Online Journal of Applied Knowledge Management

A Publication of the International Institute for Applied Knowledge Management

Volume 13, Issue 1, 2025

The ontology for SOC creation assistance and replication

Justin M. Novak, Software Engineering Institute, Carnegie Mellon University, USA, jnovak@cert.org

Angel L. Hueca, Software Engineering Institute, Carnegie Mellon University, USA, alhueca@cert.org

Samuel J. Perl, Software Engineering Institute, Carnegie Mellon University, USA, sjperl@cert.org

Christopher I. Rodman, Software Engineering Institute, Carnegie Mellon University, USA, cirodman@cert.org

Abstract

A Security Operations Center (SOC) is an indispensable tool for any modern organization or enterprise to secure its digital data and information assets. Developing SOCs and SOC capabilities to meet organizational needs in today's threat environment is an often laborious, time-consuming, and expensive task that (if not done correctly) may leave organizational goals unfulfilled. In this paper, we introduce the Ontology for SOC Creation Assistance and Replication (OSCAR), which organizations can use to aid in developing SOCs and in planning and evaluating SOC capabilities. We developed OSCAR using a purpose-built dataset created by extracting the knowledge of numerous SOC expert practitioners. OSCAR is organized into a knowledge hierarchy that includes people, process, and technology classes, but also emphasizes planning and functional considerations. OSCAR accomplishes two things. First, it fills a gap in existing cyber ontology literature by including classes for the initial development of SOCs in addition to those for security operations capabilities. Second, its domain-specific knowledge is derived from a unique dataset gathered directly from experts working in the field. Taken together, these unique traits make OSCAR an ideal tool for planning, building, and evaluating SOCs.

Keywords: SOC, security operations center, cybersecurity, CSIRT, ontology, framework.

Introduction

Security operations is a domain that addresses an organization's readiness to detect, respond to, recover from, and withstand the effects of attacks or incidents that affect its information systems and data. Functionally, for an organization, a security operations center (SOC) implements this readiness by combining elements of a cybersecurity team, an information technology (IT) department, and other relevant business functions. Thus, a SOC is a highly specialized capability within an organization. It has its own mission, a unique set of goals and needs, and a corresponding set of capabilities that it will need to fulfill these requirements.

This set of SOC requirements and capabilities may vary widely from one organization to the next, reflecting the overall goals or mission of the organization or perhaps some other criteria. As a

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34
Accepting Editor: Meir Russ

result, each deployment of a SOC capability will be different from all others, a reality that can make developing a SOC difficult. There is no one-size-fits-all solution, no single blueprint, and no 'SOC in a Box' that can meet the needs of organizations with different goals, budgets, and threat profiles. However, we can frame SOC requirements and capabilities as a unique domain of knowledge, which is not duplicated by any other domain. By framing it as such, we can say that its knowledge can be organized into a structured set of concepts and relationships, which is to say, an ontology.

In this paper, we describe the Ontology for SOC Creation Assistance and Replication (OSCAR), proposed previously by Novak et al. (2025) as a solution for the problem posed above. We demonstrate that by organizing the knowledge domain that covers creating and developing SOCs, it is possible to do more than simply replicate a SOC capability. Instead, we can replicate the process by which such a capability is developed. In practice, this allows OSCAR to supplant much of the domain expertise needed for SOC development, the rarity of which is what makes SOC development difficult and expensive. Further, unlike a framework or set of standards, OSCAR (as an ontology) can reason relationships among requirements and capabilities as well as other areas. These added insights go beyond what is easily observed and offer new insights into SOC development.

We recognize that, while all SOCs share a common domain of knowledge, the different iterations of SOC types (e.g., on-premises vs. off-premises, government vs. private sector) may have their own sub-domains. To properly define OSCAR, we have applied the following constraints in the development of this ontology:

- OSCAR is applicable to SOCs that support government organizations. While the knowledge is likely to be applicable to many other SOC types, we focused on building a full depth of knowledge in this specific sub-domain.
- OSCAR specifically addresses the needs of on-premises SOC capabilities. While this does not discount the possibility that such an SOC may use cloud-based or off-premises technology or tools, we assume that this knowledge is most applicable to SOCs deployed on-premises and operated by the host organization.
- We define a SOC as an organization that must, at a minimum, perform incident response and network monitoring functions. Therefore, OSCAR assumes that a SOC must perform these functions, although it may (or may not) perform other functions as well.

The rest of this paper is organized as follows: In the first section, we describe the SOC development knowledge domain and outline existing ontologies and other knowledge bases in this area. The second section outlines the process used for developing OSCAR. The third section reviews OSCAR's knowledge class hierarchy, data relationships, and inferences. The fourth section discusses the effectiveness and utility of OSCAR, including validating the data used. Section five concludes the paper.

Background and Related Work

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

In this section, we examine the related works that discuss the current state of SOC composition as well as methodologies and tools that are being used for SOC development. We also briefly review relevant ontologies and ontological approaches within the cybersecurity domain.

A SOC is "a centralized team in a single organization that monitors [the] information technology environment for vulnerabilities, unauthorized activity, acceptable use/policy/procedure violations, intrusions into and out of the network and provides direct support of the cyber incident response process" (Murdoch, 2019). We view this definition and the duties loosely defined therein as a specific set of responsibilities for maintaining the cybersecurity of an organization.

Torres offers another model for SOC team operations, emphasizing a "triad of people, process, and technology" as a model for understanding what the composition of an effective SOC should look like (Torres, 2015). This triad of people, processes, and technology (PPT) may depend on the services and functions that a SOC provides via a series of definitions and relationships. Establishing a security operations capability for any service invariably results in challenges that arise when aligning the skills and talents of people, regulation and policy changes over time, and the rapid evolution of the technology landscape (Torres, 2015).

Advancing the security operations capability over time requires assessment activities to be regularly conducted. Majid and Ariffi (2019) described the need to continually improve a SOC as a vital factor in maintaining its relevancy over time. The SOC Capability Maturity Model (SOC-CMM) (van OS, 2018) is one tool that is used for the continual evaluation and improvement of security operations teams. The SOC-CMM process also utilizes the PPT triad and aligns it with the services that the SOC provides to develop a repeatable assessment methodology. The capstone of the SOC-CMM process is the definition of a SOC Target Operations Model (SOCTOM) or a "desired state that the SOC needs to move forward" (van OS, 2022). A key component of the SOC-CMM is the SOCTOM assessment tool itself, which allows expert cybersecurity practitioners to track a SOC's maturity within individual "domains" of the SOC-CMM, their "aspects," and specific "elements." These maturity measures are similar to those offered by Onwubiko and Ouazzane (2019), Novak et al. (2021), and in assessment tools such as SIM3 (Stikvoort, 2015).

Methodologies and guidelines such as these are designed to result in either an assessment framework or an assessment procedure. However, other considerations, such as the unique aspects of each organization (Mansfield-Devine, 2016) and business value chain tie-ins (Murdoch, 2019), require that security experts carefully apply these maturation methodologies within each SOC instantiation, which often requires specific insight and customization for each use case. This case-specific customization suggests that truly understanding the development and maturation of the SOC cannot be reduced to a single methodology or framework.

To solve this need for singular solutions, we propose employing an ontology. According to Uschold and Gruninger (1996), an ontology describes the world view of a domain, which includes entities, attributes, and processes of the domain, and then defines the concepts and their interrelationships. Uschold and Gruninger (1996) generally described the challenges that ontologies aim to address and cite specific examples of entities and their relationships within an IT system.

There is a body of research demonstrating approaches for applying ontologies to cybersecurity. For example, Georgesecu and Smeureanu (2017) demonstrated use cases for correlating data sources and attack methods, while Ferreria et al. (2023) described using an ontology to build a recommender system for cybersecurity. Some larger-scale cybersecurity ontologies also exist. Syed et al. (2016) proposed the Unified Cybersecurity Ontology (UCO), which is intended to promote situational awareness for organizations based on cybersecurity standards and information exchange. UCO has been mapped to several existing ontologies and aims to serve as the core knowledge base for the cybersecurity domain at large.

Focusing on specific sub-domains of cybersecurity, Wang et al. (2023) proposed an ontology for network situational awareness called the System Security Assurance Ontology (SSAO), while Onwubiko (2018) described the Cybersecurity Operations Centre Ontology for Analysis (CoCoa). These sub-domain-specific ontologies are a useful parallel to the proposed OSCAR ontology.

Methods

Requirements Analysis

Several considerations need to be addressed when working with the types of government SOCs that the OSCAR ontology proposed by this research is constrained to. For example, government entities may have unique regulatory or compliance requirements, and the SOC function may therefore be responsible for addressing these. For the purposes of this research, guidance and best practices were identified from cybersecurity standards published by organizations such as the National Institute of Standards and Technology (NIST), the International Organization for Standardization (ISO), and MITRE. The OSCAR ontology engineering process then used this guidance to develop a specific focus on organizational demographics to help identify the organization's unique requirements, such as the regulatory requirements for a government entity, in addition to any functional requirements. Identifying these organization-specific requirements helped us identify granular organizational needs based on the organization's government sector or other necessary data-protection levels. The OSCAR ontology helps to answer these questions, thereby providing the organization with a template for establishing a compliant SOC.

Development Methodology

As we have highlighted in prior research (Novak et al., 2025), a SOC-development dataset has been developed based on the experiences and expert knowledge of SOC, computer emergency response team (CERT), and computer security incident response team (CSIRT) leaders from across academia, government, and private industry. These experts participated in structured interviews to identify challenges and solutions encountered during the development of a modern SOC. These structured interviews were recorded and transcribed, resulting in more than 100,000 words from more than 20 hours of interviews. In our research, we leverage this dataset to extract and define several hundred unique classes of knowledge, which are part of a structured body of SOC-development knowledge. We identify and code those classes. A visualization of the code is available in Figure 1.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

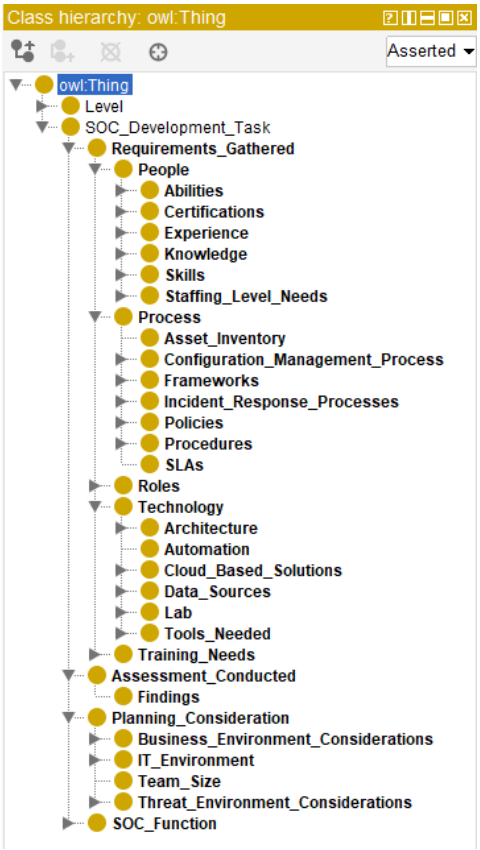

Figure 1. Coded Data Classes Used to Develop OSCAR. Captured from Protégé Model

This knowledge coding forms the basis of the OSCAR ontology. Common classes from the dataset, which we used to form the highest levels of OSCAR's hierarchy, include the following:

- level of SOC maturity for a given SOC function
- SOC development tasks
- planning considerations
- requirements gathering
- SOC functions

These high-level classes include a series of nested subclasses. We discuss the complete class hierarchy for OSCAR in further detail in the Results section of this paper.

SOC Maturity Levels

A 2021 Software Engineering Institute (SEI) report defined what is known as the CSIRT Capacity Development Continuum. This continuum identifies six levels of development for CSIRTs (Novak et al., 2021). While this artifact refers to CSIRTs, the levels themselves are functionally agnostic and refer only to levels of maturity. Therefore, we adapted this continuum for the purpose of defining SOC maturity. In our adaptation, we omit the first level, *Level 0: Nascent*, since organizations that are building their cyber capacity already have some type of CERT or CSIRT operations. The remaining SOC development levels range from *Level 1: Developing* to *Level 5:*

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

Leading Edge. Table 1 shows the complete mapping of the identified development criteria for the remaining five levels to SOC development.

Table 1. OSCAR CSIRT Framework – SOC Categorization Mapping

CSIRT Framework Level	SOC Level
1: Developing	1 – The SOC has limited capacity and/or technical capabilities to perform SOC tasks.
2: Capable	2 – The SOC has limited leadership direction, and ad hoc processes, policies, and procedures are in development.
3: Sustaining	3 – The SOC has some established policies/practices and performs some monitoring and response functions in accordance with organizational requirements.
4: Contributing Partner	4 – The SOC has established policies/practices, performs monitoring and response in accordance with organizational requirements, and performs information sharing within the country, assisting with developing critical information infrastructure (CII) sectors and sector CSIRTs within the country.
5: Leading Edge	5 – The SOC has established policies/practices, performs monitoring and response in accordance with organizational requirements, is a leader in the region, coordinates information sharing within the country and with international partners and develops CII sectors, and supports the sector CSIRTs network.

Note. Adapted from Novak et al. (2021)

Using the SEI CSIRT Capacity Development Continuum as a model, we mapped existing SOC capacity to a level of expected maturity and regulatory compliance. This mapping includes reviewing existing cybersecurity standards and guidelines, such as the Forum of Incident Response and Security Teams (FIRST) CSIRT Services Framework (FIRST, 2019), NIST guidance (NIST 800-61r2, NIST 800-86), and ISO frameworks (ISO 27001).

We used these common cybersecurity standards to identify the necessary components of a high-performing SOC that is able to meet all requirements to successfully operate and conduct any of the numerous cybersecurity functions identified in applicable cybersecurity standards, guidelines, and frameworks. We then considered, filtered, and assigned data to specific SOC maturity levels (ranging from level 1 to level 5). We collected this data through structured interviews; requirements from cybersecurity standards, guidelines, frameworks, and the research teams on the ground; and hands-on experience.

Based on the data collected in the interview process described above, we looked at the existing standards and frameworks previously mentioned. These standards and frameworks provided a template of best practices and expectations for fully functioning SOCs at different maturity levels in the different service areas. This is discussed further in the Results section.

Overview of the OSCAR Ontology

In developing the OSCAR ontology, we used a modified version of the *Stanford Seven Step Ontology Development Methodology* (Noy & McGuinness, 2001), which answers questions that apply to OSCAR.

The first step in this methodology is to determine the domain and scope of the ontology. To do so, you must answer a series of basic questions, which we do briefly here, and in further depth in Appendix 1:

A. *What is the domain that the ontology will cover?* Creating SOCs and related SOC-like capabilities, primarily in the government sector.
B. *For what are we going to use the ontology?* Addressing the challenges that government agencies face when standing up a SOC.
C. *For what type of questions, the information in the ontology should provide answers?* What PPTs are required for a given SOC or SOC capability?
D. *Who will use and maintain this ontology?* Developers of SOCs and the community of SOCs and incident response teams

The second step in the *Seven Step Ontology Development Methodology* is to consider reusing existing ontologies. As we already discussed, there are existing ontologies that offer insight and perspectives valuable to OSCAR. We reviewed them and brought insights from them into the development of OSCAR as appropriate. Remaining ontology development steps, including enumerating terms, defining the classes and class hierarchy, defining properties of classes, and defining facets of the slots, are discussed later in this paper.

Results

OSCAR uses description logics (DLs) to formally represent the knowledge contained in the ontology. In DLs, the three main components modeled are concepts, roles, and individuals. In our research, we used Protégé, which expresses ontologies in DL using the Web Ontology Language (OWL) (Antoniou and Harmelen, 2003; Noy et al., 2003; Horridge et al., n.d.). In OWL, DL roles are called *properties*, and DL concepts are called *classes*. OSCAR is made up of more than 400 classes and more than 1,000 axioms, including more than 500 logical axioms, which demonstrate relationships between classes.

The overall relational structure of OSCAR's class hierarchy is constructed using data from two main sources. The first source is the data gathered from interviews, as we already described. The second source is our existing knowledge, built from observational evidence we gained over years of experience in the field of SOC development. This *a posteriori* knowledge is used to shape and complete the ontology, not to form its basis or fact base. For example, if the interview data reveals that an expert cites a SOC Level I Analyst as a role needed for SOC operations, the team's prior knowledge about what a SOC Level I Analyst is and does will be important in classifying that concept.

The main purpose of OSCAR is to aid organizations, particularly government agencies, in developing SOC capabilities. The classes of knowledge within the ontology are therefore

organized around the main pillars of SOC development, which had traditionally focused on the PPTs required for operating an effective SOC capability (Torres, 2015). More recently, we have identified planning and organizational considerations as additional core pillars of SOC development (Novak et al., 2025). In the following sections, we describe the pillars and how OSCAR functions.

OSCAR Relationships and Defining the Pillars of SOC Development

We consider the core pillars of PPT, planning, and organizational knowledge classes to be the functional requirements of SOC development. Therefore, we initially treat each of these areas as its own distinct group of concepts. We organize these concepts into subclasses and give them names based on how they are identified in the data. Figure 2 shows the subclass hierarchy for the Technology functional area. It contains further subclasses for concepts such as Tools, Architecture, and Data Sources. These, in turn, have additional subclasses that describe various types of tools, architectures, or data sources. In this way, the entire functional area of SOC development is described, and as each sub-area is added, the entire SOC development knowledge domain is classified and represented in the ontology.

In the ontology, we further define a set of relationships among the concepts and classes of the knowledge it contains. In OWL, object properties are used to define or state these relationships. In OSCAR, we use object properties to describe how a SOC uses or applies things that have been defined as a class in the ontology. For example, a SOC must have PPTs to function. Therefore, we define object properties for each of these:

hasPeople
hasProcess
hasTechnology

We defined a total of 16 such object properties in OSCAR. Using these object properties, we can make a simple assertion, such as stating that a SOC must have a SOC Manager:

SOC = hasPeople.SOCManager

Additional object properties are used to describe other relationships, including training needed, skills required, and tools or processes used by various other entities. For example, we may state that a SOC analyst must have the skills needed to triage incidents:

SOCLevel1Analyst = hasSkill.IncidentTriageSkills

These assertions are used to describe the *SOC development* knowledge domain in OSCAR.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

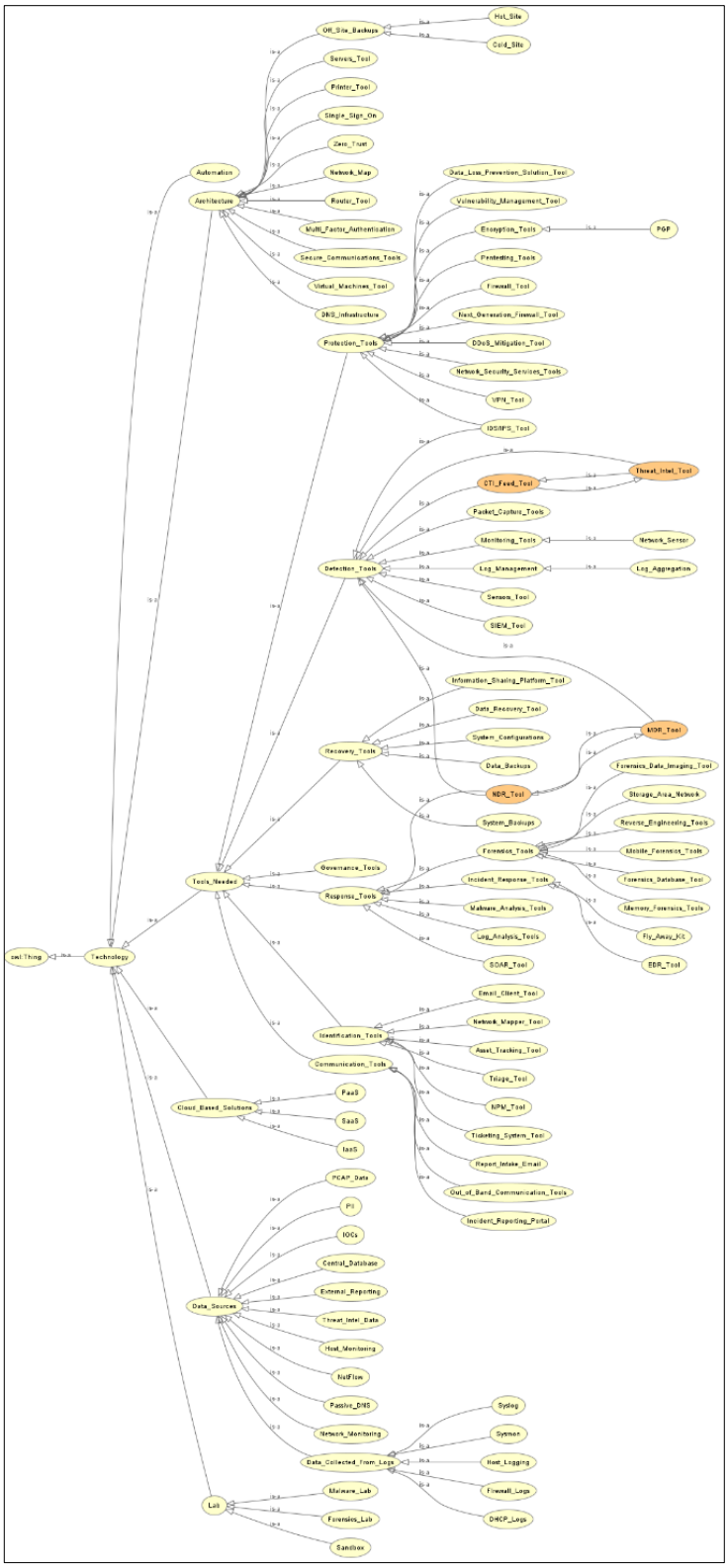

Figure 2. Technology Class Hierarchy. Screenshot from Protégé

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

Service Area Functional Levels

Describing only the SOC Development knowledge domain is a necessary but insufficient task for developing a full ontology that can serve as a useful tool in developing SOC capabilities. In addition to describing the domain, we must also address prescriptively what an effective SOC capability should look like. Put another way, we must define which concepts or classes must be present for the SOC to be effective.

To do so, first, we must consider which services a SOC must offer. There are many models for SOCs, as we described above, and it is widely accepted that not every SOC will offer the same set of services. So, we must identify and describe each service distinctly in OSCAR. To do this, we applied the *CSIRT Services Framework*, which was developed by FIRST (FIRST, 2019). This framework breaks incident response services into five high-level areas and a total of 23 distinct service areas. Within these areas, FIRST also provides a *Team Types Within the Context of Services Frameworks* guide (FIRST, 2024). The guide identifies an information security event management service as being required for SOC teams and an information security incident management service as being required for CSIRT teams. Because SOC and CSIRT teams are closely related, we included both services and all their service areas (a total of eight) in OSCAR. These are shown in Table 2.

Next, we must identify levels of capability for each service area. As described above, we adopted a five-level maturity hierarchy for SOCs when developing OSCAR. This hierarchy allowed us to map SOC capabilities in two dimensions—horizontally according to service and vertically according to capability or maturity level. Importantly, in our mapping, it is not necessary for the maturity levels to be uniform across service areas. A SOC with a level 5 capability in one service area may be a Level 2 in a different service area.

Table 2. OSCAR Service Areas

Service	Service Area
Information Security Event Management	Monitoring and Detection
	Event Analysis
Information Security Incident Management	Information Security Incident Report Acceptance
	Information Security Incident Analysis
	Artifact and Forensic Evidence Analysis
	Mitigation and Recovery
	Information Security Incident Coordination
	Crisis Management Support

Note. Adapted from CSIRT Services Framework Version 2.1 (n.d.)

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

Requirements for SOC Levels

With service areas and corresponding capability levels mapped, we examined which PPT requirements are necessary conditions to satisfy at each level. Different services require different combinations of PPTs, and different levels of capability add additional PPTs to the capability levels lower in the categorization. To identify the correct PPT requirements for each level and service area, we returned to the data collected from SOC experts as described previously. This dataset provides a basis for understanding what is required to operate different SOC functions. To translate this knowledge into PPT requirements in OSCAR, we first defined the concepts in the ontology. For example, we can say that a SOC Analyst is a class of People or that a SOC Level I Analyst is a subclass of SOC Analyst. We can similarly define a Network Monitoring Tool as a subclass of Tool or an Asset Management Policy as a subclass of Process. We can also define other restrictions, such as declaring that an individual cannot be both a SOC Level I Analyst and a SOC Level II Analyst. Expressed in DL, these concepts would be expressed as follows:

$$SOCAnalyst \sqsubseteq People$$
$$SOCLevel1Analyst \sqsubseteq SOCAnalyst$$
$$SOCLevel2Analyst \sqsubseteq SOCAnalyst$$
$$SOCLevel1Analyst \sqcap SOCLevel2Analyst \rightarrow \bot$$
$$AssetManagementPolicy \sqsubseteq Process$$
$$NetworkMonitoringTool \sqsubseteq Technology$$

Once these concepts were defined in OSCAR, we introduced axioms. Axioms in DL are statements that outline a constraint or set of constraints that must be satisfied for some assertion to be true. For OSCAR, the axioms define which PPTs must be present for a SOC to satisfy all requirements, meaning that it is capable of offering a particular service at a specified capability level.

For example, we can demonstrate in OSCAR the ability to define which PPTs are required for a SOC to perform a given service at a particular level of capability. In the case of a lower capability level, that may be a simple list, such as Level 2 of the Forensics Analysis Service Area (FASA). For this level of service, a SOC must simply have legal authority to operate, some basic open-source tools, and a SOC Level I Analyst. Expressed in DL, this is expressed as follows:

$$FASALevel2 \equiv hasPeople.SOCAnalyst \sqcap hasTechnology.\ OpenSourceTool \sqcap hasProcess.LegalAuthority$$

Using Protégé, we then write this class axiom in OWL:

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

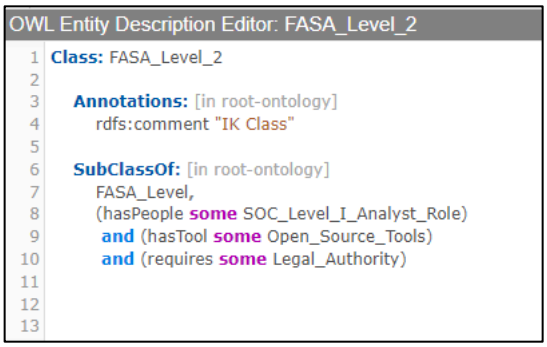

Figure 3. A Simple Service Level Axiom. OWL Image Capture from Web Protégé

In a more complex case, we say that for a SOC to perform the Incident Coordination Service Area (ICSA) at the highest level (level 5), that SOC would need a much larger range of PPTs. This range includes policies for network acceptable use, information sharing, and information classification. There is also a need for a security information and event management (SIEM) tool, an information sharing platform, and a vulnerability management tool. Finally, this level requires a SOC manager, access to legal counsel, and a SOC Level I Analyst with appropriate training in the incident coordination role. We can express this list of PPT requirements in DL as follows:

$$ICSALevel5 \equiv hasProcess.\ (NetworkAcceptableUsePolicy \sqcap InfoSharingPolicy \sqcap InfoClassificationPolicy) \sqcap hasPeople.\ ((SOCLevel1Analyst \sqcap hasRoleBasedTraining) \sqcap SOCManager \sqcap LegalCounsel) \sqcap \exists hasTechnology.\ ((InformationSharingPlatform \sqcap Automation) \sqcap SIEMTool \sqcap VulnerabilityManagmentTool)$$

Again, using Protégé, we then write this class axiom in OWL:

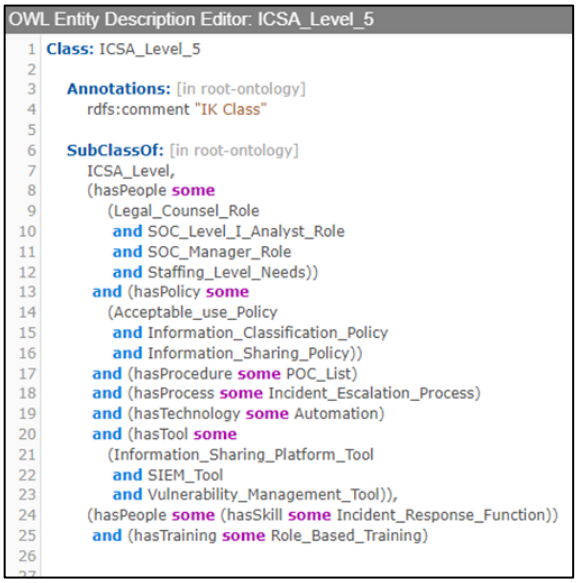

Figure 4. Axiom for Incident Coordination Service Area Level 5. OWL Image Capture from Web Protégé

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

We can also introduce additional restrictions or requirements for the SOC service areas and levels. For example, we can use cardinality restrictions to illustrate a required staffing level (e.g., more than a certain number of analysts required) or to denote a required level of experience (e.g., more than so many years of experience).

For example, we say that Event Analysis Service Area (EASA) level 5 requires a SOC Manager to have a minimum of five years of experience. In DL, this would be represented as follows:

$$EASALevel5 \equiv hasPeople.SOCManager \sqcap \geq 5\ hasExperience.Years$$

This yields the following OWL axiom:

(hasPeople some SOC_Manager_Role) and (Years min 5 xsd:integer)

We first defined Years as a data property and indicated in OSCAR that the data type is an integer. We can add additional cardinality restrictions and axioms in OSCAR as needed to identify other requirements for each service area and level. Some of these restrictions include upper limits on time *(Years max 5 xsd:integer)* and limits regarding the number of people in a given role *(hasPeople min 2 SOC_Level_I_Analyst_Role)*. In this latter case, the use of a data property and data type definition is not required.

By using the axioms described above, we begin to develop a complete view of what is required for each service area at each level of service. We can demonstrate dependencies within the PPTs for each service level. For example, Level 2 of FASA has only three requirements: a SOC Level I Analyst, Legal Authorities, and Open-Source Tools. However, identified relationships among these requirements suggest, for example, that open source tools will depend on an IT environment, that the SOC Level I Analyst can perform additional functions (e.g., incident triage), and that legal authorities are a business consideration for the organization. Figure 5 shows a visualization of these dependencies, again looking at level 2 of the FASA. From this simple example, which only has 3 requirements—a SOC Level I Analyst, Legal Authorities, and Open-Source Tools—we can see that relationships begin to form. These relationships show us the following:

- Open source tools will be dependent on an IT environment.
- The SOC Level I Analyst can also perform additional functions, such as incident triage.
- Legal authorities are a business consideration for the organization.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

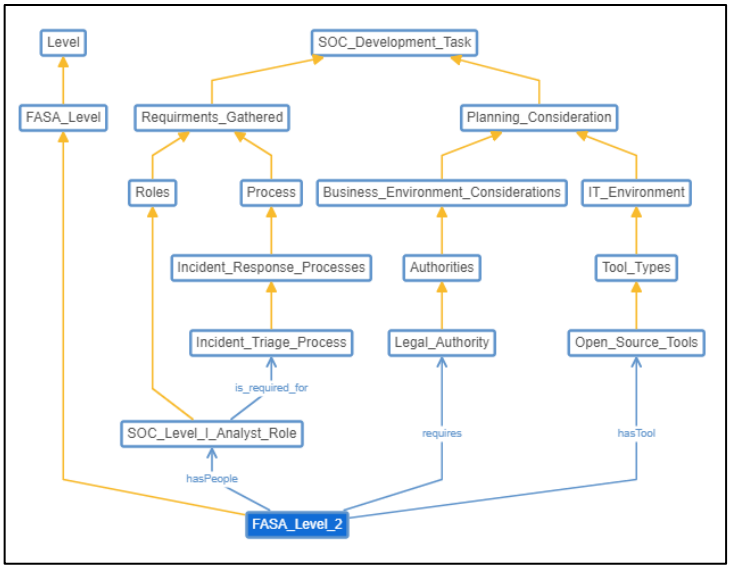

Figure 5. Hierarchy of Relationships. OWL Image Capture from Web Protégé

OSCAR collects and formalizes the knowledge required to build a SOC capability into a single structured knowledge base. We add to this knowledge base by categorizing SOC service areas and defining a level of capability for each one. By mapping the structured knowledge of OSCAR to these service areas and levels, we demonstrate the value of this approach by providing a tool that can be used to define current SOC capability levels and identify which PPTs are required to either add new services to a SOC function or perform those services at a higher capability level.

Discussion

Many choices exist for how to validate OSCAR. Each choice contains a number of tradeoffs, including cost, access to data, risk to human subjects, time, access to experts, and size of population. These choices affect the strength of the resulting claim. Since our ontology is driven by expert interview data, we chose to validate the resulting ontological artifact against the original dataset. This type of validation is a commonly used standard for validating ontological research artifacts (Gangemi and Presutti, 2009; McDaniel & Storey, 2020).

Our expert interviews and the resulting dataset comprise a collection of descriptions about how to plan for a SOC that helps protect an organization's networks, digital assets, and people, given a variety of common threat scenarios. OSCAR is a representation of the knowledge contained in our SOC expert interview dataset using description logic and axioms (Ellison et al., 2019). OSCAR also contains a few assumed concepts that help connect existing ideas the interview subjects expressed, even if we inferred those concepts when the interviewees did not name them explicitly.

During the ontological design and construction of OSCAR, we also made certain choices and used certain techniques. We also attempted to avoid the 24 common pitfalls of ontology construction as listed by Póveda-Villalón et al. (2010). These researchers grouped each of these pitfalls into three categorical types: Consistency, Completeness, and Conciseness. We also followed the community-accepted ontology design guidelines in Póveda-Villalón et al. (2012). Later in this section, we use

OSCAR to show its application during SOC assessment and planning while also validating OSCAR against the three categorical types.

Completeness

We assessed OSCAR for completeness against our dataset of interviews from SOC experts. We performed interviews by showing experts different threat scenarios and asking them how they would or could build SOCs to protect, detect, or respond to threats to the organization. The interviews were transcribed into documents, and the documents contained over 100,000 words in total. We analyzed the documents and developed a "coding" process. Each sentence was analyzed by our team and assigned one or more codes. In some cases, a thought or idea might be articulated in multiple sentences, all of which would receive the same code. In total, we developed over 736 codes, which were organized into a hierarchical structure. We then mapped most of the parent-level classes against the ontology. Table 3 shows how complete the coverage of the parent codes was against their types, subclasses, and property relationships in OSCAR.

Table 3. Summary of Mapping of Interview Data Codes to the Classes and Subclasses in OSCAR

Interview Data Code	OSCAR Class - SubClass
External Service	Planning - IT Environment
Malicious Activity	Planning - Threat Environment
Motivation (7 subcodes)	Planning Considerations
Organization (7 subcodes)	Planning - Business Environment
People (14 subcodes)	Requirements Gathered - People
Planning	Planning
Process (8 subcodes)	Process
SOC Function	SOC Function
Technology (3 subcodes)	Technology

Consistency

We compared the consistency of our conversion of interview knowledge into classes and axioms in OSCAR by using a reasoner. We built OSCAR using Protégé (Protégé, 2025), which provides multiple choices of reasoners. We used the data mapping in Table 3 along with the practical example of assessing a SOC security information event management (SIEM) solution's relevant capabilities to determine if OSCAR represents our target domain (i.e., SOC services in government organization settings) appropriately. This practical example demonstrates how OSCAR represents SOC services in government organization settings appropriately. The collection of observations from a notional assessment of a SIEM in our example SOC can be compared to a simulated SOC, which is subjected to the reasoner, to understand how consistent the observations are across instances. This example is discussed further later in this paper.

We plan to continue our consistency validation efforts by using OSCAR during future SOC assessments. We also plan to investigate whether OSCAR aligns with using SOC services in new contexts (Noy & McGuinness, 2001; HermiT, n.d.).

Conciseness

OSCAR is relatively "flat" because it contains a depth of only five subclasses. OSCAR is intended for planning, operations, and assessment settings. A deeper hierarchy would likely not be as usable or applicable in these settings. Research articles by Ferreira et al. (2023) and Pinotte et al. (2016) recommend using a "shallow" hierarchy.

Impact of OSCAR on SOC Creation Assistance

We built OSCAR to help assess and plan SOC development in government environments. Many SOCs are in different stages of planning and development or operational maturity. The following scenario is built from our team's collective experience in building and assessing SOCs. We demonstrate how OSCAR can code aspects of the PPTs that support the SOC and identify when key elements are missing by using the collection of observations from a notional assessment of a SIEM solution in our example SOC discussed above. We then demonstrate how to use these and other sample observations as data inputs for OSCAR.

In OSCAR, we first created a placeholder class for the sample SOC in our scenario and defined that class as equivalent to the union of other classes in each relevant SOC service area. We named this class SIM SOC (short for *Simulated SOC*) and used axioms to populate the placeholder class with basic class additions or individual additions. Figure 6 shows a visualization of the resulting list of classes and individuals that our fictional SOC contained once it was entered into OSCAR.

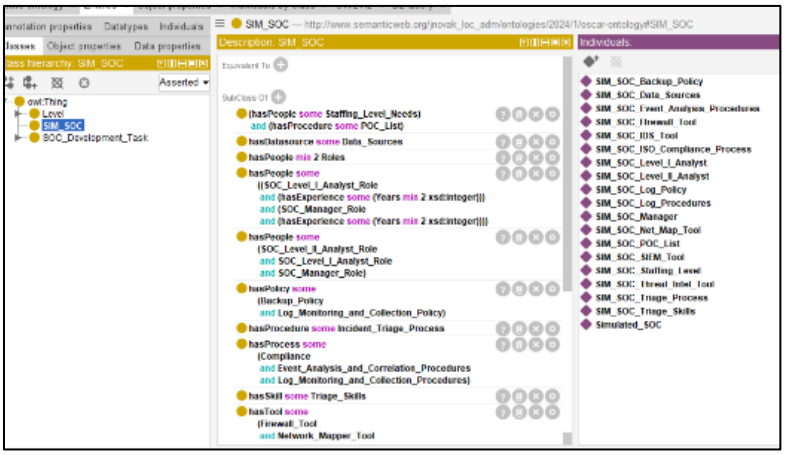

Figure 6. Visualization of Fictional SOC Assessment Data Entered into OSCAR. Captured from Protégé Model

After we loaded the assessment data into OSCAR, we could use a HermiT Reasoner to identify the levels of services provided by the SOC (HermiT, n.d.). This approach allows OSCAR to assess each SOC service area at a different level. Figure 7 shows our example SOC after we used OSCAR to categorize service levels using the HermiT Reasoning algorithm. Relationships and classes that are inferred by the HermiT Reasoner are highlighted (Glimm et al., 2014).

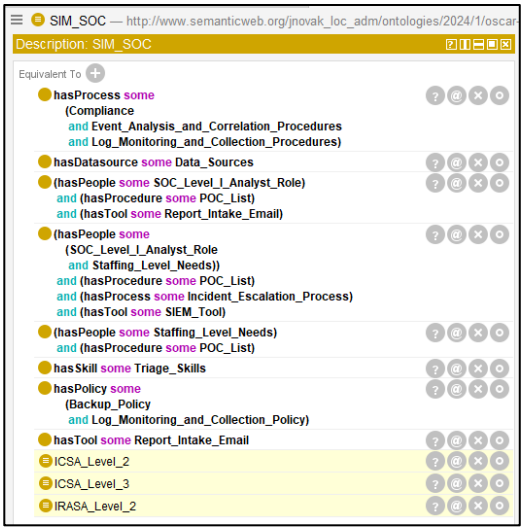

Figure 7. Automated SOC Service Level Classification Using OSCAR and Hermit Reasoner. Captured from Protégé Model

OSCAR used the input and perceived the inferred equivalencies of SIM SOC. Then it assessed the example SOC as meeting ICSA level 2, ICSA level 3, and incident report acceptance service area (IRASA) level 2 across all the chosen areas.

Limitations and Future Work

Our initial OSCAR research has focused on supporting a limited number of service areas for government on-premises SOCs, and we plan to expand beyond this construct. The remaining service areas within the CSIRT Services Framework are all relevant areas that are candidates for future work. Additional research could add other SOC operational sub-domains that we did not include initially. OSCAR could be refined to support hybrid or off-premises SOCs for other public and private sectors. Lastly, OSCAR could be used to obtain standard measurable impacts for planning and assessing SOC services in government settings.

Conclusion

SOCs remain a vital aspect of any organization's cybersecurity or information security plan. This is particularly true of organizations in the government space and other areas where outsourced solutions (e.g., SOC as a service, managed security service providers) are unavailable or available in only a limited capacity due to data restrictions or other sensitivity issues. In those cases, it is incumbent on the organization to ensure it can develop and operate its own SOC capability.

In this paper, we presented an ontology, OSCAR, that can serve as a tool to help organizations better identify the PPTs required to build a SOC capability. OSCAR provides a comprehensive mapping of the domain knowledge required for SOC development. It also contains SOC capabilities categorized into service areas and classifications of maturity levels for many service areas in the industry standard CSIRT development framework. Finally, OSCAR includes a mechanism for mapping the necessary PPTs to these service areas and maturity levels, which organizations can use to build their capability according to their needs.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

Acknowledgement

Carnegie Mellon University 2025. This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

References

Antoniou, G., & Harmelen, F. van. (2009). Web ontology language: OWL. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 91–110). Springer. https://doi.org/10.1007/978-3-540-92673-3_4

CSIRT Services Framework Version 2.1. (n.d.). FIRST — Forum of Incident Response and Security Teams. https://www.first.org/standards/frameworks/csirts/csirt_services_framework_v2.1

Ellison, D., Ikuesan, A. R., & Venter, H. (2019, July). Description logics and axiom formation for a digital forensics ontology. *Proceedings of the European Conference on Cyber Warfare and Security*. https://www.proquest.com/docview/2261008533

Ferreira, L., Silva, D. C., & Itzazelaia, M. U. (2023). Recommender systems in cybersecurity. *Knowledge and Information Systems*, *65*(12), 5523–5559. https://doi.org/10.1007/s10115-023-01906-6

Georgescu, T. M., & Smeureanu, I. (2017). Using ontologies in cybersecurity field. *Informatica Economica*, 21(3/2017), 5–15. https://doi.org/10.12948/issn14531305/21.3.2017.01

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., & Wang, Z. (2014). HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, *53*(3), 245–269. https://doi.org/10.1007/s10817-014-9305-1

HermiT Reasoner: Home. (n.d.). http://www.hermit-reasoner.com/

Horridge, M., Drummond, N., Goodwin, J., Rector, A., & Wang, H. H. (n.d.). The Manchester OWL Syntax. The University of Manchester.

Majid, M., & Ariffi, K. (2019). Success factors for cyber security operation center (SOC) establishment. *Proceedings of the 1st International Conference on Informatics, Engineering, Science and Technology,* Bandung, Indonesia. https://doi.org/10.4108/eai.18-7-2019.2287841

Mansfield-Devine, S. (2016). Creating security operations centres that work. *Network Security*, *2016*(5), 15–18. https://doi.org/10.1016/s1353-4858(16)30049-6

McDaniel, M., & Storey, V. C. (2020). Evaluating domain ontologies: Clarification, classification, and challenges. *ACM Computing Surveys*, *52*(4), 1–44. https://doi.org/10.1145/3329124

Murdoch, D. (2019). *Blue team handbook: SOC, SIEM, and threat hunting use cases notes from the field* (v1.02).

Novak, J., Hueca, A., Rodman, C., Perl, S., Breaux, T., & Valdengo, J. (2025). Building a better SOC: Towards the ontology for security operations center assistance and replication (OSCAR). *Digital Threats: Research and Practice*, *6*(1), 1-22. https://doi.org/10.1145/3722233

Novak, J., Manley, B., Mcintire, D., Mudd, S., Hueca, A., & Bills, T. (2021). *The sector CSIRT framework: Developing sector-based incident response capabilities.* Carnegie Mellon University. https://doi.org/10.1184/R1/13624148.V1

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). *Protégé-2000: An open-source ontology-development and knowledge-acquisition environment*. *Proceedings of the 2003 AMIA Annual Symposium,* 953.

Noy, N., & Mcguinness, D. (2001). Ontology development 101: A guide to creating your first ontology. *Knowledge Systems Laboratory*, (32). https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Ontology Design Patterns. (2009). In A. Gangemi & V. Presutti, *Handbook on ontologies* (pp. 221–243). Springer Berlin Heidelberg. http://link.springer.com/10.1007/978-3-540-92673-3_10

Onwubiko, C. (2018). CoCoa: An ontology for cybersecurity operations centre analysis process. *Proceedings of the 2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA)*, 1–8. https://doi.org/10.1109/CyberSA.2018.8551486

Onwubiko, C., & Ouazzane, K. (2019). Challenges towards building an effective cyber security operations centre. *International Journal on Cyber Situational Awareness*, *4*(1), 11–39. https://doi.org/10.22619/ijcsa.2019.100124

Pinotte, G. N., Cury, D., & Zouaq, A. (n.d.). *OntoMap: From concept maps to shallow OWL ontologies*.

Poveda-Villalón, M., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (n.d.). *A double classification of common pitfalls in ontologies*.

Poveda-Villalón, M., Suárez-Figueroa, M. C., & Gómez-Pérez, A. (2012). Validating ontologies with OOPS! In A. Ten Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, & N. Hernandez (Eds.), *Knowledge Engineering and Knowledge Management* (Vol. 7603, pp. 267–281). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33876-2_24

Protégé. (n.d.). https://protege.stanford.edu/

Stikvoort, D. (2015). SIM3: Security incident management maturity model. *Open CSIRT Foundation*.

Syed, Z., Padia, A., Finin, T., Mathews, L., & Joshi, A. (2016). UCO: A unified cybersecurity ontology. *Proceedings of the 13th AAAI Conference on Artificial Intelligence*.

Team Types Within the Context of Services Frameworks. (n.d.). FIRST — Forum of Incident Response and Security Teams. https://www.first.org/standards/frameworks/csirts/team-type

Torres, A. (2015). *Building a world class security operations center: A roadmap* (p. 12) [White Paper]. SANS Institute.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, *11*(2), 93–136. https://doi.org/10.1017/S0269888900007797

van Os, R. (n.d.). *SOC-CMM - Measuring capability maturity in security operations centers*. https://www.soc-cmm.com/

van Os, R. (2022). *Defining and operationalising a SOC target operating model using the SOC-CMM*. SOC-CMM. https://www.soc-cmm.com/downloads/SOCTOM%20whitepaper.pdf

Wang, Y., Zhao, B., Li, W., & Zhu, L. (2023). An ontology-centric approach for network security situation awareness. *Proceedings of the 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 777–787. https://doi.org/10.1109/COMPSAC57700.2023.00107

Appendix 1: Domain Definition Questions

A. *What is the domain that the ontology will cover?* The ontology will address cybersecurity and creating SOCs and related SOC-like capabilities, primarily in the government sector.
B. *For what are we going to use the ontology?* This ontology will address the challenges that government agencies and supporting international partners face when standing up a SOC. These challenges can range from negative organizational views on cybersecurity, to staffing, to funding, and even to political influences. The goal of the ontology is to identify the root cause of these challenges, determine the desired to-be state of the SOC, and offer potential solutions to reach the determined maturity level for a given point in time.
C. *For what type of questions, the information in the ontology should provide answers?* We developed over 250 questions from the categories discovered during the structured interview process. The question-development process considered the codes developed from the transcribed interviews and viewed the process from a regulatory requirement perspective. These categories became classes within the ontology. A full visualization of the classes used to develop the ontology questions is available in Figure 1.
D. *Who will use and maintain this ontology?* In 2001, Noy and McGuinness noted that an ontology may be developed "to share common understanding of the structure of information among people or software agents" and "to separate domain knowledge from the operational knowledge" (Noy & McGuinness, 2001). The objective of the OSCAR ontology is to create a tool that assessors can use to determine the current maturity of an organization's security posture in relation to its SOC. Accomplishing this objective requires doing the following:
    a. Extracting key areas identified in the structured interviews;
    b. Mapping those areas to existing standards, guidelines, and best practices;
    c. Identifying the stakeholders' as-is and to-be states through an assessment instrument;
    d. Determining the delta between the as-is and to-be states;
    e. Producing an output that recommends implementations and operationalizes the PPTs needed to attain the to-be state.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ

Authors Biographies

Justin M. Novak, Ph.D. is a Senior Security Operations Researcher at the CERT Division of the Software Engineering Institute, leading a team as part of the Security Operations Division supporting the US Department of State, Department of Defense, and United States Treasury. In this role, his main focus is on capacity building for incident responders - both at the individual and organizational level. At the SEI, he is also involved in research on the development and operation of CSIRTs, Sector CSIRTs, and Security Operations Centers, focusing on incident response and incident management. Justin holds a bachelor's degree in physics from the University of Pittsburgh and a PhD in Science and Technology Policy from George Mason University with a focus on the impacts of development of innovative open-source software. Justin also serves as an adjunct professor at George Mason University's College of Engineering and Computing.

Angel L. Hueca, Ph.D. is a Senior Cybersecurity Operations Researcher in the CERT® Coordination Center of Carnegie Mellon University's Software Engineering Institute (SEI). He has over 25 years of combined experience in Systems Administration and Cybersecurity. Angel has worked extensively in the private and public sector implementing intrusion detection systems (IDS) and systems auditing solutions. Currently, his focus in on international CSIRT initiatives. Angel holds a Ph.D. in Information Systems, focusing on information security and insider threat.

Samuel J. Perl is a senior member of the technical staff on the CSIRT development team within the CERT® Program at the Software Engineering Institute (SEI), a unit of Carnegie Mellon University in Pittsburgh, PA. He has been at CERT since 2011 and has performed research in a variety of areas including insider threat, vulnerability assessment, security incident and threat data analysis, threat modeling, information sharing, artificial intelligence, cognitive processes, and incident management team development. Mr. Perl holds a M.S. in Information Security Management from Carnegie Mellon University and a B.S in Information Systems from Carnegie Mellon University. He has also held appointments as an adjunct instructor in the Carnegie Mellon University's Information Systems (IS) program, Heinz College of Information Systems Policy and Management and in the West Virginia University Honors College.

Christopher I. Rodman is currently a Sr. Cybersecurity Operations Researcher within the CERT division. In this role he works with various US government agencies to build capacity and capability of international Computer Security and Incident Response Teams for US allies and other partner nations. Christopher obtained his Master of Science degree in Information Security and Assurance from Robert Morris University in 2016. He also teaches Digital Forensics as an adjunct instructor for CMU's Information Networking Institute, serves on the board for the Pittsburgh chapter of the ISC2.

DOI: https://doi.org/10.36965/OJAKM.2025.13(1)13-34

Accepting Editor: Meir Russ