SISS: Semantic Interoperability Support System for the Internet of Things

Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz

***Abstract*—The Internet of Things (IoT) landscape is hindered by a critical challenge: the lack of semantic interoperability among diverse data models. Existing IoT solutions often function as isolated "data silos", impeding the seamless integration of heterogeneous data sources crucial for informed decision-making and streamlined processes. This research addresses this issue by introducing a pioneering solution: the semantic interoperability support system (SISS). SISS is an innovative tool designed to bridge the semantic divide between disparate data models within a common application domain. To address the lack of interoperability between current IoT platforms, devices, and solutions that use native data models, SISS facilitates integration by enabling the generation of gateways or translator components. These components establish mappings between the semantic properties of source and target data models, leveraging advanced semantic analysis and inference techniques. The core principle underpinning SISS is its ability to discern and map the semantic content of data models. Through a meticulous analysis of the temporal and spatial dimensions inherent in the data, SISS establishes meaningful connections. This innovative approach fosters interoperability and enables a deeper understanding of the underlying information, enhancing the potential for data-driven insights. This article delves into the pervasive issue of semantic interoperability in the current IoT paradigm and presents SISS as a transformative solution. By emphasizing its ability to transcend the limitations of existing solutions and its methodology to generate mappings between disparate data models, this research contributes to the achievement of global semantic interoperability in IoT.**

***Index Terms*—Big data, data science, Internet of Things (IoT), machine learning (ML), precision agriculture, semantic interoperability.**

Received 15 January 2025; revised 2 May 2025; accepted 30 May 2025. Date of publication 9 June 2025; date of current version 8 August 2025. This work was supported in part by the DEMETER Project through INDUSTRIAL LEADERSHIP—Leadership in enabling and Industrial Technologies–Information and Communication Technologies (ICT) under Grant 857202; ICT receives support from the European Union's Horizon 2020 Research and Innovation Program, and Italy, Luxembourg, Spain, Germany, Greece, United Kingdom, Norway, Czechia, Montenegro, Belgium, Romania, Ireland, Finland, Serbia, Portugal, Poland, Georgia, and Slovenia.
(Corresponding author: Mario San Emeterio de la Parte.)

The authors are with the Group of Next-Generation Networks and Services, the Departamento de Ingeniería Telemática y Electrónica, and the Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, Universidad Politécnica de Madrid, 28031 Madrid, Spain (e-mail: mario.sanemeterio@upm.es; jf.martinez@upm.es; nestor.lucas@upm.es; vicente.hernandez@upm.es).
Digital Object Identifier 10.1109/JIOT.2025.3577776

I. Introduction

INTERNET of Things (IoT) offers a paradigm populated by static and mobile devices, capable of generating an enormous amount of raw data that, after processing and analysis, support the development of all kinds of intelligent solutions. However, the heterogeneity of devices, as well as the great diversity of existing protocols and data models, require a great effort in the development of intermediation or translation components. The lack of interoperability in IoT is slowing its current evolution, limiting integration between existing platforms and/or solutions.

The concept of Interoperability in IoT is very broad; therefore, to perform an adequate analysis, it must be stratified in levels. The architecture of IoT solutions itself offers the appropriate prism for the specification of the existing IoT interoperability levels. The communication infrastructure, data format and syntax, communication protocols, semantics, and business or legal processes define the bases of the five levels of IoT interoperability: Transport, Syntactic, Semantic, Behavioral, and Legal or Policy Interoperability [1].

Enabling interoperability at the infrastructure, syntax, and data format levels is essential for the transmission of data between IoT solutions and components. However, for complete integration between solutions, effective communication must be enabled, for which a common semantics or understanding is needed. This study focuses on interoperability at the semantic level, as it is the main challenge to provide interoperability to solutions of the modern IoT ecosystem.

IoT devices offer native data models, mostly developed by the manufacturers themselves. To take advantage of the information inherent in the raw data generated by the devices, they must first be integrated into the corresponding platform or solution. In addition, the data models used in communication between components of the same system, platform, or solution are largely specific, which makes it difficult to integrate them with other systems, platforms, or solutions.

To solve the challenge imposed by the lack of semantic interoperability in IoT, the current literature describes four main strategies, 1) the development of standards; 2) the development of translator components or gateways; 3) technologies for ontology alignment (OA); or 4) automation of the translation process (data integration approach) (see Fig. 1).

The development of standard data models and ontologies offers the main strategy to achieve global semantic interoperability. However, this approach does not solve the current need for integration between solutions that have

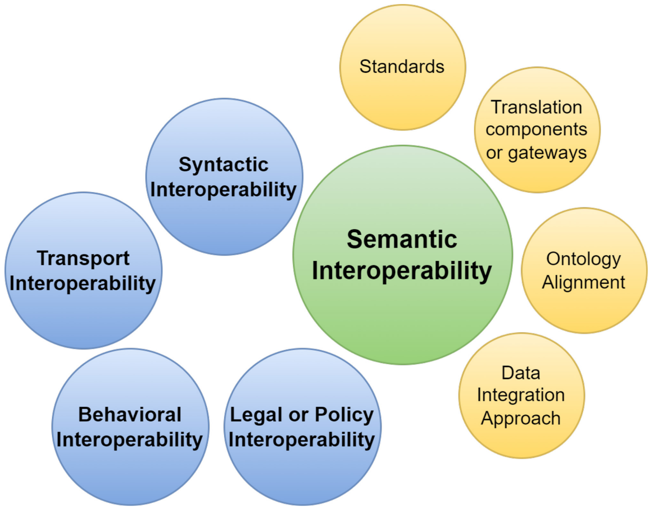


Fig. 1. Interoperability levels and semantic interoperability approaches.

already been developed based on native or specific data models. Additionally, the wide variety of specific application domains requires specific terminology. Ontologies or data models designed with a modular structure are able to cope with the variety of application domains, but require constant development or updating of the solutions implemented on the basis of these models.

The development of translator components or gateways is currently the main strategy for IoT solution developers. However, their implementation requires a previous effort in the study of the source and target models for subsequent translation or mapping, and in the implementation of the component or gateway itself. In addition, these solutions are totally specific, so they require constant updates and analysis.

Technologies for the alignment of ontologies or the automation of the process of mapping between data models show the potential needed to provide IoT scenarios with the necessary semantic interoperability. However, these technologies are not yet mature enough, and in some cases, they pursue the wrong strategies. First, OA technologies represent a complex process and user interaction, lacking the necessary automation in a dynamic paradigm, such as the IoT. Second, many existing approaches are overly reliant on syntactic matching techniques [2], [3]. Third, many alignment methods are designed with static data models in mind, which is a mismatch for the dynamic and evolving nature of IoT ecosystems. Finally, many of the current approaches do not incorporate domain-specific knowledge, which is essential for accurate semantic interpretation in IoT [4].

Due to the current trend in the use of machine learning (ML) models for process automation, some proposals aim to automate the translation between data models through ML-based solutions. The problem lies in the very nature of the technology; the model has to be previously trained and for that, an effort of study and preparation of the specific dataset for training is needed. Generating a dataset formed by equivalent data and modeled under different ontologies or models involves a greater effort than the implementation of the translator component itself. Moreover, due to the variety of existing specific data models, training a model to offer a generic solution is currently an unrealistic strategy.

This article presents a novel proposal, the *Semantic Interoperability Support System* (SISS), designed and developed as a support tool for the generation of the semantic translation middleware necessary for the integration of solutions. This system is capable of automating the analysis and mapping of properties with equivalent semantic content between two data models. This strategy significantly reduces the effort of implementing translator components to provide semantic interoperability to current solutions. The spatio-temporal characteristics inherent in the data generated by IoT devices serve as query parameters for the extraction of semantic information about the data. In addition, the SISS has a REST API to enable its integration into current platforms and a simple GUI to improve accessibility to the tool. The study discusses the capabilities of SISS to nurture a knowledge database, which can be used to train ML models, capable of fully automating the translation process between data models.

The SISS proposed in this article is based on an embryonic and theoretical version published in article "Breaking Down IoT Silos: Semantic Interoperability Support System for the Internet of Things" [5] and presented at the *International Conference on Electrical, Computer, Communications and Mechatronics Engineering* (ICECCME) conference.

To evaluate the performance of the SISS, eight of the standard or most widely accepted ontologies, data models, or vocabularies for data modeling in IoT scenarios and sensor networks have been selected. In addition, a real device from the H2020 DEMETER project [6] has been selected as a data source. With the dataset prepared, the system is subjected to a set of tests to evaluate its ability to establish the correct equivalences between the different properties of the selected models.

Section II presents the current state-of-the-art of interoperability in the Internet of Things. This section defines the layers of interoperability on which this article is based and presents current solutions and challenges that are intended to be solved through the proposal contained in this article. Section III presents the different approaches in which the current proposals are framed to achieve semantic interoperability. Section IV presents the proposed SISS. Section V describes the evaluation of the SISS. For the evaluation, a real device is selected from a precision agriculture scenario, modeled against a selection of eight different data models, and given to the SISS to evaluate its association or mapping capabilities. Section VI draws some conclusions obtained from the design and evaluation of the proposed system.

II. Related Work

The Internet of Things offers a paradigm based on the collection and sensing of data generated by devices deployed in action scenarios, for subsequent analysis, processing, and development of intelligent solutions. However, its mainly distributed architecture and the heterogeneity of the generated data represent one of the main challenges in its development and implementation. Interoperability capabilities between existing platforms and solutions are hampered by the diversity of existing data models [7]. The use of native data models, developed specifically by the device manufacturer/supplier, or designed for specific platforms or projects, results in the generation of *data silos*.

Enabling the cooperation and integration of IoT solutions requires making them interoperable [8]. However, the solutions developed in the current scenario offer resistance at the level of communication, data syntax, interpretability, understanding, and at the legal or political level between organizations. To understand the current state-of-the-art in the search and development of technologies capable of providing interoperability to IoT solutions and platforms, it is necessary to understand the levels at which the concept of interoperability is stratified.

The ISO/IEC 21823-1:2019 standard [1] defines the five levels of interoperability on which this study is based.

1) *Transport interoperability* involves establishing a communication infrastructure that allows data to be sent from one entity to another even if they are connected to different networks. Quality of Service (QoS) requirements, such as timeliness, durability, order, lifetime, failure tolerance, and data delivery are managed at this level.
2) *Syntactic Interoperability* focuses on data exchange in a common format and protocol. At this level, the correct understanding of the meaning of the information provided by the data is not managed.
3) *Semantic Interoperability* focuses on the correct understanding of the information interchanged by all entities. Metadata and shared information models (ontologies) are the main tools for ensuring semantic interoperability.
4) *Behavioral Interoperability* refers to the integration of devices and systems through aligned business processes to enable efficient and effective data exchange, automation, and decision-making.
5) *Legal or Policy Interoperability* allows for information exchange and effective collaboration between organizations that operate with different strategies, legal frameworks, and policies.

Transport interoperability is a major focus of interest due to the very nature of IoT technology and the interconnection between "things". To address the connectivity and bandwidth limitations of IoT scenarios, various network technologies and protocols have been developed. Some of the most widely used include MQTT, Zigbee, 6LoWPAN, or LoRaWAN. Even the use of 5G [9] and its Network Slicing features for latency optimization, bandwidth management, and security are approaches capable of dealing with the limitations of the most complex IoT scenarios.

Several scientific papers and articles discuss systems, technologies, or solutions to achieve transport interoperability in the IoT. Abdelouahid et al. [10] discussed connectivity requirements to improve interoperability between devices that make up the IoT. The book "The Data Distribution Service" by Corsaro and Schmidt [11] provided a set of QoS policies that control the availability of data to domain participants, including durability, lifespan, and history policies. It also provides a comprehensive understanding of DDS, its functionalities, and its importance in ensuring efficient and reliable data exchange in distributed systems. This knowledge is crucial for developing and integrating IoT applications that require transport interoperability.

In [12] Alanazi and Elleithy discussed the design of an efficient QoS routing protocol, focusing on reliability and guarantee of end-to-end delay while conserving energy.

In Speer's Ph.D. thesis [13] the development of fault-tolerant adaptive QoS control algorithms is presented. These algorithms ensure reliable and timely data delivery in wireless sensor networks, a critical aspect for achieving transport interoperability in IoT systems.

Yacchirema and Palau [14] outlined a smart IoT gateway architecture designed to enable transport and syntactic interoperability among heterogeneous devices. This architecture likely focuses on the integration of different communication protocols and data formats to allow for seamless data exchange without necessarily interpreting the meaning of the data.

Ahmed [15] proposed a transparent translator to solve interoperability issues at two layers of an IoT system, including the communication protocol layer. This approach focuses on syntactic interoperability by ensuring that different IoT applications can exchange messages regardless of the messaging protocols used.

The article "IoT Communication Protocols—IoT Data Protocols" [16] provides an overview of IoT data protocols, including MQTT, HTTP, CoAP, DDS, WebSocket, AMQP, XMPP, and OPC UA. These protocols are used for low-level data communication in IoT systems and are crucial to achieving syntactic interoperability.

The current IoT landscape is severely affected by the generation of *data silos* as a result of the development of non-interoperable devices, platforms, and solutions. However, enabling IoT integration and cooperation requires not only an interoperable infrastructure, protocols, and data format or syntax, but also a correct common understanding of the semantics or information contained in the generated and exchanged data. In an environment with countless heterogeneous IoT devices and components, interoperability at the semantic level is a critical challenge. The interest in achieving interoperability at the semantic level is especially motivated and justified by the European data strategy [17], which aims to make Europe a leader in the data-driven society.

Kovacs et al. [18] outlined a system architecture to achieve global semantic interoperability, using international standards, such as oneM2M and the OMA NGSI-9/10. The system architecture offers the potential for global semantic interoperability and leverages established standards and semantic models. However, challenges related to standardization, complexity, and additional performance considerations should be carefully evaluated in the practical implementation of the proposed solution.

Semantic interoperability is the tool to ensure interoperability between IoT platforms. The European Horizon 2020 INTER-IoT initiative [19], [20], proposes a solution to enable interoperability between IoT platforms at all levels or layers, with special attention to semantic interoperability. It is an open-source solution, with the objective of enabling integration between heterogeneous IoT technologies. The goal is to facilitate the discovery, orchestration, and integration of different applications and services from different platforms. For this purpose, the solution offers the *inter platform semantic mediation component* (IPSM) in charge of performing ontology-to-ontology translations.

The Open Data in Agriculture platform [21] offers a proposal focused on data integration to provide interoperability at the semantic and syntactic levels in the domain of precision agriculture. It consists of a Web-based platform.

This article does not provide a detailed literature review of current solutions for enabling IoT interoperability at the behavioral and legal layers, as their achievement depends fundamentally on achieving interoperability at lower layers. However, it is recommended to read [22], which provides a systematic review of international standards for IoT interoperability. In addition, this article describes in more detail the nature and solutions of interoperability at the behavioral and legal levels.

In summary, numerous proposals considered as examples of interoperable IoT platforms are described in the literature. Some of the most notable proposed architectures are FIWARE [23], OpenIoT [24], and OneM2M [25]. Some have been implemented, constituting functional IoT platforms (e.g., FIWARE or OneM2M). These platforms are designed to provide a middleware solution that allows applications to interact with the device layer, enabling interoperability.

However, the solutions presented in the current literature try to define standards or develop standards-based platforms to facilitate interoperability. But what about the current data silos? And for the data models specialized in certain IoT application domains, how can they be integrated or enrich the vocabulary of the new "interoperable platforms for IoT" and their new ontologies and data models?

Current solutions focus on the development of translator components or gateways or on the complex development of technologies for OA. Proposals based on mapping or translation between data models or ontologies are known as data-integration approaches. Automating the process of generating translator components would provide a novel solution to enable semantic interoperability between different existing data models.

A notable strategy for automating the mapping process between various data models, facilitating subsequent translation, is grounded in the application of machine learning (ML) models. The paper titled "An Analysis of Machine Learning-Based Semantic Matchmaking" [26] outlines key methodologies for leveraging ML to automate semantic mapping in the context of Internet of Things (IoT) data models. The study introduces and evaluates three proposed solutions (LEX-DB, W2VEC, and K-MEANS), based on parameters, such as Accuracy, Time to Completion, and Peak Memory.

However, ML-based solutions encounter a significant challenge inherent in the technology itself: the necessity for a training dataset. The generation of a dataset, structured under various data models yet containing equivalent semantic information, poses a comparable or greater level of effort compared to the development of specific translator components. This underscores a crucial consideration in the adoption of ML-based approaches to semantic mapping automation.

Another important approach is the development of technologies or algorithms for OA. The well-known evaluation initiative, OA evaluation initiative (OAEI) [27], aims to evaluate the performance of OA techniques and assesses the strengths and weaknesses of current alignment/matching systems.

In line with this approach, the literature presents proposals capable of performing matching between ontologies, for the integration of context in environments characterized by the diversity and heterogeneity of devices. In [28], the OntoPhil, an ontology-to-ontology matching algorithm developed for the Smart Cities application domain, is presented. This algorithm is evaluated according to the OAEI criteria [27]. However, the proposals expose specific solutions for specific application domains. The absence of generic/adaptive solutions represents one of the main problems of this approach.

Table I presents the different works in the current literature. In addition, the interoperability layer(s) in which each proposal is framed is indicated. The solutions framed in the semantic layer have been classified, analyzing if they present a data integration approach.

To solve the current problems in achieving a next-generation and interoperable IoT ecosystem, this article presents SISS. The proposed system offers a tool to support the process of mapping or translation between data models and aims to serve as a starting point for the full automation of these processes.

III. Approaches to Address Semantic Interoperability

Semantic interoperability in IoT involves ensuring that data exchanged between devices and systems are correctly interpreted and understood. For data to be syntactically compatible, the format and syntax used in communication must coincide, be transformed by middleware, or be compatible. For data to be semantically compatible, there must be a common understanding of their meaning. Without semantic interoperability, there is no effective communication. IoT devices use a variety of data models, naming conventions, units of measurement, or coding schemes, making integration and interpretation difficult.

To solve the current problem derived from the lack of semantic interoperability, current solutions can be classified into four main approaches: 1) the definition of standards; 2) the development of OA or mapping solutions; 3) the development of ML models to automate mapping or translation between different models or ontologies; and 4) approaches based on the integration of data under currently existing data models.

A. Standards

The definition and development of standard data models can significantly alleviate the lack of semantic interoperability in the IoT by providing a common framework for data representation and integration across heterogeneous devices, platforms, and applications.

TABLE I
INTEROPERABILITY IN IoT

Ref	Year	Interoperability Layers	Major Contribution	Data-integration
[10]	2021	Transport	Definition of the connectivity requirements to improve interoperability between IoT devices	N/A
[11]	2012	Transport	Detailed exploration of the DDS protocol, which is fundamental for ensuring efficient, reliable, and real-time data exchange in IoT applications across different networks.	N/A
[12]	2015	Transport	Development of routing protocols that support real-time traffic and reliable data delivery	N/A
[13]	2008	Transport	Develops a hop-by-hop data delivery mechanism and an Adaptive Fault Tolerant Quality of Service Control algorithm	N/A
[14]	2023	Transport, Syntactic	Definition of a Smart IoT Gateway architecture to enable transport and syntactic interoperability of heterogeneos devices	N/A
[15]	2021	Syntactic	Propose a transparent translator to achieve interoperability between the transport layer (protocols) and the syntactic layer through the SSN ontology	N/A
[16]	2022	Transport, Syntactic	Overview of current and more used IoT data protocols for low-level data communication.	N/A
[18]	2016	Semantic	Worldwide semantic interoperability, using international standards such as oneM2M and the OMA NGSI-9/10	N
[20]	2021	Semantic	Support universal semantic interoperability among platforms by simplifying the process of aligning and matching ontologies	Y
[21]	2020	Syntactic, Semantic	Presents the Open Data in Agriculture platform, that consist on a set of components and tools based on semantic web technologies to provide syntactic and semantic interoperability from a data-integration approach	Y
[23]	2019	Semantic	Presents the FIWARE platform. Semantic Interoperability through NGSI-LD standarized by the ETSI and the Semantic Mediation Gateway (SMG) technology for translation between oneM2M and NGSI	Y
[24]	2015	Semantic	Presents the OpenIoT platform. Provides Semantic Interoperability through the use and extension of the standard W3C SSN ontology	N
[25]	2015	Semantic	OneM2M standard. Presents the IoT-O, an ontology to achive semantic interoperability through the alignment of well-known IoT ontologies and standards	N
[28]	2014	Semantic	Presents the OntoPhil, an ontology matching algorithm to garantee the automatic information exchange between agents and the smart cities	N
Our solution	2024	Semantic	Support or automation of the translation between existing data models. Generation of a dataset of equivalences between data models suitable for training ML-based solutions or for defining and updating ontologies or metaontologies.	Y

Standard data models establish a shared vocabulary and semantics for the description of IoT data, ensuring that all stakeholders have a common understanding of the meaning and context of the data. This reduces ambiguity and facilitates communication and collaboration between different entities in the IoT ecosystem.

Standards enable the harmonization of data from heterogeneous sources through the definition of consistent structures, formats, and schemas. This simplifies the effort required for data integration and enables interoperability between disparate devices and systems.

IoT entities adhering to the same data model can exchange data without the need for extensive mapping or transformation, streamlining communication, and collaboration in IoT environments. Furthermore, standard data models facilitate the integration between solutions from different IoT ecosystems and domains, resulting in greater scalability. New devices and data types can be easily accommodated within the existing standard data model, ensuring compatibility and consistency as the IoT ecosystem evolves and grows.

Standard data models can be enriched with semantic annotations, such as new ontologies and vocabularies, to increase interoperability between IoT data. Semantic annotations provide context and meaning to additional data elements, enabling sophisticated processing and reasoning capabilities. In addition, standard models designed from modular structures facilitate the addition of new modules with a specific vocabulary to describe concepts about new application domains. This approach enables the extension of such models without the need to update or vary the systems or solutions that have already been implemented from the core module or other specific modules previously developed.

Fig. 2 illustrates the advantages of the approach based on formalization and development of standard models to enable semantic interoperability in the IoT.

However, this approach also presents some problems. First, the definition and generation of standard data models is a future-focused solution. This means that current IoT devices, solutions, systems, and platforms remain isolated. The integration and cooperation of these components depend on extensive efforts of analysis, adaptation, translation, and mapping to the new standards. Second, several standard data models have now been defined for different IoT application domains [18], [29], however, the literature and current status show that their success depends purely on their adoption by providers and developers. In general, device manufacturers continue to use native or device-specific data models.

B. Ontology Alignment or Matching

Ontology or data model alignment solutions and techniques play a crucial role in achieving semantic interoperability in IoT by facilitating the integration and harmonization of heterogeneous data models. This type of solution makes it possible to identify the equivalence between concepts and terminology of apparently non-interoperable vocabularies.

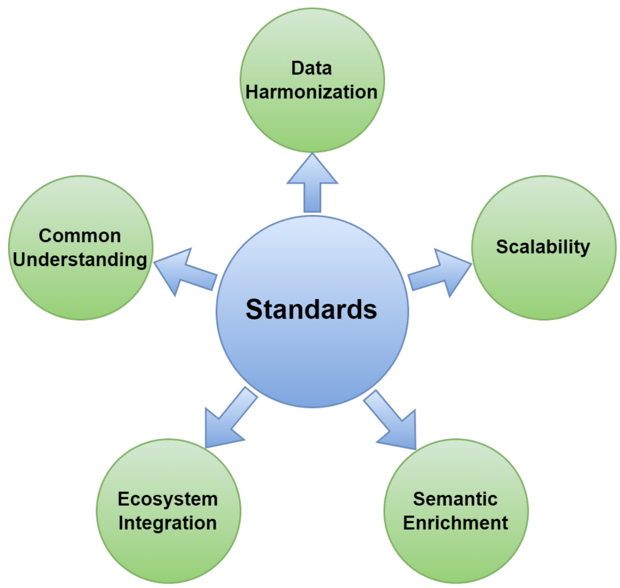


Fig. 2. Advantages of the standard data model-based approach for semantic interoperability in IoT.

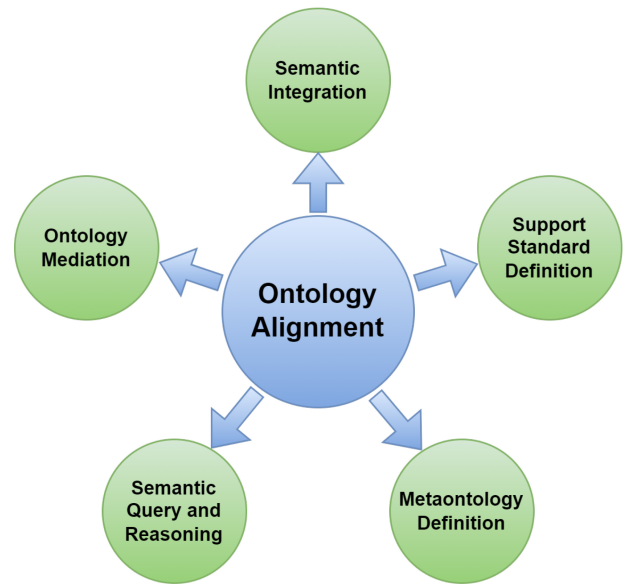


Fig. 3. Advantages of ontology or data model alignment techniques for semantic interoperability in IoT.

OA enables the identification of correspondences between concepts, properties, and relationships from different ontologies or data models. By aligning or mapping ontology elements, such as classes, attributes, or relationships, different IoT systems can establish a common understanding of the generated data.

Ontology mediation techniques provide an abstraction layer that enables interoperability at the semantic level, overcoming current fragmentation. Mediation techniques translate and reconcile differences between ontologies by mapping concepts and relationships. In addition, these techniques allow the location of possible gaps in the different ontologies, enabling the extension of vocabularies and the integration of new concepts.

The alignment of ontologies facilitates semantic integration by enabling the aggregation and fusion of data from heterogeneous IoT sources. Aligning ontology elements resolves potential semantic conflicts and allows IoT platforms to integrate new solutions while preserving rich semantics and meaning of the data.

Through the alignment or mapping of ontologies, data transformation is supported by enabling data conversion between different models. The mapping between the concepts defined in the various ontologies of a specific domain enables the subsequent generation of gateways for the translation of data representations between one ontology and another, ensuring compatibility and consistency between IoT ecosystems.

Furthermore, OA supports the development and adoption of standards to achieve global interoperability in IoT data exchange. By providing mechanisms for the alignment or harmonization of ontologies, these technologies facilitate the creation of standardized ontology environments and broader and more complete vocabularies for cross-platform interoperability. Thanks to these techniques, the main problem of the approach based on the formalization of standard data models can be solved, providing these standards with interoperability among other previously defined models. This approach is the main tool for the definition and development of a *meta-ontology* to describe all concepts, entities, and relationships in IoT.

Finally, OA techniques enable semantic querying and reasoning, allowing IoT systems to perform advanced data analysis and decision making. By resolving semantic ambiguities in data, IoT platforms can infer implicit knowledge, discover relationships, and derive new insights or information from heterogeneous IoT data.

Fig. 3 graphically enumerates the advantages derived from ontology and data model alignment and matching techniques to achieve semantic interoperability.

The main problems or difficulties of this approach lie in the complexity and high costs of the analysis and alignment process between ontologies. Currently, a large number of ontologies are available, even for the same application domain, so the alignment process to achieve global semantic interoperability in IoT is a very extensive work. Furthermore, if a given ontology is updated, extended, or modified, the analysis and alignment process must be repeated. In addition to the problem, the tools developed for OA support in the current literature are not very powerful, so the effort required for data scientist analysis is very high. To solve this problem, an approach that pursues the automation or semi-automation of the process should be followed.

C. Machine Learning Automation

To reduce analysis efforts and the costly process of OA or mapping between data models, one of the most novel approaches consists of the automation of the process through ML models. This approach aims to automate the translation process between data models, generating global interoperability in the IoT domain across different perspectives.

ML models can be trained to automatically learn the semantic relationships between the properties of different data models. This process includes the analysis of the structure and semantics of various data models and the corresponding identification between their elements (e.g., classes, attributes, relationships). ML techniques, such as neural networks, graph embedding, and natural language processing can be employed to learn these mappings based on labeled and unlabeled data.

ML model training can facilitate ontology matching and alignment by learning measures of similarity between concepts, properties, and relationships. Techniques, such as similarity learning, clustering, and classification can be used to compare and match ontological elements. These techniques would enable the automatic generation of mappings between heterogeneous ontologies designed for the same application domain.

Once trained, ML models can automate the process of translating or transforming data between different representations or formats by learning translation functions or algorithms. For example, sequence-to-sequence models, such as RNN and transformers, can be trained to convert data from one ontology or data model into another while preserving semantic consistency and meaning.

Ontology mediation and integration based on learning mediation rules or strategies to reconcile semantic conflicts and differences between ontologies is another novel approach in the use of ML models. Reinforcement learning, probabilistic inference, and rule-based systems can be used to learn mediation policies and algorithms, enabling interoperability between heterogeneous IoT systems.

ML models can also be used to automatically annotate IoT data with semantic metadata or tags based on pattern and association learning. Techniques, such as NER or entity linking can be used to extract semantic information from raw data and enrich it with additional context and meaning.

Finally, ML models can be designed for continuous learning, adapting, and evolving data models and ontologies dynamically in IoT environments. By leveraging online learning, transfer learning, and reinforcement learning techniques, ML models can adapt to changes in data structures, semantics, and usage patterns. This approach would ensure semantic interoperability and long-term scalability by solving the problem of alignment between ontologies that undergo updates or variations.

Fig. 4 shows the main advantages of applying ML models in automating the OA or translation processes between data models to achieve semantic interoperability in IoT.

This new approach arises from advanced technologies in data science. However, for the training of these models, a prior knowledge and training dataset is needed. Therefore, the main problem exposed by this approach lies in the very nature of machine learning, training. For a successful development, a training set on equivalent data that have been modeled under different vocabularies or ontologies is mainly required. However, no such public datasets are currently available and the effort required for their generation exceeds that of the most recurrent solutions in the current landscape, the development of specific translator components. However, data integration-based solutions provide the backbone for generating the datasets (relationships and semantic mapping) needed for future success in developing models capable of fully automating the process, enabling global semantic interoperability in IoT and other domains.

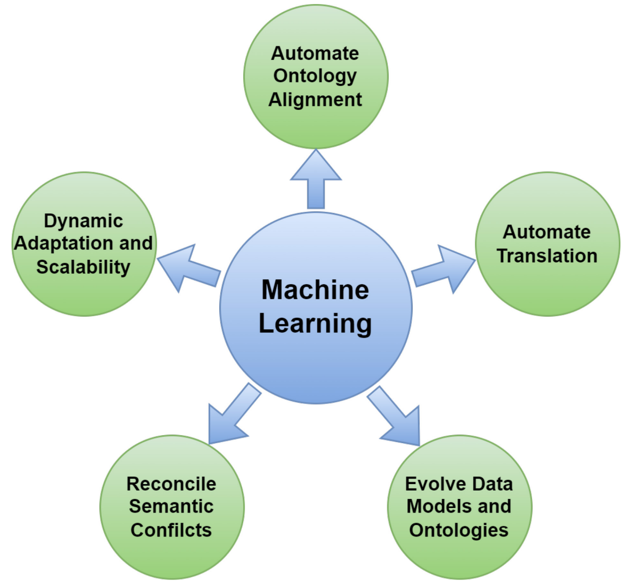


Fig. 4. Advantages of ML models for semantic interoperability in IoT.

D. Data Integration Approach

The approach based on data integration consists of pure translation between data with different formats and syntax, which have been modeled under different vocabularies, but whose semantic content is equivalent. This approach is more primitive than the use of ML models for process automation. However, as previously discussed, the automation of the translation process or the generation of translator components from ML models is not a realistic approach without a prior data set (semantic relations and mappings) previously prepared for training. The data integration approach explores different algorithms and conversion techniques to semi-automate or support the analysis and translation process. In this way, it constitutes the pillar or basis for the generation of the necessary datasets to feed the ML models in their training and turn it into a realistic approach in the future.

The basic principle of the data integration approach is based on the nature of the entities that make up a data model. A given entity is made up of two basic units, the "semantic container" or "key" and the "semantic content" or "value". Taking into account that these two units are basically nothing more than strings or numbers, the aim of the translation is to establish their equivalence among different data models.

Analyzing the different properties of existing IoT data models, it is observed that the main difference between the wide variety of heterogeneous data models lies in the "semantic container". In contrast, "semantic content" has a higher similarity factor, except in the case of conversions between formats, such as between temporal marks (ISO 8601 or Unix Epoch) or spatial marks (Latitude and longitude pairs, asWKT, geohash, etc.). For this reason, algorithms based on string distance metrics present a valuable approach to establish relationships between the properties of data modeled under different vocabularies, models, or ontologies.

There is an extensive list of commonly known algorithms for establishing equivalences and relationships between data or text properties. The main string distance metrics are described below.

1) *Levenshtein Distance:* Measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to convert one string into another.
2) *Jaccard Similarity:* Measures the similarity between two data sets to see which members are compatible and which are dissimilar. Jaccard's similarity is calculated by dividing the number of observations in both sets by the number of observations in each set.
3) *Cosine Distance:* Measures the cosine of the angle between two vectors. It is often used to calculate the similarity between text documents, by representing them as vectors in a high-dimensional space.
4) *LCS:* Similarly to the Levenshtein distance, it measures the number of edits needed to transform one string into another. However, only the number of insertions and deletions, not substitutions, is considered in the calculation of this distance.
5) *Hamming Distance:* Consider only the number of substitutes; therefore, it only applies to strings of the same length.
6) *Damerau-Levenshtein Distance:* Measures the minimum number of edits (insertions, deletions, and substitutions) and allows the swapping of two adjacent characters.
7) *Jaro Distance:* Only considers the transposition of characters.

The development of translation systems between models based on the use of this type of algorithms allows to automate or semi-automate the mapping process between properties of the different existing data models in IoT, without the need of having a previous analysis. Furthermore, this approach constitutes the starting point for the generation of the previously mentioned data sets or equivalences, for the subsequent training of systems or solutions based on the use of ML models.

The approach based on data integration forms the basis for the future automation of OA and translation processes and even the generation of meta-ontologies that enable a complete description of the agents, environment, and relationships in IoT. Therefore, data integration is considered the first step toward achieving global semantic interoperability in the IoT.

IV. SISS Design

In this article, SISS is proposed as a contribution to the achievement of semantic interoperability in IoT. SISS is a system designed to perform the analysis and mapping of equivalences between the different properties of two specific data models. The objective of the design of this system is to offer support in the development of translator components or gateways, enabling communication and integration between different systems, applications, solutions, or platforms. Its development aims to streamline the process of analysis and mapping between properties, generating the necessary support for the achievement of semantic interoperability between the wide variety of heterogeneous data models for the different IoT application domains. The source code of the system is accessible in the zenodo repository [30], with the assigned DOI: https://doi.org/10.5281/zenodo.1139506210.5281/zenodo.11395062.

SISS integrates a knowledge database (KDB) or triplestore, which allows the injection and storage of the relationships between previously validated properties. This database provides the necessary information for the subsequent training and development of ML models. For this reason, this contribution is understood as a starting point for future research and development of systems that allow the process to be fully automated.

A. Architecture

SISS architecture has been designed with respect to two fundamental concepts that characterize the data generated in the IoT scenarios. The first concept is the spatio-temporal semantic nature of the data generated by the different agents that populate the IoT scenarios. The second concept is the very structure of the properties or entities that form the data models, ontologies, and vocabularies with which the IoT data are modeled. This refers to two basic units, "semantic container" and "semantic content" (key-value pairs).

The intrinsic temporal and spatial marks in the data generated by the various heterogeneous devices provide explicit information through the location and date on which the data itself were generated. In addition, IoT scenarios are made up of numerous sensors that generate observations of weather, soil conditions, terrain, or the environment. This basic principle allows extracting certain knowledge through the information provided by the temporal and spatial marks of the data, which can then be used as an aid in tracing relationships between properties or knowledge inference. To take advantage of this information, SISS integrates a component that performs spatio-temporal queries on open datasets of environmental conditions and weather stations. This information allows to establish a set of semantic rules for the classification of data properties.

Fig. 5 presents the architecture of the proposed SISS.

The main workload in the development of translator components or gateways lies in the previous analysis and mapping between the properties of the data models between which the translation will be performed. The objective of the proposed architecture is to semi-automate this process, reducing the necessary effort. SISS consumes as input two documents containing the observations generated by IoT devices or sensors and the information associated with them. Each document is expected to contain *semantically equivalent information* but expressed against a different data model or vocabulary. In this way, the SISS will generate the set of mapping rules for translation between the different models or vocabularies.

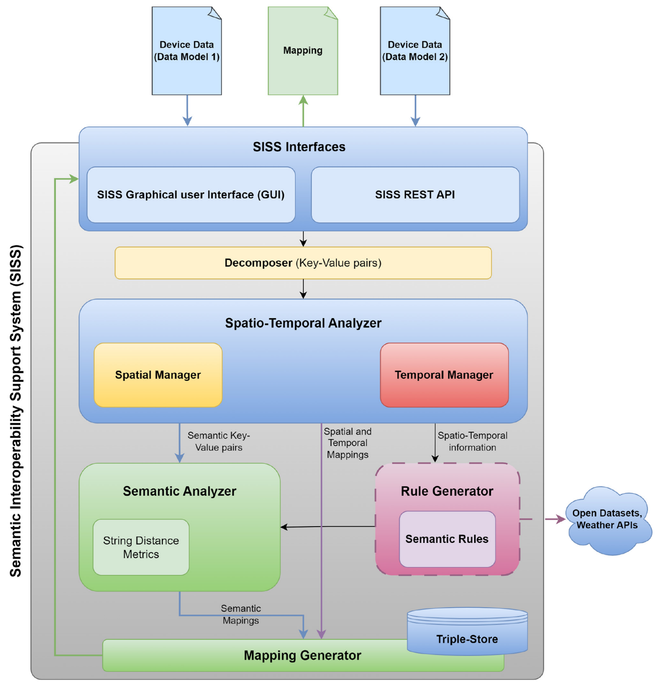


Fig. 5. SISS architecture.

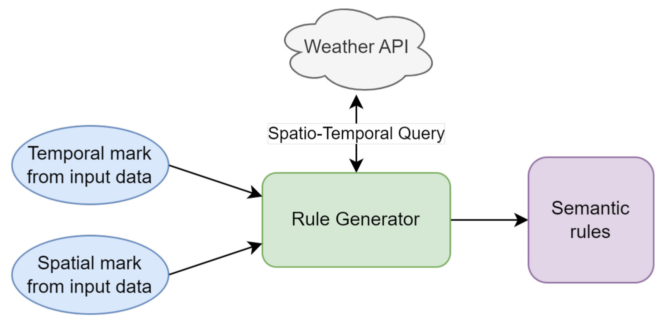


Fig. 6. Spatio-temporal query from rule generator.

For the generation of equivalences between properties and conversion rules between formats, the SISS architecture is composed of four main components: 1) a temporal information manager; 2) a spatial information manager; 3) a semantic rule generator; and 4) a semantic analyzer.

The spatial and temporal information managers are integrated into a component that acts as a wrapper, handling the transmission of temporal and spatial data, along with the remaining semantic information, to the other components.

1) Temporal Manager: This component is responsible for analyzing the key-value pairs of which each of the input documents to the system is composed in search of properties whose content corresponds to timestamps. To do this, the component performs a search among the different data properties and compares them with the different formats for the expression of timestamps or dates.

The formats covered by the component include those defined in the ISO 8601 standard [31] and the Unix Epoch.

Once the key-value pairs corresponding to the time stamps of both documents provided at the system input have been located, it categorizes them according to the format in which they are expressed. In the event that each data model uses a different format, this component generates the algorithm or expression necessary for translation between both formats.

Finally, the *Temporal Manager* component isolates the key-value pairs associated with timestamps from the entire set of pairs within each document. The resulting set of pairs, minus those related to timestamps, is then forwarded to the spatial manager.

2) Spatial Manager: Once the timestamp information has been extracted from the input data, the *Spatial Manager* component processes the remaining pairs to identify geoposition marks. The formats contemplated by the component include latitude–longitude pairs, WKT [32], and geohash.

Once the key-value pairs containing information related to the geopositioning marks have been identified, the component extracts them from the remaining set of pairs. In the case that in each data model the spatial conditions are expressed in a different format, the component facilitates the transformation function between both formats. For example, if the spatial marks in the "source" document are expressed in WKT format and in the "target" document they are presented as two separate properties, latitude and longitude, the component will identify the relationship between these entities and report a composition association.

Finally, the wrapper component, *Spatio-Temporal Analyzer*, sends the spatial and temporal information to the semantic rule generator, the mappings between temporal and spatial marks to the *Mapping Generator* component, and the rest of the key-value pairs to the *Semantic Analyzer* component.

3) Rule Generator: The *Rule Generator* takes advantage of the information provided by the temporal and spatial marks of the data provided at the input of the system. This component queries open data sources and external weather stations to extend the semantic information about the data.

The *Rule Generator* is decoupled from the SISS architecture to facilitate its integration, exclusion from the system process flow, or adaptation to the specific application domain. It should be noted that this component performs queries to open APIs to extract complementary information through the timestamp and geopositioning in which the input data were generated (see Fig. 6). This information allows the generation of semantic rules that facilitate the identification of different data properties.

The objective of this component is to facilitate the identification of properties that contain information about a measurement or observation generated by a given sensor. For example, through a spatio-temporal query to a given meteorological API, the component obtains the minimum and maximum temperature values for a given date and geopositioning. From these values, a rule is generated that identifies that if a certain property from the input documents to the system has a numerical value as semantic content and is within the ranges extracted from the API (minimum and maximum values of temperature), it is presumably a temperature value.

4) Semantic Analyzer: The *Semantic Analyzer* constitutes the core or main component of the SISS. This component is in charge of the semantic analysis of the remaining key-value pairs of the documents provided at the input of the system. Once the spatio-temporal information has been extracted, residing in the data in the form of temporal and geopositioning marks, the *Semantic Analyzer* is in charge of establishing the mapping between the remaining properties. For the analysis and mapping between the semantic content properties, the component uses two main operations, the use of string distance metrics (in this case the Levenshtein distance has been implemented) and the analysis with respect to semantic rules.

Depending on their nature, the semantic rules are either generated by the *Rule Generator* or directly specified in a property file. The SISS itself contains a property file in which certain semantic rules have been specified. These rules are used for the generation of regular expressions to support the determination of commonly specified properties in the data generated by IoT devices, such as units of measurement [33].

Finally, the *Mapping Generator* is in charge of generating the equivalence file between the different spatial, temporal, and semantic properties. This component will deliver the result as a response to the request originated in the interfaces and, if validated, it will store these relations in a KDB or triple-store, in the form of RDF graphs. Additionally, the component will deliver a list with all properties that could not be mapped. In this way, the user or developer can evaluate if the system has not been able to link them, or otherwise evaluate their inclusion in the different vocabularies.

B. Description of SISS Operation

The following are the five high-level steps performed by the SISS to establish the mapping between the data provided as input to the system.

1) The first step consists of decomposing the user-supplied data files into $<$ key $>$: $<$ value $>$ pairs. This process facilitates the subsequent classification and association between the semantic contents (value) and the semantic containers (key) of the input and output models.
2) Second, the pairs are processed to extract the keys whose semantic content specifies the timestamp of the data. For this purpose, a simple search based on regular expressions is used to identify attributes whose semantic content matches the main standards for representing timestamps. Some of the syntaxes or formats covered by the system are the ISO 8601 standard [31] or the Unix Epoch. The *Temporal Manager* identifies the correspondence between the input and output model timestamps and specifies the type of conversion required for the final mapping.
3) Third, the remaining pairs are processed to identify geoposition marks. Some of the formats currently considered are latitude and longitude, WKT, and geohash. For this purpose, the semantic content of the remaining values after temporal identification of the data is processed, looking for continuous values in the range of plausible values in the coordinate definition, or character strings for the geohash. Finally, the *Spatial Manager* generates the conversion function between the spatial formats of the input and output models for the final mapping.
4) Once the temporal and spatial marks have been identified and excluded, a first semantic comparison is performed in search of identical values or values with a high degree of coincidence between the input and output models. The Levenshtein distance is used to calculate the degree of coincidence. After this treatment, two files will be generated, one with the direct relations traced between the input and output data, together with the keys and the associated syntax for each model. Additionally, a file will be generated with the <key>:<value> pairs whose linkage has not been established at this stage ("differences").
5) In the last step, the mapping of indirect equivalences between the remaining pairs of the previous process is performed. To establish their relationships, a set of semantic rules will be used, the objective of which is the classification of values according to the content and semantic relationships between attributes.

C. Semantic Analyzer: String Distance Metric

The main tool of SISS to establish the relationships between the different semantic properties of the two data sources introduced at the input of the system consists of the use of the string distance metric "Levenshtein Distance".

The Levenshtein distance between two strings of given lengths (lev(a, b)), is expressed according to

$$\operatorname{lev}(a,b)=\begin{cases}|a| & \text{if } |b|=0\\ |b| & \text{if } |a|=0\\ \operatorname{lev}(\operatorname{tail}(a),\operatorname{tail}(b)) & \text{if } \operatorname{head}(a)=\operatorname{head}(b)\\ \quad\operatorname{lev}(\operatorname{tail}(a),b) & \\ 1+\min \operatorname{lev}(a,\operatorname{tail}(b)) & \text{otherwise.}\\ \quad\operatorname{lev}(\operatorname{tail}(a),\operatorname{tail}(b)) & \end{cases} \tag{1}$$

In this equation.

1) a and b are two given strings.
2) $|x|$ is the length of a given string x.
3) *tail* of a given string x, corresponds to the same string without its first character.
4) *head* of a given string x, corresponds to the first character of x.

To determine whether one property is equivalent to another, the *Semantic Analyzer* calculates the Levenshtein distance between the semantic contents of the different properties of the input data to the system. If the distance is less than 15% of the sum of the lengths of the values (semantic content) of both properties, the relationship between them is established [see (2)]

$$p_a \equiv p_b \quad \text{if} \quad \operatorname{lev}(v_a, v_b) \leq (|v_a| + |v_b|) * 0.15. \tag{2}$$

In this equation:

1) p_a and p_b are two properties determined from two different data models.
2) v_x is the value or semantic content of a given property x.
3) $|v_x|$ is the length of the value or semantic content of a given property x.

4) $lev(v_a, v_b)$ is the Levenshtein distance between the values or semantic content of two given properties a and b.

To clarify the use of these equations in the validation of the system presented in Section V, the following example is described.

Two certain data models A and B (M_A and M_B) include a specific property for the description of the device identifier (Did_A and Did_B). Based on the vocabularies of the data models selected for the evaluation in Section V, suppose that these properties are "*Device*": "*urn*:*demeter*:*Gamelby*:*environmentalObservations*:*UPM*:*weatherStation*:*weatherStationUPM*″ for Did_A, and "*resourceId*″:"*urn*:*afc*:*AS*01:*environmentalObservations*:*UPM*:*weatherStation*:*weatherStationUPM*″ for Did_B. SISS calculates the Levenshtein distance between the values or semantic content of both properties ($lev(v_{Did_A}, v_{Did_B})$), taking into account (1) and obtaining the value 14, as the distance between the two strings. The *Semantic Analyzer* then calculates the value corresponding to 15% of the sum of the lengths of both strings, obtaining $(|v_a| + |v_b|) * 0.15 = 23.55$. In this case, the value obtained in the calculation of the Levenshtein distance is smaller, so the system will identify that the semantic content of the *Device* and *resourceId* properties of the M_A and M_B models are equivalent to each other.

Additionally, this component verifies whether the literality of the semantic content of a given property is part of the semantic content of another property in the opposite data model. This considers situations where one property is nested within another, or where combining the semantic content of multiple properties in one model yields the semantic content of another property in the alternative model.

D. Example of Rule Generator Operation

To illustrate the functioning of the *Rule Generator* component, a specific real example is presented, on an IoT farm located in the Ylivieska region, Finland, within the framework of the European Research Project AFarCloud [34], [35].

Through the SISS interface, two documents have been submitted in which the observations generated by a given device are displayed. These observations contain measurements of climatological conditions, such as ambient temperature and humidity. These documents contain semantically equivalent information; however, they have been generated by different devices from different providers. For this reason, the data have been modeled with different vocabularies or against different data models or ontologies. The mapping between properties needs to be established to develop a component that allows translation between these models, providing semantic interoperability between them.

The *Temporal Manager* and *Spatial Manager* components have detected the temporal and geoposition marks in the input data and provided them to the *Rule Generator* component. In this particular scenario, this component is connected to the open API of the *Finnish Meteorological Institute* (FMI) [36]. Thanks to the open API of the FMI, the *Rule Generator* is able to execute queries to know the observations generated by the different meteorological stations in Finland. Through a query with the spatio-temporal information extracted from the SISS input data, this component extracts the different measurements generated, at the specified instant, by the meteorological stations adjacent to the established geoposition. Fig. 7 shows the information extracted for this specific example.


```
{
  "coord" : {
    "lon" : 24.506916,
    "lat" : 64.056475
  },
  "weather" : [ {
    "id" : 804,
    "main" : "Clouds",
    "description" : "overcast clouds"
  } ],
  "base" : "stations",
  "main" : {
    "temp" : 274.75,
    "feels_like" : 272.12,
    "temp_min" : 271.75,
    "temp_max" : 275.45,
    "pressure" : 1017,
    "humidity" : 89,
    "sea_level" : 1017,
    "grnd_level" : 1014
  },
  "visibility" : 10000,
  "wind" : {
    "speed" : 2.4,
    "deg" : 72,
    "gust" : 4.24
  },
  "clouds" : {
    "all" : 99
  },"dt" : 1710236975,
  "sys" : {
    "country" : "SE",
    "sunrise" : 1710220688,
    "sunset" : 1710262212
  },
  "timezone" : 3600,
  "id" : 2712988,
  "name" : "Ylivieska"
}
```


Fig. 7. Meteorological information from the API of the Finnish meteorological institute.

With this information, the component knows the temperature and humidity values, among others, between which these observations oscillate. From this information, the component generates a series of semantic rules, in which it specifies, for example, that if a certain numerical value is between the ranges "271.75" and "275.45", it may be a temperature value, and consequently the unit of measurement will be Kelvin degrees. Finally, these semantic rules are given to the semantic analysis component to facilitate the identification of these properties in the input data and subsequently generate the mapping between them.

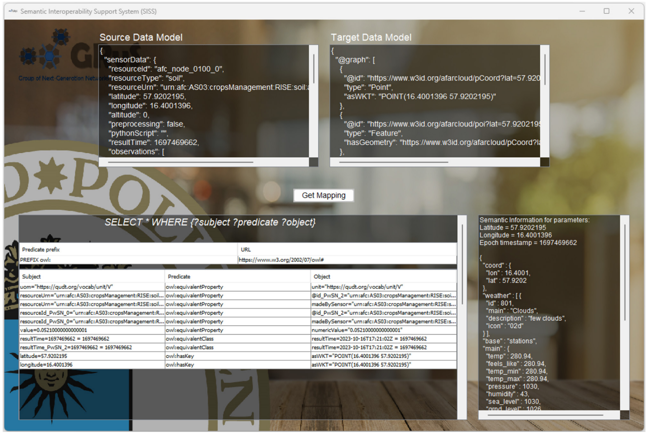


Fig. 8. SISS: GUI.

It should be noted that this is a specific case that has been used to evaluate the capabilities of this component within the framework of the AFarCloud project. Of course, for any other domain or country the *Rule Generator* has to be previously adapted. For this reason, in the SISS architecture, this component is kept decoupled, to facilitate its adaptation or update.

E. Accessibility

The main objective in the design of SISS is to enable semantic interoperability between IoT data models, reducing the effort and cost of analysis and development. For this reason, improving accessibility to the tool is a derived objective, reducing the time needed to understand its operation, resulting in a useful tool for the developer or data scientist.

To achieve this goal, two types of interfaces are defined in the tool design. First, the system integrates a graphical user interface (GUI), to allow its use as a desktop or standalone application. Second, a properly documented REST-type interface is provided to facilitate the integration and deployment of the tool on other platforms, applications, or specific solutions.

To facilitate the customization and extension of the integrated system interfaces, the source code has been released as an open-access project. As detailed in Section IV, the source code is available for consultation and download in the corresponding Zenodo repository [30].

1) Graphical User Interface: The GUI designed and integrated in the SISS offers the possibility of deploying the system as a desktop application. This enables its use as a support tool in the analysis or alignment of data models or ontologies.

It should be taken into account that SISS has been conceived as a support tool in the development of translator components to achieve semantic interoperability. For this reason, facilitating its use as a standalone application, without the need to integrate it into other platforms or solutions, enables support in the analysis process.

Fig. 8 illustrates the look and feel of the SISS GUI. It can be seen that it is a very simple interface oriented toward a quick and easy understanding of the tool.

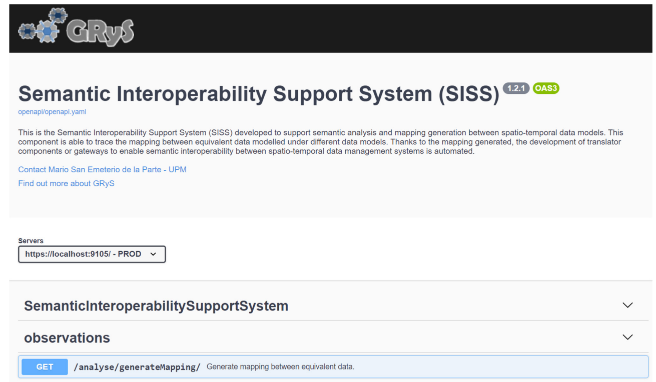


Fig. 9. SISS: REST API swagger documentation.

2) REST API: The SISS provides a REST API interface to facilitate its deployment as a Web resource, as well as the integration of the service into other applications, platforms, or solutions. This REST API has a graphical interface in which the documentation of the specific method for the generation of mapping between two messages in JSON or JSON-LD formats is exposed. This interface or documentation has been developed with *Swagger* [37] and allows testing and execution of the mapping operation, exposed as a method of type *GET*.

Fig. 9 shows the documentation of the REST interface, accessible from any browser.

The REST API provides a single operation to resolve or obtain the mapping between the properties of two JSON entered by the user. This interface will only check that the syntax of both JSONs is correct and will execute the request to the SISS. In addition, the documentation directly provides two examples of observations generated by IoT devices modeled under the specification of the AIM [38] and the spatio-temporal semantic data model proposed in [39] for the Precision Agriculture (PA) application domain.

Fig. 10 shows the documentation of the operation provided by the REST API of the SISS. It is a request of type *GET*, exposed through the path/analyze/generateMapping/.

As can be seen in the documentation, the *GET* method consists of only two parameters corresponding to the two documents between which the mapping is to be performed. Moreover, both parameters are mandatory to be able to perform the operation. Table II details the *GET* method exposed by the REST API, as well as the HTTP codes configured for the possible responses to the operation.

F. System Scalability Considerations

Due to the specific nature of the ontology or information model alignment process, the concept of scalability is not directly applicable to the present system. The system's primary objective is to provide a tool for establishing correspondences between properties and concepts, thereby facilitating the subsequent development of translation components. This alignment process is not continuous, as the system does not function as a real-time data translator between different representation languages.

TABLE II
SISS REST API OPERATION TO GENERATE MAPPINGS BETWEEN THE PROPERTIES OF TWO JSONS

GET /analyze/generateMapping/	Generates the mapping between the properties of two given JSONs. It exposes all non-equivalent properties and those formed by combination of more than one source or target property.
Query parameters:	
sourceJSON, type: JSON; **Required**	One of the JSON between which the mapping is to be performed. It can be a JSON or JSON-LD.
targetJSON, type: JSON; **Required**	One of the JSON between which the mapping is to be performed. It can be a JSON or JSON-LD.
Response codes:	
200:	"Successful operation"
405:	"Invalid input: not JSON-compliant"
415:	"Invalid input: not a JSON"
5XX:	"Unexpected error"

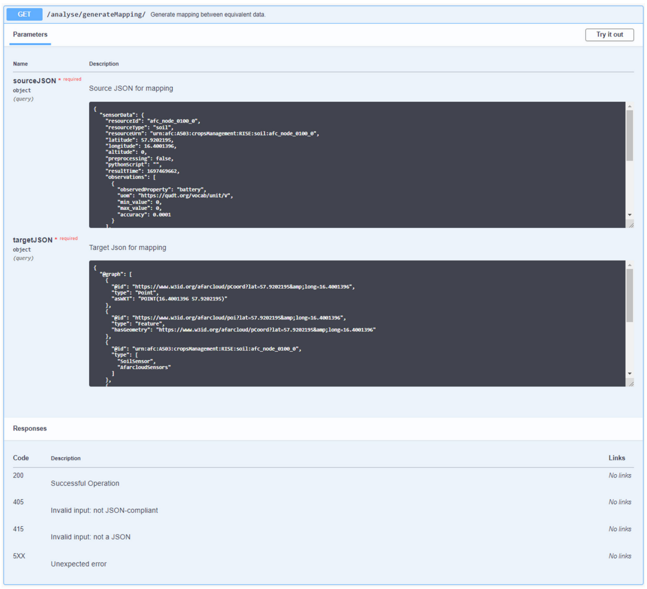


Fig. 10. SISS: GET /analyze/generateMapping.

Nevertheless, the system's design is grounded in concurrent programming principles, enabling it to efficiently manage multiple simultaneous requests. This architecture ensures the system's ability to perform effectively under concurrent usage scenarios, enhancing its adaptability and operational efficiency within the specific requirements of its application domain.

V. SEMANTIC INTEROPERABILITY IN PRECISION AGRICULTURE FOR EVALUATION OF SISS

There are currently numerous application domains for IoT technology, such as Smart Cities, Industry 4.0, eHealth, and precision agriculture, among others. These application domains have a common component, sensor networks, as the main source of raw data and knowledge after processing and analysis. The nature of the data generated by the devices deployed presents information in three dimensions.

1) *Temporal:* By means of a timestamp or a date, the device reports the instant of time when the data were generated or, alternatively, the instant when they were sent to the platform.
2) *Spatial:* Geoposition of the device at the time of data collection. For both static and mobile devices, geopositioning offers a particularly interesting component for further processing, analysis, and inference of the generated raw data.
3) *Semantic:* This dimension comprises the data and any additional related information, such as units of measurement, device identifiers, type, service, or scenario to which the device belongs. Additionally, the semantic dimension includes any information related to the data, the device, and its relationships with other devices, agents, or environment.

To evaluate *SISS* against the needs and problems of a real environment, a consistent evaluation process has been designed. First, real devices, deployed in the scenarios of the European H2020 project DEMETER [6], have been selected. Second, the most representative ontologies for sensor networks and the IoT have been selected by modeling the data generated by the devices. The sample generated for the evaluation is composed of data with equivalent spatial, temporal, and semantic information (it is the same data, modeled under different vocabularies). Finally, the SISS interface is fed with the different generated data, and its ability to map and establish the relationships between the properties of the different data models is evaluated.

The device used in the experimentation is a weather station. It is provided by the *Universidad Politécnica de Madrid* (UPM) and is used to collect environmental observations. This station incorporates two sensors: one to measure ambient temperature and the other to measure relative humidity. At the time the temperature and humidity observations were acquired, the device was located in the region of Gamleby, Sweden. However, to anonymize the data, the georeferenced point with a latitude of 57.91199875 and a longitude of 16.40600014 will be used. The date and time at which the measurements are generated is December 20, 2023 at 17:22:22 UTC.

A. Selected Data Models and Ontologies

A total of eight data models and ontologies have been selected for the evaluation of SISS. These models include the most well-known and widespread standards and vocabularies in the IoT domain. Among them, two information models specific to the application domain of Precision Agriculture have been included. First, the AIM information model [38]

```json
{
  "@context": {
    "ssn": "http://www.w3.org/ns/ssn/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "geo": "http://www.w3.org/2003/01/geo/wgs84_pos#",
    "time": "http://www.w3.org/2006/time#",
    "qu": "http://purl.org/NET/ssnx/qu/qu#"
  },
  "@type": "ssn:Observation",
  "ssn:observedBy": {
    "@type": "ssn:Sensor",
    "ssn:hasManufacturer": "UPM",
    "ssn:hasModel": "weatherStation",
    "ssn:hasID": "weatherStationUPM"
  },
  "ssn:observedProperty": [
    {"@type": "ssn:Property", "ssn:hasName": "air_temperature", "ssn:hasUnit": "qu:DegreeCelsius"},
    {"@type": "ssn:Property", "ssn:hasName": "air_humidity", "ssn:hasUnit": "qu:Percent"}
  ],
  "ssn:observationResult": [
    {"@type": "ssn:SensorOutput",
      "ssn:hasValue": {"@type": "xsd:float", "@value": 14 }
    },
    {"@type": "ssn:SensorOutput",
      "ssn:hasValue": {"@type": "xsd:float", "@value": 50}
    }
  ],
  "ssn:resultTime": {"@type": "time:Instant","time:inXSDDateTime": "2023-12-20T17:22:22Z"},
  "geo:location": {
    "@type": "geo:Point",
    "geo:lat": 57.91199875,
    "geo:long": 16.40600014
    }
}
```


Fig. 11. Device WeatherUPM modeleded under SSN ontology.


```json
{
  "@context": {
    "iot-lite": "http://purl.org/iot/vocab/lite#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "geo": "http://www.w3.org/2003/01/geo/wgs84_pos#",
    "time": "http://www.w3.org/2006/time#",
    "qu": "http://purl.org/NET/ssnx/qu/qu#"
  },
  "@type": "iot-lite:Observation",
  "iot-lite:madeBySensor": {"@type": "iot-lite:Sensor","iot-lite:manufacturer": "UPM","iot-lite:model": "weatherStation","iot-lite:id": "weatherStationUPM"},
  "iot-lite:hasResult": [
    {
      "@type": "iot-lite:Result",
      "iot-lite:value": {"@type": "xsd:float","@value": 14},
      "iot-lite:observedProperty": {
        "@type": "iot-lite:Property",
        "iot-lite:name": "air_temperature",
        "iot-lite:hasUnit": {"@id": "qu:DegreeCelsius"}
      }
    },
    {
      "@type": "iot-lite:Result",
      "iot-lite:value": {"@type": "xsd:float","@value": 50},
      "iot-lite:observedProperty": {
        "@type": "iot-lite:Property",
        "iot-lite:name": "air_humidity",
        "iot-lite:hasUnit": {"@id": "qu:Percent"}
      }
    }
  ],
  "iot-lite:resultTime": {"@type": "time:Instant","time:inXSDDateTime": "2023-12-20T17:22:22Z"},
  "iot-lite:location": {"@type": "geo:Point","geo:lat": 57.91199875,"geo:long": 16.40600014}
}
```


Fig. 12. Device WeatherUPM modeleded under IoT-Lite Ontology.

developed in the DEMETER project. Second, the spatio-temporal semantic data model integrated in the European research project AFarCloud [39].

In the following sections, each of the selected models and ontologies are presented, and the purpose of their design and the capabilities of each one are described. Finally, a simple example of the modeling of the data generated by a *Weather Station* device on the observations of an ambient temperature sensor is presented. These data contain information on spatial, temporal, and semantic dimensions. In this way, it will be evaluated if the SISS is able to extract the temporal and geopositioning marks against the diversity of formats and syntax exposed by the different models. In addition, the system is expected to be able to link the different properties that expose equivalent semantic content between the different models.

B. Semantic Sensor Network Ontology

The semantic sensor network (SSN) ontology is design for the description of sensors and the observations they make, the involved procedures, the studied features of interest, the samples used, and the observed properties, as well as actuators. It provides a common framework for describing and integrating sensor data across different domains and applications. The SSN ontology is modularized to support the judicious use of the “just enough” ontology for specific applications, including satellite imagery, scientific monitoring, infrastructure, citizen observers, and the Web of Things [40], [41]. The SSN ontology has also been extended to incorporate additional requirements to describe observations, and the extensions are defined in a new RDF namespace [42].

The data generated by the device under experimentation, modelled with the terminology and parameters defined in the SSN ontology, is presented in Fig. 11.

C. IoT-Lite Ontology

The IoT-Lite ontology is a lightweight ontology designed to represent Internet of Things (IoT) resources, entities or agents, and services [43], [44]. It is an instantiation of the SSN ontology and is designed to be integrated with other ontologies, such as the SAO ontology [45]. To enable discovery and interoperability between heterogeneous IoT platforms, IoT-Lite is designed to be used with a quantitative taxonomy, such as qu-taxo, or m3-lite [46]. IoT-Lite reduces the complexity of other IoT models, as it is limited to the description of the main IoT concepts and is designed to be extended with other models that increase its expressiveness. This ontology describes concepts with respect to three main classes: 1) Objects; 2) Systems or resources; and 3) services.

The IoT-Lite ontology is available under the W3C Document License and the Creative Commons Attribution 3.0 license. It is an ontology still under development and is subject to change. IoT-Lite is based on the SSN ontology to describe concepts and relationships between IoT agents [47].

Fig. 12 shows the data from the experimental device modelled according to the IoT-Lite ontology definition.

```json
{
  "@context": "http://iotschema.org/",
  "@type": "DataObservation",
  "sensor": {
    "@type": "DataSensor",
    "manufacturer": "UPM",
    "model": "weatherStation",
    "id": "weatherStationUPM"
  },
  "observation": [{
      "@type": "Property",
      "name": "air_temperature",
      "value": 14,
      "unitCode": "CEL"
    },
    {
      "@type": "Property",
      "name": "air_humidity",
      "value": 50,
      "unitCode": "P1"
    }],
  "resultTime": "2023-12-20T17:22:22Z",
  "location": {
    "@type": "Place",
    "geo": {
      "@type": "GeoCoordinates",
      "latitude": 57.91199875,
      "longitude": 16.40600014
    }
  }
}
```


Fig. 13. Device WeatherUPM modeleded under Schema.org Extension for IoT ontology.

D. Schema.org Extension for IoT

Schema.org is a collaborative and community activity, with the aim of creating, maintaining, and promoting schemas for structured Internet data [48]. Schema.org provides a vocabulary that supports representation under the RDFa, Microdata, and JSON-LD syntaxes for the markup of Web pages, emails, and others. This vocabulary allows for the description of entities, relationships between entities, and actions. The Schema.org vocabulary can be extended through a well-documented extension model.

The IoTSchema extension has been developed as an initiative to extend the Schema.org vocabulary to describe and model IoT services and devices. The IoTSchema extension consists of a repository of semantic definitions for the description of connected *things* [49]. This repository is publicly accessible and open [50]. It aims to establish a bridge layer between device ecosystems and semantic Web technologies. This model normalizes OCF and oneM2M device definitions and reuses properties and relationships from other ontologies and models, such as SSN, SOSA, SAREF, QUDT.

The goal of IoTSchema is to provide device vendors and platform providers with a common vocabulary to enable semantic interoperability between platforms and devices. In addition, experts are expected to use IoTSchema to create models specific to their application domains.

The IoTSchema extension is an ongoing effort with contributions from a community of developers and experts, organized by the Schema Extensions for IoT Community Group, which is part of the World Wide Web Consortium (W3C) [51]. Fig. 13 represents the data generated by the experimental device, modeled with the vocabulary of the IoTSchema extension.

E. SAREF Ontology

The smart applications reference (SAREF) [52], [53] ontology is a semantic reference framework designed to facilitate interoperability between smart applications and devices in the context of IoT. SAREF provides a set of classes, properties, and semantic relationships to describe in a standardized way common concepts used in smart applications, such as sensors, actuators, home devices, smart buildings, smart cities, and more.

The main objective of SAREF is to establish a common and coherent vocabulary that allows IoT developers, manufacturers, and service providers to understand and share data in a consistent and accurate way, regardless of the specific devices, platforms, or applications involved. To this end, the ontology defines various semantic classes and properties that represent relevant entities and attributes in the IoT domain.

SAREF is designed to be compatible with other relevant standards and ontologies, such as SSN (Section V-B), SSN extension ontology (SSN-EXT) [54], and others. This facilitates integration and interoperability between different initiatives and projects using specific ontologies and data models.

Some of the key concepts contemplated by SAREF includes,

1) *Devices and Services:* SAREF defines classes and properties to represent physical devices, services, and associated functions.
2) *Observed Properties and Measurements:* SAREF provides a specific vocabulary for the description of properties observed by sensors, as well as the associated units of measurement.
3) *Location and Context:* SAREF includes classes and properties to represent the geospatial location of devices and other relevant contextual aspects.
4) *Events and Actions:* SAREF enables the description of events and actions that can occur in the context of smart devices and IoT applications.

Fig. 14 represents the information generated by the experimental device in the vocabulary, structure, and specification of the SAREF ontology.

F. AFarCloud Model

The agri-food sector is at risk due to the high demand for food from a growing world population. The agricultural sector must increase production and reduce its impact on the environment. With the aim of solving this problem, the H2020 - Aggregate Farming in the Cloud (AFarCloud) project offers a distributed platform for autonomous agriculture, enabling the integration of intelligent agricultural systems in real-time [35].

Communication between the various devices and components that participate in the AFarCloud intelligent platform communicates under the vocabulary defined in the data model developed for the project. The data model emphasizes spatio-temporal aspects, lightweight syntax design, and data structuring to reduce processing times and increase performance. It also involves mechanisms for making sensors and IoT devices interoperable with existing agri-solutions and federating data and services between different platforms. The

```json
{
  "@context": {
    "saref": "https://w3id.org/saref#",
    "qu": "http://purl.org/NET/ssnx/qu/qu#"
  },
  "@type": "saref:WeatherObservation",
  "saref:providedBy": {
    "@type": "saref:WeatherStation",
    "saref:manufacturerName": "UPM",
    "saref:hasLocation": {
      "@type": "saref:Location",
      "saref:latitude": 57.91199875,
      "saref:longitude": 16.40600014
    }
  },
  "saref:observedProperty": [
    {
      "@type": "saref:Temperature",
      "qu:hasUnit":
        "http://purl.org/NET/ssnx/qu/degreeCelsius",
      "saref:hasValue": 14
    },
    {
      "@type": "saref:Humidity",
      "qu:hasUnit": "http://purl.org/NET/ssnx/qu/percent",
      "saref:hasValue": 50
    }
  ],
  "saref:observationTime": "2023-12-20T17:22:22Z"
}
```

Fig. 14. Device WeatherUPM modeleded under SAREF ontology.

```json
{
  "resourceId": "urn:afc:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM",
  "location": {
    "latitude": 57.91199875,
    "longitude": 16.40600014,
    "altitude": 0.0
  },
  "observedProperty": "air_temperature",
  "resultTime": 1703092942,
  "result": {
    "value": 14,
    "uom": "http://qudt.org/vocab/unit/DEG_C"
  },
  "observedProperty": "air_humidity",
  "resultTime": 1703092942,
  "result": {
    "value": 50,
    "uom": "http://qudt.org/vocab/unit/PERCENT"
  },
  "sequenceNumber": 0
}
```

Fig. 15. Device WeatherUPM modeleded under the AFarCloud Ontology.

AFarCloud data model is used as a source of structured and formatted data.

The AFarCloud data model and ontology are integral to the project's goal of enabling efficient data management and interoperability within the agricultural IoT project landscape, ultimately contributing to improved decision-making and operational processes in precision agriculture.

The data generated by the experimental device modeled under the AFarCloud ontology are shown in Fig. 15.

G. NGSI-LD

NGSI-LD, an acronym for "next generation service interfaces - linked data", comprises an information model and an API designed to manage, publish, query, and subscribe to contextual information in a standardized way. It facilitates the open exchange and sharing of structured information between different systems and services. The NGSI-LD information model represents context information as entities composed of properties and relationships with other entities. It is based on property graphs with semantics formally defined using resource description framework (RDF) and the Semantic Web framework. This information model can be serialized using javascript object notation for linked data (JSON-LD).

The NGSI-LD ontology is a cross-domain ontology that defines the temporal keys or the characteristics of the system composition of entities. It defines a small number of types, but is open to any type defined by users, enabling a flexible and extensible information model. The ontology includes sets of primitive concepts related to the temporal dimension, mobility, system states, and system composition.

NGSI-LD has been standardized by the European Telecommunications Standards Institute (ETSI) and aims to enable the use of various vocabularies, taxonomies, thesaurus, and external ontologies in an NGSI-LD context. Provides recommendations on how to link external data models in NGSI-LD, taking into account the graph-based metamodel and the cross-domain ontology [55].

The NGSI-LD data model includes basic concepts, such as entities, relationships, and properties. Entities are informative representations of physical or virtual resources. Each entity is uniquely identified by a URI. Relationships describe links between subjects (entities, properties, and other relationships) and objects (entities). Properties associate characteristics with entities or relationships [56].

The NGSI-LD API is defined using the JSON-LD specification to promote interoperability at the syntactic level. The JSON-LD *@context* is used to define terms and abbreviations, allowing developers to use short attribute names while ensuring that they can be extended to full URIs for universal understanding, providing interoperability at the semantic level [57].

NGSI-LD is designed to allow the user to reuse domain-specific ontologies in modeling, although these ontologies are not part of the NGSI-LD specification. The guidelines for modeling with NGSI-LD suggest how to model a domain-specific system, process, or environment, and how to couple entity instances with types/classes using relationships and properties.

In general, NGSI-LD provides a structured and standardized way of representing and exchanging contextual information, using the principles of linked data and enabling interoperability between different systems and domains. Fig. 16 shows the data generated by the experimental device, modeled according to the specifications described in the *Guidelines for Modelling with NGSI-LD* [58] document.

H. FIWARE (SmartDataModels)

The FIWARE *Smart Data Models* initiative is a cooperative global program aimed at standardizing data models compatible with the NGSI-LD specifications. This initiative is led by the FIWARE Foundation, TM Forum, india urban data exchange (IUDX), and open & agile smart cities (OASCs) [59].

A Smart Data Model consists of four elements: 1) the schema, or technical representation of the data model that

```json
{
  "@context": [
    "https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context-v1.3.jsonld"
  ],
  "id": "urn:ngsi-ld:WeatherObserved:weatherStationUPM",
  "type": "WeatherObserved",
  "dataProvider": {
    "type": "Organization",
    "id": "urn:ngsi-ld:Organization:UPM"
  },
  "dateObserved": {
    "type": "Property",
    "value": {"@type": "DateTime","@value": "2023-12-20T17:22:22Z"}
  },
  "location": {
    "type": "GeoProperty",
    "value": {"type": "Point","coordinates": [16.40600014, 57.91199875]}
    },
  "temperature": {
    "type": "Property",
    "value": 14,
    "unitCode": "CEL"
  },
  "humidity": {
    "type": "Property",
    "value": 50,
    "unitCode": "percent"
  }
}
```


Fig. 16. Device WeatherUPM modeleded under NGSI-LD.

defines the technical data types and structure; 2) the specification of a document written for human readers; 3) a URI containing a URL with the basic attribute data or the entity it identifies; and 4) a set of example payloads for the NGSIv2 and NGSI-LD versions.

The data models are grouped by themes, each of which is constituted by a git repository [60]. These themes belong to one or more application domains, representing different industrial sectors. So far, some of the domains collected in the initiative comprise Smart Cities, Smart Agrifood, Smart Water, Smart Energy, Smart Sensoring, Smart Health, Cross sector, etc. The life cycle of the data models includes three stages 1) Official; 2) Harmonization; and 3) Incubated.

The Smart Data Models initiative constitutes an important approach to break down vertical data silos within organizations by enabling cross-organizational data exchange within data spaces. It provides an asset for developers looking for means to ensure interoperability between different solutions and platforms. In addition, the initiative also incorporates existing ontologies into its data models. For example, the European Union's Railway Agency has published an ontology for railroads, which has been incorporated into Smart Data Models.

The initiative provides a set of tools and resources to make it easier for developers to use and contribute to data models. For example, it offers an online editor to create new data models [61], guidelines for adopting intelligent data models in projects [62], and a service to generate an @*context* based on external ontologies [63].

All data models are public and copyright-free, so users have the right to use, modify, and share them freely.

Fig. 17 shows the data generated by the experimental device, modeled according to the specifications of the


```json
{
  "id": "Sweden-WeatherObserved-weatherStationUPM-Gamleby-2023-12-20T17:22:22.00Z",
  "type": "WeatherObserved",
  "address": {
    "addressLocality": "Gamleby",
    "addressCountry": "SE"
  },
  "dataProvider": "UPM",
  "dateObserved": "2023-12-20T17:22:22.00Z",
  "location": {
    "type": "Point",
    "coordinates":
    [16.40600014, 57.91199875]
  },
  "relativeHumidity": 50,
  "source": "http://www.aemet.es",
  "temperature": 14,
  "refDevice": "weatherStationUPM-0A3478",
}
```


Fig. 17. Device WeatherUPM modeleded under FIWARE smart data models ontology.

"WeatherObserved" data model [64], which belongs to the SmartDataModels model set.

I. Agriculture Information Model

The agriculture information model (AIM) is a data model and ontology to describe the different sensors, devices, agents, and processes involved in smart farming solutions. The AIM provides the basis for semantic interoperability between smart farming solutions and platforms. The AIM has been developed within the framework of the H2020 DEMETER project [6].

AIM has been designed using a modular and layered approach. This approach allows profile and/or extend well-known generic and agriculture-related ontologies, such as those published by the open geospatial consortium (OGC). The OGC intends to create an agriculture information model standards working group (AIM SWG) to develop, publish, and maintain AIM to support the interoperability of agricultural applications [65].

DEMETER's AIM architecture is open and will be available to all interested parties [66]. This model constitutes the common vocabulary of the DEMETER project that provides the basis for semantic interoperability between smart farming solutions. It encompasses the integration of multiple data sources considering legacy systems, open data, geographic, and satellite information to provide an open and interoperable data integration model.

Fig. 18 represents the observations generated by the experimental device, modeled according to the AIM ontology specification for the description of observations generated by a given sensor/device [66].

J. Evaluation

For its evaluation, the system is subjected to a set of tests to align or map the selected data models. Each test consists of the evaluation of the association capabilities between equivalent properties, expressed under the data models selected in this work. In each test, the SISS is executed with two of the examples shown in the Figs. 11–18. Finally, the percentage of *linked* properties versus properties that the system has not been able to associate is calculated.

```
{   "@graph": [{
        "@id": "http://www.w3id.org/demeter/pCoord?lat=57.91199875&amp;long=16.40600014",
        "type": "Point",
        "asWKT": "POINT(57.91199875 16.40600014)"
    }, {"@id": "urn:demeter:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM",
        "type": ["weatherStation"]
    }, {"@id": "urn:demeter:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM:air_temperature-1709043696415",
        "type": "Observation",
        "hasFeatureOfInterest": "http://www.w3id.org/demeter/poi?lat=57.91199875&amp;long=16.40600014",
        "hasResult": "urn:demeter:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM:air_temperature-1709043696415/q1",
        "madeBySensor": "urn:demeter:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM",
        "observedProperty": "http://www.w3id.org/demeter/air_temperature",
        "resultTime": "2023-12-20T17:22:22Z"
    }, {"@id": "urn:demeter:AS01:environmentalObservations:UPM:weatherStation:weatherStationUPM:air_temperature-1709043696415/q1",
        "type": "QuantityValue",
        "identifier": "weatherStationUPM:air_temperature-1709043696415/q1",
        "numericValue": "14",
        "unit": "http://qudt.org/vocab/unit/DEG_C"
      }],"@context": ["https://w3id.org/demeter/agri-context.jsonld"]
}
```


Fig. 18. Device WeatherUPM modeleded under the agriculture information model.

To simplify the execution of the evaluation scenario tests and taking into account that the system offers a REST API to facilitate its deployment and integration, a small script has been developed to execute the requests. Thus, the first step in the evaluation consists of deploying the SISS as a Web resource, accessible through its REST API. At this point, the programmed script executes the different requests to the interface in the form of *HTTP GET* requests (*GET /analyze/generateMapping/* see Table II).

The body of the response message provided by the system contains the relationships between the different properties in the form of RDF graphs. An example of the relationships in the form of RDF graphs provided by the system is represented in Table III. The relationships or mappings established by the system have been analyzed manually, validating whether the relationships are correct or not.

It should be taken into account that the vocabularies of the different models have specific capabilities. That is to say, some properties of a certain model, which express specific semantic information, have no equivalence in other models since their vocabulary does not contemplate the modeling of such information. This type of property will be excluded from the success percentage calculation in the mapping performed by the system, except in the case that the system has identified them erroneously.

The percentage of success of SISS ($R(\%)$) in mapping between properties is specified by the number of properties that the system has been able to map (L_{siss} – SISS Links), divided by the number of total properties that actually have a relationship between the two models (L_{tot} – Total Links). This calculation is represented in

$$R(\%) = \frac{L_{\text{siss}}}{L_{\text{tot}}}. \tag{3}$$

Additionally, this calculation considers cases where, for example, the system identifies that a certain property in the source model is *equivalent* to two properties in the target model, but in reality one of the two properties is not *equivalent*, but contains the key (*hasKey*). In such cases, the correct relationship is excluded from the calculation, penalizing the system's response.

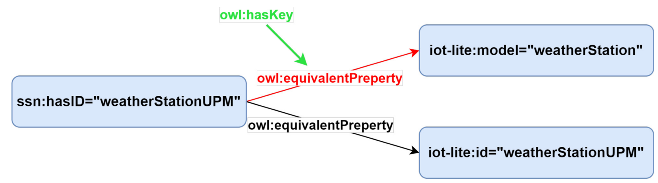


Fig. 19. Example of incorrect relationship between properties.

Fig. 19 graphically represents an example between the SSN and IoT-Lite models. In this example, the system has detected a relationship between the property *ssn:hasID = "weatherStationUPM"* of the message modeled under the SSN specification, with the properties *iot-lite:model = "weatherStation"* and *iot-lite:id = "weatherStationUPM"*, but only one of the relationships is correct. In this case, the correct relationship will also be taken as erroneous, impairing the calculation of the system's hit percentage.

Table IV shows the evaluation results. Each cell represents the percentage of success in the automatic mapping performed by the system, concerning the different models and ontologies presented above.

The evaluation results show that the proposed system is able to automatically map 78.08% of the equivalent properties present in the most representative IoT information models. This result demonstrates the effectiveness of the SISS in reducing the cost and complexity of the OA process. The system's ability to automate a significant portion of the process enables users to focus on more critical and strategic tasks, thereby mitigating the manual burden inherent to such activities.

Furthermore, the SISS incorporates a graphical user interface, presented in Section IV-E, designed to maximize usability and transparency in the mapping process. This interface allows users to immediately visualize the results generated by the system, including both successful alignments and properties that have not been automatically mapped or lack equivalence in the provided models. This approach enhances the user experience and provides a mechanism for manually validating and adjusting the results, ensuring a higher level of accuracy in cases where the system does not achieve a fully automated mapping.

Identification of unmapped properties or those without equivalences is a critical feature to enable full semantic interoperability. Through this functionality, users can iteratively enrich or update the ontologies, contributing to the

TABLE III
RDF MAPPING GRAPH

SELECT * WHERE {?subject ?predicate ?object}		
PREFIX owl:		https://www.w3.org/2002/07/owl#
subject	**predicate**	**object**
asWKT="POINT(16.4001396 57.9202195)"	owl:hasKey	latitude=57.9202195 longitude=16.4001396
resultTime=$2023-10-16T17:21:02Z=$ 1697469662	owl:equivalentClass	resultTime=1697469662
numericValue="0.0521"	owl:equivalentPreperty	value=0.0521
unit="https://qudt.org/vocab/unit/V"	owl:equivalentPreperty	uom="https://qudt.org/vocab/unit/V"
madeBySensor="$urn : afc : AS03 : cropsManagement : RISE : soil : afc_node$_0100_0"	owl:equivalentPreperty	resourceUrn="$urn : afc : AS03 : cropsManagement : RISE : soil : afc_node$_0100_0"
@id="urn:afc:AS03:cropsManagement: RISE:soil:afc_node_0100_0"	owl:hasKey	resourceId=afc_node_0100_0 resourceType=soil

TABLE IV
EVALUATION RESULTS FROM SISS

Percentage of associated properties	SSN	IoT-Lite	Schema.org for IoT	SAREF	AFarCloud	NGSI-LD	SDM $FIWARE$	AIM $DEMETER$
SSN	X	86.67%	90%	100%	55.56%	71.43%	85.71%	72.73%
IoT-Lite	86.67%	X	83.33%	91.67%	63.64%	42.86%	71.43%	80%
Schema.org for IoT	90%	83.33%	X	77.78%	81.82%	87.5%	100%	80%
SAREF	100%	91.67%	77.78%	X	55.56%	66.67%	85.71%	77.78%
AFarCloud	55.56%	63.64%	81.82%	55.56%	X	66.67%	71.43%	100%
NGSI-LD	71.43%	42.86%	87.5%	66.67%	66.67%	X	87.5%	77.78%
SDM $FIWARE$	85.71%	71.43%	100%	85.71%	71.43%	87.5%	X	75%
AIM $DEMETER$	72.73%	80%	80%	77.78%	100%	77.78%	75%	X

continuous refinement of the system and enhancing the quality of alignments in future implementations.

However, the results of the evaluation reveal significant differences in the percentage of equivalences obtained for the different information models. The main causes of these differences are described below.

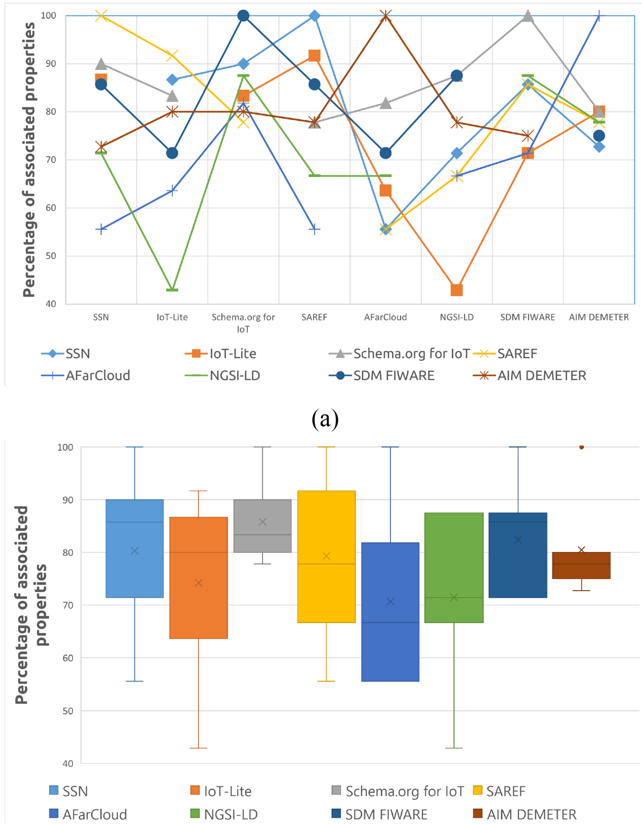
As a basis for analysis, Fig. 20 graphically presents the results. This figure consists of two subfigures.

The graph Fig. 20(a) illustrates the success rate achieved during the experimentation for each information model relative to the other models. Note that the connecting lines between results exhibit a discontinuity at points representing the equivalence of an information model with itself.

Graph Fig. 20(b) presents the Box & Whiskers diagram of the results. This diagram facilitates the identification of the most representative experiments or experimental limits. Specifically, the AFarCloud model exhibited lower alignment rates compared to the other models. In contrast, the AIM model achieved the most constrained results, showing minimal variation between its mapping outcomes with other models. Finally, IoT-lite demonstrated the greatest variability in the experimental results.

The AFarCloud information model was specifically developed for the precision agriculture domain. In addition, its syntactic design focuses on efficient resource usage, enabling data management systems (DMSs) to support real-time execution capabilities. For this reason, it adopts a simple JSON-based syntax. Furthermore, the description of observations generated from this model prioritizes the characterization of associated spatial and temporal information. By facilitating the extraction of spatiotemporal data, this design feature enhances the functionality of the embedded rule engine in the proposed system and supports subsequent semantic reasoning.

During the experimentation, a lower performance in the alignment of the AFarCloud model with the SSN and SAREF models has been detected. Both models employ a more verbose syntax than AFarCloud, based on linked data. However, this syntactic difference did not prevent the SISS from correctly identifying relationships between properties containing spatial and temporal information in both experiments. Furthermore,



(a)

(b)

Fig. 20. Representative graphs of the evaluation results. (a) Evaluation results in line graph. (b) Box and Whiskers of evaluation results.

the experiments conducted with these models demonstrate that the SISS effectively handles models with linked data-oriented syntax without encountering difficulties.

However, when examining the properties that were not correctly mapped between the SSN and SAREF models compared to the other models studied, a common issue is identified: the semantic content of the properties "*ssn:hasUnit*" and "*qu:hasUnit*" defined in the SSN and SAREF vocabularies, respectively. These models employ a specific format to define measurement units that differs from the standards and formats initially considered in the SISS knowledge database. Specifically, "*qu:DegreeCelsius*" in SSN and "*http://purl.org/NET/ssnx/qu/degreeCelsius*" in SAREF represent the syntax used to denote degrees Celsius as the unit of measurement for a given observation. A similar pattern is observed with other measurement units whose semantic content and structure differ significantly from those stored in the system's knowledge database, which primarily includes standards and the most widely used and accepted formats. Therefore, to improve the performance and effectiveness of the SISS, its knowledge database must be expanded to enhance identification capabilities, supporting more comprehensive semantic reasoning and mapping.

However, experimentation with the AFarCloud model continues to show a lower average automatic mapping rate compared to other models. This is attributed to the system's inability to correctly decompose properties embedded within the resource identifier defined in the AFarCloud model (the "resourceId" property). The core issue lies in the use of opaque identifiers, such as UUIDs, which do not present an exploitable structure but only provide a unique identification of the resource. However, in specific cases, as in the AFarCloud model, instead of opaque identifiers, complex composite identifiers are used. These complex identifiers are generated from a composition of various entity-specific or observation-specific properties. As future work, the implementation of a dedicated module for the identification and management of unique identifiers is proposed.

In contrast to the results obtained for the AFarCloud model, the AIM model exhibits minimal variation in the experimental results. This stability is due to the AIM model being designed and implemented based on the leading IoT standards, which are also evaluated in this study. Consequently, AIM reuses properties, structures, formats, and syntax from other models. This design feature translates into greater consistency in the experimental outcomes.

Following the analysis of the evaluation results, the current limitations of the proposed system have been identified. Table V provides a list of these limitations and describes potential system improvements as guidance for future work.

In conclusion, the results of this evaluation position the SISS as a robust tool for addressing one of the key challenges in the IoT domain: efficient semantic integration across heterogeneous models. The evaluation results have facilitated an in-depth analysis of the system's current capabilities and limitations, providing a clear foundation for future work and research in the field. The system demonstrates itself as a practical solution that balances automation and manual control, representing a significant advancement in the state-of-the-art for semantic interoperability.

TABLE V
CURRENT LIMITATIONS AND FUTURE WORK

Detected Limitations	**Future work**
Syntax and format variability in the representation of measurement units for various observations.	Extension of the SISS knowledge database.
Processing challenges in the management of opaque identifiers and complex composite identifiers, for subsequent alignment.	Development and integration of a specific module for the identification and processing of opaque identifiers and complex composite identifiers.

VI. CONCLUSION

The absence of semantic interoperability in IoT is the main culprit for a fragmented ecosystem. The progress of this technology is currently slowed by the generation of "data silos" and the difficulty of integrating current intelligent systems, solutions, and platforms. To solve this problem, this article discusses and analyzes the main proposals to achieve interoperability in the Internet of Things. After analyzing the literature, four main approaches have been identified:

1) the development of standards; 2) OA technologies; 3) the automation of the mapping or translation process between models with machine learning techniques; and 4) the approach based on data integration. However, these approaches present numerous challenges and pending issues. To address current problems, a novel support system, the *SISS*, has been proposed and evaluated in this article.

For the evaluation of the proposed system, eight of the most well-known and widely used data models, ontologies, or vocabularies for sensor networks and IoT have been selected. The evaluation of SISS has provided promising results in supporting semantic interoperability in the Internet of Things (IoT) domain. By analyzing messages generated under different ontologies or data models, SISS has demonstrated the ability to map equivalences between properties whose semantic content is equivalent, using metrics, such as *Levenshtein Distance*, regular expressions, and semantic rules.

The results of the experiment reveal that SISS has been able to map more than 78% of the equivalent properties between different models, indicating its effectiveness in identifying semantic relationships between heterogeneous data. This achievement is critical in facilitating interoperability between IoT systems and devices, enabling efficient and accurate data integration and sharing.

The proposed system incorporates a GUI and a well-documented REST-type interface, which improves the accessibility of the system and allows it to be integrated into other solutions or platforms. The design of these interfaces has been focused on facilitating access and understanding of the tool, reducing the costly process of analysis and mapping between the different concepts and properties that make up the data models and ontologies in IoT.

However, the evaluation also highlighted some limitations and challenges for the system. Some representations of units of measure contemplated by existing models or ontologies were observed to be not present in the SISS knowledge database, making it difficult to identify relationships in these cases. This limitation suggests the need to expand the knowledge database of the system to cover a greater diversity of measurement units used in the IoT context.

In addition, SISS faces difficulties when dealing with complex composite identifiers that incorporate altered information or combine multiple properties from other models. These cases represent a challenge for the system, as the complexity of the structure makes it difficult to identify semantic relationships automatically. In such situations, human intervention or increased system sophistication is required to adequately resolve these scenarios. To solve the processing of identifiers, it is proposed, as a future work of the proposal, the implementation and integration of a specific module for their processing.

In summary, SISS offers a promising tool for achieving global semantic interoperability in IoT by providing an automated mechanism for mapping and aligning ontologies and data models. Although it faces challenges in the coverage of units of measurement and the complexity of some data structures, these can be addressed by continuously expanding the knowledge base and improving semantic analysis algorithms. In addition, the triplestore integrated into the system stores the different relationships between the analyzed data models. This database provides the necessary resources for training ML models to fully automate the alignment between data models or ontologies and the generation of translator components. For this reason, the SISS is considered to provide the basis for future research and the achievement of a defragmented and interoperable IoT ecosystem.

Author Contributions

The contributions of the authors according to the CRediT taxonomy are as follows: Conceptualization, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Data curation, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Formal analysis, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Funding acquisition, José-Fernán Martínez-Ortega; Investigation, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Methodology, Mario San Emeterio de la Parte and Vicente Hernández Díaz; Project administration, José-Fernán Martínez-Ortega and Mario San Emeterio de la Parte; Resources, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Supervision, Mario San Emeterio de la Parte and José-Fernán Martínez-Ortega; Validation, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Visualization, Mario San Emeterio de la Parte and Néstor Lucas Martínez; Writing—original draft, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz; Writing—review and editing, Mario San Emeterio de la Parte, José-Fernán Martínez-Ortega, Néstor Lucas Martínez, and Vicente Hernández Díaz. All authors read and approved the final manuscript.

References

[1] *Internet of Things (IoT)—Interoperability for IoT Systems*, ISO/IEC Standard 21823-1:2019, 2019. [Online]. Available: https://www.iso.org/standard/71885.html

[2] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 158–176, Jan. 2013. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/ 6104044?casa_token=WLfXpm5X8J8AAAAA:vAClZfJ-p4Zr- z9rMLxgh6QU3O9QxLtymAZf2BGnbZ6jd7sOSdvyprunBdFuhBjpQ5vL8oeo0S0

[3] N. F. Noy and M. A. Musen, "The PROMPT suite: Interactive tools for ontology merging and mapping," *Int. J. Human-Comput. Stud.*, vol. 59, no. 6, pp. 983–1024, Dec. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S107158190300137X

[4] J. Euzenat and P. Shvaiko, *Ontology Matching*. Heidelberg, Germany: Springer, 2013. [Online]. Available: https://link.springer.com/10.1007/978-3-642-38721-0

[5] M. S. E. De La Parte, J.-F. Martínez-Ortega, N. L. Martínez, and V. H. Díaz, "Breaking down IoT Silos: Semantic interoperability support system for the Internet of Things," in *Proc. 3rd Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Jul. 2023, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10253024

[6] “Building an interoperable, data-driven, innovative and sustainable European agri-food sector | DEMETER project | fact sheet | H2020,” 2024. [Online]. Available: https://cordis.europa.eu/project/id/857202

[7] M. Jacoby, A. Antonić, K. Kreiner, R. Łapacz, and J. Pielorz, “Semantic interoperability as key to IoT platform federation,” in *Interoperability and Open-Source Solutions for the Internet of Things* (Lecture Notes in Computer Science), I. Podnar Žarko, A. Broering, S. Soursos, and M. Serrano, Eds. Cham, Switzerland: Springer Int., 2017,pp. 3–19.

[8] M. Noura, M. Atiquzzaman, and M. Gaedke, “Interoperability in Internet of Things: Taxonomies and open challenges,” *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 796–809, Jun. 2019. [Online]. Available: https://doi.org/10.1007/s11036-018-1089-9

[9] S. Li, L. D. Xu, and S. Zhao, “5G Internet of Things: A survey,” *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X18300037

[10] R. A. Abdelouahid, O. Debauche, and A. Marzak, “Internet of Things: A new interoperable IoT platform. Application to a smart building,” *Procedia Comput. Sci.*, vol. 191, pp. 511–517, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921014666

[11] A. Corsaro and D. C. Schmidt, “The data distribution service–the communication middleware fabric for scalable and extensible systems-of-systems,” in *System of Systems*, vol. 19. London, U.K.: IntechOpen, 2012, [Online]. Available: https://web.archive.org/web/20190502181322id_/https://cdn.intechopen.com/pdfs/30416.pdf

[12] A. Alanazi and K. Elleithy, “Real-time QoS routing protocols in wireless multimedia sensor networks: Study and analysis,” *Sensors*, vol. 15, no. 9, pp. 22209–22233, Sep. 2015. [Online]. Available: https://www.mdpi.com/1424-8220/15/9/22209

[13] N. A. P. Speer, “Design and analysis of adaptive fault tolerant QoS control algorithms for query processing in wireless sensor networks,” Ph.D dissertation, Comput. Sci. Appl., Virginia Tech, Blacksburg, VA, USA, 2008. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/27221

[14] D. Yacchirema and C. Palau, “Internet of Things Interoperability: A smart IoT gateway approach,” in *Proc. 18th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10211617

[15] N. H. Ahmed, A. M. Sadek, H. Al-Feel, and R. A. AbulSeoud, “Internet of Things multi-protocol interoperability with syntactic translation capability,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 1–11, 2021. [Online]. Available: https://www.academia.edu/download/83390606/Paper_69-Internet_of_Things_Multi_protocol_Interoperability.pdf

[16] “IoT communication protocols—IoT data protocols—Technical articles.” 2024. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/internet-of-things-communication-protocols-iot-data-protocols/

[17] “Communication From the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions a European Strategy for Data,” 2020. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CEL%3A52020DC0066

[18] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. Le Gall, and M. Zhao, “Standards-based worldwide semantic interoperability for IoT,” *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 40–46, Dec. 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7785888?casa_token=rGcfWfTyxuMAAAAA:ILRu-j5qo0sJPUb1sZrh1Y-l_OeDbBWey-v2amvgsSC1emSDIkC5bQGq73gX7HHgEBLiNBfztg

[19] “Interoperability of heterogeneous IoT platforms | INTER-IoT project | Fact Sheet | H2020,” 2024. [Online]. Available: https://cordis.europa.eu/project/id/687283

[20] C. E. Palau et al., *Interoperability of Heterogeneous IoT Platforms: A Layered Approach* (Internet of Things). Cham, Switzerland: Springer Int., 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-82446-4

[21] S. Aydin and M. N. Aydin, “Semantic and syntactic interoperability for agricultural open-data platforms in the context of IoT using crop-specific trait ontologies,” *Appl. Sci.*, vol. 10, no. 13, p. 4460, Jan. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/13/4460

[22] E. Lee, Y.-D. Seo, S.-R. Oh, and Y.-G. Kim, “A survey on standards for interoperability and security in the Internet of Things,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1020–1047, 2021. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9381989?casa_token=ftHHNzvzdaQAAAAA:4H66WVTNQwTws-4XEoDYKx5nMeISXYiuCR4JIJwNTDQnZ2sbAAQ7PLrwrtMAUQX8XJ9tBjE0Dw

[23] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, “A standard-based open source IoT platform: FIWARE,” *IEEE Internet Things Mag.*, vol. 2, no. 3, pp. 12–18, Sep. 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8950963?casa_token=9vE_5BfalZsAAAAA:u3z-67dOEvcv2GUMymCanMfe2oopIqJCAlGVqTs7me-tuxuoqbbEnV1GfhK5pthyk21f1mHfyw

[24] J. Soldatos et al., “OpenIoT: Open source Internet-of-Things in the cloud,” in *Interoperability and Open-Source Solutions for the Internet of Things* (Lecture Notes in Computer Science). I. Podnar Žarko, K. Pripužić, and M. Serrano, Eds.Cham, Switzerland: Springer, 2015, pp. 13–25.

[25] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, “Toward semantic interoperability in oneM2M architecture,” *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 35–41, Dec. 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7355582

[26] E. Karabulut and R. C. Sofia, “An analysis of machine learning-based semantic matchmaking,” *IEEE Access*, vol. 11, pp. 27829–27842, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10076794

[27] “Ontology alignment evaluation initiative::Home,” 2024. [Online]. Available: https://oaei.ontologymatching.org/

[28] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, “Definition of an ontology matching algorithm for context integration in smart cities,” *Sensors*, vol. 14, no. 12, pp. 23581–23619, Dec. 2014. [Online]. Available: https://www.mdpi.com/1424-8220/14/12/23581

[29] K. Dickerson, R. García-Castro, P. Kostelnik, and M. Paralič, “Standards for the IoT,” in *IoT Platforms, Use Cases, Privacy, and Business Models: With Hands-On Examples Based on the VICINITY Platform*, C. Zivkovic, Y. Guan, and C. Grimm, Eds.Cham, Switzerland: Springer, 2021, pp. 125–147. [Online]. Available: https://doi.org/10.1007/978-3-030-45316-9_6

[30] M. San Emeterio de la Parte, J.-F. Martínez-Ortega, N. Lucas Martínez, and V. Hernández Díaz, “GRYS-UPM/SISS-semantic-interoperability-support-system: Semantic interoperability support system,” May 2024. [Online]. Available: https://zenodo.org/records/11395062

[31] *Date and Time Format*, ISO Standard 8601, Feb. 2017. [Online]. Available: https://www.iso.org/iso-8601-date-and-time-format.html

[32] R. Lott (Open Geospatial Consortium, Arlington VA, USA). *Geographic Information—Well-Known Text Representation of Coordinate Reference Systems*. (Aug. 2018). [Online]. Available: https://docs.ogc.org/is/18-010r7/18-010r7.html#8

[33] FAIRsharing Team, “FAIRsharing record for: Quantities, units, dimensions and types,” 2015. [Online]. Available: https://fairsharing.org/FAIRsharing.d3pqw7

[34] “AFarCloud,” 2018. [Online]. Available: http://www.afarcloud.eu/

[35] “Aggregate farming in the cloud | AFarCloud Project | Fact Sheet | H2020,” Nov. 2021. [Online]. Available: https://cordis.europa.eu/project/id/783221

[36] “Observations in Finland—Finnish Meteorological Institute,” 2024. [Online]. Available: https://en.ilmatieteenlaitos.fi/observations-in-finland

[37] “API documentation & design tools for teams | Swagger.” 2024. [Online]. Available: https://swagger.io/

[38] R. Palma et al., “Agricultural information model,” in *Information and Communication Technologies for Agriculture—Theme III: Decision* (Optimization and Its Applications), D. D. Bochtis, C. G. Sørensen, S. Fountas, V. Moysiadis, and P. M. Pardalos, Eds.Cham, Switzerland: Springer Int., 2022, pp. 3–36. [Online]. Available: https://doi.org/10.1007/978-3-030-84152-2_1

[39] M. S. E. de la Parte, S. L. Serrano, M. M. Elduayen, and J.-F. Martínez-Ortega, “Spatio-temporal semantic data model for precision agriculture IoT networks,” *Agriculture*, vol. 13, no. 2, p. 360, Feb. 2023. [Online]. Available: https://www.mdpi.com/2077-0472/13/2/360

[40] “Semantic sensor network ontology,” W3C, 2024. [Online]. Available: https://www.w3.org/TR/vocab-ssn/

[41] A. Haller et al., “The modular SSN ontology: A joint W3C and OGC standard specifying the semantics of sensors, observations, sampling, and actuation,” *Semantic Web*, vol. 10, no. 1, pp. 9–32, Jan. 2019. [Online]. Available: https://content.iospress.com/articles/semantic-web/sw320

[42] “Extensions to the semantic sensor network ontology,” W3C, 2024. [Online]. Available: https://www.w3.org/TR/vocab-ssn-ext/

[43] “IoT-lite ontology,” 2024. [Online]. Available: https://www.w3.org/submissions/iot-lite/

[44] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, “IoT-Lite: A lightweight semantic model for the Internet of Things,” in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Jul. 2016, pp. 90–97. [Online]. Available: https://ieeexplore.ieee.org/document/7816831

[45] "Quality ontology," 2013. [Online]. Available: https://mobcom.ecs.hs-osnabrueck.de/cp_quality/
[46] A. Gyrard et al. "M3-lite taxonomy." Accessed: Dec. 29, 2024. [Online]. Available: http://iot.ee.surrey.ac.uk/ontology/m3-lite.owl
[47] "Linked open vocabularies (LOV)," 2024. [Online]. Available: https://lov.linkeddata.es/dataset/lov/vocabs/iot-lite
[48] "Schema.org," 2024. [Online]. Available: https://iot.schema.org/
[49] K. Michael, A. Darko, and T. Aparna, "Schema.org extensions for IoT (IoTschema)," 2023. [Online]. Available: https://www.w3.org/WoT/ws-2019/Presentations
[50] "IoT-schema-collab/IoTschema," Jan. 2024. [Online]. Available: https://github.com/iot-schema-collab/iotschema
[51] "Schema extensions for IoT community group," Jan. 2019. [Online]. Available: https://www.w3.org/community/iotschema/
[52] L. Daniele, R. Garcia-Castro, M. Lefrançois, and M. Poveda-Villalon, "SAREF: The smart applications REFerence ontology," May 2020. [Online]. Available: https://saref.etsi.org/core/v3.1.1/
[53] "ETSI portal—The smart applications reference ontology (SAREF)," 2024. [Online]. Available: https://saref.etsi.org/
[54] M. Schildhauer et al., "OBOE: The extensible observation ontology," Sep. 2019. [Online]. Available: https://github.com/NCEAS/oboe/tree/OBOE_1_1
[55] "Cross-cutting context information management (CIM); usage of external data models with NGSI-LD API," ETSI, Sophia Antipolis, France, document ETSI GR CIM 021, 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/CIM/001_099/021/01.01.01_60/gr_CIM021v010101p.pdf
[56] A. Abid, J. Lee, F. Le Gall, and J. Song, "Toward mapping an NGSI-LD context model on RDF graph approaches: A comparison study," *Sensors*, vol. 22, no. 13, p. 4798, Jan. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/13/4798
[57] "Understanding @context—NGSI-LD smart farm tutorials," 2024. [Online]. Available: https://ngsi-ld-tutorials.readthedocs.io/en/latest/understanding
[58] "Guidelines for modelling with NGSI-LD—Buscar con Google," 2024. [Online]. Available: https://www.google.com/search?q=Guidelines+for+Modelling+with+NGSI-LD&rlz=1C1CHBF_esES977ES977&oq=Guidelines+for+Modelling+with+NGSI-LD&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBBzM0OWowajeoAgCwAgA&sourceid=chrome&ie=UTF-8
[59] "Smart data models—FIWARE," Mar. 2020. [Online]. Available: https://www.fiware.org/smart-data-models/
[60] "Smart data models · GitHub." 2024. [Online]. Available: https://github.com/smart-data-models
[61] "Learning zone," Mar. 2020. [Online]. Available: https://smartdatamodels.org/index.php/learning-zone/
[62] "How to use smart data models in your projects," 2024. s [Online]. Available: https://fiware.github.io/data-models/specs/howto.html
[63] "Generate a@context based on external ontologies IRIs," Oct. 2021. [Online]. Available: https://smartdatamodels.org/index.php/generate-acontext-based-on-external-ontologies-iris/
[64] A. Abella, "Smart data models—Weather observed," Dec. 2023. [Online]. Available: https://github.com/smart-data-models/dataModel.Weather/blob/master/WeatherObserved/README.md
[65] "OGC to create agriculture information model standards working group; Public comment sought on draft charter," 2024. [Online]. Available: https://www.ogc.org/requests/public-comment-requested-agriculture-information-model-standards-working-group-charter/
[66] "DEMETER-AIM summary," 2024. [Online]. Available: https://agroportal.lirmm.fr/ontologies/DEMETER-AIM/?p=summary

Mario San Emeterio de la Parte received the Ph.D. degree (Highest Hons.) is entitled as "Contributions to Data Engineering in Spatio-Temporal Semantic Data Management for IoT" from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2024.

He is a Professor with the Department of Telematics and Electronic Engineering, UPM. His research focuses on data engineering, big data, data science, artificial intelligence (AI), Internet of Things (IoT), and cyber–physical systems. His work also includes spatio-temporal semantic data management through distributed semantic middleware architectures. He has contributed to several European research projects, including H2020-AFarCloud and H2020-DEMETER.

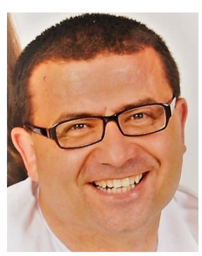
Prof. San Emeterio de la Parte is a member of the Next-Generation Networks and Services Research Group.

José-Fernán Martínez-Ortega received the Ph.D. degree in telematic engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2001.

He is a Full Professor with the Department of Telematics and Electronic Engineering, UPM, and the Head of the Next-Generation Networks and Services Research Group. He also served as a Guest Professor with Mälardalen University, Västerås, Sweden, from August 2013 to 2018, following an invitation as an international expert. He has authored several national and international publications included in the Science Citation Index in his interest areas, and responsible of several International, European, and National Projects, including research contracts with different IT companies. His main interest areas and expertise are ubiquitous computing and Internet of things (IoT), networked cooperating agent, and cyber–physical systems.

Néstor Lucas Martínez received the Ph.D. degree in telecommunications engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2021.

He is an Associate Professor with the Department of Telematics and Electronic Engineering, UPM, and a researcher with the Next-Generation Networks and Services Research Group. His research focuses on the design and application of adaptive models and artificial intelligence techniques in middleware systems for mission management in autonomous agents. He has authored several international publications indexed in the Science Citation Index in his areas of expertise. He has also participated in multiple international research projects on intelligent and adaptive systems, most recently in the Horizon 2020 projects DEMETER and AFarCloud, and previously in SWARMs, ACCUS, DEMANES, and WoO. He is particularly interested in intelligent middleware that supports coordination and cooperation among distributed agents operating in dynamic environments. His work spans the Internet of Things, cyber–physical systems, decision support systems, and the edge–fog–cloud continuum.



Vicente Hernández Díaz received the M.Sc. degree in electronic engineering from the Universidad de Alcalá, Alcalá de Henares, Spain, in 2013.

He is an Associate Professor with the Department of Telematics and Electronic Engineering, Universidad Politécnica de Madrid, Madrid, Spain, and a researcher with the Next-Generation Networks and Services Research Group. He has participated in several European research projects, and recently in AFARCloud, DEMETER, and SWARMS. The AFARCloud project, funded by the ECSEL-2017 European Research and Development Program, focused on Smart Agriculture and IoT; DEMETER, funded by the H2020 European Research and Development Program, was also focused on Smart Agriculture and more precisely on bringing down the farmers reluctance to adopt IT; SWARMS, funded by ECSEL-2024 European Research and Development Program, was focused on providing an infrastructure and framework for fostering the use of collaborative robots. He lectures on diverse subjects in the Telematics Engineering domain concerning computer networking, distributed applications and Internet of Things, Web Technologies, cyber-security and cloud computing. He has supervised more than 40 degree thesis and 15 M.Sc. thesis. His research activity has been assessed by the Spanish national agency.

Dr. Díaz has received two six-year research activity award.