingly important role in the quality of our lives and the efficiency of businesses they support. Typically software-intensive systems are clusters of closely interdependent *systems of systems* where software contributes to a significant portion of the system functionality [24]. Such systems of systems rely on complex interdependencies among themselves as well as with their operational environment to satisfy the required behavior. To cope with such increasingly complex and dynamic interactions, software-intensive systems demand advanced software engineering processes to effectively predict, control and evolve their resulting emergent behavior. We believe that software-intensive systems require an essential way to build a *common language* that creates a shared understanding between stakeholders and promotes cohesiveness between the information gathered from diverse sources to guide their software engineering processes. Such a common language allows to gather a holistic view which includes relationships between the system attributes, their environment and the nexus of causal chains [7] spanning multiple dimensions and levels of abstractions that exist in Universe of Discourse (UoD) to satisfy the real world goals of the associated users, business and organization.

A comprehensive understanding of the problem domain is fundamental to communicate and engineer quality requirements for software-intensive systems. Keeping this idea central to our work, in this paper, we introduce the Onto-ActRE framework. The Onto-ActRE framework adopts a mixed-initiative approach to elicit, represent and analyze the diversity of factors associated with software-intensive systems. It combines various RE modeling techniques with complementary semantics in a unifying ontological engineering process. We choose different RE modeling techniques that capture complementary dimensions of the problem domain as a way to identify the emergent behavior of the software-intensive system working as a whole, under a certain configuration, with various technical and non-technical factors, and their relationships in the UoD. The overarching ontological engineering process then models the gathered information using a uniform representation scheme that promotes cohesiveness between the artifacts generated from different modeling techniques and creates a shared understanding from multiple dimensions. To enable participation from diverse stakeholders and experts alike, the ontological engineering process is supported with appropriate tool support that provides rich modeling constructs with easily understandable semantics.

The rest of the paper is structured as follows. In Section 2 we outline the related work in this area followed by Section 3 that describes the Onto-ActRE framework along with its constituent models and features. Section 4 outlines the GENeric Object Model (GenOM) [12] tool support for various ontological engineering processes in the framework. In Section 5, we motivate the feasibility of our framework by providing a brief overview and examples of its application to the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) [4] automation [11]. Finally we discuss our conclusions and future work in Section 6.

## 2. Related Work

The need to understand the domain, the interface between the "machine" and the "environment" and the interdependencies between them, has been well documented in the RE literature [7], and realized in the research community [17].

Ontologies [22] have long been used and accepted as a means to perform conceptual domain modeling in the knowledge engineering community. In this context, ontologies are interpreted as "an explicit specification of a conceptualization" [23]. To assist autonomous agent interactions, the use of ontologies resulting from the RE process has been suggested by Brietman et al. in [1]. They outline a Language Extended Lexicon (LEL) [14] based approach to structured ontology construction. The LEL by itself does not carry any semantics unless it is instantiated using a conceptual model. More recent efforts for ontology based object-oriented domain modeling have been explored in [6].

We believe that building a *common language* and understanding should not be restricted to specific constructs or techniques which may result in the RE process becoming too narrow-focused or stove-piped that it may fail to capture the key aspects required to engineer comprehensive and accurate requirements. Although popular RE methods of goal-driven approaches [25], viewpoints-oriented approaches [20] [9], scenario-based approaches [18] [21] and other techniques that are a combination of them [15] [19] have been developed and experimented with, their applicability and selection often restricts the requirements engineer to work with a limited set of constructs and tools. Furthermore, such methods offer no support for interaction from other approaches that may be more suitable for communicating and representing a particular set of requirements.

## 3. Ontology-based Active Requirements Engineering Framework

Traditionally, software developers have restricted their focus only to the software system attributes, but the software system itself is embedded within a socio-technical environment that caters to the real world goals of the associated users, business and organization. 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. Figure 1 provides a conceptual overview of the Onto-ActRE framework that takes a step in this direction. The Onto-ActRE framework provides means to understand and evaluate the effects of system functions and constraints in light of the concepts, properties their relationships that exist in the UoD from the perspectives of the real world goals, technology, organization, and business/mission requirements.

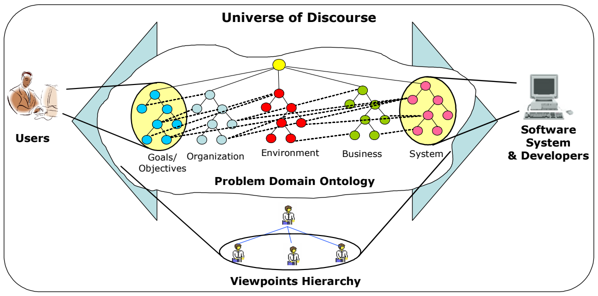

**Figure 1: The Onto-ActRE Conceptual Overview**

We believe that a RE process for software-intensive systems and its underlying representation models should be flexible enough to leverage the strengths of multiple complementary philosophies along with a comprehensive theoretical foundation that supports the interactions between them. To address such needs, the Onto-ActRE framework through its theoretical foundations as a mixed-initiative approach, supports the collaboration between several RE modeling techniques by harnessing the expressiveness of ontologies in a structured way. Within the framework, ontological engineering processes are the primary method of representing and analyzing the information gathered from complementary RE modeling techniques necessary to adequately understand the system under consideration. The Onto-ActRE framework includes contributions from popular and well-studied RE modeling techniques based on the notions of goals, viewpoints, scenarios, and their combinations. More specifically, the Onto-ActRE framework includes models and methods for 1) Goal-driven scenario composition; 2) Requirements domain model; 3) Viewpoints hierarchy, and 4) Other domain specific taxonomies to hierarchically organize the application domain concepts, properties and their relationships.

The product of applying the Onto-ActRE framework is a Problem Domain Ontology (PDO) that provides the definition of a common language. The PDO is a machine understandable, hierarchical representation, engineered using object-oriented ontological domain modeling techniques. Figure 2 depicts several possible elements of the Onto-ActRE PDO. The PDO construction involves classification and categorization of information available from various sources during the requirements engineering phase to create hierarchical representations. These structured representations include generic concepts at the top which are decomposed into specific concepts at the bottom. The inherent benefits of the PDO lie in the uniformity of its representation that promotes cohesiveness between artifacts generated throughout the RE lifecycle. Onto-ActRE is an "active" requirement engineering framework as it moves away from static requirement repositories to active ones that link requirements to the concepts, properties and their relationships in the UoD at different levels of abstractions from multiple dimensions. Such an approach provides a systematic way to understand and predict the emergent behavior of the system through its traceable rationales and supporting infrastructures.

We now elaborate more on various Onto-ActRE models, methods and their interactions to provide guidelines for RE practice using this framework.

### 3.1. Goal-Driven Scenario Composition

The goal-driven scenario composition method of the Onto-ActRE framework results in the capture of the real world goals and objectives of the system at different levels of abstractions to form a goal hierarchy. From a process perspective such goals can also represent tasks and activities at different levels of detail that need to be performed. The possible realization criteria for the goals in the hierarchy are captured using leaf node scenarios. Through a systematic derivation of scenarios from the goals or vice versa, the coverage of the scenarios over the application domain can be established or in the other case, the goals selected for a scenario provides constraints to restrict its scope. This method can

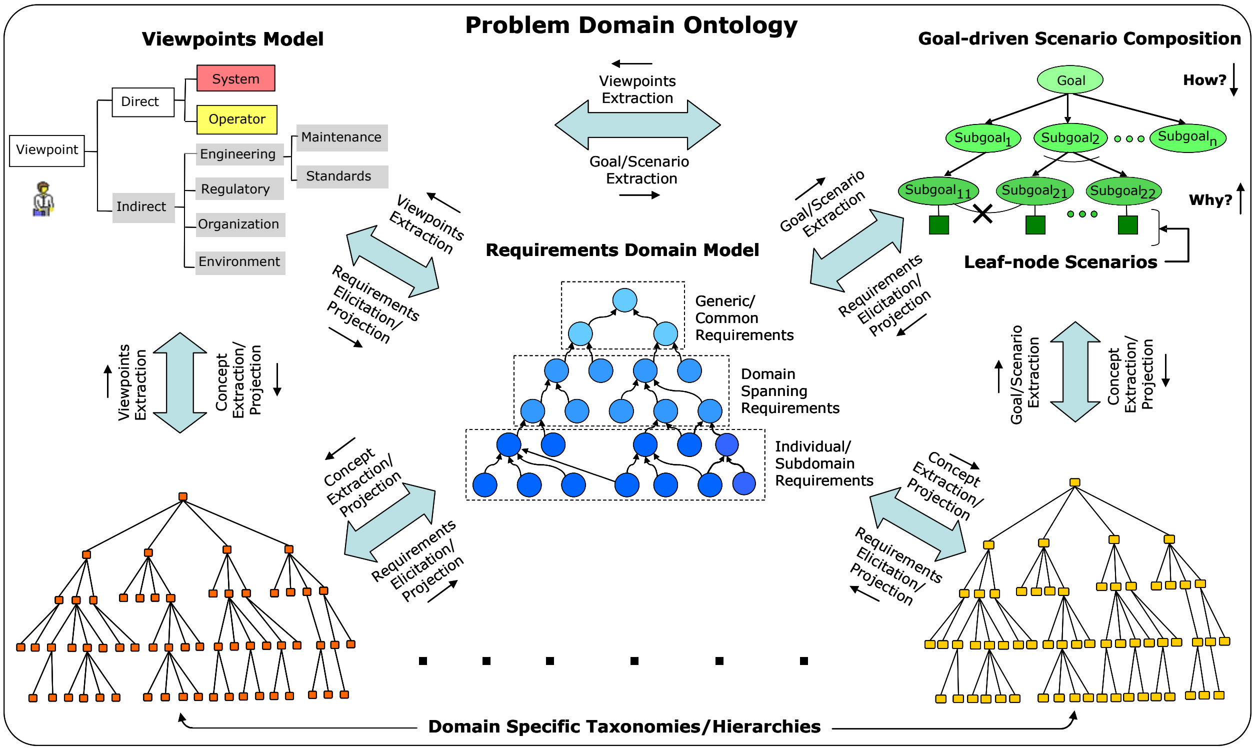
Problem Domain Ontology
Viewpoints Model
Viewpoint
Direct
System
Operator
Indirect
Engineering
Maintenance
Standards
Regulatory
Organization
Environment
Viewpoints Extraction
Goal/Scenario Extraction
Goal-driven Scenario Composition
Goal
How?
Subgoal$_1$
Subgoal$_2$
Subgoal$_n$
Subgoal$_{11}$
Subgoal$_{21}$
Subgoal$_{22}$
Why?
Leaf-node Scenarios
Viewpoints Extraction
Requirements Elicitation/ Projection
Requirements Domain Model
Goal/Scenario Extraction
Requirements Elicitation/ Projection
Generic/ Common Requirements
Domain Spanning Requirements
Individual/ Subdomain Requirements
Viewpoints Extraction
Concept Extraction/ Projection
Goal/Scenario Extraction
Concept Extraction/ Projection
Concept Extraction/ Projection
Requirements Elicitation/ Projection
Concept Extraction/ Projection
Requirements Elicitation/ Projection
Domain Specific Taxonomies/Hierarchies

**Figure 2: The Onto-ActRE Problem Domain Ontology**

leverage the effectiveness of existing well defined techniques for goal [25] and scenario [21] based approaches in a cohesive and integrated fashion. Figure 2 shows how the resulting hierarchical model participates in the overall Onto-ActRE framework.

### 3.2. Requirements Domain Model (RDM)

The RDM organizes the problem domain requirements through a hierarchical representation that includes top-level generic requirements, mid-level domain spanning requirements and leaf-node sub-domain requirements as shown in Figure 2. Such an organization of requirements allows for their exploration to be conservative in nature i.e. to be more inclusive rather than exclusive. The scope of the RDM spans over the system requirements (functional and non-functional) and its related entities in the environment such as organization, business/mission requirements and other domain-specific considerations. Examples of sources for requirements in the RDM can be natural language requirements, requirements from C&A standards, taxonomies, transcripts, manuals, organizational policies, domain knowledge, environmental constraints, laws and regulation, etc. The requirements in the RDM also have several properties that help to characterize them. Furthermore, the PDO allows to utilize the relationships that exist between requirements in the RDM with other concepts in the PDO such as goals in the goal hierarchy, associated stakeholders in a viewpoints hierarchy, and other domain specific taxonomies as a natural way to organize the diversity of factors associated with requirements.

### 3.3. Viewpoints Hierarchy and other Domain Specific Taxonomies

Requirements usually capture ideas, perspectives and relationships at various levels of detail and they are interpreted differently from different viewpoints [8]. For simplicity in identifying such viewpoints we advocate the use of the VORD [9] viewpoints class template, as shown in Figure 2, to create a viewpoints hierarchy consisting of various stakeholders, services or concerns associated with the system and the environment. Viewpoints in the hierarchy can be extracted from the goals or the RDM or vice versa depending on the domain of application or the availability of the sources. The higher level non-leaf nodes in the viewpoints hierarchy specifically consists of viewpoints, such as organizational viewpoints which map to generic requirements in the RDM, and the leaf nodes represent viewpoints such as those of specific system stakeholders, services or concerns that are related to specific requirements in the leaf nodes of the RDM.

The PDO can also contain other domain specific taxonomies or hierarchies of concepts which help to understand or analyze the problem domain. For example, a domain that involves security requirements may include a risk assessment taxonomy and/or a network vulnerabilities taxonomy. Such additional dimensions also help to synergistically understand and link to various knowledge artifacts in the Onto-ActRE PDO. The synergy between various models in the PDO is also indicated in Figure 2.

## 4. Onto-ActRE Tool Support

The ontological engineering processes to represent and analyze the information gathered through models and methods outlined in the previous sections are supported by the GENeric Object Model (GenOM) [12] tool support. 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 [3] as well as the OWL representation [16] format. The conceptual architecture of GenOM is shown in Figure 3.

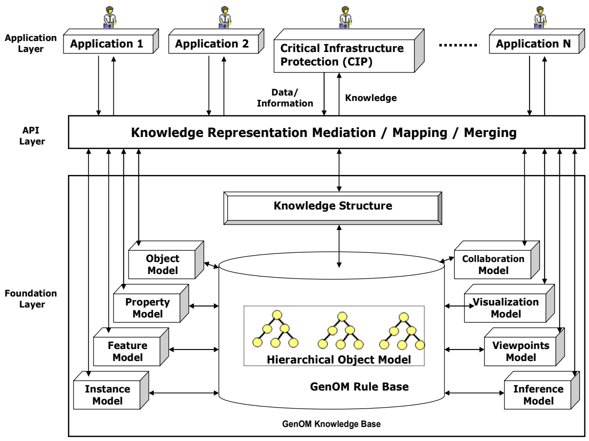

**Figure 3: GenOM Conceptual Architecture**

The GenOM meta-language consists of *objects*, *properties*, and *features* with semantics that effectively support knowledge acquisition and representation. GenOM *objects* with support for single or multiple inheritances are used to model hierarchical structures that describe the concepts in a domain. GenOM *properties* are used to describe the characteristics or attributes of *objects* and *features*. Finally, GenOM *features* are used to describe the relationship or dependencies that exist between objects. Once the *objects*, *properties* and *features* are defined, they are instantiated to represent specific *instances* that exist in a problem domain. GenOM is also associated with an inference engine [2] which 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).

# 5. The DITSCAP Automation Case Study

The Onto-ActRE framework and GenOM tool support are currently being applied to the automation of the DITSCAP [11]. The Defense Information Infrastructure (DII) connects the Department of Defense (DoD) mission support, command and control, and intelligence computers and users through voice, data, imagery, video, and multimedia services, and provides information processing and value-added services. For such a critical infrastructure, the DoD requires all constituent systems in the DII to be certified and accredited following the DITSCAP [4]. The DITSCAP provides an excellent platform to assess the security of software-intensive systems from organizational, business, technical and human aspects while supporting an infrastructure-centric approach [5]. However, DITSCAP itself can be quite overwhelming due to its long and exhaustive process of information gathering, documentation and analysis suggested by a multitude of DoD directives and security requisites. The complexity of software-intensive systems, their diverse operational environments and the exhaustive breath and depth of the certification process significantly diminish the effectiveness of DITSCAP in the real world.

To overcome the shortcomings of the DITSCAP in understanding and evaluating software-intensive systems, several hierarchical models are constructed following the models and methods of the Onto-ActRE framework. Based on our approach, the creation of a DITSCAP PDO helps to elicit, represent and analyze the information gathered throughout the C&A process within a unifying framework. The DITSCAP PDO specifically includes structured and well-defined representations of: 1) A requirements domain model based on DITSCAP-oriented directives, security requirements and enforced policies with leaf-node compliance assessment questionnaires; 2) Overall DITSCAP process aspect knowledge that includes a hierarchy of C&A goals with process realization criteria as leaf-node questionnaires (scenarios); 3) A viewpoints hierarchy that includes system stakeholders and information assurance services; 4) A risk assessment taxonomy that aggregates a broad spectrum of risks from the DITSCAP domain; 5) Meta-knowledge about information learned from network discovery/monitoring tools; and 6) Interdependencies between concepts in the DITSCAP PDO. It is easy to see how these models map to models within the Onto-ActRE framework.

In the context of the C&A process, the DITSCAP PDO helps to gather information from different dimensions to assess system compliance with the certification criteria.

## 5.1. Multi-Dimensional Link Analysis

Lee et al [13] suggest that "Individual pieces of information finally become valuable knowledge when they establish 'links' with each other from various aspects/dimensions based on a certain set of goals". Using this as an underlying theory we introduce the concept of Multi-Dimensional Link Analysis (MDLA) within the Onto-ActRE framework. Through MDLA, the core pieces of information required to predict and control the emergent behavior of the system can be systematically identified from the definition of a common language and its underlying models. In the DITSCAP domain, understanding such behavior contributes to effectively estimate and enforce security requirements for software-intensive systems. A motivating example of MDLA in the DITSCAP domain is shown in Figure 4. The GenOM instance visualization shown in Figure 4 depicts the relationships of DITSCAP-enforced security requirements with C&A goals, related/dependent requirements, associated stakeholders in the domain, compliance criteria, requirements source, and related risk factors.

The boxes and their interconnections in Figure 4 represent *instances* of various domain *objects* and *features* modeled in GenOM for the DITSCAP PDO. The security requirement *instance* under consideration in Figure 4 is "Enclave Boundary Defenses" requirement which concerns with firewalls being installed at appropriate places in a network for the target system. This requirement is an *instance* of the "Network Controls" category *object* of the requirements

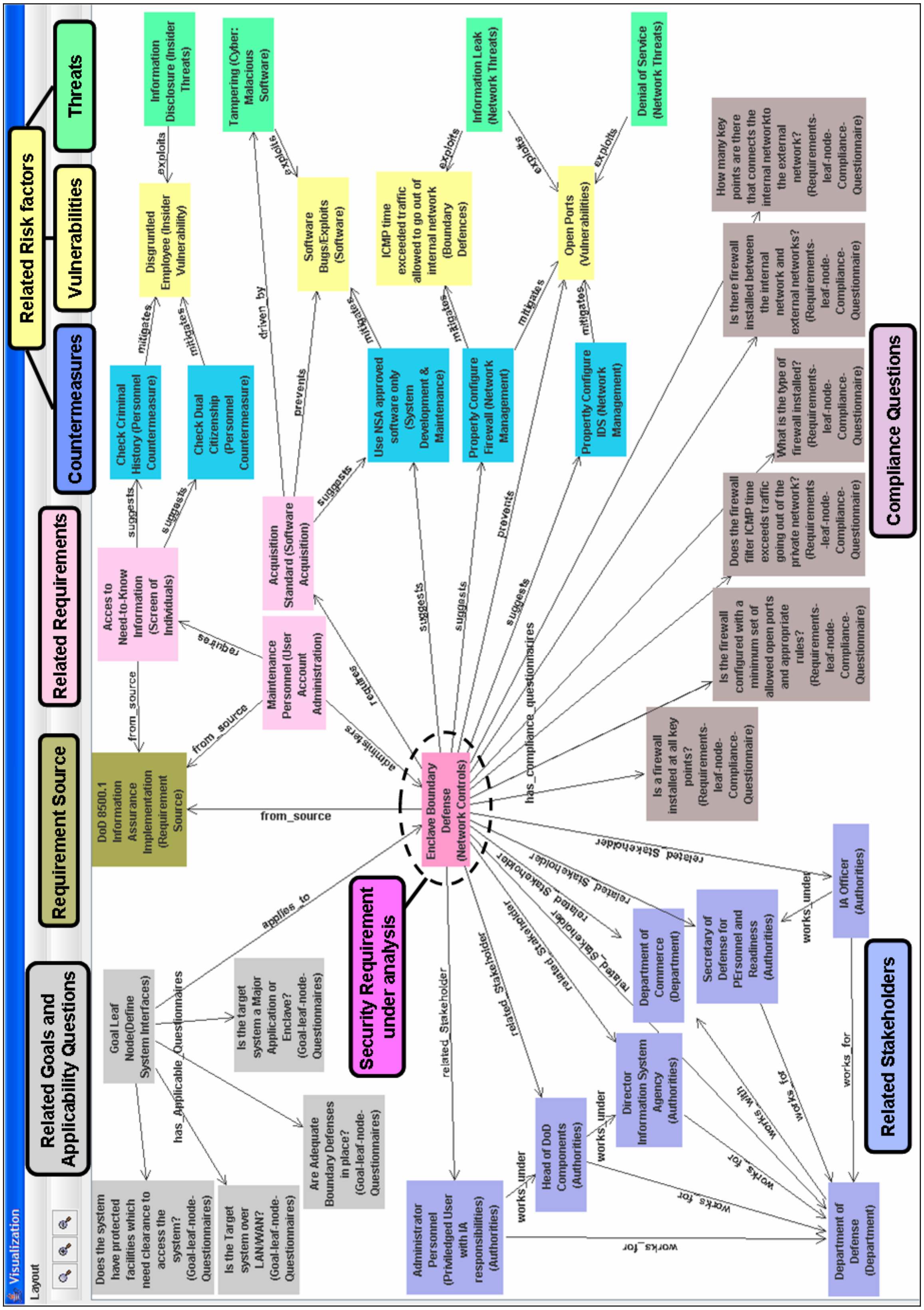
Visualization
Layout
Related Goals and Applicability Questions
Requirement Source
Related Requirements
Countermeasures
Related Risk factors
Vulnerabilities
Threats
Security Requirement under analysis
Compliance Questions
Related Stakeholders
Does the system have protected facilities which need clearance to access the system? (Goal-leaf-node-Questionnaires)
Is the Target system over LAN/WAN? (Goal-leaf-node-Questionnaires)
Are Adequate Boundary Defenses in place? (Goal-leaf-node-Questionnaires)
Goal Leaf Node(Define System Interfaces)
Is the target system a Major Application or Enclave? (Goal-leaf-node-Questionnaires)
has_Applicable_Questionnaires
applies_to
DoD 8500.1 Information Assurance Implementation (Requirement Source)
from_source
Acces to Need-to-Know Information (Screen of Individuals)
Maintenance Personnel (User Account Administration)
Acquisition Standard (Software Acquisition)
requires
administers
Check Criminal History (Personnel Countermeasure)
Check Dual Citizenship (Personnel Countermeasure)
suggests
mitigates
Disgruntled Employee (Insider Vulnerability)
exploits
Information Disclosure (Insider Threats)
Tampering (Cyber: Malacious Software)
driven_by
prevents
Software Bugs/Exploits (Software)
Use NSA approved software only (System Development & Maintenance)
ICMP time exceeded traffic allowed to go out of internal network (Boundary Defences)
Properly Configure Firewall (Network Management)
Information Leak (Network Threats)
Open Ports (Vulnerabilities)
Properly Configure IDS (Network Management)
Denial of Service (Network Threats)
Enclave Boundary Defense (Network Controls)
has_compliance_questionnaires
Is a firewall installed at all key points? (Requirements-leaf-node-Compliance-Questionnaire)
Is the firewall configured with a minimum set of allowed open ports and appropriate rules? (Requirements-leaf-node-Compliance-Questionnaire)
Does the firewall filter ICMP time exceeds traffic going out of the private network? (Requirements -leaf-node-Compliance-Questionnaire)
What is the type of firewall installed? (Requirements-leaf-node-Compliance-Questionnaire)
Is there firewall installed between the internal network and external networks? (Requirements-leaf-node-Compliance-Questionnaire)
How many key points are there that connects the internal networkto the external network? (Requirements-leaf-node-Compliance-Questionnaire)
related_Stakeholder
Administrator Personnel (Priviledged User with IA responsibilities) (Authorities)
Head of DoD Components (Authorities)
Director Information System Agency (Authorities)
Department of Commerce (Department)
Secretary of Defense for PErsonnel and Readiness (Authorities)
IA Officer (Authorities)
Department of Defense (Department)
works_under
works_for
works_with

**Figure 4: Overview of MDLA in DITSCAP Automation Tool supported by GenOM Visualization**

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

The "Enclave Boundary Defenses" requirement *instance* also relates to *instances* of other models within the DITSCAP PDO through several *features* that represent non-taxonomic relationships among them. In Figure 4, the "Enclave Boundary Defenses" *instance* relates to *instances* of the DITSCAP C&A goal hierarchy as well as *instances* of the viewpoints hierarchy through the "applies-to" and "related-stakeholder" *features* respectively. Other *features* such as "from-source" relate the requirement to the source from which it was extracted. Also the relationships of the "Enclave Boundary Defenses" requirement with other requirements *instances* are captured through the "administers" and "requires" *features*.

The ontological characteristics of the DITSCAP PDO also helps in utilizing the relationships that exist between security requirements and the various factors considered for risk assessment [10]. By taking advantage of the synergy between these models, we can systematically identify the threats, vulnerabilities and countermeasures associated with the target system from the compliance information gathered for the security requirements. From a requirements perspective, the relationships between risk factors can help to identify and elaborate on the interdependencies between requirements which may not be readily apparent. Figure 4 depicts the relationships between "Enclave Boundary Defenses" requirement and risk factors with the *features* "suggest", "prevents" and "driven by" for countermeasures, vulnerabilities and threats respectively. The *features* "mitigate" and "exploits", capture the relationships between risk factors in the DITSCAP PDO risk assessment taxonomy. The information available through such relationships when combined with asset value and mission criticality provides the basis to perform cost-benefit analysis and requirements prioritization necessary to establish *adequate security*.

We believe that the information gathered from such diverse sources helps to understand various aspects of DITSCAP-oriented requirements to facilitate their realization, and evaluation in the real world. MDLA's integrated framework for analytical analysis promotes assurance for a comprehensive coverage of the requirements space by actively assisting the process of discovering missing, conflicting, and interdependent pieces of information as well as establishing metrics and measures based on common understanding and the reflected language from multiple dimensions.

## 6. Conclusions and Future Work

In this paper, we present a novel framework, Ontology-based Active Requirements Engineering, that integrates various RE and knowledge engineering techniques to address the complexities of software-intensive systems. Although the scope of this paper does not allow us to elaborate on all aspects of our framework, it has been engineered as an integrated and unique combination of techniques that facilitates eliciting and capturing of requirements and specifications, modeling of system environments and domain knowledge, managing software evolution and adaptability to change and, supporting analysis and design processes through decision support and traceability.

To foster the systematic usage of the Onto-ActRE framework in other applications and domains we identify the need for formal definitions and empirical evaluations and consider them as our future work. Our experiences with DITSCAP automation and other application domains will help us further evolve the Onto-ActRE framework.

**Acknowledgements.** 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, Charleston, SC. USA. The authors acknowledge the support from Scott West, John Linden, Bill Bolick, Gail-Joon Ahn and Bill Chu. Finally, the authors thank Deepak Yavagal, Divya Muthurajan, and Swapnil Brahmankar for their contributions to this research.

## 7. References

[1] Breitman, K. K., and Leite, J. C. S. P., "Ontology as a Requirement Engineering Product," In Proceedings of the 11th IEEE International Requirements Engineering Conference, Monterey Bay, California, USA, September 2003, pp. 309-319.

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

[3] Chaudhri, V. K., Farquhar, A., Fikes, R., Karp, P. D., and Rice, J. P., "OKBC: a programmatic

foundation for knowledge base interoperability," In Proceedings of the 15th National/10th Conference on Artificial intelligence/innovative Applications of Artificial intelligence, AAAI, Menlo Park, CA, 1998, pp. 600-607.

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

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

[6] Evermann, J., Wand, Y., "Ontology based object-oriented domain modeling: fundamental concepts," Requirements Engineering, Springer-Verlag London Ltd., 2005.

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

[8] Kotonya, G., and Sommerville, I., "Requirements Engineering - Processes and Techniques," New York, John Wiley, 1998.

[9] Kotonya, G. and Sommerville, I., "Viewpoints for requirements definition," BCS/IEE Software Engineering Journal, Vol. 7(6), 1992, pp. 75-87.

[10] Lee, S. W., Gandhi, R. A., and Ahn, G., "Security Requirements Driven Risk Assessment for Critical Infrastructure Information Systems", In Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS 05), RE '05, Paris, France, August, 2005.

[11] Lee, S. W., Gandhi, R. A., and Ahn, G.: Establishing Trustworthiness in Services of the Critical Infrastructure: Automating the DITSCAP. In Proceedings of the workshop on Software Engineering for Secure Systems (SESS05), 27th International Conference on Software Engineering, 2005, pp. 43-49.

[12] Lee, S.W. and Yavagal, D., "GenOM User's Guide," Technical Report TR-SIS-NISE-04-01, Knowledge Intensive Software Engineering Research Group, Dept. of Software and Information Systems, UNC Charlotte, 2004.

[13] Lee, S.W. and, Rine D.C., "Missing Requirements and Relationship Discovery through Proxy Viewpoints Model," STUDIA INFORMATICA UNIVERSALIS: International Journal on Informatics, Regular Issue Vol. 3, No. 3, December 2004, pp. 315-342.

[14] Leite, J.C.S.P., Franco, A.P.M., "A strategy for conceptual model acquisition," In 1st IEEE international symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, CA, 1993, pp 243–246.

[15] Liu, L., Yu, E., "Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach," Information Systems Journal, Vol. 29(2), Elsevier, Apr. 2004, pp. 187-203.

[16] McGuinness, D., and van Harmelen, F. (editors), "OWL Web Ontology Language Overview", W3C Recommendation, 10th February, 2004, http://www.w3.org/TR/owl-features/

[17] Offen R., "Domain Understanding is key to successful to system development" Requirements Engineering Journal, Vol. 7(3) Springer-Verlag London Ltd., 2002, pp.172 – 175.

[18] Potts, C., Takahashi, K., Anton A.I., "Inquiry-Based Requirements Analysis," IEEE Software, 1994, pp. 21-32.

[19] Rolland, C., Souveyet, C., Achour, C. B., "Guiding Goal Modeling Using Scenarios," In IEEE Transactions on Software Engineering, Vol. 24(12), 1998, pp. 1055-1071

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

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

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

[23] T.R. Gruber, "A translation approach to portable ontology specification," Knowledge Acquisition 5, 1993, pp. 199–220.

[24] The Integration of Software-Intensive Systems (ISIS) initiative. Carnegie Mellon Software Engineering Institute (SEI), URL: http://www.sei.cmu.edu/isis/isis-main.html

[25] van Lamsweerde, A., "Goal-oriented requirements engineering: a guided tour," In Proceedings of the 5th International Symposium on Requirements Engineering, Toronto, Canada, August 2001, pp. 249-262.