DOI: https://doi.org/10.36910/6775-2524-0560-2026-62-07
UDC 004.8:004.94:351.862
Kunanets Nataliia, Professor
https://orcid.org/0000-0003-3007-2462
Burov Evgen, Professor
http://orcid.org/0000-0001-8653-1520
Pasichnyk Volodymyr, Professor
http://orcid.org/0000-0002-5231-6395
Zakharia Oleg, postgraduate student
https://orcid.org/0009-0008-6979-129X
Zhovnir Yurii, PhD
https://orcid.org/0009-0006-6186-2861
National University «Lviv Polytechnic», Lviv, Ukraine

USING ACTIVE CONCEPTUAL MODELS FOR REASONING IN INTELLIGENT PHYSICAL SECURITY SYSTEMS BASED ON CASE KNOWLEDGE

Kunanets N., Burov E., Pasichnyk V., Zakharia O., Zhovnir Yu. Using active conceptual models for reasoning in intelligent physical security systems based on case knowledge. The article presents a knowledge-intensive approach to intelligent physical security systems that integrates Case-Based Reasoning (CBR) with Active Conceptual Models (ACMs) in a distributed multi-agent architecture. It argues that experiential case knowledge alone cannot ensure effective operation in dynamic environments and must be operationalized through executable, context-aware models. ACMs act as a mediating layer that transforms static case knowledge into agent-oriented models supporting situation interpretation, prediction of situation development, and real-time decision-making. A methodology for deriving ACMs from case knowledge is proposed, including case clustering, abstraction into prototypical structures, extraction of ontological elements, and formal execution. A mapping between the case space and executable conceptual models enables the systematic transformation of historical incidents and procedures into operational agent behavior models. The system architecture consists of Functional Agents, Local Agents, and a Coordinating Agent. Functional agents interpret sensor data, local agents execute ACMs and maintain situational awareness within specific zones, and the coordinating agent performs global reasoning and dynamic deployment of ACMs. The proposed approach enhances scalability, adaptability, and explainability, demonstrating the feasibility of integrating experiential knowledge, conceptual modeling, and agent-based execution into a unified intelligent security infrastructure.
Keywords: Active Conceptual Models; Case-Based Reasoning; intelligent security systems; multi-agent architecture; situational awareness; ontological modeling; Situation Calculus; Event Calculus; semantic reasoning; cyber-physical systems.

Кунанець Н.Е., Буров Є.В., Пасічник В.В., Захарія О.В., Жовнір Ю.І. Використання активних концептуальних моделей для аргументації в інтелектуальних системах фізичної безпеки на основі знань про конкретні випадки. У статті запропоновано знання-інтенсивний підхід до побудови інтелектуальних систем фізичної безпеки, що інтегрує Case-Based Reasoning (CBR) з Active Conceptual Models (ACMs) у межах розподіленої багатоагентної архітектури. Обґрунтовано, що накопичене експериментальне знання саме по собі є недостатнім для забезпечення оперативного реагування в динамічних середовищах житлових спільнот і потребує операціоналізації у вигляді виконуваних, контекстно-чутливих моделей. ACM розглядаються як медіаційний шар, що трансформує статичні представлення випадків у семантично узгоджені, агентно-орієнтовані моделі, придатні для інтерпретації ситуацій, прогнозування їх розвитку та прийняття рішень у реальному часі. Запропоновано формальну методологію виведення ACM із бази випадків, що охоплює етапи відбору та кластеризації випадків, їх абстрагування до прототипових і концептуальних структур. Архітектурно система реалізується як ієрархія Functional Agents, Local Agents та Coordinating Agent. Функціональні агенти забезпечують семантично збагачене сприйняття даних сенсорів; локальні агенти виконують ACM, підтримуючи ситуаційну обізнаність у межах конкретних зон; координуючий агент здійснює глобальне міркування, інтерпретацію багатолокаційних випадків, генерацію та динамічне розгортання ACM. Такий підхід забезпечує масштабованість, адаптивність, пояснюваність і стійкість системи до змін контексту.
Ключові слова: Active Conceptual Models; Case-Based Reasoning; інтелектуальна система фізичної безпеки; багатоагентна архітектура; онтологічне моделювання; Situation Calculus; Event Calculus; семантичне міркування; концептуальні моделі; кіберфізичні системи.

Постановка наукової проблеми. Modern physical security systems are undergoing a rapid transformation. With the proliferation of intelligent sensors, pervasive connectivity, and advances in artificial intelligence, security environments are no longer passive infrastructures but dynamic systems. Traditional rule-based mechanisms struggle to cope with emerging patterns, contextual ambiguity, and the variability inherent in human behavior. As a result, security decision-making increasingly requires models capable of integrating experiential knowledge, contextual awareness, and interpretable reasoning.

Case-Based Reasoning has emerged as a promising candidate to meet these needs. Its reliance on experiential knowledge aligns naturally with how security professionals operate: decisions are often grounded in analogies to prior incidents, standard operating procedures, and accumulated practical experience.

While a well-structured case-base provides the experiential foundation for decision-making, an intelligent security system requires an additional layer of reasoning to operate effectively in real-world environments. In practice, operational reasoning involves answering questions about what happens in system, what actions agents should take and why, what are the alternative action paths, and how multiple agents coordinate their behavior. Active-Conceptual Models offer precisely this capability.

Unlike traditional procedural rules, ACMs capture the temporal, causal, and contextual dependencies that govern security-relevant behavior. In the architecture of the intelligent physical security system, ACMs act as a bridge between high-level case knowledge and low-level operational execution.

This article develops a framework for constructing, using, and reasoning with ACMs within a case-based intelligent physical security system. Building on the conceptual and methodological foundations established in our earlier work [1], we introduce ACMs as a central representational mechanism for deriving action structures from case knowledge, predicting and tracking case development, supporting interpretive, abductive, temporal, and goal-directed reasoning, and enabling coordination across distributed intelligent agents. We show how ACMs can be mathematically formalized and linked to the hierarchical structure of the case-base. We discuss how ACMs provide the foundations for representing state transitions, action effects, and temporal relationships. Finally, we demonstrate how ACMs enhance both the operational and interpretive reasoning capabilities of the system, resulting in a more adaptive, transparent, and context-sensitive security infrastructure.

Аналіз досліджень. Case-Based Reasoning is an artificial intelligence paradigm that solves new problems by recalling and adapting solutions from similar past experiences. Instead of representing knowledge through abstract rules or statistical models, CBR systems maintain a repository of concrete problem–solution pairs (cases) and reason through similarity-based retrieval and adaptation [2].

CBR relies on two fundamental principles: Similarity-based reasoning, according to which similar situations tend to require similar responses, and experience groundedness, where knowledge is stored as concrete episodes rather than abstract rules. As new experiences accumulate, the case base grows incrementally, enabling systems to learn continuously from operational feedback.

Compared with traditional rule-based systems, CBR avoids the knowledge acquisition bottleneck associated with extracting large rule sets from experts. Rule-based approaches are effective when domain knowledge is well structured but often become brittle as complexity increases.

CBR also differs from data-driven machine learning approaches, particularly deep neural networks. While neural models excel at discovering patterns in large datasets, they often lack interpretability and require substantial training data. CBR systems, by contrast, provide inherent explainability because decisions can be justified through reference to similar past cases. They also operate effectively with smaller datasets and adapt incrementally through the revise–retain learning cycle, allowing new experiences to be incorporated without retraining.

These properties make CBR particularly suitable for domains characterized by contextual complexity and experiential expertise, such as medical diagnosis, legal reasoning, design, planning, and operational decision support [3]. Early work demonstrated analogical reasoning mechanisms capable of combining case reuse with general problem-solving strategies, as illustrated by systems such as PRODIGY [4]. Later developments improved case construction and representation, for example through systems that extract structured cases from operational data and incorporate techniques such as rough and fuzzy sets to better model human reasoning [5].

Recent research has increasingly explored hybrid architectures combining CBR with modern AI techniques, particularly large language models (LLMs). These systems employ LLMs for semantic case retrieval, enabling conceptual matching beyond traditional similarity metrics [6]. Other approaches focus on improved case representations and fine-tuning strategies that capture essential elements of prior experiences, thereby enhancing case retrieval and adaptation [6-8]. CBR principles are also being integrated with Retrieval-Augmented Generation (RAG) pipelines to improve knowledge retrieval and reasoning processes [9]. In parallel, theoretical work seeks to formalize CBR processes mathematically and integrate them with probabilistic and machine-learning frameworks [10]. Extensions incorporating cognitive capabilities such as self-reflection, curiosity, and goal-driven autonomy further enable agents to adapt their behavior in dynamic environments [6,11].

While CBR provides a powerful mechanism for leveraging experiential knowledge, an intelligent system must also translate this knowledge into operational behavior. A case base alone describes what happened in the past but does not directly specify how agents should act in evolving situations. Bridging this gap requires a representation capable of linking situations, actions, and outcomes in a form suitable for runtime reasoning.

Active Conceptual Modeling addresses this challenge by extending conceptual models from static descriptions to runtime operational artifacts. Traditionally, conceptual modeling techniques such as entity–relationship diagrams, UML models, and process diagrams served primarily as design-time tools supporting analysis and documentation. However, the increasing complexity of modern cyber-physical systems has created demand for models that remain active during system operation. Active Conceptual Modeling refers to an approach in which conceptual models participate directly in monitoring, reasoning, and adaptation processes [12,13]. Rather than acting as passive representations, ACM artifacts can be interpreted or executed during runtime, allowing them to respond dynamically to events and influence system behavior.

Several characteristics distinguish ACM from traditional conceptual modeling [12,14]. Active models support runtime execution, enabling them to react to operational events; dynamic adaptation, allowing models to evolve with changing environments; and a mediating role between physical systems, digital infrastructures, and human stakeholders.

A key theoretical development in ACM is the shift from viewing models as representations to viewing them as mediators. The triptych framework proposed by Mayr and Thalheim conceptualizes models as entities connecting the real world, digital systems, and human understanding [15]. Complementary research on mediation scripts and behavior-oriented modeling frameworks introduces explicit notions of states, transitions, and behavior into conceptual models, enabling their integration into system dynamics [12].

Executable modeling frameworks introduce formal execution semantics that bridge conceptual abstraction with operational behavior, supporting compositional reasoning, state management, and verification capabilities necessary for runtime systems [16]. Developments in artificial intelligence have also influenced ACM methodologies. Large language models enable conversational and iterative model construction, allowing conceptual models to be generated, refined, and visualized through natural language interaction [17]. Knowledge-driven approaches incorporating ontologies and knowledge graphs facilitate semi-automated model construction and runtime feedback through simulation or execution [10].

Applications of ACM span domains such as service-oriented systems, manufacturing, enterprise transformation, ecological modeling, and digital twins [14,18].

Despite these advances, important challenges remain. These include the lack of standardized execution semantics, limited large-scale empirical validation, and difficulties in verifying adaptive model behavior. Usability and accessibility issues in modeling tools also remain insufficiently addressed [20].

Taken together, this research highlights a complementary relationship between Case-Based Reasoning and Active Conceptual Modeling. CBR provides the experiential knowledge necessary for reasoning about past situations, while ACM offers the operational structures required to translate this knowledge into real-time system behavior.

Aim of the study. The purpose of the article is to develop a theoretically grounded and formally defined approach to the operationalization of case-based knowledge in intelligent physical security systems through the systematic derivation of Active Conceptual Models (ACMs) from the case base and their implementation within a multi-level multi-agent architecture to ensure situational awareness, interpretive reasoning, and adaptive real-time decision-making.

Main material and justification of the research results. To operationalize the knowledge-intensive CBR framework, we employ a distributed multi-agent architecture that links sensor perception with case-driven reasoning. The design emphasizes modularity, local autonomy, cognitive specialization, and edge computing, enabling scalable and resilient operation.

The system is organized as a three-layer hierarchy of agents. At the lowest level, Functional Agents (FAs) process raw sensor data and convert it into ontology-aligned perceptual events. By embedding interpretation near sensors, FAs reduce data volume and provide semantically meaningful observations such as detected persons, identities, or anomalous behaviors.

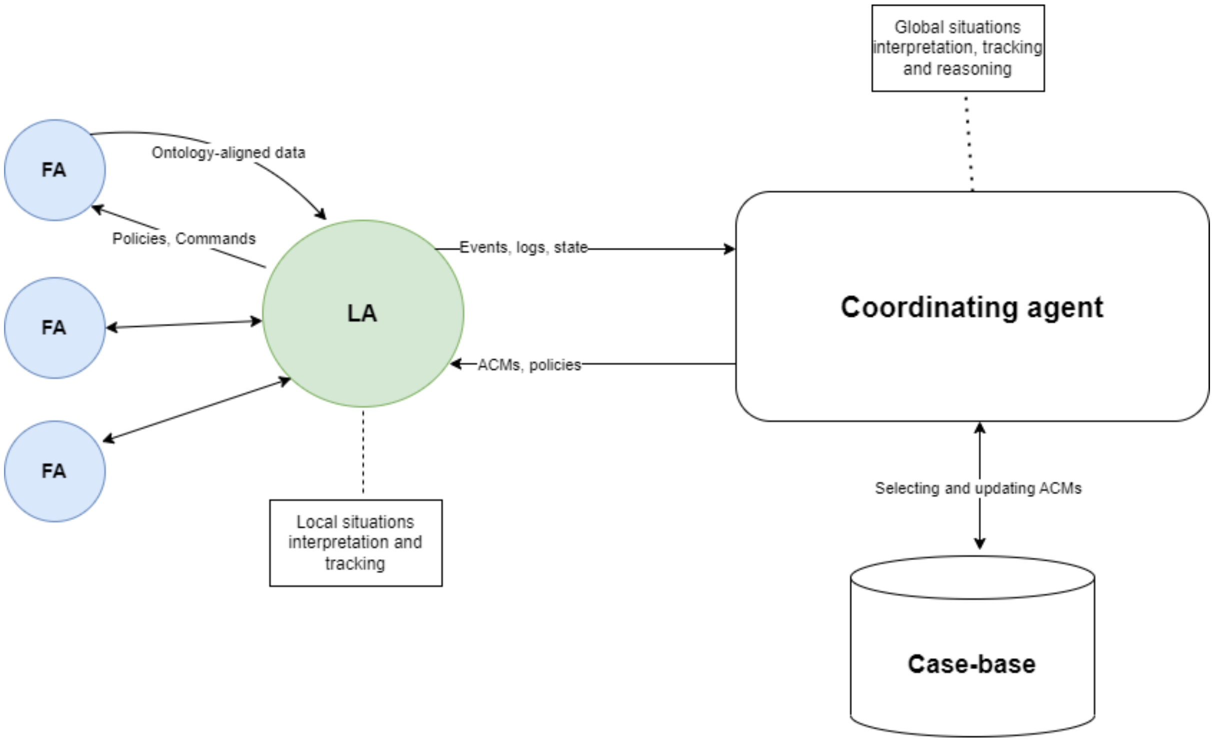

Figure 1. General architecture of multi-agent intelligent security system

Above them, Local Agents (LAs) supervise specific physical zones such as gates, lobbies, or parking areas. Each LA executes an ACM describing the expected behavior and security policies for that location. By integrating multiple FA inputs, LAs detect local situation patterns, trigger appropriate responses, and maintain situational awareness within their spatial domain.

At the top level, the Coordinating Agent (CA) performs global reasoning and system-wide coordination. Acting as the interface between the operational agents and the CBR knowledge base, it analyzes aggregated situational information, selects appropriate ACMs, and orchestrates coordinated responses such as alerts, policy adjustments, or additional data collection.

Together, these three layers form an integrated architecture where sensor-level perception, location-level situation interpretation, and case-driven global reasoning operate in a coordinated manner. Let's consider the parts of proposed architecture in more detail. Functional Agents. Functional Agents (FAs) form the perceptual layer of the intelligent security system and connect physical sensors with the conceptual reasoning processes executed by Local Agents. Their role is to transform heterogeneous raw sensor inputs into ontology-aligned semantic events and fluents that can be used by Active Conceptual Models. Unlike LAs, which perform situation interpretation and decision-making, FAs focus on sensing, preprocessing, feature extraction, device management, and semantic event generation. This separation cleanly distinguishes perception from reasoning, improving modularity and scalability. Each FA is associated with a particular sensor modality or device cluster (e.g., cameras, RFID readers, audio sensors, LiDAR units). A typical FA includes several components. The sensor interface layer manages communication with hardware devices, handling protocols, buffering, and error management. The signal processing and feature extraction module performs modality-specific analysis such as object detection in video streams, authentication event parsing for access devices, acoustic classification, or motion detection from radar or LiDAR. These processes may use embedded machine-learning models enabling near-sensor intelligence. A semantic event generator converts extracted features into ontology-referenced events and fluents that can be directly used for reasoning. FAs also include a health and diagnostics module that monitors sensor status and reports anomalies such as device failure, occlusion, or tampering. Communication with LAs uses lightweight asynchronous protocols that transmit only semantically meaningful events, reducing network load and supporting real-time operation.

FAs operate under the coordination of their corresponding LA. While FAs provide bottom-up perceptual updates, LAs supply top-down contextual directives that influence sensing priorities during critical situations. Through this interaction, Functional Agents bridge the physical sensing environment and the conceptual reasoning layers of the system.

Local Agents represent the core operational reasoning units of the system. Each LA supervises a specific spatial zone—such as a gate, lobby, or parking area—and maintains situational awareness by integrating semantic events received from Functional Agents. The LA architecture consists of two parts: a stable software core and a replaceable Active Conceptual Model. This separation provides operational stability while allowing the conceptual knowledge and behavior logic to evolve dynamically. The ACM defines how the agent interprets events, recognizes situations, and selects actions. It is location-specific and can be updated by the Coordinating Agent as new knowledge or security scenarios emerge. Each ACM includes four main elements: Ontology slice – a subset of the system ontology describing relevant entities, relations, and contextual concepts for the location; Procedural library – a collection of location-specific response procedures linked to local actuators; Communication protocols – structured interaction patterns between the LA, FAs, and the CA; Execution logic – reasoning mechanisms that govern state transitions and action triggering.

The software core provides the runtime infrastructure for executing ACMs. It includes the reasoning environment, local data storage for situational state and case fragments, communication services, and controlled interfaces to physical actuators. Through this design, each Local Agent maintains context-aware situational reasoning for its location while remaining adaptable to evolving operational knowledge.

The Coordinating Agent performs global reasoning and governance across the entire security system. While Local Agents manage individual zones, the CA integrates information from all agents to maintain a community-wide situational model and detect patterns that span multiple locations.

The CA performs four principal functions. First, it maintains global situation awareness, aggregating situational updates from LAs and identifying distributed incident patterns such as coordinated intrusions or repeated appearances of suspicious individuals. Second, it performs ACM derivation and deployment, generating or updating ACMs for Local Agents based on case knowledge, ontological structures, and operational context. Third, it manages system policies and governance, ensuring compliance with global safety rules, privacy constraints, and operational protocols. Fourth, it performs learning and system optimization by analyzing performance metrics and refining case structures, procedures, and reasoning strategies. The CA does not replace local autonomy but complements it. Local Agents provide rapid responses within their zones, while the CA supplies global context, strategic reasoning, and conceptual coordination.

Derivation of Active Conceptual Models from case knowledge represents a key mechanism in the proposed multi-agent security architecture, in which case knowledge stored in the case base is transformed into executable Active Conceptual Models. The purpose of the case-to-ACM derivation process is to convert historical and analytical case descriptions into structured models that can guide the behavior of Local Agents and the Coordinating Agent. The process of transformation can be formalized as a sequence of mappings between several structured knowledge spaces.

Let:

- C be the global set of cases stored in the case-base;
- L the set of locations within the security domain (gates, lobbies, parking areas);
- A^L the set of Local Agents;
- A^C the Coordinating Agent;
- O the domain ontology (including the GFO foundation and domain ontologies);
- S the set of situation descriptions;
- P the set of procedural schemas (goals, procedures, methods, actions);
- R the set of reasoning rules used during execution.

Each case $c \in C$ is modeled as a structured representation:

$$c = < S_c, A_c, T_c, G_c, \Xi_c >,$$

where $S_c = \{s_0, s_1, \dots, s_n\} \subseteq S$ is the finite set of states (situations), $A_c = \{a_0, a_1, \dots, a_{n-1}\}$ represents actions or tasks connecting states, $T_c \subseteq S_c \times A_c \times S_c$ is the transition relation, G_c is the set of goals associated with the case, Ξ_c contains meta-information such as provenance or risk evaluation.

Thus, each case represents a trajectory of evolving situations:

$$\mathrm{S}_0 \underset{a_0}{\rightarrow} S_1 \underset{a_1}{\rightarrow} \dots \underset{a_{n-1}}{\longrightarrow} S_n,$$

where the initial state is triggered by an observed event and the final state corresponds to case closure or resolution.

Each situation is described as a structured situational entity:

$$s =< \Gamma_s, \tau_s, \theta_s, \kappa_s >,$$

where Γ_s is the situation graph, τ_s the temporal extent (Chronoid), $\theta_s -$ the spatial extent (Topoid), and κ_s the configuration information.

Standard operational procedures (SOPs) are represented as normative cases that provide idealized templates of correct operational behavior. Together with empirical cases, they form the knowledge basis from which ACMs are derived.

An ACM defines the operational reasoning model for an agent. For a Local Agent operating at location $\ell \in \mathcal{L}$ the ACM is defined as:

$$ACM_{LA}(\ell) =< O_\ell, P_\ell, R_\ell, Prot_\ell, M_\ell >,$$

where $O_\ell \subseteq \mathcal{O}$ – ontology slice relevant to the agent, $P_\ell \subseteq \mathcal{P}$ – procedural library (goals and procedures), $R_\ell \subseteq \mathcal{R}$is – execution logic, $Prot_\ell$ – communication protocols, M_ℓ – metadata.

Similarly, the Coordinating Agent maintains:

$$ACM_{CA} =< O_{CA}, P_{CA}, R_{CA}, Prot_{CA}, M_{CA} >.$$

The derivation process consists of four main stages: case selection and clustering, case abstraction and generalization, knowledge extraction, ACM assembly. Case selection and clustering represent the first step in which cases relevant to a particular agent are identified, while for Local Agents the selection is based primarily on spatial criteria. Each agent is responsible for a domain consisting of a set of spatial regions:

$$Dom(\ell) \subseteq \{\theta_s | s \in S\}.$$

A case is considered relevant to location ℓ if at least one of its states occurs within this domain:

$$C_\ell = \{c \in C | \exists s \in \mathcal{S}_c : \Theta_s \in Dom(\ell)\}.$$

The Coordinating Agent selects cases that span multiple locations or are explicitly marked as global or SOP cases:

$$C_{CA} = \{c \in C || span(c)| > 1 \vee c\ is\ global\}.$$

This step partitions the case-base into clusters aligned with agent responsibilities.

Case abstraction and generalization represent a process in which cases within each cluster are generalized in order to identify recurring patterns, while a hierarchical abstraction structure is constructed to organize the resulting knowledge representations:

Level 0 represents case instances that correspond to unique historical events recorded in the case base.

Level 1 represents prototypical cases that are aggregated representations capturing common sequences of states and actions across similar cases.

Level 2 represents conceptual cases that provide abstract descriptions emphasizing state classes, transition types, and outcome categories.

Level 3 represents meta-cases that form higher-level structures describing relationships among conceptual cases.

Formally, if $\sim$ denotes a similarity relation over cases, equivalence classes $[c]_\sim$ define clusters of similar cases. A prototypical case is derived as:

$$proto([c]) =< S^*, A^*, T^*, G^*, E^* >,$$

where elements are parameterized abstractions of the corresponding elements across all cases in the class. Conceptual cases further abstract these structures to capture generalized situation patterns and decision structures.

Knowledge extraction represents the next stage, during which ACM components are extracted from conceptual and prototypical cases and four types of knowledge are derived. Ontology slices correspond to ontology elements referenced in the states, actions, and goals of each conceptual case (k) that are collected to form a structured representation of the domain knowledge:

$$Ont(k) = \{o \in) | o\ appears\ in\ S, A, or\ G\}.$$

The ontology slice for an agent is the union of ontology elements referenced by its conceptual cases:

$$O_\ell = \bigcup_{k \in K_\ell} Ont(k).$$

A closure operation adds required ontological dependencies.

Procedural libraries are derived from conceptual cases in which transitions correspond naturally to procedural knowledge, where actions represent operations, transition sequences correspond to methods, chains of transitions correspond to procedures, and case goals correspond to higher-level agent goals.

Thus, each conceptual case can be mapped to one or more procedures:

$$\Pi: C_{conc} \to P(P).$$

For Local Agents:

$$P_\ell = \bigcup_{k \in K_\ell} \Pi(k).$$

SOP cases contribute canonical procedures, while empirical cases contribute variations and exception handling.

Execution logic is derived from case trajectories using Situation Calculus and Event Calculus. Three main types of rules are produced: State recognition rules detect when current sensor-derived situations correspond to case states; action feasibility rules**.** Determine when actions are possible:

$$Poss(a, s) \leftrightarrow Precond(a, s).$$

Successor state axioms describe how actions change the states of the world. Execution logic for a Local Agent is defined as:

$$R_\ell = \bigcup_{k \in K_\ell} \Lambda(\mathrm{k}, \Pi(k)).$$

These rules allow agents to interpret sensory events, activate procedures, and infer outcomes.

Communication protocols derived from conceptual cases specify the required perceptual events for state recognition, reporting obligations, and coordination dependencies, where EEE denotes the observable event types and MMM denotes the message schemas. For a conceptual case k, Obs(k) defines the required events, while Rep(k) defines the reporting messages.

Protocols are derived as:

$$Prot_\ell = \bigcup_{k \in K_\ell} \{\Phi_{\mathrm{FA}}(\mathrm{k}), \Phi_{\mathrm{CA}}(\mathrm{k})\}.$$

These specify interactions between Functional Agents, Local Agents, and the Coordinating Agent.

ACM assembly and deployment represent the final stage in which the ACM package includes an ontology slice, a procedural library, execution logic, communication protocols, and metadata. Local Agents receive ACMs specialized for their location, while the Coordinating Agent receives a global ACM supporting cross-location reasoning. This modular design allows ACMs to evolve without modifying agent software cores.

Example of a tailgating attempt scenario considers a case describing a tailgating attempt at Gate 1 in which the case instance includes in particular the following states S_0 where a resident approaches the gate and swipes a card, S_1 where the gate opens, S_2 where an unknown person follows closely behind, S_3 where the gate closes before entry representing a successful outcome, and S_3' where an unauthorized person enters representing a failure.

Transitions include verifying the card, tracking movement, detecting following behavior, and triggering responses. Authorized access management goals include allowing authorized access, preventing unauthorized entry, and maintaining smooth system operation. A prototypical case is derived from multiple similar incidents across different gates and represents a generalized pattern of system behavior in which generalized states include authorized access initialization, entry tracking, detection of close following, countermeasure activation, and case closure, while entities such as residents, followers, and gates are treated as parameters. A conceptual case provides an abstract representation of the scenario by generalizing entities, states, and transitions and by emphasizing the structural relationships among system components and operational outcomes. The conceptual abstraction represents the Tailgating Pattern:

$$k =< S, A, T, G, E >,$$

with goals such as MaintainSecurity, EnforceAuthorization, MinimizeDisruption.

ACM derivation from this conceptual case produces the Local Agent ACM for Gate 1, which extracts an ontology slice containing classes such as Person, Resident, UnknownPerson, Gate, and behavioral concepts including Tailgating and AuthorizedEntry. The derived ACM also includes procedures implementing tailgating detection and response mechanisms, execution rules supporting the recognition of close-following behavior and the triggering of countermeasures, and communication protocols that regulate interaction with Functional Agents providing sensor data as well as reporting to the Coordinating Agent when anomalies occur.

Висновки. This article proposes a knowledge-intensive framework for intelligent physical security systems that integrates case-based reasoning with Active Conceptual Models in a distributed multi-agent architecture. The central idea is that case knowledge provides a rich experiential basis for reasoning but must be operationalized through executable, context-sensitive models to support real-time perception, interpretation, and response. ACMs serve as the mediating layer that transforms static case representations into active reasoning structures usable by agents. The work therefore bridges three perspectives: experiential knowledge stored in a case base, conceptual modeling grounded in ontology and situation theory, and operational reasoning in cyber-physical security systems.

Cases are modeled as temporally ordered trajectories of states, transitions, and goals. This view treats cases not merely as problem–solution pairs but as evolving behavioral patterns that can be monitored and anticipated during system operation. ACMs are then derived from this knowledge as executable conceptual artifacts that integrate ontology fragments, procedural knowledge, execution logic, and communication protocols.

A formal methodology for case-to-ACM derivation is introduced. The process maps case instances to prototypical and conceptual cases and subsequently derives ontology slices, procedural libraries, execution rules, and communication protocols. These elements are assembled into ACMs that can be deployed across the system’s multi-agent architecture. This layered architecture supports scalability, specialization, and coordinated reasoning across distributed security components.

Conceptually, the approach reframes physical security reasoning as continuous situation interpretation rather than isolated event detection. By tracking evolving situations through ACM structures, the system can distinguish between superficially similar behaviors—such as legitimate access versus tailgating—using temporal patterns and contextual knowledge derived from previous cases. Operationally, ACM-driven agents enable adaptive behavior without retraining.

Compared with purely rule-based systems, the approach reduces brittleness and knowledge engineering overhead by grounding behavior in accumulated experience. Compared with purely data-driven machine learning systems, it offers stronger explainability and incremental learning in data-scarce or evolving environments.

Nevertheless, several challenges remain. The growth of case bases requires careful governance to maintain coherent ACM libraries. Distributed deployment introduces issues of temporal synchronization and coordination among agents. Computational complexity in similarity assessment, abstraction, and global reasoning must be addressed through efficient indexing and incremental reasoning mechanisms.

Future research should focus on empirical validation of the framework in real residential communities, as well as on verification techniques ensuring the safety of dynamically generated ACMs. Finally, the integration of automated learning methods, such as improved case abstraction and hybrid use of large language models for conceptual enrichment, represents a promising direction.

In summary, intelligent physical security systems require models that connect perception, reasoning, and action in an explainable and experience-driven manner. Active Conceptual Models derived from case knowledge provide such a foundation within a distributed multi-agent architecture. Beyond the domain of physical security, the proposed framework may also be applicable to a wide range of cyber-physical and socio-technical systems requiring adaptive and interpretable decision making.

Список бібліографічних описів

1. Burov E., Zhovnir Y., Zakhariya O., Kunanets N. Situation anticipation and planning framework for intelligent environments. *Radio Electronics, Computer Science, Control*. 2025. № 2. P. 94–109. https://doi.org/10.15588/1607-3274-2025-2-8
2. Bergmann R. Experience management: foundations, development methodology, and internet-based applications. Berlin; Heidelberg : Springer, 2002.
3. Yan A., Cheng Z. A review of the development and future challenges of case-based reasoning. *Applied Sciences*. 2024. Vol. 14, № 16. Art. 7130. https://doi.org/10.3390/app14167130
4. Veloso M. M., Carbonell J. G. Derivational analogy in PRODIGY: automating case acquisition, storage, and utilization. *Case-Based Learning*. Boston : Springer, 1993. P. 55–84. DOI: https://doi.org/10.1007/978-1-4615-3228-6_3
5. Patterson D., Anand S. S., Dubitzky W., Hughes J. G. Towards automated case knowledge discovery in the M2 case-based reasoning system. *Knowledge and Information Systems*. 1999. Vol. 1. P. 61–82. DOI: https://doi.org/10.1007/BF03325091.

6. Natalis K., Christou D., Kondapalati V. Review of case-based reasoning for LLM agents: theoretical foundations, architectural components, and cognitive integration. *arXiv*. 2025. Available at: https://arxiv.org/html/2504.06943v2 (accessed: 20.02.2026). https://doi.org/10.48550/arXiv.2504.06943.
7. Leake D., Wilkerson Z., Crandall D. J. Combining case-based reasoning with deep learning: context and ongoing case feature learning research. *Neuro-Symbolic Learning and Reasoning in the Era of Large Language Models*. 2023. 11 December. Available at: https://openreview.net/forum?id=nc9XqnaLys (accessed: 20.02.2026).
8. Pradeep P., Caro-Martínez M., Wijekoon A. A practical exploration of the convergence of case-based reasoning and explainable artificial intelligence. *Expert Systems with Applications*. 2024. Vol. 255 (Part D). https://doi.org/10.1016/j.eswa.2024.124733
9. Smiti A., Elouedi Z. Dynamic maintenance case base using knowledge discovery techniques for case based reasoning systems. *Theoretical Computer Science*. 2020. Vol. 817. P. 24–32. https://doi.org/10.1016/j.tcs.2019.06.026
10. Wu Y., Zhou J. A contextual information-augmented probabilistic case-based reasoning model for knowledge graph reasoning. *International Conference on Case-Based Reasoning*. Cham : Springer, 2023. https://doi.org/10.1007/978-3-031-40177-0_7
11. Muñoz-Avila H., Jaidee U., Aha D. W., Carter E. Goal-driven autonomy with case-based reasoning. *Conference proceedings.* Berlin; Heidelberg : Springer, 2010. P. 228–241. https://doi.org/10.1007/978-3-642-14274-1_18
12. An S., Rugaber S., Hammock J., Goel A. K. Contextualizing large-scale domain knowledge for conceptual modeling and simulation. *arXiv Preprint*. *35th International Workshop on Qualitative Reasoning, IJCAI ECAI Conference*, July 23-29, 2022, Messe Wien, Vienna, Austria. 2022. arXiv:2209.02579.
13. Storey V. C., Lukyanenko R., Rodríguez Castellanos A. Conceptual modeling: topics, themes, and technology trends. *ACM Computing Surveys*. 2023. https://doi.org/10.1145/3589338
14. Schewe K.-D., Bósa K., Buga A., Nemeş S. T. Conceptual modelling of service-oriented software systems. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*. 2018. Vol. 13. P. 216–233.
15. Mayr H. C., Thalheim B. The triptych of conceptual modeling: a framework for a better understanding of conceptual modeling. *Software and Systems Modeling*. 2021. Vol. 20, № 1. P. 7–24. https://doi.org/10.1007/s10270-020-00836-z
16. Lukyanenko R., Samuel B. M., Storey V., Sturm A. Conceptual modeling systems: a vision for the future of conceptual modeling. *ER Forum / PhD Symposium*. 2022.
17. Härer F. Conceptual model interpreter for large language models. *arXiv* Preprint. 2023. arXiv:2311.07605. https://doi.org/10.48550/arXiv.2311.07605.
18. Robinson S. Conceptual modelling for simulation: progress and grand challenges. *Journal of Simulation*. 2020. Vol. 14, № 1. P. 1–20. https://doi.org/10.1080/17477778.2019.1604466.
19. Buckl S., Matthes F., Monahov I., Schweda C. M. Modeling enterprise architecture transformations. *Proceedings of the 2011 IEEE International Enterprise Distributed Object Computing Conference Workshops*. 2011. P. 372–381.
20. Bork D., Ali S. J., Roelens B. Conceptual modeling and artificial intelligence: a systematic mapping study. *arXiv Preprint*. 2023. arXiv:2303.06758. https://doi.org/10.48550/arXiv.2303.06758

References

1. Burov E., Zhovnir Y., Zakhariya O., Kunanets N. Situation anticipation and planning framework for intelligent environments. *Radio Electronics, Computer Science, Control*. 2025. № 2. P. 94–109. https://doi.org/10.15588/1607-3274-2025-2-8
2. Bergmann R. Experience management: foundations, development methodology, and internet-based applications. Berlin; Heidelberg : Springer, 2002.
3. Yan A., Cheng Z. A review of the development and future challenges of case-based reasoning. *Applied Sciences*. 2024. Vol. 14, № 16. Art. 7130. https://doi.org/10.3390/app14167130
4. Veloso M. M., Carbonell J. G. Derivational analogy in PRODIGY: automating case acquisition, storage, and utilization. *Case-Based Learning*. Boston : Springer, 1993. P. 55–84. DOI: https://doi.org/10.1007/978-1-4615-3228-6_3
5. Patterson D., Anand S. S., Dubitzky W., Hughes J. G. Towards automated case knowledge discovery in the M2 case-based reasoning system. *Knowledge and Information Systems*. 1999. Vol. 1. P. 61–82. DOI: https://doi.org/10.1007/BF03325091.
6. Natalis K., Christou D., Kondapalati V. Review of case-based reasoning for LLM agents: theoretical foundations, architectural components, and cognitive integration. *arXiv*. 2025. Available at: https://arxiv.org/html/2504.06943v2 (accessed: 20.02.2026). https://doi.org/10.48550/arXiv.2504.06943.
7. Leake D., Wilkerson Z., Crandall D. J. Combining case-based reasoning with deep learning: context and ongoing case feature learning research. *Neuro-Symbolic Learning and Reasoning in the Era of Large Language Models*. 2023. 11 December. Available at: https://openreview.net/forum?id=nc9XqnaLys (accessed: 20.02.2026).
8. Pradeep P., Caro-Martínez M., Wijekoon A. A practical exploration of the convergence of case-based reasoning and explainable artificial intelligence. *Expert Systems with Applications*. 2024. Vol. 255 (Part D). https://doi.org/10.1016/j.eswa.2024.124733.
9. Smiti A., Elouedi Z. Dynamic maintenance case base using knowledge discovery techniques for case based reasoning systems. *Theoretical Computer Science*. 2020. Vol. 817. P. 24–32. https://doi.org/10.1016/j.tcs.2019.06.026.
10. Wu Y., Zhou J. A contextual information-augmented probabilistic case-based reasoning model for knowledge graph reasoning. *International Conference on Case-Based Reasoning*. Cham : Springer, 2023. https://doi.org/10.1007/978-3-031-40177-0_7
11. Muñoz-Avila H., Jaidee U., Aha D. W., Carter E. Goal-driven autonomy with case-based reasoning. *Conference proceedings*. Berlin ; Heidelberg : Springer, 2010. P. 228–241. https://doi.org/10.1007/978-3-642-14274-1_18

12. An S., Rugaber S., Hammock J., Goel A. K. Contextualizing large-scale domain knowledge for conceptual modeling and simulation. *arXiv Preprint 35th International Workshop on Qualitative Reasoning, IJCAI ECAI Conference*, July 23-29, 2022, Messe Wien, Vienna, Austria. *arXiv Preprint*. 2022. arXiv:2209.02579.
13. Storey V. C., Lukyanenko R., Rodríguez Castellanos A. Conceptual modeling: topics, themes, and technology trends. *ACM Computing Surveys*. 2023. https://doi.org/10.1145/3589338
14. Schewe K.-D., Bósa K., Buga A., Nemeş S. T. Conceptual modelling of service-oriented software systems. *Enterprise Modelling and Information Systems Architectures (EMISAJ)*. 2018. Vol. 13. P. 216–233.
15. Mayr H.C., Thalheim B. The triptych of conceptual modeling: a framework for a better understanding of conceptual modeling. *Software and Systems Modeling*. 2021.Vol.20, № 1.P. 7–24. https://doi.org/10.1007/s10270-020-00836-z
16. Lukyanenko R., Samuel B. M., Storey V., Sturm A. Conceptual modeling systems: a vision for the future of conceptual modeling. *ER Forum* / PhD Symposium. 2022.
17. Härer F. Conceptual model interpreter for large language models. *arXiv Preprint*. 2023. arXiv:2311.07605. https://doi.org/10.48550/arXiv.2311.07605.
18. Robinson S. Conceptual modelling for simulation: progress and grand challenges. *Journal of Simulation*. 2020. Vol. 14, № 1. P. 1–20. https://doi.org/10.1080/17477778.2019.1604466.
19. Buckl S., Matthes F., Monahov I., Schweda C. M. Modeling enterprise architecture transformations. *Proceedings of the 2011 IEEE International Enterprise Distributed Object Computing Conference Workshops*. 2011. P. 372–381.
20. Bork D., Ali S. J., Roelens B. Conceptual modeling and artificial intelligence: a systematic mapping study. *arXiv Preprint*. 2023. arXiv:2303.06758. https://doi.org/10.48550/arXiv.2303.06758

Історія статті:
Отримано: 16.02.2026 Доопрацьовано: 20.03.2026 Прийнято до друку: 23.03.2026 Опубліковано: 29.03.2026