UDC 004

Comparison analysis between strict ontologies and fuzzy ontologies

Oleh Kozlenko[1]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*

Annotation

Ontological modeling has been important in the field of cybersecurity, but with the growing use of artificial intelligence in various processes related to cybersecurity, it has become an increasingly relevant area for research every new year. Ontologies can serve as a primary source of knowledge for artificial intelligence models and as a "sequence of actions" in different processes. Typically, strict ontologies were used due to their formalized structure, but they did not fully capture processes that involve fuzzy contexts of actions or results. The aim of this article is to present and analyze different ontologies, both strict and fuzzy, that are used or could be used in the field of cybersecurity and related processes, demonstrating their similarities, differences, and areas of application.

Keywords: Cybersecurity, risk, fuzzy logic, ontology, fuzzy ontology

Introduction

An ontology consists of elements such as constraints, class instances, taxonomy of concepts, and relationships between elements [51]. Typically, constraints take the form of logical theories of concepts and relations. All elements of an ontology and their relationships are used in software and formal structures, such as databases [11]. Currently, ontologies are classified into three categories: domain-oriented, middle-level ontologies, and upper-level (general) ontologies [16]. Middle-level ontologies contain knowledge that arises and interacts during the operation of the system [18]. A general ontology describes high-level concepts. The most effective approach is one that includes all three ontology classes, hierarchically organized as follows: the top level — general ontology, the second level — domain-oriented ontologies, and the third level — task ontologies. This approach enables a comprehensive analysis of tasks that need to be implemented by the software [18]. Standard ontologies require verification and adjustment by specialists, even if their initial versions are automatically generated. This is important not only because of the limitations of automatic generation but also because ontologies depend on specific applications and often contain practical information that can be controlled by an expert. The primary goal of developing fuzzy ontology methods is to overcome these problems.

Ontology Definition

Ontologies are a formalized and structured method of organizing and representing knowledge, which simplifies communication and information exchange. The formal and well-structured nature of ontologies contributes to improving communication and the reuse of information between organizations [3]. Additional advantages mentioned in [1] include the ability to separate domain knowledge from operational knowledge. Through the introduction of relationships, ontologies allow knowledge exchange across different areas. Ontological languages provide a common representation of information and simplify the process of reusing it.

Ontologies can be classified into three main categories based on their level of abstraction [9]:

• Upper-level ontologies have the highest level of abstraction and are independent of specific domains. They provide general knowledge bases, based on which more specific ontologies for particular domains can be developed. These ontologies are also known as fundamental or universal [10].

• Middle-level ontologies are less abstract and typically include several domain ontologies. They provide more specific representations of

abstract concepts found in upper ontologies. While the distinction between upper and middle levels is not always clear, middle-level ontologies are often used to represent commonly used concepts, such as location [11].

• Domain ontologies focus on concepts specific to a particular domain and represent concepts and their relationships within the context of that domain. They can be created by importing middle-level ontologies or by extending concepts from upper-level or middle-level ontologies [13-15].

To create large ontologies, it is possible to integrate or accumulate existing ontologies that analyze specific aspects of software [16,17]. The main ontology can be extended to describe specific aspects of the software that are of interest to researchers. Explicit assumptions between relationships and concepts in the implementation domain allow for adapting knowledge about the software. Explicit knowledge specifications about the software are useful for new users who need to understand the concepts used [14]. One application of ontologies is to filter general knowledge of the subject area from operational knowledge in software [16,18]. This can include the configuration of products from their components according to specifications, as well as the development of component ontologies for configuring custom computer systems. Such an algorithm can be applied to configure objects if there is a system component ontology [12]. Knowledge analysis within the system is comprehensive and useful when there is a formal description of terms and their specifications, which influences the choice or modification of ontologies. The ontology should include concepts from other domains, such as people, time, space, and events. Ontological engineering also encompasses concepts such as temporal intervals and moments of time. If possible, it implements existing ontologies to describe these concepts. The ontology should also include temporal concepts to describe time-related instances and intervals, as well as concepts related to clocks and calendars. Well-known standard ontologies for temporal concepts are W3C's OWL-Time [30] and Time Ontology [31,32]. Both provide a vocabulary for describing time intervals and moments, including events. They also contain classes and relations for describing intervals and instances in temporal and calendar expressions. The ontology may also require geospatial concepts to describe physical locations of people or infrastructure. One source of geospatial ontologies is SOCoP [33]. If necessary, the GeoNames ontology [34] may be useful for inclusion in a cyberontology, as it provides textual references to toponyms such as countries and cities.

Analysis and examples of strict ontologies

The lack of a systematic approach has led to a variety of ontologies. For example, Fentz and Eckelhardt [5] propose an ontology based on four components: security and reliability taxonomy, basic risk analysis methodology, IT infrastructure concepts, and business concepts that facilitate modeling. This ontology analyzes different policy scenarios, defining risk as the "probability of a successful attack." However, despite the clear definition of risk, dependencies between attacks, system vulnerabilities, defenders' skills, and financial losses due to data breaches are not considered [6]. Overall, the methodology for ontology development in this context is a "staircase" approach that combines top-down and bottom-up analysis. Bottom-up analysis involves understanding the semantics of the basic data sources for integration, while top-down analysis involves understanding the needs of end users who will use the ontology and the semantically integrated dataset, i.e., the questions users may ask about system security [8].

More specifically, the methodology used for the current development of ontologies is based on the principles of reuse [8] and includes the following steps:

- Identifying existing ontologies in the relevant area, including foundational and mid-level ontologies.
- Developing a modern cyber-ontology, including classes and properties (and their definitions) already represented in the best ontologies.
- When the number of classes and properties from the existing ontology grows, this ontology needs to be implemented, and equivalence relationships between the classes of the ontology and the cyber-ontology must be established.

Available structured and defined resources are used as knowledge in the domain. These resources analyze the types of entities, relationships, properties, features, and ranges of values expressed in the resource. Where appropriate, and as related to other cyber-scheme

databases, the relevant entities, relationships, properties, and values are included in the ontology after clarification according to the principles of ontology engineering [7].

The developers, led by Gao [4], created an attack model based on ontology to assess the security of information systems from the perspective of a potential attacker. The aim of this assessment is to determine the risk of an attack, which allows comparing the system's performance before and after that event. The evaluation process consists of four stages. First, system vulnerabilities are identified using automated tools that analyze computer systems, programs, or networks for weaknesses and generate scan results. In the second stage, the developed ontology is used to identify possible attacks that could be triggered by the discovered vulnerabilities. The third stage involves querying the ontology to obtain potential effects of the attacks. Finally, in the fourth stage, the effect of the attack is determined.

The ontology [1] includes five main classes: attack vulnerability, attack vector, attack target, system vulnerability, and defense. The attack vulnerability encompasses security principles such as confidentiality, integrity, availability, authentication, authorization, and audit, which are key security characteristics and attack objectives. The attack vector describes the path through which the attack occurs. The "attack target" class contains potential objects of the attack, such as tools, software, and people. System vulnerability refers to defects in the system, such as design or implementation flaws. The "defense" class describes countermeasures against attacks. The classes in this ontology share similarities with those used in the AVOIDIT taxonomy. Gao [4], in turn, applied the relationships defined in [2] and complemented them with new ones. An attack has one or more vectors, depending on the type of vulnerability. An attack threatens the security properties defined as a result of the attack. The attack vector threatens the target that has vulnerabilities and may be part of another target. Defensive strategies are aimed at protecting targets and security properties. The relationships between attack vectors are realized through the "ifSuccessfulLeadsToThreat" relationship.

The detailed process of ontology development for the cybersecurity field is described in the work [18]. This research is based on the Diamond model, which defines malicious activity in a system [19]. Subsequent ontologies developed according to this framework include CRATELO, created [20] as a three-layer ontology to describe various threats to network security. Its structure includes ontologies for secure operations [15], which combine domain ontologies, a middle-level ontology extending security concepts [16], and a upper-level ontology [17]. A simplified version of the DOLCE ontology [21] demonstrates the detection of SQL injections [22]. Researchers in [23] has developed a hybrid model that combines a network packet ontology with an adaptive cognitive agent that learns based on instances and uses reinforcement learning to improve decision-making in defense against attacks. They [23] also discusses a comprehensive ontology for determining the behavior of malicious software. Not all attacks go through all seven steps during their lifecycle; some are autonomous and do not require command control. Confidence that an attack is occurring increases with the number of observed indicators [18]. There are many advantages to representing information about attacks in the context of cybersecurity through a chain. To limit the system, several assumptions about the attacker must be made. First, the attacker does not have complete internal knowledge of the system they are attacking, indicating some form of probing. Second, not all attacks are completely new; attackers often exploit published vulnerabilities to carry out various attacks such as denial-of-service (DoS), data pre-filtering, or unauthorized access. Finally, it is assumed that the system has sufficient traditional sensors to detect basic network behavior (NIDS) and HIDS. Attackers are classified into three categories based on their knowledge and level of expertise: script kiddies, advanced, and professionals/state actors. Script kiddies usually use known methods and tools, performing simple variations of known approaches [24]. Advanced attackers modify existing attacks or tools to avoid detection, but their behavior generally remains similar. State actors or experts discover new vulnerabilities and develop new attacks, such as zero-day attacks. Systematic attempts to classify malicious software include one ontology [22] and three XML description languages [24, 25]. It is also worth mentioning the attempt to classify features of malicious software [25].

Swimmer ontology [22] was created to facilitate data exchange between security products. Its hierarchy of malicious software classes is quite simple and organizes malware

into well-known categories such as Trojans, viruses, and worms. However, this structure may be insufficient for cases of malware exhibiting behavior that spans multiple classes or new behavior that does not match any known class. In Swimmer's taxonomy, the characteristics of malicious software are divided into three main classes:

- Payload: Software with malicious intent.
- Vector: The method of deployment or spread of malware.
- Evasion: Characteristics for detection avoidance.

Swimmer introduces the term "situationality" to describe a state in which malware is defined by its actions.

MAEC (Malware Attribute Enumeration and Characterization) is a language for describing all known types, variants, and manifestations of malware, focusing on attribute patterns such as behavior, artifacts, and attack patterns. This provides a more flexible method of characterizing malware compared to signatures based on metadata (e.g., file hashes). MAEC has a multi-layered architecture:

- Lower level: Describes actions of malware, such as access to equipment and changes to system state.
- Mid-level: Describes the behavior of malware, organizing and defining the purpose of lower-level actions.
- Higher level: Generalizes malware by mechanisms, which are organized groups of behaviors, helping to understand the composition of malware at a very high level.

Other resources, such as the Malware Exchange Metadata Format (ICSG) [23] and Zelcer's categories of common malware features [25], limit the scope of cyberspace. Early attempts to create a common language for describing incidents in computer and network security were presented by Howard and Longstaff [26]. Since then, several standard languages for describing security incidents have emerged.

OpenIOC is an XML format for exchanging cybersecurity incident information, organized as indicators of compromise (IOCs), which represent patterns of malicious activity. Developed by MANDIANT [24], OpenIOC provides an open standard that simplifies data processing in the defense industrial base (DIB). The format includes approximately 30 XML schemas to describe different object classes, such as MD5 hashes, registry keys, IP addresses, and more. OpenIOC objects have been integrated into MAEC and later became the foundation for CybOX objects. IODEF [27] is an XML specification developed by the IETF Extended Handling (INCH) working group [28] that serves as an information exchange format for Computer Security Incident Response Teams (CSIRTs). VERIS [30] from Verizon Business [29] is used for collecting security incident data, based on the A4 threat model, which views security incidents as a series of events affecting an organization's information assets. These events are described through four dimensions:

- Agent: Who's actions affected the asset.
- Action: What actions impacted the asset.
- Asset: Which assets were affected.
- Attribute: How the action affected the asset.

Events describe the occurrence of actions and changes in the real world, while situations represent stories of action occurrence. They are dynamic and difficult to model. Logical formalisms, such as event calculus [35] and situation calculus [36], have been created to represent events and situations. The concepts of events and situations are included in several described ontologies, such as DOLCE, GFO, and Cyc, which have event classes. GFO contains a History class that corresponds to the notion of a situation, while Cyc has a Situation class. The ProcessualEntity class in BFO has subclasses that correspond to events and situations. DOLCE's extension for describing events and situations [37], the upper-level ontology [38], and the ontology for linking open event descriptions (LODE) [39] also consider these aspects.

MITRE developed the Network Operations (NetOps) ontology in OWL format to support the interests of the NetOps Community Interest (COI). The NetOps ontology includes entities and events that reflect the missions and interests of the U.S. federal government's network management. Several other resources may be useful for concepts, abstractions, and relationships between entities that can be included in cybersecurity ontologies. Specifically, the Common Event Expression (CEE) [40] was created to standardize the way events in computer systems are described, logged, and exchanged. Some of these events may correspond to actions and behavior of malicious software. The most significant for ontology development are the Common Event

Expression Dictionary (CDET) and taxonomy. The dictionary defines the event fields and value types used in CEE to clarify properties associated with specific events. The taxonomy, in turn, classifies event types such as user logins, service restarts, network connections, privilege escalation, and account creation. No less important is the Cyber Observable Expression (CybOX) [27], which provides a specification for capturing, characterizing, and transferring observed events or state properties in cyberspace to support various use cases. Both MAEC and CEE use CybOX to describe cyber objects, actions, and events. Malware is also included in the TTP header, and STIX refers to other schemas and cyber information, such as MAEC, CybOX, CVE, and CPE. The Security Content Automation Protocol (SCAP) [41] is a set of specifications for standardizing the format and terminology used by security software products for reporting vulnerabilities and security configurations. In its current form [42], SCAP includes seven specifications:

- eXtensible Configuration Checklist Description Format (XCCDF) [43], for creating checklists and assessment result reports.
- Open Vulnerability and Assessment Language (OVAL) [44], for representing system configuration information and machine state assessments.
- Open Checklist Interactive Language (OCIL) [45], for formulating a set of questions and corresponding procedures to interpret responses.
- Common Platform Enumeration (CPE) [46], terminology for hardware, operating systems, and applications.
- Common Configuration Enumeration (CCE) [47], terminology for security software configurations.
- Common Vulnerabilities and Exposures (CVE) [48], terminology for software flaws related to security.
- Common Vulnerability Scoring System (CVSS) [49], specification for measuring the relative severity of software vulnerabilities.

For the development of ontologies in the cybersecurity field, the most significant are OVAL, CPE, CCE, and CVE. In [50], the author described semantic frameworks for these four standards based on modular ontologies. The Parmelee framework is aimed at simplifying data compatibility in automated security systems based on OVAL, CPE, CCE, and CVE standards

Fuzzy Ontologies and Their Features

Fuzzy ontologies are not part of the W3C (World Wide Web Consortium) standards, so new methods need to be developed for their restoration. As stated in [6], a fuzzy ontology can be reduced to an equivalent strict one and justified using existing mechanisms. However, there are currently no tools for storing, building, and utilizing fuzzy ontologies. Since ontologies must represent knowledge in a well-structured format, it is almost impossible to find a single formal structure that all application developers would agree on. The transformation of strict ontological structures into fuzzy ones is seen as a potential solution to this problem and has attracted the attention of several research groups. Wallace and Avritis [52] expanded the concept of knowledge representation based on ontologies by including fuzzy degrees of membership to a set of conceptual relations defined in the ontology, which are used to assess the context of a set of entities, user context, and the query for intelligent information retrieval. The defined set is commonly found in semantic relations, and their combinations are used to create fuzzy quasi-taxonomic relations. Gottroy [3] focuses on knowledge extraction from databases using fuzzy rules to refine ontologies, but his approach remains somewhat unclear and lacks formal semantics. There are also related studies in the area of fuzzy OWL ontologies [1] and fuzzy reasoning DL [4]. However, these approaches continue to use traditional logical representations of knowledge, which are not optimal for reasoning with learned ontologies. Artificial Intelligence (AI) methods, heuristic [5] or analogous [6] methods analyze alternative paradigms, but they have not been integrated into the real-world automatic knowledge acquisition mechanism. Several researchers provide descriptions of fuzzy logic, including fuzzy OWL extensions [7].

Kwan and his team [7] developed an automatic system for generating fuzzy ontologies – the fuzzy ontology generation algorithm (FOGA). They integrated fuzzy logic into formal concept analysis to handle uncertainty in information during conceptual clustering and the generation of concept hierarchies. However, the quality of clustering depends on manually

assigned meaningful labels for class names, attributes, and relations, which requires domain-specific knowledge.

Many researchers, including Chandrasekaran [9], have studied the definition and necessity of ontologies, which are the foundation of any knowledge representation system in a particular domain. Petr Musílek proposed a new approach to integrating fuzzy concepts and approximate reasoning into ontologies, allowing a better reflection of the advantages and perception of semantic web services from the users' perspective [53]. Silvia Kalegari, in her work, demonstrates how fuzzy logic can be incorporated into the process of ontology creation using the KAON (Karlsruhe Ontology) editor, which provides a foundation for the development of ontology-based applications [53]. Muhammad Abulaish presented a model for ontology expansion for imprecise concepts, where fuzzy relations are used to encode the degree of value of properties [10]. Huirali proposed a method for ontology creation based on fuzzy theory with two levels of uncertainty, where the combination of uncertain models and degrees of uncertainty is the key contribution of his work [25]. Aarti Singh and colleagues developed a fuzzy integrated ontology model capable of processing uncertain information provided by users [11]. Akinribido created a fuzzy ontology information retrieval system (FOIRS), which evaluates document relevance to user queries based on dominant words in each document. Jiang proposed a method for building ontologies based on FCA using a natural language processing (NLP) module [13, 27]. The main knowledge source for this ontology is a set of cardiology patient discharge reports and a Japanese language dictionary. The NLP module has three stages: a diagnostic term dictionary, a morphological analysis system, and a model for linking the morphological analyzer. The diagnostic dictionary extracts medical terms, the morphological analysis system (ChaSen) uses this dictionary, and the ChaSen connection model integrates with Protégé-2000 to build a formal ontology.

Haav [13] proposed a new method combining a rule-based language with FCA for semi-automatic domain ontology construction. In this approach, formal contexts for the domain are first extracted from input data using natural language processing. Using FCA and simplification procedures, an initial ontology is created in the form of a concept lattice, which is then represented as a set of rules and visualized. The designer can expand the ontology by adding new concepts and relations using the rule language. The advantages of this method include full attention to ontological considerations and non-taxonomic relations, as well as the ability to represent the ontology in first-order logic. However, the drawbacks include the complexity of transforming the initial ontology into first-order predicate logic and problems with conceptual expansion due to lexical gaps in domain representation.

Obitko [14] developed a new ontology design method using FCA, which has advantages such as creating a distributed ontology environment and providing visualization of the concept lattice. However, the main disadvantage of this method is that the extraction of formal contexts is done manually, making it unsuitable for the creation of larger domain ontologies. Peng [15] proposed an alternative FCA method for building component search ontologies based on semantics. In this gradual method, experts participate in creating a shared ontology by providing various input/output semantics, including business objects and their characteristics. Additional FCA is built based on the new objects. FCA-based ontologies can be created as new concepts, or concepts may be absent. If new concepts are created, the component performs a new business function, and concepts corresponding to the component and compatible predicate concepts can be created. Some of these concepts may be insignificant for the industry and, in such cases, can be deleted after expert evaluation. Concepts labeled "No Action" do not alter business logic and share the same input/output semantics as existing "Action" concepts. Eventually, the system contacts suppliers to confirm the component property specifications.

Teng [16] continued Peng’s work by developing a new method called the "Tourism Ontology Construction Method (TOCM)" using FCA. TOCM includes modules for preprocessing tourism information, FCA, and ontology construction. The first stage involves collecting tourism information from websites, processing it to remove irrelevant information, and structuring the data. In the second stage, saved terms are used to create formal contexts and a concept lattice. In the ontology construction module based on the concept lattice, the ontology is created, where class slot values can be entered manually or automatically. The advantages of this method include the integration of contextual

and linguistic knowledge using FCA, providing a rich set of information for knowledge engineers.

Diyu [15] introduced an enhanced formal algorithm for concept analysis to create domain ontologies. This method uses a threshold value (T), along with information gain (IG) and entropy (E). Categories, objects, and attributes from the field of computer science and engineering, collected from ACM, are introduced into the algorithm. After calculating entropy and information gain, attributes are added to the ontology if IG exceeds T. Otherwise, the attribute is added to a new category. The process continues for all attributes, and at the end of the algorithm, the ontology is formed from recovery attributes within category C.

Liu Ning [54] proposed a new method for creating marine ontologies using FCA, which includes four stages. The first stage involves calculating the initial ontology from a thesaurus with the participation of marine experts. In the second stage, a marine ontology is created based on FCA, where text processing with NLP allows identifying objects and attributes, and then a concept lattice and hierarchy are created. The third stage involves compiling new concepts that combine the initial and newly created ontologies. In the final stage, the ontology is formally described using Protege. This method has advantages such as focusing on objects and their characteristics, allowing for the generation of new objects and improving the automation of ontology construction.

Conclusion

As can be seen from the analysis of various ontologies in the field of cybersecurity and not only that, there is no clear answer to the question of the best choice of ontology for use in the process of formalizing software applications or in artificial intelligence. Clear ontologies of any level provide a strictly formalized representation of a specific application domain but may not cover the full range of behavioral or process variability, especially if the human factor is involved. Fuzzy ontologies, on the other hand, were specifically developed to address the aforementioned problem but have their own complexities in the construction process. Unfortunately, there is no single approach to building fuzzy ontologies (unlike strict ontologies, where there is). As shown in the second part, various researchers have attempted to develop algorithms for generating fuzzy ontologies (and automating this process), but the two most effective approaches remain – the generation of a fuzzy ontology based on a clear one and FOGA. Specifically for the field of cybersecurity, FOGA, in my opinion, is a more relevant algorithm due to the flexibility of the approach, although its downside is the complexity of the process and control. Generating a fuzzy ontology based on a clear one is a simpler option, but it may contain inaccuracies when transitioning from the clear to the fuzzy domain and require further consultation with experts.

References

[1] Gangemi, A. and Presutti, V. "Ontology design patterns," in Handbook on Ontologies.: Springer , 2009, pp. 221-244.

[2] Morris, T.I., Mayron, L.M., Smith, W.B., Knepper, M.M., Reg, I., Fox, K.L. "A perceptually-relevant model-based cyber threat prediction method for enterprise mission assurance," in IEEE Multi-disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, Miami Beach, 2011, pp. 60-65.

[3] Gonzalez, C., Ben-Asher, N., Oltramari, A., Lebiere, C. "Cognitive Models of Cyber Situation Awareness and Decision Making," in Cyber Defense and Situational Awareness, A., Wang, C., Erbacher, R. Kott, Ed.: Springer, 2014, vol. 62

[4] Cure O. Merging expressive spatial ontologies using Formal Concept analysis with uncertainty considerations. Methods for Handling Imperfect Spatial Information. Spinger-Verlag. 2010; 256:188–209.

[5] Chen RC, Bau CT, Yeh VJ. Merging domain Ontologies based on the WordNet system and Fuzzy Formal Concept Analysis techniques. Applied Sot Computing. 2011 Mar; 11(2):1908–23.

[6] Ganter B, Stumme G. Creation and merging of Ontology top-levels. International conference on Conceptual structures; 2003 Jul; 2746. p. 131–45.

[7] Quan, T. T., S. C. Hui, A.C.M. Fong, and T.H. Cao. 2006. “Automatic Fuzzy Ontology Generation for Semantic Web.”

IEEE Transactions on Knowledge and Data Engineering, 18(6): 842-856.

[8] Ramanauskaitė S. et al. Security ontology structure for formalization of security document knowledge //Electronics. – 2022. – Т. 11. – №. 7. – С. 1103.

[9] B.Chandrasekaran, J.R. Josephson, and V.R. Benjamin's, "Ontologies: What are they? Why do we need them?" IEEE Intelligent Sy stems and Their App lications, vol.14, pp. 20–26, 1999.

[10] Silvia Calegari and Davide Ciucci, "Integrating Fuzzy Logic in Ontologies," International Conference on Enterprise Information Sy stems,pp . 66-73, 2006

[11] AartiSingh, Dimple Juneja and A.K. Sharma, "A Fuzzy Integrated Ontology Model to Manage Uncertainty in Semantic Web: The FIOM ", International Journal on Comp uter Scien ce and En gineerin g, vol.3, no.3, p p .1057-1062, M ar 2011.

[12] Akinribido, Babajide S. Afolabi, Bernard I. Akhigbe and Ifiok J.Udo, "A Fuzzy-Ontology Based Information Retrieval Sy stem for Relevant Feedb ack" International Journal of Computer Science Issues, vol. 8, issue 1, pp: 382-389, January 2011.

[13] Haav, A semi-automatic method to Ontology design by using FCA. In: V. Snasel, R. Belohlavek (Eds.). Concept Lattices and their Applications. International work-shop on Concept Lattices and their Applications; 2004.p. 13–24.

[14] Obitko M, Snasel V, Smid J. Ontology Design with Formal Concept Analysis. Concept Lattices and theirApplications; 2004. p. 111

[15] Djouadi, Y., and H. Prade. 2010. "Interval-Valued Fuzzy Galois Connections: Algebraic Requirements and Concept Lattice Construction." Fundamenta Informaticae, 99(2): 169-186.

[16] Zadeh, L.A. 1965. "Fuzzy sets." Information and control, 8: 338-353.

[17] E. Turban, E. A. Jay. T. P. Liang(2005). Decision Support System and Intelligent System. Ed 7, Yogyakarta: Andi Offset. Penggunaan Metode Fuzzy Inference System Fis Mamdani Dalam Pemilihan Peminatan Mahasiswa Untuk_TugasAkhir 6 accessed Oct 24 2018.

[18] Schumacher, M. "Toward a Security Core Ontology," in Security Engineering with Patterns. Berling-Heidelberg: Springer-Verlag, 2003, pp. 87-96.

[19] Chirillo J. (2001), 'Hack attacks revealed: A complete reference with custom security hacking toolkit', Wiley Computer Publishing, New York. Defense Research Center, SRI International, position paper presented at the Proceedings of a Workshop with title 'Research on Mitigating the Insider Threat to Information Systems', at RAND, Held August 2000, Arlington VA.

[20] MAEC - Malware Attribute Enumeration and Characterization. [Online] http://maec.mitre.org/.

[21] Laboratory for Applied Ontology - DOLCE. [Online] http://www.loa-cnr.it/DOLCE.html.

[22] Swimmer, M. Towards An Ontology of Malware Classes. [Online] January 27, 2008. http://www.scribd.com/doc/24058261/Towards-an-Ontology-of-Malware-Classes.

[23] Ingle, J. Organizing Intelligence to Respond to Network Intrusions and Attacks. Briefing for the DoD Information Assurance Symposium. Nashville, TN, 2010.

[24] MANDIANT: Intelligent Information Security. [Online] http://www.mandiant.com.

[25] Zeltser, L. Categories of Common Malware Traits. Internet Storm Center Handler's Diairy. [Online] Sept. 25, 2009. http://isc.sans.edu/diary.html?storyid=7186.

[26] Howard, J. D. and Longstaff, T. A Common Language for Computer Security Incidents. [Technical Report]. Sandia National Laboratories, 1998.

[27] CybOX – Cyber Observable Expression. [Online] http://cybox.mitre.org/

[28] Internet Engineering Task Force. [Online] http://www.ietf.org/.

[29] VERIS Framework. [Online] https://verisframework.wiki.zoho.com/.

[30] Verizon Business. [Online] http://www.verizonbusiness.com/.

[31] Hobbs, J. R. and Pan, F. An Ontology of Time for the Semantic Web. CM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing. 2004. Vol. 3, 1, pp. 66-85.

[32] Pan, F. and Hobbs, J. R. Time in OWL-S. Proceedings of the AAAI Spring Symposium on Semantic Web Services. s.l. : Stanford University, 2004. pp. 29-36.

[33] Spatial Ontology Community of Practice (SOCoP). [Online] http://www.socop.org/.

[34] GeoNames Ontology - Geo Semantic Web. [Online] http://www.geonames.org/ontology/documentation.html.

[35] Kowalski, R. and Sergot, M. A Logic-based Calculus of Events. New Generation Computing . 1986. Vol. 4, pp. 67–95.

[36] Reiter, R. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. [ed.] Vladimir Lifshitz. Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy. San Diego, CA : Academic Press Professional, Inc., 1991. pp. 359-380.

[37] Gangemi, A. and Mika, P. Understanding the Semantic Web through Descriptions and Situations. Proceedings of CoopIS/DOA/ODBASE. 2003. pp. 689-706.

[38] Kaneiwa1, K. Iwazume, M. and Fukuda, K. An upper ontology for event classifications and relations. AI'07 Proceedings of the 20th Australian joint conference on Advances in artificial intelligence . 2007.

[39] LODE: Linking Open Descriptions of Events. [Online] http://escholarship.org/uc/item/4pd6b5mh.

[40] Common Event Expression: CEE, A Standard Log Language for Event Interoperability in Electronic Systems. [Online] http://cee.mitre.org/.

[41] The Security Content Automation Protocol (SCAP) – NIST. [Online] http://scap.nist.gov/.

[42] Quinn, Waltermire, Johnson, Scarfone, Banghart. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT). Gaithersburg, MD : NIST, 2011. SP800-126.

[43] XCCDF - The eXtensible Configuration Checklist Description Format - The Security Content Automation Protocol (SCAP) - NIST. [Online] http://scap.nist.gov/specifications/xccdf/.

[44] OVAL - Open Vulnerability and Assessment Language. [Online] http://oval.mitre.org/.

[45] OCIL - The Open Checklist Interactive Language - The Security Content Automation Protocol (SCAP) - NIST. [Online] http://scap.nist.gov/specifications/ocil/.

[46] CPE - Common Platform Enumeration. [Online] http://cpe.mitre.org/.

[47] Common Configuration Enumeration (CCE): Unique Identifiers for Common System Configuration Issues. [Online] http://cce.mitre.org/.

[48] CVE - Common Vulnerabilities and Exposures. [Online] http://cve.mitre.org/.

[49] Common Vulnerability Scoring System (CVSS-SIG). [Online] http://www.first.org/cvss/.

[50] Parmelee, M. Toward an Ontology Architecture for Cyber- Security Standards. George Mason University, Fairfax, VA : Semantic Technologies for Intelligence, Defense, and Security (STIDS) 2010.

[51] Niles, I., and Pease, A. Towards a Standard Upper Ontology. [ed.] Chris Welty and Barry Smith. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). 2001.

[52] Lee C.S. A fuzzy ontology and its application to news summarization / C.S. Lee, Z.W. Jian, L.K. Huang // IEEE Transactions on Systems, Man and Cybernetics (Part B). – 2005. - 35(5) –pp. 859- 880.
DOI: 10.1109/TSMCB.2005.845032

[53] Gómez-Pérez, A., Fernández, M. and de Vicente, A. Towards a Method to Conceptualize Domain Ontologies. ECAI '96Workshop on Ontological Engineering. Budapest, Hungary : s.n., 1996. pp. 41-52.

[54] Masolo, C., Guizzardi, G., Vieu, L., Bottazzi, E., Ferrario, R. "Relational Roles and Qua Individuals". In AAAI Fall Symposium on Roles, an Interdisciplinary Perspective, Virginia, USA. 2005