https://doi.org/10.31891/csit-2025-1-19
UDC 004.4

Anatoliy MELNYK
IT Step University
Bohdan ZIMCHENKO
Lviv Polytechnic National University

CRISP-FUZZY RULE MANAGEMENT IN THE CLOUD: ENABLING SCALABLE DECISION-MAKING FOR CYBER-PHYSICAL SYSTEMS

Today, many cyber-physical systems (CPS) rely on local decision-making frameworks that often fail to address both precise thresholds and the ambiguity inherent in sensor data. There is a need to develop a scalable, cloud-based decision support system (DSS) that unifies crisp rule evaluation with fuzzy logic to improve decision accuracy and responsiveness across diverse applications. The aim of this paper is to design and implement a cloud-hosted crisp-fuzzy rule management system that supports centralized rule administration and asynchronous processing for multiple CPS domains.

Our approach employs a microservices architecture within a Microsoft Azure environment, comprising three core APIs: User Management, Knowledge Management, and Decision Support. The system integrates secure multi-tenant access using external identity providers and leverages a PostgreSQL database with a multi-tenant schema. Sensor data from various devices are transmitted via HTTP and queued through Azure Service Bus, thereby decoupling data ingestion from intensive rule evaluation. A background worker, known as the Decision Relay Consumer, processes each incoming message by applying direct threshold comparisons for crisp rules and linear interpolation for fuzzy membership functions, thus handling uncertain sensor readings effectively.

Experimental validation using a smart garden simulation demonstrates that the integration of crisp and fuzzy rule evaluations enhances the system's ability to prioritize and trigger appropriate actions in real time. The results confirm that the proposed architecture not only improves decision-making reliability under ambiguous conditions but also reduces on-device computational burdens, facilitating centralized management and scalability.

The novelty of this work lies in its unified framework that seamlessly combines crisp thresholds with fuzzy logic in a cloud-based environment, enabling cross-domain applicability and adaptive rule management. The practical significance extends to various industries—including agriculture, manufacturing, and smart buildings—where timely and robust decision-making is essential.

Keywords: cyber-physical systems, cloud computing, decision support, crisp-fuzzy logic, microservices, rule management.

Анатолій МЕЛЬНИК
ІТ Степ Університет
Богдан ЗІМЧЕНКО
Національний університет «Львівська політехніка»

УПРАВЛІННЯ ЧІТКО-НЕЧІТКИМИ ПРАВИЛАМИ В ХМАРІ: ЗАБЕЗПЕЧЕННЯ МАСШТАБОВАНОГО ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ КІБЕРФІЗИЧНИХ СИСТЕМ

Сьогодні багато кіберфізичних систем (КФС) спираються на локальні фреймворки прийняття рішень, які часто не враховують як точні порогові значення, так і неоднозначність, притаманну для даних із сенсорів. Існує необхідність розробки масштабованої хмарної системи підтримки прийняття рішень (СППР), що об'єднує оцінювання чітких правил із нечіткою логікою для підвищення точності рішень та оперативності у різноманітних застосуваннях. Метою цієї роботи є розробка та впровадження системи управління чітко-нечіткими правилами, розміщеної в хмарі, яка підтримує централізоване адміністрування правил та асинхронну обробку даних для численних доменів КФС.

Наш підхід базується на архітектурі мікросервісів у середовищі Microsoft Azure, що складається з трьох основних API: Управління Користувачами, Управління Знаннями та Підтримка Прийняття Рішень. Система інтегрує безпечний багатокористувацький доступ із застосуванням зовнішніх провайдерів ідентифікації та використовує базу даних PostgreSQL із мульти-схемою. Дані із сенсорів з різних пристроїв передаються через HTTP та розміщуються у черзі за допомогою Azure Service Bus, що дозволяє відділити процес прийому даних від інтенсивної оцінки правил. Фоновий процес, що називається Decision Relay Consumer, обробляє кожне отримане повідомлення шляхом застосування прямих порогових порівнянь для чітких правил, та лінійної інтерполяції для нечітких функцій членства, таким чином ефективно працюючи з невизначеними значеннями сенсорів.

Експериментальна перевірка за допомогою симуляції розумного саду демонструє, що інтеграція оцінювання чітких і нечітких правил покращує здатність системи визначати пріоритетність та ініціювати відповідні дії в режимі реального часу. Отримані результати підтверджують, що запропонована архітектура не лише підвищує надійність прийняття рішень в умовах неоднозначності, а й зменшує обчислювальне навантаження на пристрої, сприяючи централізованому адмініструванню та масштабованості.

Новизна даної роботи полягає у створенні єдиного підходу, що безшовно поєднує чіткі порогові значення з нечіткою логікою в хмарному середовищі, що забезпечує застосовність у різних доменах та адаптивне управління правилами. Практична значимість розробки охоплює різноманітні галузі, зокрема сільське господарство, виробництво та розумні будівлі, де своєчасне та надійне прийняття рішень є критично важливим.

Ключові слова: кіберфізичні системи, хмарні обчислення, підтримка прийняття рішень, чітко-нечітка логіка, мікросервіси, управління правилами

Introduction

Autonomous CPSs can make decisions and execute actions independently of human intervention. These systems encounter numerous challenges, such as managing real-time constraints, processing extensive streams of sensor data, addressing uncertainty in measurements and operational settings, and ensuring safety and security [1]. Facilitating efficient decision-making within these systems presents a considerable challenge. Diverse technological methodologies are utilized to get autonomy, contingent upon the system's architecture and the tasks it executes.

In simple scenarios, a clear threshold-based decision rule may be adequate. Nevertheless, numerous real-world scenarios necessitate more nuanced reasoning. Fuzzy methodologies facilitate the representation of partial truth and yield resilient outcomes in response to ambiguous or overlapping sensor data. Previous studies have shown the feasibility of integrating crisp and fuzzy logic to tackle uncertainty in CPSs. Nonetheless, the effective management of these rule sets across many autonomous systems continues to be a challenging endeavor.

To create a more scalable decision-making framework in CPSs, it is advantageous to dissociate DSS from the particular architecture and functions of any unique CPS instance, as previously suggested in our research [2]. A centralized decision support service can function autonomously, allowing for adaptability across various applications, rather than integrating decision logic within each individual system. This separation improves both flexibility and scalability, allowing CPS instances with diverse requirements to engage with a common decision-making framework without necessitating unique logic for each deployment. By transferring rule administration to an external service, we establish a standardized approach for managing decision logic that is independent of specific system limitations while yet addressing their requirements. Consequently, numerous CPSs can effectively utilize a shared decision support service.

A critical necessity emerges when several CPSs, each representing distinct organizations (e.g., various manufacturing lines or agricultural locations), must collaborate on a unified decision support service. These CPS instances may function within different knowledge domains. For example, one oversees soil moisture, while another regulates industrial vibration levels. Nonetheless, they all require a centrally administered and unified set of logic definitions. This project aims to provide a cloud-based crisp-fuzzy rule management system capable of simultaneously supporting the rules of different organizations, enabling each CPS to authenticate, submit sensor data, and obtain decisions on demand. Through the integration of user management, we enable domain experts to independently manage their rule definitions without disrupting the configurations of other organizations, while ensuring that devices may safely acquire tokens for automated requests.

This study proposes a cloud-based, microservices-oriented strategy in which the decision logic is hosted independently of the specific CPS architecture. An exclusive user management module oversees authentication and identity control, enabling each CPS to log in using machine-to-machine (M2M) credentials. A knowledge management API allows domain specialists to establish or modify precise and ambiguous rules, whereas a decision support API perpetually analyzes incoming sensor data. A background worker assesses sensor inputs against established rule sets to provide scalable, asynchronous processing and triggers corresponding actions. To assess the viability and efficacy of our methodology, we implement a smart garden scenario, illustrating how a CPS organization might utilize both crisp thresholds and fuzzy logic for enhanced environmental control.

State-of-the-art

Numerous new CPSs demand resilient and reusable DSSs capable of efficiently managing real-time data streams and accommodating sensor uncertainties. An expanding corpus of research underscores fuzzy logic as an appropriate approach for describing partial truths and uncertain inputs. However, existing implementations frequently adhere to domain-specific designs, constraining their adaptability for CPSs deployed in varied contexts.

A fuzzy wavelet neural network is presented in [3] for the optimization of energy usage and traffic forecasting in intelligent transportation systems. The authors aim to minimize power consumption and enhance real-time responsiveness by incorporating fuzzy logic to manage uncertainties in traffic density.

A DSS based on Markov decision processes for disruptive occurrences in smart buildings is presented in [4]. This approach prioritizes incident-driven scenarios, such as power outages or fires, highlighting the necessity for swift responsiveness within a delineated framework.

Research initiatives in [5] focus on a knowledge-driven design guidance system for cloud-based DSSs in intricate engineering applications. The paper delineates a systematic approach of "formulation-refinement-exploration-improvement" to encapsulate domain knowledge and facilitate iterative decision-making procedures.

A systematic literature review in [6] examines software designs for healthcare-oriented CPSs, emphasizing the essential requirement for dynamic rule-based decision support that integrates real-time data and interoperability. Their assessment highlights how multi-tier architectures and customized middleware may connect sensor data from patient-monitoring devices to cloud servers.

A thorough overview of edge-cloud computing for cyber-physical systems is provided in [7], emphasizing solutions for latency, resource utilization, and system dependability. The authors categorize advanced techniques, such as heuristic optimization and machine learning, to enhance performance in dispersed settings.

An alternative methodology utilizing Pythagorean fuzzy sets is described in [8]. This study illustrates advanced decision-making for digital economy initiatives by optimizing computational loads and evaluating various performance metrics.

The study in [9] examines real-time safety monitoring on construction sites, where a CPS amalgamates architectural information modeling with sensor data to identify hazards and notify workers. The authors demonstrate how continuous data streams can be integrated into a DSS to mitigate on-site hazards.

The study in [10] examines human-in-the-loop decision-making for complicated CPSs, highlighting interactive data processing in which human operators work alongside automated control systems. This architecture is advantageous in contexts where complete automation is unfeasible.

Another challenge in industrial CPS pertains to the allocation of data analysis duties between cloud and edge servers. Reference [11] presents a multi-criteria fuzzy logic recommendation system to choose the optimal location for processing incoming sensor data, hence ensuring efficient and timely analytics.

A multi-tenant cloud-based DC nano grid is presented in [12] to enhance energy efficiency and self-sustainability in smart buildings. This paradigm spreads power from renewable sources across several structures, diminishing reliance on conventional grids.

Although various studies illustrate the potential of fuzzy logic to improve DSS functionality in particular CPS contexts, they generally remain limited to specific use cases or singular designs. A unified, cloud-hosted crisp-fuzzy rule engine is required to serve numerous CPS domains, ensuring seamless integration, scalability, and adaptation. By decoupling rule definitions from specific systems and situating them in a secure, multi-tenant environment, CPS instances can utilize shared logic for real-time decision-making while maintaining domain-specific requirements.

Overall system architecture

Figure 1 depicts the cloud-based implementation of the proposed crisp-fuzzy rule management system in an Azure environment. Each service operates autonomously to perform a designated function, while jointly constituting a unified platform for many organizations requiring unique decision-making logic. Utilizing microservices architecture enables the system to scale individual components according to demand, facilitate asynchronous processing of sensor data, and securely interface with external identity providers.

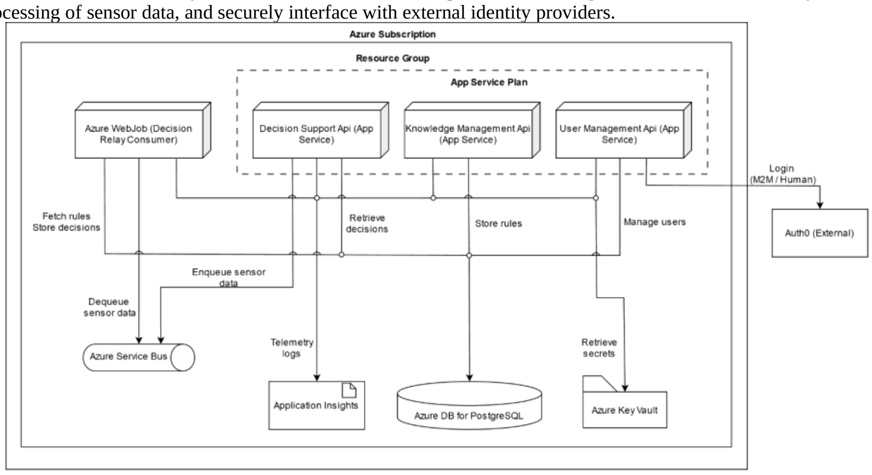

Fig. 1. Azure deployment diagram

The design centers around three principal APIs, each deployed as an Azure App Service within a consolidated App Service Plan. The User Management API governs all facets of user registration, role assignment, and organizational management. In this design, a singular "organization" pertains to a specific CPS domain or deployment, such as a manufacturing line or an agricultural environment. Auth0 functions as the external identity provider, producing JSON Web Tokens (JWTs) for both human users and M2M logins. Administrators can create new organizations, whereas experts are granted permissions to manage rules within their designated domain. The User Management API fetches secrets, including database connection strings and Auth0 credentials, from Azure Key Vault, thereby guaranteeing that sensitive information is not stored in application code.

The Knowledge Management API concentrates on the definition, modification, and storage of the rule sets. These rules may be entirely crisp, fuzzy, or mixed. Experts from each organization access this API under permitted roles, ensuring that rules remain distinct and secure. Upon the addition or modification of a rule, the API stores it in an Azure PostgreSQL database. This database contains all organizations, user accounts, and additional metadata, while row-level ownership and foreign key relationships ensure the segregation of each organization’s data.

While the first two APIs handle user identity and rule definitions, the Decision Support API functions as the gateway for sensor data submissions from any CPS. Instead of processing decisions instantaneously, it places the incoming readings into an Azure Service Bus queue. This asynchronous process enables the system to manage an influx of sensor data without overloading the API. Enqueuing data decouples ingestion from processing, so ensuring the main API stays responsive to incoming queries. The queue is administered within the same Azure Resource Group, and additional throughput or partitions can be allocated as sensor volumes increase.

A background process, the Decision Relay Consumer, operates as an Azure WebJob to retrieve messages from the Service Bus queue and implement the crisp-fuzzy decision logic. This worker extracts the relevant rules from the database by correlating the incoming organization identifier with the stored rule sets. It analyzes distinct thresholds or assesses membership functions for fuzzy rules and subsequently documents the chosen actions - alongside any priority or confidence values - in a "Decisions" table. If the volume of incoming data increases, additional processing instances can be created to scale out the worker, allowing the system to manage bigger workloads while preserving timeliness.

Numerous supplementary Azure services enhance this configuration. Azure Key Vault safeguards credentials, including database passwords and Auth0 secrets, preventing the presence of unencrypted secrets in configuration files. Application Insights records logs, performance measurements, and telemetry, enabling administrators to monitor request latencies, response trends, and potential bottlenecks across all services. Collectively, these services offer comprehensive monitoring and enable swift troubleshooting or optimization of the solution.

The system attains significant modularity by organizing each key function into distinct, independently deployable microservices. Each service can be modified, redeployed, or scaled independently without affecting the others. Similarly, the interface with Auth0 provides detailed control over user roles, while the multi-tenant data model in PostgreSQL guarantees separation between enterprises utilizing the same platform. This architecture decision facilitates gradual updates and improves reliability, as failures in one component do not inherently compromise overall operations.

The architecture utilizes the advantages of a cloud-native method to deliver a flexible and secure solution for crisp-fuzzy rule administration. Numerous CPSs - each recognized as a separate entity in Auth0 and the User Management API - can utilize the identical infrastructure for decision-making, acquire or modify domain-specific regulations, and securely convey real-time sensor data for asynchronous assessment.

Relational database

To support multiple CPSs under one shared platform, the system adopts a multi-tenant relational schema. All tenants - referred to as organizations - store data in common tables yet remain isolated through foreign keys that reference each organization’s unique identifier. This approach ensures that rules, decisions, and user accounts for one CPS never overlap or interfere with those of another. Figure 2 (the relational database diagram) provides a graphical overview of the primary tables and their relationships.

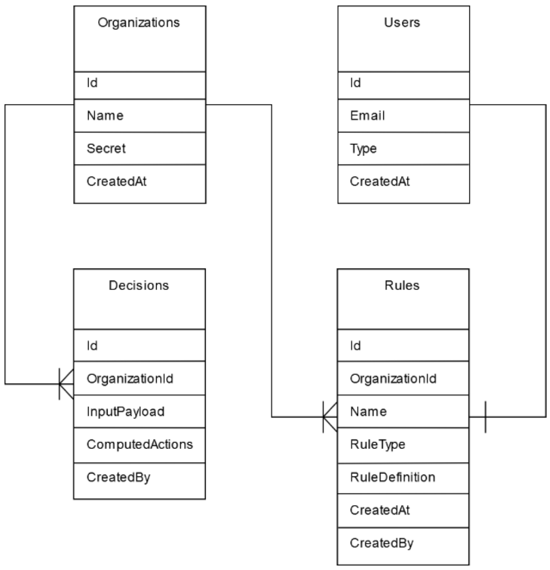

Fig. 2. Relational database diagram

At the center of this design is the Organizations table, whose rows denote distinct CPS domains (e.g., one might represent an agricultural setup, another a manufacturing line). Each organization entry includes an Id - a GUID that doubles as the primary key, a descriptive Name, and a Secret used in M2M authentication. Whenever a CPS device logs in, it presents this pair (OrganizationId and Secret) to retrieve a JWT.

User accounts - whether they belong to Administrators or Experts - reside in the Users table. Each row is linked to an organization via an OrganizationId foreign key. This arrangement ensures an expert user within one organization cannot modify rules or see data belonging to another. Storing both admin and expert roles in the same table, differentiated by a Type field, centralizes user identity management. For instance, an admin might create an expert account for a domain specialist, who will then access the system via Auth0-based credentials and modify only the relevant rule sets.

Crucial to the decision-making process, the Rules table defines how sensor data is interpreted for each organization. A rule can be crisp, fuzzy, or mixed. Internally, the rule definition is stored as a JSON document that captures conditions, thresholds, or membership curves. This JSON structure is accessed whenever the background worker processes queued sensor readings. Since each rule row also references an OrganizationId, a single database can hold multiple rules from disparate domains, ensuring no collisions between the logic of different CPSs.

When the system evaluates sensor data, it produces an action or a set of actions, stored in the Decisions table. Each row corresponds to a single decision event, including references to the originating OrganizationId and the raw input payload. By recording every decision, the system maintains a running log of historical outcomes that can be analyzed or audited later.

By structuring the database in this manner, the system both unifies and isolates data: all CPS domains benefit from the same robust schema and microservice logic, while each domain's integrity is enforced by foreign key constraints. Through this relational design, multi-tenancy becomes seamless, enabling easy onboarding of new CPS organizations and simple expansions to handle additional rule types or functionalities.

CPS interaction workflow

A core requirement of this architecture is facilitating autonomous decision-making for cyber-CPSs while keeping the cloud-based services both scalable and secure. Figure 3 illustrates the high-level communication flow between a CPS (represented as a "Machine"), the system's microservices, and the underlying database. From the CPS perspective, four main steps occur to authenticate, submit data, and retrieve the resulting decisions.

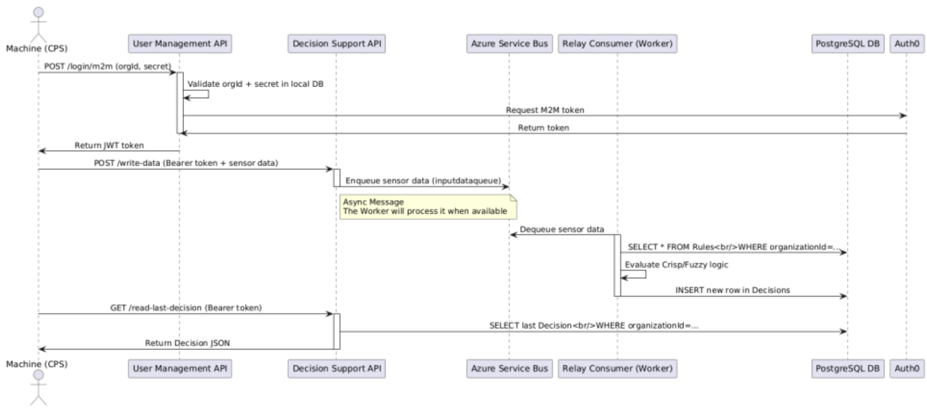

Fig. 3. CPS interaction workflow

In the first step, the CPS device obtains a valid token for M2M communication. It sends a request to the User Management API, providing its OrganizationId and the corresponding Secret. The User Management API validates these credentials - checking the database for a matching organization entry - and, if successful, interacts with Auth0 to fetch a JWT. This JWT encodes the CPS's organization identifier and authorized permissions. After receiving the token, the CPS can prove its identity to other services.

With a valid JWT in hand, the CPS proceeds to the second step: sending sensor data to the Decision Support API. This is done by a POST /write-data call, carrying the token in an Authorization header and a payload of key-value sensor readings (e.g., temperature, humidity, or vibration metrics). The Decision Support API, after confirming that the JWT is valid, defers the intensive logic by placing the sensor data into an Azure Service Bus queue. This asynchronous operation means the API responds quickly, and any surge in sensor submissions can be buffered without overwhelming the service.

Once data lands in the queue, the third step unfolds behind the scenes, as the Relay Consumer, which is a background worker, continuously listens for new messages. Upon dequeuing a message, the worker checks which organization submitted the data, queries the relevant rules from the PostgreSQL database and evaluates whether crisp thresholds or fuzzy membership functions are triggered. Although this stage is essential for determining the recommended actions, the details of crisp-fuzzy computation will be outlined in a later section. For now, it suffices that once the worker completes its evaluation, a new decision record is stored in the "Decisions" table, noting the triggered actions and any relevant priorities.

Finally, the CPS can retrieve the most recent decision in the fourth step by calling GET /read-last-decision on the Decision Support API, again using the bearer token. The API queries the database for the last decision entry associated with the organization's ID and returns a JSON response. This might include an action label such as "ActivateIrrigation," "ReduceSpeed," or another domain-specific command, along with supplemental metadata that allows the CPS to act accordingly.

Experiments

To demonstrate how crisp and fuzzy rules are combined into a single decision process, this section explores a smart garden domain scenario. In this scenario, a single database record contains four sub-rules designed for a smart garden: two crisp definitions that rely on simple threshold checks, and two fuzzy definitions that use membership functions to interpret sensor readings more flexibly. Figure 4 illustrates how these four sub-rules are organized in JSON form, reflecting a typical entry in the rule database. When the system receives new sensor data - for instance, soil moisture, temperature, and light level - it processes that data by loading this single record from the database, then examining each sub-rule within it in sequence.

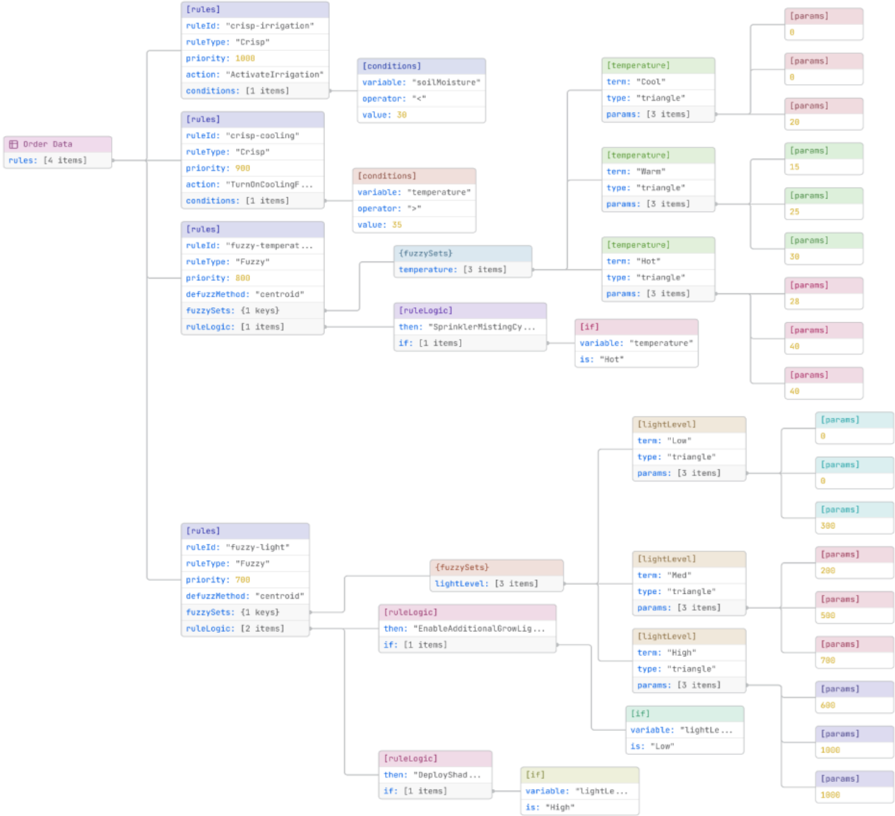

Fig. 4. Mixed crisp-fuzzy rule definition

The Relay Consumer, operating as a background service, first merges and parses all relevant rule documents for the garden's organization. This means it retrieves the JSON record, deserializes the "rules" array, and collects each sub-rule into a working list. For crisp sub-rules, the consumer checks whether the specified conditions match the current sensor data. A rule that instructs "Activate irrigation if soilMoisture < 30" will be triggered only if the sensor reading meets that strict inequality. Another crisp sub-rule might activate cooling fans if the temperature exceeds 35 °C. These checks are direct comparisons, so either the data satisfies the conditions or it does not.

Fuzzy sub-rules, by contrast, require additional interpretation. A sub-rule might define triangular membership sets for "Cool," "Warm," and "Hot," then stipulate that if "temperature is Hot," a "SprinklerMistingCycle" should be triggered. To determine the degree of membership in "Hot," the Relay Consumer applies a linear interpolation formula, computing a value between 0 and 1. Once it has a membership value for each "if" clause, it combines them (in this implementation, using a basic fuzzy AND operation) and deems the sub-rule triggered if the combined membership remains above 0.5. Another fuzzy sub-rule might address "lightLevel," labeling ranges such as "Low," "Med," and "High," then triggering actions like "EnableAdditionalGrowLights" or "DeployShading" if the membership crosses the threshold. Because the fuzzy conditions allow partial truths, the decision logic becomes less abrupt, letting the system react smoothly to intermediate sensor values.

Each triggered rule supplies both an action name (for instance, "TurnOnCoolingFans") and a numerical priority. After all sub-rules are evaluated, the consumer sorts any triggered actions by priority, placing the most crucial items first. In the example shown in Figure 5, the input data might specify a soil moisture of 40, a temperature of 38, and a light level of 250. The crisp irrigation condition, requiring soil moisture below 30, is not met so that action is omitted. The temperature rule with threshold 35 is triggered, pushing "TurnOnCoolingFans" into the final list. At the same time, the fuzzy temperature sub-rule finds "Hot" membership sufficiently high (since 38 °C falls near the upper range), adding "SprinklerMistingCycle" as well. The fuzzy light sub-rules, meanwhile, see that 250 lies between "Low" and "Med," thus failing to exceed the 0.5 membership threshold for either "Low" or "High," leaving those actions untriggered.

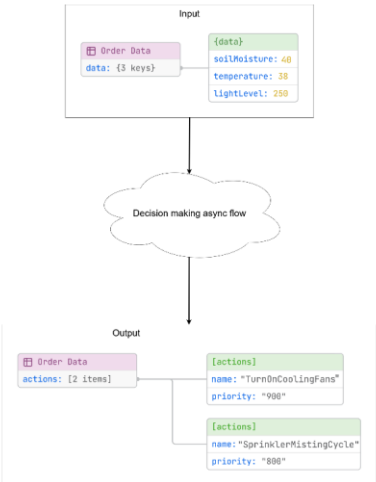

Fig. 5. Input and output for decision relay

Once the consumer finalizes this ordered list, it serializes the outcomes (for example, "TurnOnCoolingFans" with priority 900 and "SprinklerMistingCycle" with priority 800) along with the original sensor data in the "Decisions" table. This decision record captures exactly which conditions were satisfied and how the system responded, and it includes a timestamp for future reference. Whenever the garden controller subsequently queries for the "last decision," the Decision Support API returns this result so that the controller can carry out the recommended actions. By uniting crisp and fuzzy rules within a single record, the system accommodates both straightforward triggers and more adaptive thresholds without requiring multiple separate rule sets, making the environment simpler to administer and more versatile to the variability of real-world measurements.

Conclusions

This work has introduced a cloud-based DSS that employs both crisp thresholds and fuzzy membership functions to facilitate autonomous decision-making in CPSs. Utilizing a microservices architecture, we delineated user authentication, knowledge management, and real-time decision processing, enabling many organizations to utilize the same platform while maintaining the separation of their data and logic. A straightforward smart garden example illustrated how crisp and fuzzy conditions can coexist within a single rule specification, providing CPS devices with exact instructions while maintaining minimal onboard complexity.

Despite the encouraging results, many restrictions must be recognized. The system relies on stable network connectivity - any interruption in the cloud connection may hinder sensor updates or obstruct the prompt acquisition of new decisions. Secondly, due to the asynchronous nature of the DSS rule processing, it may be challenging to ensure very stringent real-time restrictions. Third, fuzzy logic requires subject expertise to establish suitable membership functions, and inaccuracies in these definitions may result in poor or perplexing outputs. Ultimately, if the quantity of organizations or rules increases, maintaining optimal performance may demand meticulous scaling tactics and continuous oversight of workloads. Notwithstanding these limitations, the suggested methodology provides a versatile and sustainable framework for autonomous CPSs to utilize both crisp and fuzzy reasoning within a shared, cloud-based environment.

While the existing solution encompasses the fundamental aspects of multi-tenant crisp-fuzzy decision-making, several upgrades could augment its functionalities. Initially, enhancing the fuzzy logic module with supplementary membership function types, such as trapezoidal or Gaussian, could better accommodate intricate real-world data distributions and facilitate more nuanced decision-making. Secondly, the implementation of a streamlined user interface would facilitate domain specialists in visualizing and modifying rules more intuitively, hence diminishing dependence on direct API calls. Third, enhancing the system's elasticity via automatic scaling policies in Azure would guarantee constant performance during peak sensor input, especially in extensive CPS installations. Another approach entails assessing how machine learning methodologies, such as anomaly detection or reinforcement learning, could enhance the existing rule-based framework, resulting in adaptive tactics that progress with historical data. Ultimately, exploring integration with edge computing may produce a hybrid model, delegating certain logic or pre-processing tasks to local devices while reserving the cloud for intensive computations. By exploring these pathways, the platform could transform into a more adaptable and dependable decision-making framework, accommodating a wider array of autonomous CPS scenarios.

References

1. El-Kady, A. H., Halim, S., El-Halwagi, M. M., & Khan, F. (2023). Analysis of safety and security challenges and opportunities related to cyber-physical systems. Process Safety and Environmental Protection, 173, 384-413.
2. Melnyk, A. O., Zimchenko, B. (2022). A Web-Based Application for Modeling and Deploying Fuzzy Logic-Based Decision Support Systems for Autonomous Cyberphysical Systems (keynote). In IntelITSIS (pp. 23-32).
3. Jain, D. K., Neelakandan, S., Veeramani, T., Bhatia, S., & Memon, F. H. (2022). Design of fuzzy logic based energy management and traffic predictive model for cyber physical systems. Computers and Electrical Engineering, 102, 108135.
4. Zaman, M., Eini, R., Zohrabi, N., & Abdelwahed, S. (2022, September). A Decision Support System for Cyber Physical Systems under Disruptive Events: Smart Building Application. In 2022 IEEE International Smart Cities Conference (ISC2) (pp. 1-7). IEEE.
5. Wang, R., Milisavljevic-Syed, J., Guo, L., Huang, Y., & Wang, G. (2021). Knowledge-based design guidance system for cloud-based decision support in the design of complex engineered systems. Journal of Mechanical Design, 143(7), 072001.
6. Plaza, A. M., Díaz, J., & Pérez, J. (2018). Software architectures for health care cyber-physical systems: A systematic literature review. Journal of software: evolution and process, 30(7), e1930.
7. Cao, K., Hu, S., Shi, Y., Colombo, A. W., Karnouskos, S., & Li, X. (2021). A survey on edge and edge-cloud computing assisted cyber-physical systems. IEEE Transactions on Industrial Informatics, 17(11), 7806-7819.
8. Song, Z., Mishra, A. R., & Saeidi, S. P. (2023). Technological capabilities in the era of the digital economy for integration into cyber-physical systems and the IoT using decision-making approach. Journal of Innovation & Knowledge, 8(2), 100356.
9. Fang, Y., Roofigari-Esfahan, N., & Anumba, C. (2018, January). A knowledge-based cyber-physical system (CPS) architecture for informed decision making in construction. In Construction Research Congress 2018 (pp. 662-672).
10. Hu, C. L., Wang, L., Chen, M. L., & Pei, C. (2024). A real-time interactive decision-making and control framework for complex cyber-physical-human systems. Annual Reviews in Control, 57, 100938.
11. Queiroz, J., Leitão, P., & Oliveira, E. (2022). A fuzzy logic recommendation system to support the design of cloud-edge data analysis in cyber-physical systems. IEEE Open Journal of the Industrial Electronics Society, 3, 174-187.
12. Kumar, N., Vasilakos, A. V., & Rodrigues, J. J. (2017). A multi-tenant cloud-based DC nano grid for self-sustained smart buildings in smart cities. IEEE Communications Magazine, 55(3), 14-21.

Anatoliy Melnyk **Анатолій Мельник**	Vice-Rector for Research and Innovation, Doctor of Technical Sciences, Professor, IT Step University, Lviv, Ukraine e-mail: melnyk_anatoliy@itstep.org https://orcid.org/0000-0002-8981-0530	Проректор з науки та інновацій, доктор технічних наук, професор, IT Степ Університет, Львів, Україна
Bohdan Zimchenko **Богдан Зімченко**	PhD Student of Computer Engineering Department, Lviv National Polytechnic University, Lviv, Ukraine e-mail: bohdan.v.zimchenko@lpnu.ua https://orcid.org/0000-0002-4574-4068	Аспірант кафедри комп'ютерної інженерії, Національний Університет «Львівська політехніка», Львів, Україна