Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2024, № 4 (116) https://doi.org/10.33108/visnyk_tntu2024.04
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDC 004.8:004.42:004.9:006.32:351.74

DESIGNING THE ONTOLOGY FOR INTELLIGENT SECURITY SYSTEM OF RESIDENTIAL COMMUNITY

Yevhen Burov; Yurii Zhovnir; Oleh Zakharia

Lviv Polytechnic National University, Lviv, Ukraine

Abstract. *Intelligent Security systems represents a challenging domain for the implementation of Artificial Intelligence. They are inherently dynamic systems, aware of changes in their environment and able to react intelligently. The ontology provides the common vocabulary, the foundation for specification of objects included in a system and their interactions. We consider the ontology as software document, which is developed alongside the security system. In this sense it is a local ontology, reflecting the current version of the application. However, to be reusable, we build it based on GFO foundational ontology, allowing to model spatial, temporal and situational dynamics. The first version of ontology is built based on scenarios supported by the system. It is initially represented as a set of patterns in pattern language. The final version of ontology is represented in OWL and RDF formats.*

Key words: *Intelligent Security System, Artificial Intelligence (AI), Ontology, General Formal Ontology (GFO), Web Ontology Language, Resource Description Framework, Pattern Language, Situational Awareness, Spatial and Temporal Dynamics, Residential Community.*

https://doi.org/10.33108/visnyk_tntu2024.04.111 *Received 31.10.2024*

1. INTRODUCTION

The important trend in the development of information systems today is the introduction of artificial intelligence (AI) features in all areas of our life. The success of generative AI, using large language models provides the notable increase in human productivity giving us simplified access to information and knowledge, helping to create content and learn.

The domain of security systems represents a significant area for the deployment of artificial intelligence, primarily due to the need to rapidly evaluate circumstances and render instantaneous decisions across various contexts, which can often be highly unpredictable. In the realm of security, it is particularly challenging to predefine all conceivable usage scenarios. Consequently, it is rational to present the intelligent security system as a situation-aware entity, capable of monitoring its surroundings, forecasting alterations within them, and engaging in analysis and reasoning regarding those changes. The realization of an intelligent security system necessitates both reactive and proactive actions, which further complicates the development of a situation-aware system, encompassing the utilization of experiential and contextual knowledge, real-time actions, evaluation of outcomes, and the continual updating of the knowledge base.

In the foundation of knowledge base is the ontology, which provides the common vocabulary for the system. Using a common ontology in an intelligent security system offers several advantages, especially when integrating AI, access control, surveillance, and cybersecurity components. These benefits span across improved interoperability, system scalability, enhanced decision-making, and better knowledge representation.

Therefore, the task of creating and maintaining such an ontology is among the essential requirements for the intelligent security system. However, the realization of such a task brings several challenges, stemming from the contextual nature of concept meaning,

Corresponding author: Yevhen Burov; e-mail: vpasichnyk@gmail.com

the shifting of this meaning depending on the situations, the need to support the changes in ontology in learning process. In our work we consider an ontology as software artifact, which is developed alongside the development of security systems and supports its development and operation.

This article discusses the architectural solutions, development process and the first version of the ontology for intelligent security system being developed by authors.

The paper is structured as follows. In the Background research section, we discuss the current problems in ontology design and the works about ontologies in security domain. In the Methods and materials part we formulate the assumptions and guiding principles for the ontology design. We also provide the structure of intelligent security system and our vision of its functioning, including processing knowledge. The Results section is dedicated to the actual ontology development process. In the Discussion and Conclusions section we summarize the contributions of this article and discuss the intelligent features which could be addressed in the future versions of intelligent security system.

2. EXPERIMENTAL METHODS

The analysis of current trends in ontology design allows us to formulate the main requirements and assumptions we will use as guiding principles in the ontology design for security systems for the large residential community.

1. We are considering the ontology as a software artifact, a document supporting the development of security system. As such it should contain the minimal required number of concepts and relationships reflecting the functions of the system in the current state of its development. However, the ontology is updated, when the new version of system, having more functionality is created, resulting in new ontology version.

2. Created ontology is a local ontology in the sense that it reflects the objects and relationships implemented in specific system. On the other hand, to be reusable, we intend to build it on the ground of foundational ontology. Such an ontology will provide the basis for ontology expansion in future and the possibility for communication with external intelligent services, using the common or mapped object semantics.

3. This foundational ontology should be 4d ontology, allowing to model spatial and temporal phenomena, because security incidents happen and are developing in time and space.

4. The system's ontology will be built as a set of patterns, summarily representing the pattern language. Such form of representation allows to better implement the logic and systems functions and pass the knowledge to developers.

5. We consider the intelligent security system as a situation aware system, because of the need of constantly monitoring the changing environment and detect possible threats in it.

6. The system will use historical, experiential knowledge for situation detection and decision-making, enabling the prediction of situation-development.

Software development process for any complex product should be guided by long-term vision, specifying the features and the strategy for product development. This vision serves as a reference point allowing us to compare the current state of the product with ideal, vision state, finding the gaps and planning of closing them in the future stages of development.

This idea is supported by numerous research articles and practical experience in the domain of software development. For example, article [26] says that a well-defined long-term vision is crucial for sustaining software quality and adaptability over time, suggesting that strategic planning can mitigate risks associated with software obsolescence and technical debt.

The article [27] explores the dichotomy between short-term and long-term thinking in software development. It argues that prioritizing short-term gains can lead to accumulating technical debt, which ultimately hinders long-term productivity and innovation. The author

emphasizes the importance of aligning immediate actions with long-term goals to ensure sustainable development practices.

The ontology of intelligent systems is grounded in its functions and structure. Therefore, in the process of its development it is important to have a clear understanding of them. The analysis of literature [22–25] allows us to specify several functional domains in security system including Access Control, Surveillance and Monitoring, Cybersecurity, Emergency response management, Environmental monitoring, Maintenance and upgrade.

The first iteration of security system for residential community is focused on the implementation of Access control and Surveillance and Monitoring subsystems. The proposed structure of intelligent security system (fig. 1) has such parts:

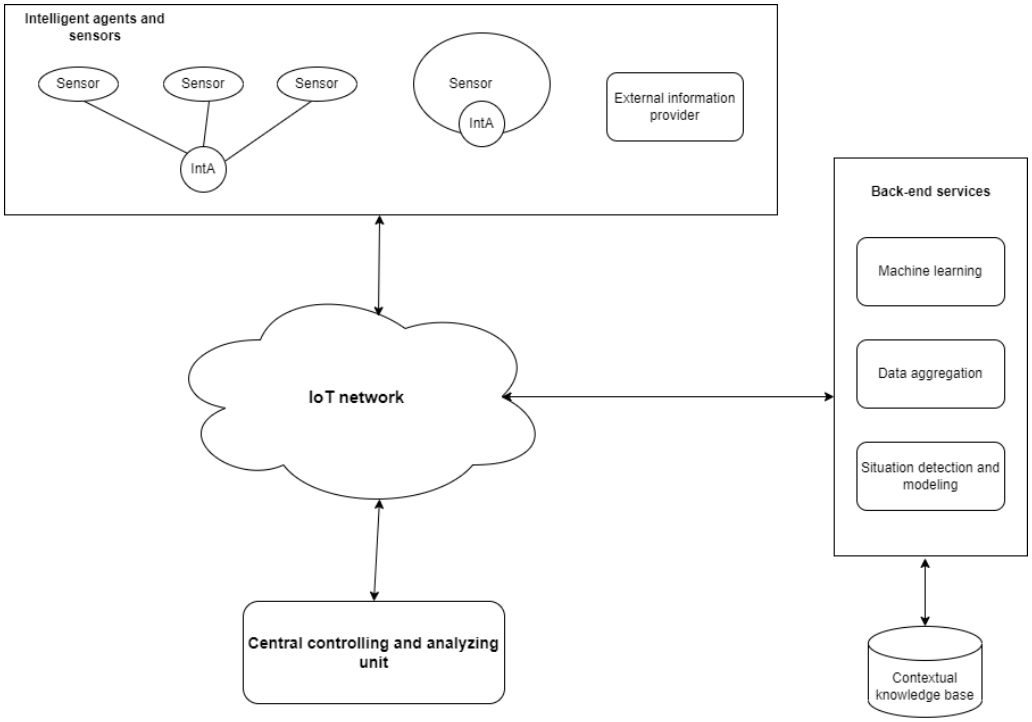


Figure 1. The structure of intelligent security system

- Intelligent agents – task-oriented, autonomous entities, frequently combined with intelligent sensors. Agents can be instantiated either locally on specific sensor apparatus or serve a groups of devices. Intelligent agents can also be functionally specialized. For instance, in executing object recognition or internet queries for information.
- The back-end is characterized by software services which are frequently implemented as cloud services. They undertake resource-intensive computations and are constructed in accordance with Service-Oriented Architecture (SOA) requirements.
- The front-end or central controlling unit synthesizes and monitors the comprehensive overview of the security landscape, utilizing the information provided by sensors and services along with insights from the knowledge base and engaging with security personnel.

For the objectives of this investigation, the General Formal Ontology (GFO) has been chosen [28]. GFO functions as a fundamental 4-dimensional ontology that provides a consistent structure for the conceptualization of forms, modalities, and entities across various tiers of abstraction and granularity. It integrates methodologies sourced from mathematical logic, philosophical examination, artificial intelligence, and linguistic scrutiny.

GFO is using the concepts of topoids, chronoids, configuroids to model spatial, temporal and structural aspects of reality. Situoids and situations are used to represent contexts.

We model the intelligent security system as situation-aware system, able to detect, recognize the important situations, make decisions in them and take appropriate actions. Thus, situoids Su and its time slices – situations Sit – become the central elements of modeling in proposed system.

Situoids exhibit the following characteristics:

- Contextual Complex. Situoids are used to encapsulate the entirety of a context or situation, encompassing all pertinent entities along with their interrelations.
- Temporal and Spatial Boundaries. A situoid is demarcated both temporally and spatially, indicating that it exists within a defined time frame and a specific spatial domain.
- Dynamic Characteristics. Situoids possess the capacity to undergo transformations over time as the entities and their interrelationships develop within the context.

A situoid Su can be specified through its goal Gl and regarded as a transition between two bounding situations $(Sit_{st}^{su}, Sit_{end}^{su})$, namely the starting state and the intended target state. Multiple intermediary situations are implied to exist between those bounding states. Each situoid is associated with a topoid, a chronoid, and a configuroid, signifying that it happens within both spatial and temporal dimensions, while the world experiences structural modifications during its duration. Situations are situoid states in specific time moments. We can consider them as slices of situoids in the time points. Each situation, like a Situoid can be considered as a whole, but also analyzed structurally.

The intelligent security system operation is organized around processing the interrelated conceptual knowledge models. There are several types of models. Environment models Cm_{env} are built based on the objects recognized in the environment and store the parameters of those objects supplied by sensors or external data sources. Contextual/task models Cm_{con} are keeping data relevant to specific task, situation, goal. They are formed as a subset $Cm_{env,tc} \supseteq Cm'_{env,tc}$ of the Environment model for current time t_c. with additional objects relevant to the intention $Gl_{int,tc}$, provided by corresponding pattern from knowledge base.

$$Cm_{con,tc} = (Cm'_{env,tc}, Gl_{int,tc}, t_c) \quad (1)$$

Intelligent agents monitor object parameters based on the information from a sensor of a group of sensors interpreted as parameters of objects from the ontology On. They maintain the local Environment model and share it with the Central Unit. The operation and decision-making of agents is dependent on the set of contextual models, describing the actions to take in various contexts. Those models are supplied and periodically updated by the system's back-end services which are running learning and pattern-matching algorithms.

Back-end service collects information from intelligent agents and creates and maintains its own global Environment model $Cm_{env,gb}$. This model is used to detect and anticipate situations, coordinate the actions of agents, make system-wide decisions, and present information to human personnel working with the Front-end unit.

The system uses experiential knowledge for detecting situations, planning and anticipating the situation's development. This knowledge is stored contextually, that is a key for retrieval is context similarity. When a system looks for information in a knowledge base it looks for the knowledge about the similar task in similar environment configuration. Or, in case of detecting situations it looks of possible situations/threats which can happen in contexts, like current. When similarity is established, the system makes a mapping between situation and knowledge pattern. In this way access to knowledge represented by patterns is provided.

Therefore, a function F_{sim}, measuring the distance between the current context (1) and the key-context $(\mathrm{Cm}_{\mathrm{env}}^{\mathrm{kb}}, \mathrm{Gl}_{\mathrm{int}}^{\mathrm{kb}})$in knowledge base should be implemented. In the process of searching the value of this function should be minimized:

$$\mathrm{F_{sim}}: \left(\left(\mathrm{Cm'_{env,tc}}, \mathrm{Gl_{int,tc}} \right), \left(\mathrm{Cm^{kb}_{env}}, \mathrm{Gl^{kb}_{int}} \right) \right) \rightarrow \min \quad (2)$$

Back-end services access, maintain and update contextual knowledge base. They develop, maintain and update policies and models used by intelligent agents and sensors.

Situation detection is done by a service monitoring the current Environment model $Cm_{env,tc}$ on cues and patterns related to situations, which could happen in current context $Cm_{con,tc}$. For this it monitors the set of cues. Each cue is a condition (pattern) with weight, reflecting its importance.

$$Cue = (Cm_{cue}, w_{cue}) \quad (3)$$

Cues are ordered according to their weight. Cues leading to the situations with greater impact have more weight. The impact is deduced from the knowledge base. If an important cue is detected, Central unit updates the data collection policy, allowing it to confirm/decline the presence of a situation.

The intelligent security system uses situoids from GFO to model the dynamics of situations development. This development is presented as a time-ordered sequence of situations $(Sit_{t1}, Sit_{t2}, .., Sit_{tk})$ with the corresponding sequence of configurations $(Cf_{t1}, Cf_{t2}, \dots, Cf_{tk})$. Each configuration in sequence is represented as knowledge graph:

$$Cf_{ti} = (SV_{con}, SE_{rel}, t_i) \quad (4)$$

Where SV_{con} is a set of nodes, corresponding to objects and SE_{rel} is a set of relationships used in situation specification. Both objects and relationships are classified according to the system's ontology. Specific configurations in the sequence of configurations can be different, which reflects the situation configurations dynamics in the process of task execution. The transitions between situations are modeled using experiential knowledge as structures of actions or events which cause the transition.

3. RESULTS AND DISCUSSION

In this part of the article, we describe and justify the process of ontology creation. In the first stage of building the ontology it is important to select the corpus of knowledge from which the ontology is derived.

For the first iteration of intelligent security system, we decided to represent such corpus as a set of security scenarios which should be supported by system. Each scenario is a situoid, having an initial situation with specified triggering conditions, the sequence of intermediary situations, describing the actions and possible situation trajectories and the final situation, completing the situoid.

Specific subsystems and objects cooperate across scenarios. Thus, access control and video surveillance systems cooperate by sharing event triggers. For instance, an unauthorized entry attempt in the access control system will immediately activate cameras to record. Guards and security personnel continuously monitor both access control logs and video footage. In critical events, they can override automatic systems. Cameras are often triggered by access points when a person swipes their card or uses biometric identification, ensuring that both access and corresponding video are logged for cross-verification. Video analytics provide an additional layer of security by detecting anomalies (e.g., tailgating, suspicious activity) and linking with the access control system to alert guards or deny access.

The initial implementation of security system is limited to Access control and Video Surveillance subsystems, which includes the following object types:

1. Access Control Subsystem: Access Points, Access Cards/Badges, Biometric Devices, Control Panel, Residents/Guests, Guards/Staff, Emergency Response. Gates, doors, or barriers equipped with control mechanisms. Physical tokens (e.g., RFID cards) used by residents and staff. Fingerprint or face recognition devices used to identify and authenticate individuals. Central unit for managing access control, storing credentials, and logs. Individuals who are authorized (or seek authorization) to enter the community. Security personnel responsible for monitoring and decision-making. Mechanisms for handling alarms or security breaches.

2. Video Surveillance Subsystem: Cameras, DVR/NVR, Motion Sensors, Video Management Software (VMS), Monitors, Analytics Engine. Positioned at strategic locations to monitor and record video. Digital/Network Video Recorder for storing video footage. Devices that trigger recording or alerts based on detected movement. Software that displays, processes, and analyses video feeds. Screens for security staff to observe live video feeds. AI-based system for object detection, face recognition, and anomaly detection in video footage.

In the second step of ontology design we derive classes and relationships from scenarios. Since we decided to base the security ontology on General Formal Ontology, we'll follow GFO's foundational structure and align the security system objects and relationships accordingly. GFO includes categories like Objects, Processes, Roles, Relations and Time each of which can be used to model components in the security system.

Each category in GFO is used to group objects of common nature: objects are concrete entities (physical and digital) such as cameras, access points, access cards, Events and activities, such as authentication, surveillance, and alert generation; Relations specify how objects are connected, like «monitors», «grants access», or «records»; Roles show the role an object or person plays within the system, e.g., resident, guard, visitor; Time is used to reflect temporal aspects, like «Time of Entry», «Video Recording Time».

We structure the objects, derived from scenarios according to GFO:

1. Physical Objects (subclass of GFO's `Material Object`): Camera (object with the role of `Monitoring Device`); Access Point (e.g., doors, gates, classified under `Physical Access Control`); Biometric Device (object functioning as `Identification Device`); Access Card (object as `Identification Token`).

2. Digital Objects (subclass of GFO's `Information Object`): Control Panel (central system responsible for managing access), Video Management Software (VMS) (system responsible for video data analysis and management); AI Identity Verification System (digital object responsible for identity recognition and anomaly detection).

3. Agents (subclass of GFO's `Agent`): Resident (plays the role of an authorized user).

- Visitor/Guest (plays the role of an external party); Security Guard (plays the role of security personnel responsible for oversight); AI Monitoring Agent (an intelligent agent tasked with observing and analysing data in real-time).

Main processes in the system are:

1. Access Authentication (subclass of GFO's `Process`): Biometric Authentication: A process where the resident's identity is verified through a biometric device; Card Swipe Process: The sequence of a resident swiping an access card and getting verified.

2. Video Surveillance Process (subclass of GFO's `Process`): Video Capture, Motion Detection and Recording, AI Anomaly Detection. Cameras record activity based on a schedule or trigger. Initiated when a motion sensor detects movement. AI processes real-time video data to identify unusual behavio

3. Alert Generation (subclass of GFO's `Process`): Unauthorized Access Attempt, Suspicious Behavior Detection. The event of a failed access attempt triggers an alert. AI detects suspicious activities and generates real-time alerts.

Those processes can be also considered as situoids. For example, on fig.2 is a description of situoid ‘Unauthorized access attempt’.

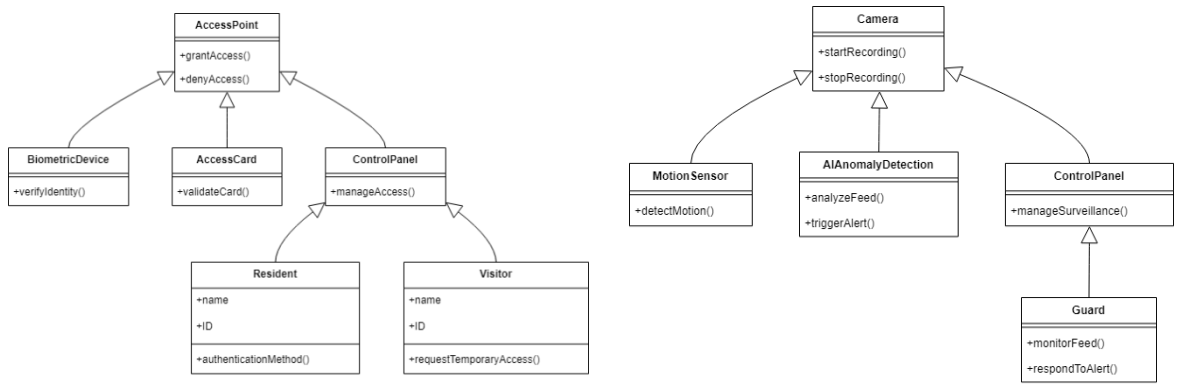
Roles are used as to abstract objects into different categories and use them in behavior patterns. Roles represent contextual Functions of Entities.

1. Security Role: Guard Role, AI Monitoring Role. Monitors and takes action on alerts. AI plays a role in continuously analyzing data for threats.
2. Access Role: Resident Role, Visitor Role, Maintenance Staff Role Plays the role of an authorized person allowed entry at specific times and locations. Plays the role of a temporary individual who must be authorized by a resident or guard. Has restricted access during certain times.

By mapping the scenario-derived concepts to GFO, security ontology would be formally grounded and capable of interoperating with other systems or domains that use the GFO framework.

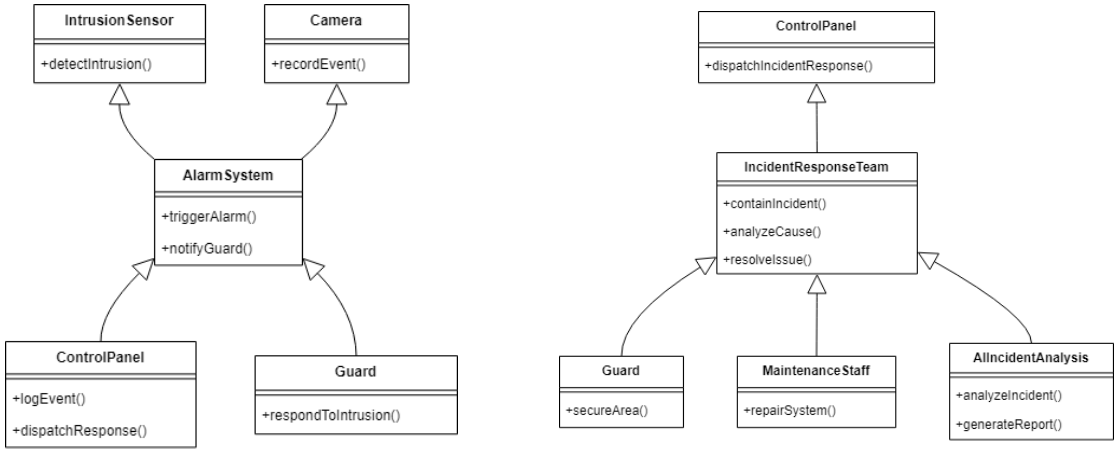
The next step in the ontology development is to group objects and relations into meaningful patterns, which form the pattern language of the system. Those patterns represent the basic repeatable interactions between objects and can be used like competency questions to test the content of ontology. Moreover, those patterns are used by developers to single out meaningful interactions in the system.

In the process of ontology development multiple patterns were found. The fig. 2–4 presents examples of such patterns.



a) b)

Figure 2. a – Access control authentication pattern, b – Surveillance monitoring pattern



a) b)

Figure 3. a – Intrusion Detection and Response Pattern, b – Incident Response Pattern

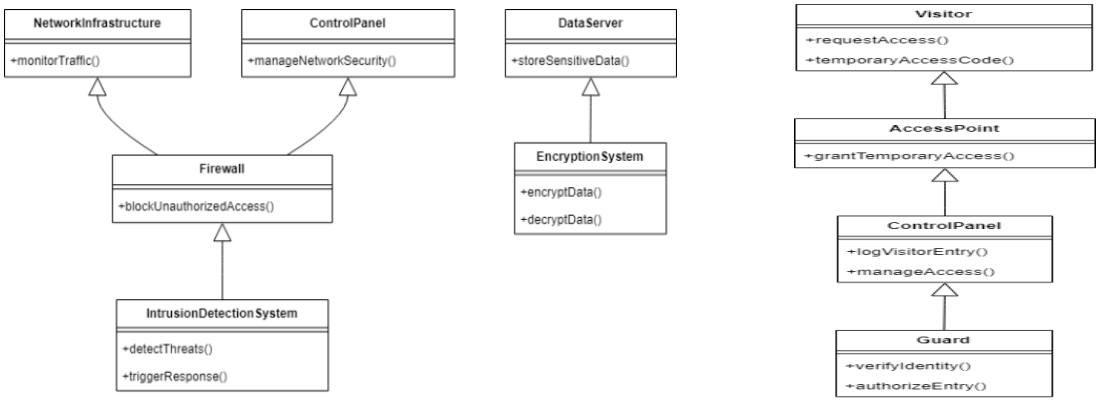


Figure 4. a – Cybersecurity Pattern, b – Visitor Management Pattern

The first version of security system ontology was designed in OWL and RDF formats (fig. 5). It is used as data scheme for developing an intelligent security monitoring system for residential communities.

Ontology is defined as specification of shared conceptualization [1]. This definition puts emphasis on the task of creating the common conceptualization, which is aimed to provide for the storage, reuse, understanding of knowledge and communication between intelligent agents.

This definition is subjected to critics in [2], stating that the notions of concept, conceptualization are phenomenological, they are understood intuitively, but lack clear and unambiguous definitions. Addressing this critic, Gruber in [3] states that ontology could be considered as a tool and product of engineering and thereby defined by its use. The authors of [2] propose to informally define an ontology as a set of relevant domain objects combined with the annotations which document them and corresponding logical theory for the vocabulary objects.

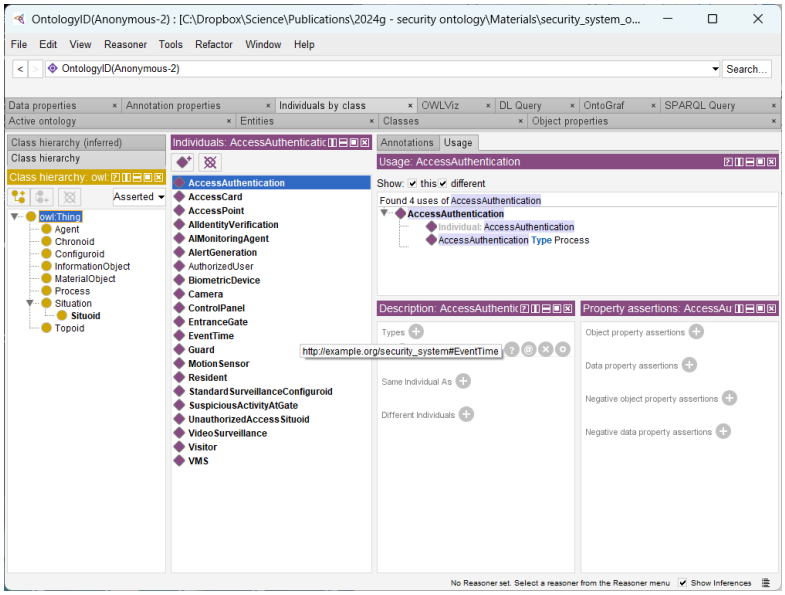


Figure 5. The developed security system ontology in Protégé

Another challenge associated with the development of ontologies is the fluency and contextual variability inherent in the meanings of concepts. For instance, within the framework

of Formal Concept Analysis, a concept is characterized as an unchanging node within the conceptual taxonomy, possessing a singular definition based on its associated attributes. The definition of a concept, as articulated in the literature [4], is formulated from an initial collection of individual objects by categorizing them according to shared attribute sets. This methodology provides a mechanism for constructing ontologies from a specified array of objects endowed with properties. While such an approach confers numerous advantages, it also presents limitations for intelligent agents operating in real-world contexts.

Intelligent agents autonomously generate and amend their own conceptual frameworks, which mirror their experiential learning and areas of expertise. This process is integral to the continuous learning of intelligent agents. The ontologies produced by these agents are inherently subjective and must be harmonized with commonly accepted ontologies when there is a necessity for knowledge exchange.

Furthermore, the interpretation of a concept is subject to evolution over time. An individual (or intelligent agent) develops an understanding of a concept through the observation of real-world objects and the application of conceptual labels to these entities. The objects that embody concepts serve as prototypes or reference points in the definition of a concept's meaning. Subsequently, objects that exhibit similarity are aggregated, thereby enriching the concept's meaning and reflecting increasingly nuanced variations and use cases. The intrinsic characteristic of concept fluency is emphasized in numerous scholarly articles [5]. The evolution of concept meaning is an ongoing process, as concepts continually acquire new interpretations, occasionally through associations with other concepts.

The versioning of ontology, described in [2] representing it as a network of ontologies with mappings between its components is a way to resolve the problem of ontology changes in the process of learning. The ontologies used by intelligent agents or systems are local, formulated, and utilized by individual agents. Given that they encapsulate interpretations unique to each agent, such local conceptualizations are frequently referred to as contextual ontologies [6]. Nonetheless, these conceptualizations address contextual variances in ontologies across agents, without accounting for context-dependent interpretations of concepts within the local ontology. Contextual ontologies [6] differentiate between local and shared conceptualizations. Local conceptualizations are preserved within the memory of a specific intelligent agent and are aligned with shared conceptualizations when the necessity for interaction with other agents emerges. In [6], contexts are delineated as local conceptualizations. Benslimane [7] subsequently defines the concepts of mono-context ontology and multi-context ontology, in which multi-context ontology encompasses concepts with multiple interpretations.

The article [8] proposes the method of representing contextually dependent shades of meaning for a concept using the prototype theory and selecting the relevant meaning dynamically, dependent on the situation. To effectively manage multiple contexts and engage in reasoning concerning them using Description Logic, an enhanced version of OWL-namely, OWL-Contology representation language has been formulated [9]. This linguistic framework is grounded in OWL syntax and provides bridging rules that facilitate the correlation of concepts, individuals, and roles at both the syntactic and semantic dimensions. Another challenge comes from the substantial size and intricate complexity of ontologies. To be practically applicable, an ontology must encompass tens or even hundreds of thousands of concepts (ontology width). For instance, the Cyc ontology comprises hundreds of thousands of concepts alongside over one million rules [10], while WordNet contains more than 117 thousand synsets (synonym groups) [11]. The quantity of relations (the depth of the ontology) escalates non-linearly in relation to the increase in the number of concepts. Consequently, reasoning, which is predominantly reliant on the processing of relations, becomes computationally unfeasible. Conversely, to effectively address specific tasks and facilitate

decision-making in particular contexts, it is often sufficient to utilize only a limited subset of concepts and relations.

The initial proposition for mitigating the structural complexity of ontology involved disaggregating the overarching ontology into segments based on the generality or granularity of the conceptual levels used. The reusable component was designated the upper ontology, while the specialized component was identified as the domain ontology. This bifurcation engendered an additional challenge associated with the integration of both upper and domain ontologies [12]. Furthermore, domain ontologies continued to exhibit excessive structural complexity and redundancy when addressing practical issues or when applied within specific business contexts.

A potential resolution for the issue of subjectivity in the construction of ontologies lies in the utilization of thoroughly examined collections of foundational ontology components. The endeavor to elucidate and formalize the most fundamental concepts and relationships, which can be consistently leveraged across a variety of ontologies, culminated in the development of foundational ontologies such as SUMO, UFO, and GFO [13,14], alongside libraries of objects and patterns, as is shown in [15]. The variety and fluency of domain ontologies called for the research of meaningful patterns in the ontology design. This work started with the exploration of conceptualization patterns in information systems design [16]. The work [17] proposes use of ontology design patterns (ODPs) as building blocks for ontology design. It presents a formal framework for collaborative ontology design that justifies the use of ODPs with explicit rationales. The main idea to battle the inherent complexity of domain is to specify the small, task and context-oriented ontologies that can be used as building blocks in ontology design. The authors focus mainly on CPs (Content Patterns) to provide readers with concrete examples and a closer view on their exploitation on the Semantic Web. The idea of ontology design patterns was developed in [18] with the introduction of the concept of ontology pattern language to organize related ontology patterns in ontology engineering. The article follows the similarity between the design patterns in the development of software and ontology. It proposes to formulate the domain ontologies as design patterns. The work [19] provides an example of building the testing ontology using patterns from Software Process Ontology Pattern Language (SP-OPL). In article [20] develops a formal definition of OntoUML, a conceptual modeling language, using a graph grammar and ontological patterns. The definition is independent of the UML meta-model and incorporates micro-theories from the Unified Foundational Ontology (UFO). In [21] three extensions to Ontology Pattern Language (OPLa) are detailed into a reorganized namespace: OPLo-core, containing the original annotations; OPLe-SD, for use in detailing schema. Recent trends in the design of domain ontologies as pattern languages emphasize adaptability, modularity, and practical application. These trends are driven by the need for more efficient and error-free ontology development, particularly in complex domains. The use of Generic Ontology Design Patterns (GODPs) and extensions to existing languages are central to these advancements.

4. CONCLUSIONS

The creation of ontology for the intelligent security system in residential communities creates the foundation for the knowledge-based application evolution and development. The ontology provides a common vocabulary of terms, and their relationships used in the process of development. The ontology is designed as local, that is geared towards the needs of specific system. On the other hand, it is based on the solid foundation of upper GFO ontology, simplifying data communication between different systems using the same ontology as basis.

The proposed ontology could be used to introduce artificial intelligence (AI) features to the future versions of security system, significantly enhancing the efficiency, scalability, and intelligence of both access control and video surveillance subsystems. AI would allow the

system to learn from past incidents, recognize patterns, and make decisions autonomously, improving response times and reducing false positives.

The functionality of security systems components and subsystems could be enhanced with the introduction of following AI modifications.

An intelligent agent that continuously analyses access patterns, detects anomalies (e.g., unusual times of entry), and flags potential security breaches based on learned behavior. AI Identity Verification. Machine learning algorithms enhance biometric identification (fingerprint, face recognition) by adapting to changes in a person's physical characteristics over time, thus reducing error rates. Behavioral Profiling Agent. This AI agent builds behavioral profiles of residents, staff, and frequent visitors, allowing the system to preemptively detect suspicious deviations from normal behavior (e.g., unusual time of access or multiple failed entry attempts).

Machine learning models for pattern recognition, object detection, and anomaly detection (e.g., identifying abandoned objects, unusual movement patterns, or unauthorized entry through unguarded areas). Facial Recognition with Deep Learning. Advanced facial recognition that not only identifies known individuals but also detects stress levels, emotional states, or intent based on facial cues. Predictive Surveillance. An AI module that processes real-time video feeds, predicts suspicious activity (e.g., loitering near critical points) based on historical data, and alerts guards before incidents occur.

A backend system that continually refines AI models using past access data and surveillance footage. This system provides automated suggestions for system improvements (e.g., better camera placement, more restrictive access rules). Automated Incident Learning. A learning module that reviews and categorizes past incidents (security breaches, false alarms, normal behavior) to train AI models, improving accuracy over time. Context-Aware System. AI that analyzes contextual data such as weather, time of day, and public events to adjust security measures accordingly (e.g., increase surveillance during festivals or modify access control rules during extreme weather events). These AI-enhanced features would make the security system more intelligent, responsive, and capable of learning from experience, providing a robust defense mechanism for the residential community.

References

1. Gruber T. R., (1993) "A translation approach to portable ontology specifications," Knowledge Acquisition, vol. 5, no. 2, pp. 199–220. https://doi.org/10.1006/knac.1993.1008
2. Neuhaus F. (2018). "What Is an Ontology?" ArXiv, October 22.
3. Tom Gruber (2009). Ontology. In Encyclopedia of Database Systems. Springer, pp. 1963–1965. https://doi.org/10.1007/978-0-387-39940-9_1318
4. Ganter B., Wille R. (2013). Formal Concept Analysis, Springer Berlin Heidelberg,
5. Hofstadter D. (1995). Fluid Concepts and Creative Analogies Computer Models of the Fundamental Mechanisms of Thought. Basic books.
6. Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L., Stuckenschmidt H. (2004) Contextualizing Ontologies. Journal of Web Semantics, vol. 1, issue 4, pp. 325–343. https://doi.org/10.1016/j.websem.2004.07.001
7. D. Benslimane, A. Arara, G. Falquet, Z. Maamar, P. Thiran, (2006) Contextual Ontologies. Springer, pp. 168–176. https://doi.org/10.1007/11890393_18
8. Burov Y. and Karpov I. (2023) "Contextual Concept Meaning Alignment Based on Prototype Theory." presented at the COLINS (3), pp. 137–146.
9. P. Bouquet, F. Giunchiglia, F.van Harmelen, L. Serafini, H. Stuckenschmidt (2003) C-Owl: Contextualizing Ontologies. Springer, pp. 164–179. https://doi.org/10.1007/978-3-540-39718-2_11
10. Cyc's knowledge base. Available at: https://www.cyc.com/archives/service/cyc-knowledge-base, last accessed: 2024/10/24
11. Wordnet. A lexical database for English. Available at: https://wordnet.princeton.edu/, last accessed: 2024/10/24.
12. Guarino N. (1998) "Formal ontology in information systems," Proceedings of the first international conference (FOIS'98), June 6–8, Trento, Italy, vol. 46, IOS press.
13. Guizzardi G., Wagner G. (2010). "Using the unified foundational ontology (UFO) as a foundation for general conceptual modeling languages," in Theory and applications of ontology: computer applications, Springer, Dordrecht, pp. 175–196. https://doi.org/10.1007/978-90-481-8847-5_8

14. Herre H. (2010) “General Formal Ontology (GFO): A foundational ontology for conceptual modelling,” in Theory and applications of ontology: computer applications. Springer, Dordrecht, pp. 297–345. https://doi.org/10.1007/978-90-481-8847-5_14
15. Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A., Schneider L. (2002). “The wonderweb library of foundational ontologies”.
16. P. Wohed, “Conceptual patterns for reuse in information systems analysis,” presented at the Advanced Information Systems Engineering: 12th International Conference, CAiSE 2000 Stockholm, Sweden, June 5–9, 2000 Proceedings 12, Springer, 2000, pp. 157–175. https://doi.org/10.1007/3-540-45140-4_12
17. Gangemi and V. Presutti, “Ontology design patterns,” in Handbook on ontologies, Springer, 2009, pp. 221–243. https://doi.org/10.1007/978-3-540-92673-3_10
18. R. de Almeida Falbo, M. P. Barcellos, J. C. Nardi, and G. Guizzardi, “Organizing ontology design patterns as ontology pattern languages,” presented at the The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26–30, 2013. Proceedings 10, Springer, 2013, pp. 61–75. https://doi.org/10.1007/978-3-642-38288-8_5
19. Souza E., Falbo R., and Vijaykumar N. “Using ontology patterns for building a reference software testing ontology,” presented at the 2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, IEEE, 2013, pp. 21–30. https://doi.org/10.1109/EDOCW.2013.10
20. Zambon E. and Guizzardi G., “Formal definition of a general ontology pattern language using a graph grammar,” presented at the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2017, pp. 1–10. https://doi.org/10.15439/2017F001
21. Hirt Q., Shimizu C., and Hitzler P., “Extensions to the Ontology Design Pattern Representation Language,” presented at the WOP@ISWC, 2019. Accessed: Jul. 20, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Extensions-to-the-Ontology-Design-Pattern-Language-Hirt-Shimizu/68392a85faf15d91310c3e21daef14a706bb4099
22. Illescas J., Ehrlinger L., Denk G., and Buchgeher G. “Towards an Ontology for Technical Security Standards,” presented at the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2023, pp. 1–8. https://doi.org/10.1109/ETFA54631.2023.10275669
23. Shoaib Farooq M. and Talha Waseem M. “Developing and Building Ontologies in Cyber Security,” arXiv e-prints, p. arXiv-2306, 2023.
24. Preuveneers D. and Joosen W. (2024) “An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications,” Future Internet, vol. 16, no. 3, p. 69. https://doi.org/10.3390/fi16030069
25. Babayeva G., Maennel K., and Maennel O. (2022). “Building an ontology for cyber defence exercises,” presented at the 2022 ieee european symposium on security and privacy workshops (euros&pw), IEEE, pp. 423–432. https://doi.org/10.1109/EuroSPW55150.2022.00050
26. Kelly D. (2009) “Determining factors that affect long-term evolution in scientific application software,” Journal of Systems and Software, vol. 82, no. 5, pp. 851–861. https://doi.org/10.1016/j.jss.2008.11.846
27. Khorikov V. “Short-term vs long-term perspective in software development,” Enterprise Craftsmanship. Accessed: Sep. 19, 2024. [Online]. Available: https://enterprisecraftsmanship.com/posts/short-term-vs-long-term-perspective/
28. Loebe, Frank, Patryk Burek, and Heinrich Herre (2022) “GFO: The General Formal Ontology.” Applied Ontology 17, no. 1, pp. 71–106. https://doi.org/10.3233/AO-220264

УДК 004.8:004.42:004.9:006.32:351.74

ПРОЕКТУВАННЯ ОНТОЛОГІЇ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ БЕЗПЕКИ ЖИТЛОВОГО МАСИВУ

Євген Буров; Юрій Жовнір; Олег Захарія

Національний університет «Львівська політехніка», Львів, Україна

Резюме. *Інтелектуальні системи безпеки є складною та перспективною областю для впровадження штучного інтелекту (ШІ). Їхня динамічна природа дозволяє системам безпеки не лише відслідковувати зміни в навколишньому середовищі, але й реагувати на них розумно та оперативно. Важливою частиною таких систем є онтологія, яка забезпечує загальний словник понять і термінів, що використовуються для описування об'єктів та їхньої взаємодії. Це дозволяє забезпечити*

узгодженість даних та ефективність комунікації між різними компонентами системи. У даному дослідженні онтологію розглянуто як програмний документ, що розробляється паралельно з основною системою безпеки, яка забезпечує її актуальність та адаптивність. Особливістю створюваної онтології є її локальний характер: вона відображає поточний стан і функціональність системи. Однак для забезпечення багаторазового використання та інтеграції з іншими рішеннями, розроблення базується на General Formal Ontology (GFO). Це дозволяє моделювати просторову, часову та ситуаційну динаміку, що особливо важливо для управління безпековими процесами в умовах постійних змін. Перша версія онтології створена на основі сценаріїв, підтримуваних системою. На початковому етапі вона представлена як набір шаблонів у вигляді мови шаблонів (Pattern Language), що спрощує її розроблення та подальшу адаптацію. Фінальна версія онтології реалізується у форматах OWL (Web Ontology Language) та RDF (Resource Description Framework), що забезпечує її відповідність сучасним стандартам обміну даними та інтеграції. Завдяки використанню цих форматів онтологія стає сумісною з іншими системами безпеки та може бути використана в різних інформаційних середовищах. Це відкриває можливості для масштабування та вдосконалення системи відповідно до нових викликів і вимог житлових комплексів. Запропонована онтологія та її інтеграція в інтелектуальну систему безпеки сприяє підвищенню рівня ситуаційної обізнаності, оперативності реагування та ефективності управління ризиками. Такий підхід дозволяє створювати гнучкі системи безпеки, що можуть адаптуватися до змін у реальному часі, та підвищує рівень надійності в сучасних житлових спільнотах.

Ключові слова: *інтелектуальна система безпеки, штучний інтелект (AI), онтологія, загальна формальна онтологія (GFO), мова веб-онтології, структура опису ресурсів, мова шаблонів, ситуаційна обізнаність, просторова та часова динаміка, житлова громада.*

https://doi.org/10.33108/visnyk_tntu2024.04.111 *Отримано 31.10.2024*