Data Information Interoperability Model for IoT-enabled Smart Water Networks

Mandeep Singh
Birmingham City University
SingularLogic
msingh@singularlogic.eu

Wenyan Wu
Birmingham City University
Birmingham, B47XG, UK
wenyan.wu@bcu.ac.uk

Stamatia Rizou
SingularLogic
Kifissia, 14564, Greece
srizou@singularlogic.eu

Edlira Vakaj
Birmingham City University
Birmingham, B47XG, UK
edlira.vakaj@bcu.ac.uk

***Abstract*—Syntactic and semantic interoperability is a fundamental requirement for the success of the Internet of Things (IoT)-enabled Smart Water Networks (SWNs). Still, whilst consuming publicly accessible IoT data, the syntactic and semantic representation of the collected data poses challenges for the success of pervasive and ubiquitous sensing in the water domain. Challenges include the heterogeneity of data representation formats, semantic models, and the adoption of domain-specific standards and ontologies. These challenges emphasise the requirement for enhanced interoperability in SWNs. To address this, we propose a Data and Information Interoperability Model (DIIM) by combining the Semantic Web technologies, widely known for overcoming interoperability issues, and Model-driven architecture (MDA) approach. DIIM facilitates syntactic interoperability by serialization conversion and adoption of domain-specific standards as well as semantic interoperability of metadata by aligning the semantic models of IoT and Smart Water Network (SWN) applications. Furthermore, it automatically creates an ontology as a semantic model if it is missing and adds references to existing domain-specific ontologies as annotation in their models. We evaluate DIIM methodology by applying it to a real-world use case of IoT-enabled applications for water quality monitoring.**

***Index Terms*—Data Interoperability, Syntactic harmonization, Semantic Model Generation and Alignment, Smart Water Networks, IoT/WoT, Ontology, Representation Standard, Water Quality Monitoring**

I. Introduction

Due to global water scarcity and water quality crisis, each year, billions of dollars are being invested in integrated water resource management to meet the water demand with the supply of affordable, sustainable, and pure water to consumers [1]. Water utilities build Smart Water Networks (SWNs) by integrating various solutions and systems that enable remote and continuous monitoring and diagnosis of problems, manage maintenance issues and optimise the water distribution network by utilising data-driven methods.

The deployment of data-enabled Internet of Things (IoT)/Web of Things (WoT) at the physical layer of a SWN, e.g. smart sensors, valve controllers, and cooling units by water utilities have offered an opportunity to build a cohesive *overlay network* in SWN. Networked IoT offer utilisation of the IoT/WoT data (as numbers, symbols, text, images, sound recordings, unit values, etc.) and information (contextualized data) to enable smart sensing beyond the initial coverage areas of a SWN. However, the SWN applications, e.g. leakage detection, water distribution, water quality management, and customer metering, must interoperate with the IoT/WoT before they can run data-driven information analysis and make decisions or operate appropriately in real-time at the top layer of a SWN.

IoT-enabled SWN applications have data/information interoperability if data providers (IoT) and data consumers (SWN applications) can exchange, deliver, or use data through sending messages in a coordinated way. Such messages must typically be transformed at each interoperability level [2], either by the sender or receiver, to a construct that can be readily consumed and thus understood by the receiver; this process is often referred to as message alignment [3]. Wasserman and Fay compare The Levels of Conceptual Interoperability Model (LCIM) [2] with the Open Systems Interconnection (OSI) model [4] and state that any of OSI based communication technologies can enable syntactic and semantic interoperability, though achieving the semantic interoperability at level 3 remains a major challenge in the Semantic Web [5].

Furthermore, diversity in data syntax formats, e.g. Comma-separated Values (CSV), JavaScript Object Notation (JSON), Extensible Markup Language (XML), and Resource Description Framework (RDF), to serialize IoT data before it can be sent over an IoT data protocol, e.g. Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), and Hypertext Transfer Protocol (HTTP), leaves IoT software developers with a tough decision on interoperability, i.e., support all possible formats that costs a lot of resources or focus on one or two formats. Similar is the situation for the developers of the SWN applications as they must also implement all possible interfaces to deserialize the received IoT data. In [6], Howell et al. recount the reason of interoperability failure from smart grid [7] as *(i) lack of machine communication protocols, (ii) lack of common data formats, and (iii) lack of the common meaning of exchanged content.* In IoT-enabled SWNs, some of the existing solutions use MQTT or CoAP as a common communication protocol and WaterML2 as a common data format. However, semantic interoperability aspects are not sufficiently addressed by these standards. Thus, IoT-enabled applications require semantic models to understand the content of the exchanged data.

Both industry and academia have acknowledged the benefits of semantic models in the field of semantic web technologies through the World Wide Web Consortium (W3C) 'semantic web stack', which shows ontologies playing a critical role [8]. Groß et al. elaborate that an ontology enables a representation of the data machine-processable, ultimately allowing reasoning, generation of new knowledge, and automatic detection of inconsistencies in the semantic models [9]. However, regarding data semantics, a similar challenge to adopt domain-specific standards and formats remains for the developers of the IoT/WoT and SWN applications, as there are so many ontologies in the IoT and the water domain. Additionally, limited knowledge of existing and appropriate domain-specific ontologies and the cumbersome task of developing or adopting an ontology result in either no application ontology or referencing a single ontology, which limits the interoperability potential.

Hence, a solution to share and use IoT/WoT data should be based on the data interoperability and without necessarily tightly coupling of IoT/WoT with the interfaces of SWN applications at the time of development. The interoperability of IoT-enabled applications is still a subject of research, as cited in [10]; *data interoperability is one of the main obstacles to promote IoT adoption and innovation.* This paper proposes a domain-specific Data and Information Interoperability Model (DIIM) for IoT-enabled applications by utilising the Semantic Web technologies and Model-driven approach. To enable interoperability between IoT applications, DIIM can automatically build a semantic model of given IoT data, adopt domain-specific standards, align to domain-specific ontologies, and transform data into various data serialization formats according to an application's requirements.

We organise the rest of the paper as follows. Section II presents related work on interoperability and semantic modelling in the IoT and water domain. In Section III, we list the identified interoperability challenges of IoT-enabled SWNs. Section IV elaborates the DIIM methodology and its application to a case study of IoT-enabled water quality monitoring. Section V concludes the work presented in this paper.

II. Related Work

This section reviews the related work on interoperability in IoT and water domain projects. It also discusses the semantic modelling in the water domain.

A. Interoperability in SWNs and IoT applications

Hatzivasilis et al. compare the major European Union (EU) funded IoT research projects, BigIoT, OpenIoT, INTER-IoT, and SEMIoTICS in terms of interoperability features. Among the IoT projects, SEMIoTICS not only offers interoperability at four levels but goes 2 steps ahead of its competitors. It utilises semi-automatic pattern-driven techniques for the cross-domain operation and interaction of applications [11].

For IoT ecosystems, BigIoT introduces five interoperability patterns: *(i) cross-platform access, (ii) cross-application domain access, (iii) platform independence, (iv) platform-scale independence, and (v) higher-level service facades.* Although these patterns help to reuse data and services from different platforms of an ecosystem, there is a need for automatic search and orchestration of services [12].

In comparison with IoT interoperability approaches, the Water Enhanced Resource Planning (WatERP) framework from the water domain proposes an architecture that harmonizes the communication between systems. These systems control, monitor, and manage the water supply distribution chain by using a Service Oriented Architecture (SOA)-Multi-Agent System (MAS) approach together with a knowledge base driven by the Water Management Ontology (WMO) [13]. This approach integrates and utilises innovative technologies, SOA, web services, MAS, and semantic web languages to handle the interoperability issue of monitoring and decision-making applications within SWNs. The framework also offers a standardized SOA-MAS-based interface and communication interpretation through WMO. Additionally, through SOA-MAS-based approach, intelligent orchestration of system functionalities within the architecture is achieved, as agents can be conceptualized with Believe Desire Intention (BDI) [14] model to become autonomous and cooperative to achieve their declarative and procedural goals [15].

The Water analytics and Intelligent Sensing for Demand Optimised Management (WISDOM) project enables the interoperability of things and software in smart water networks through a software platform that utilises ontology for semantics and web services for web-enabled sensors to integrate business operations across the water value chain. They define a *water value chain* as the artefacts, agents, and processes involved in delivering potable water to consumers from natural water sources and safely disposing of foul and runoff wastewater. Their interoperability approach integrates existing data models, which are formalized in different data formats and use heterogeneous domain perspectives. They intersect existing models and align them with the WISDOM ontology that is used as a common ontology to support the data interoperability across the existing models. They promote interoperability through semantic web technologies and by performing a schema conversion from a knowledge base of devices instantiated within the WISDOM ontology into another model, e.g. Smart Appliances REFerence (SAREF), Infrastructure for spatial information in Europe (INSPIRE), Industry Foundation Classes 4 (IFC4), Semantic Water Interoperability Model (SWIM) [6].

B. Semantic modelling in the water domain

A chronological list of semantic models in the water domain is presented by Howell [16]. There are many existing ontologies and standards. However, there is no standardized common ontology in the water domain as compared to the Gene Ontology (GO) [17] that represents the knowledge base of genes. Maedche and Staab argue that mapping existing ontologies will be easier than creating a common ontology because a smaller community is involved in the process. Their further argumentation is, ontologies must be normalized to a uniform representation to eliminate syntactic differences and make semantic differences between the source and the target ontology more apparent [18].

Figure 1 depicts the common ontologies, formats, and standards that are being used in the water domain to conceptualize the domain knowledge. All listed standards are based on markup language XML, and all listed ontologies are defined in Web Ontology Language (OWL). As XML, RDF, and OWL are Semantic Web technologies, and RDF is built on XML and OWL is built on RDF [19], we can use OWL as an interoperable language to overcome the syntactic and semantic heterogeneity of data and information that is represented in any of the listed water domain standards and ontologies. However, if IoT data is represented in another standard than Semantic Web technologies and domain-specific ontology is referred to, we require translation and mapping to achieve interoperability. Finally, we can conclude that a common standardized ontology does not exist in the water and IoT domain. However, there have been attempts to recycle and merge existing standards and ontologies rather than build something from scratch.

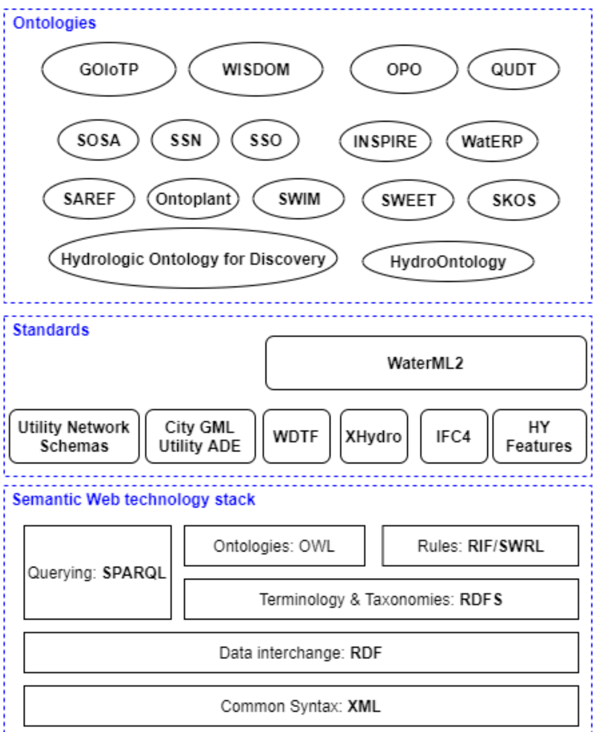

Fig. 1. Semantic Web technologies based standards and ontologies in the IoT and the water domain

WISDOM project proposes a semantic model for intelligent water sensing and analytics through a domain ontology created using Izssa's ontology integration approaches [20]. The WISDOM model integrates heterogeneous data sources and various ontologies; thus, it identifies the necessity of validation. At first, validate the domain model as an accurate, sufficient, and shared conceptualization of the domain by domain experts, then validate the ontology instantiation and deployment as a web service within a cloud-based platform through software testing [21].

In the INTER-IoT project, Generic Ontology for IoT Platforms (GOIoTP) is developed as a core ontology and reference meta-data model for IoT platforms. It offers modular data structures for describing entities like device structure, platform, observation, actuation, units, measurements, location, service, and user. Generic Ontology for IoT Platforms Extended (GOIoTPex), also developed in the INTER-IoT project, extends and fills the stub classes/concepts from GOIoTP with more specific classes, properties and individuals.

Both top-level ontologies, GOIoTP and WISDOM, bring complementary concepts and domain knowledge. Therefore, an IoT-enabled SWN will require such ontologies to build a semantic model representing data and information of IoT and water domain. Hence, an issue of ontology integration arises, Izssa's ontology integration approaches [20] can solve: *(i) Ontology mapping* to establish correspondence rules between concepts of two ontologies. *(ii) Ontology alignment* to bring two or more ontologies into a mutual agreement. *(iii) Ontology transformation* to change the structure of the ontology to make it compliant with another. *(iv) Ontology fusion* to build a new ontology from two or more existing ones.

III. Interoperability Challenges of IoT-enabled SWNs

While reviewing the related work, it becomes clear that interoperability is still a hot topic. The IoT-enabled applications require interoperability at syntactic (data exchange) and semantic (understanding the meaning of the exchanged data) layers to overcome interoperability issues. Most of the interoperability solutions for IoT-enabled applications are developed with the vertical application approach for smart networking and undermining the potential brought through the cross-domain integration of IoT-solutions in the water domain. For example, Industry 4.0 [22] cannot yield the potential of interconnected IoT if the data sent and received by IoT cannot be understood and used by consumer applications. Some of the challenges that need to be addressed to achieve IoT-data interoperability in the water domain are:

- **Transformation of data in various representation and encoding formats:** SWN application developers generally encode or represent data in their favourite data format, e.g. XML or JSON, and also publish data in their encoded format, e.g. UTF-8 or Latin1. Therefore, the data encoding format of one application could be different from the data encoding format of another application that wants to share data. So, one of the applications must have the translator/(de)serialization ability for each other's data representation and encoding format, and this must be implemented and deployed. However, when many applications want to interoperate and have different data representation and encoding formats, it will become challenging.
- **No standardized domain-specific ontology:** In the water domain, there are too many domain-specific and application-specific ontologies and no common standard water ontology. One reason is an ontology models only

a specific aspect of the real world based on the ontology engineer aim, therefore the ontology is limited by the interest and viewpoint of an ontology engineer. Another reason is that the reuse of existing ontologies is not widely practised because extending existing or merging an ontology with own ontology is a complex ontology engineering task. Therefore, each application tends to build its application-specific ontology. Additionally, to build a common SWN ontology, a consortium of organizations and companies from the public and private sectors is required. In this context, applications fail to adopt existing ontologies; thus, they cannot semantically understand the data shared by other applications without semantic mappings.

- **Adoption of the water domain-specific standards:** WaterML2 [23] is an XML-based standard that is developed by Open Geospatial Consortium (OGC) group (CSIRO, CUAHSI, USGS, BOM, NOAA, KISTERS, and others) to standardize time series data (hydro-meteorological observations and measurements) exchange in Hydrology. However, data modelled for IoT is not always in WaterML standard, as most of the IoT or SWN application developers do not know at development time which of standards they need to support ad-hoc utilization of the data.
- **Generation of missing ontology and vocabulary:** Semantic interoperability requires an understanding of the data through conceptual knowledge, generally represented through ontologies or vocabulary. However, these ontologies are mostly missing for the existing databases because ontology development remains a cumbersome manual task. In addition, developers must develop these ontologies manually while considering the schema of the represented data in different formats since schema helps to identify the structural organization of data.

These challenges demand a framework that can generate ontology from existing data while reusing the existing ontologies and adopting the existing standards, and facilitate syntactic and semantic interoperability of data and information between IoT and SWN applications.

IV. Data Information Interoperability Model

In this section, the first subsection outlines the Data and Information Interoperability Model (DIIM) that uses Model-driven architecture (MDA) approach and Semantic Web technologies to ensure syntactic and semantic interoperability of the IoT data. The second subsection describes a case study on IoT-enabled water quality monitoring. The third subsection demonstrates the application of DIIM to the case study.

A. DIIM architecture and methodology

From a technical point of view, figure 2 reveals the key components and their classification that are designed as web services for the loosely coupled DIIM architecture.

- **IoT components:** *Publication/Subscription Manager* offers the service to publish or subscribe IoT data. Every data subscription or publication creates a profile with the semantic model in the Knowledge Base (KB). *IoT Locator* scans in Thing Description (TD) of IoT and returns the list of IoT that matches the subscription. IoT data collector collects the data publisher's endpoint and delivers to KB. IoT data pusher retrieves the processed data from KB and pushes it to the subscriber's endpoint.

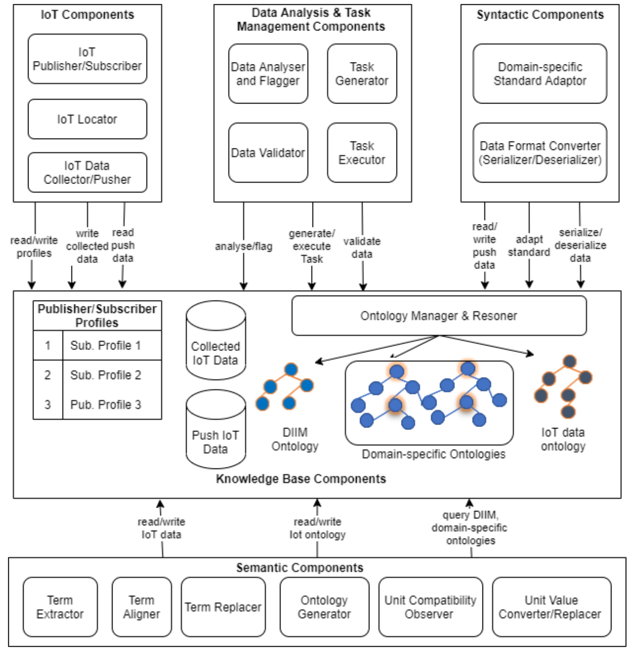

Fig. 2. Key components of DIIM

- **Data analysis and Task management components:** *Data Analyser and Flagger* components analyse the collected IoT data and create a publisher profile with a semantic model. Additionally, it sets a flag for every syntactic and semantic difference between the publisher's and subscriber's semantic model. *Task generator* creates tasks for every flag to achieve syntactic and semantic harmonization of the collected data. The *Task Executor* executes generated tasks. *Data Validator* validates the processed data and reschedules *Data Analyser and Flagger* on validation failure.
- **Semantic components:** *Term Extractor* extracts terms from the collected data. *Term Aligner* aligns the extracted terms to the terms of domain-specific ontologies and the subscriber's semantic model. *Ontology Generator* uses RDF conversion tools [24] to generate an ontology of the collected IoT data and annotate its terms with aligned terms. *Term Replacer* replaces the terms for the push IoT data. *Unit Compatibility Observer* sets a flag if the measurement unit of IoT differs from the subscriber's unit. *Unit Valuer Converter and Replacer* replaces the collected data values according to the conversion unit formula if a unit conversion flag exists.
- **Syntactic components:** *Domain-specific Standard Adaptor* translates the collected data into a domain-

specific standard. *Data Format Converter* serialize/de-serialize the collected data from OWL into another platform/application-specific data format.

- **KB:** It incorporates the knowledge of DIIM methodology in DIIM ontology written in OWL. The core part of the ontology contains domain-specific knowledge of a SWN-application, such as domain-specific ontologies, standards, serialization formats, measurement properties and their units. *IoT Data Ontology* holds the collected IoT data in OWL. *Ontology Manager* manages all ontologies in KB. *Publisher and Subscriber profiles* are also described and populated in OWL. *Term Aligner and Tagger* utilizes existing alignment tools, e.g. ALIN, MapOnto, and Yam++, to find alignment between IoT data ontology and other ontologies in the KB and annotates its terms with the aligned terms. *Ontology Reasoner*, such as Hermit or Pellet, reasons about the facts in the KB and answers the queries. KB also contains a list of available OWL/RDF translators, serializers and deserializers that can transform data from one format to other.

Figure 3 illustrates DIIM's key methodological steps that utilise a set of existing tools and technologies to offer syntactical and semantic interoperability of data and information to the IoT-enabled applications. The activity diagram highlights the procedure of the syntactic and semantic interoperability enablement between IoT/WoT and IoT-enabled applications. DIIM's procedural steps are as follows:

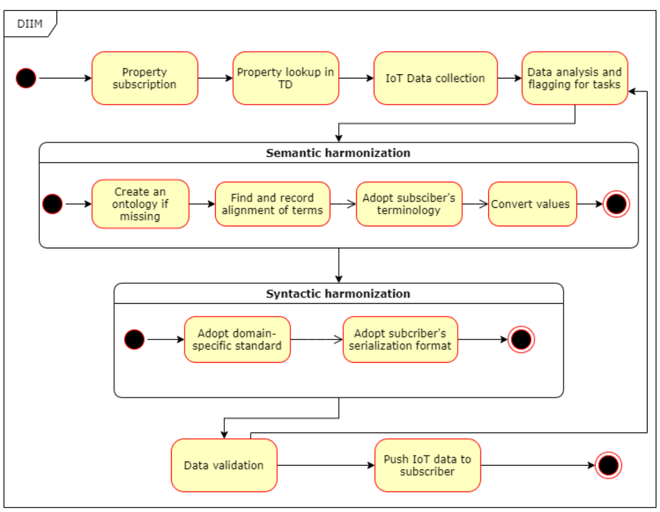

Fig. 3. DIIM's activity states

1) **Property subscription:** An IoT-enabled application subscribes its interest to receive data of a particular property, e.g. temperature. DIIM subscription interface specifies parameters as follows:

> **mandatory***{subscriberEndpoint, communicationProtocol, serializationFormat, propertyName, measurementUnit}*
> **optional***{latitude, longitude, fromDate, toDate, standardName, ontologyURI, applicationDomain}*

2) **Property lookup in TD:** DIIM keeps on scanning the TD unless a match to the subscription is found.
3) **IoT data collection:** Once DIIM finds a subscription's match, it collects IoT's available data and metadata and stores it as IoT collected data.
4) **Data analysis and flagging for tasks:** DIIM analyzes Data and flags it according to the subscription as the next step. For each flag, a task is created, scheduled and executed by DIIM.
5) **Semantic harmonization:** If IoT data do not refer to an ontology, DIIM create an ontology for the terms of the collected data. DIIM aligns these terms to the domain-specific and subscriber ontologies. All found alignments are stored as annotations in the IoT ontology. If the measurement unit of IoT does not match to subscription, the property values are converted and stored in the IoT ontology.
6) **Syntactic harmonization:** DIIM queries the data in the IoT ontology and transforms it in the domain-specific standard and serialization format that the subscriber requires. Finally, it stores harmonized data in the push IoT data.
7) **Data validation and push:** Newly harmonized data is validated according to the subscription profile. If validation fails, DIIM switches to Step *Data analysis and flagging for tasks*; otherwise, data is pushed to the subscriber.

B. A case study on IoT-enabled water quality monitoring

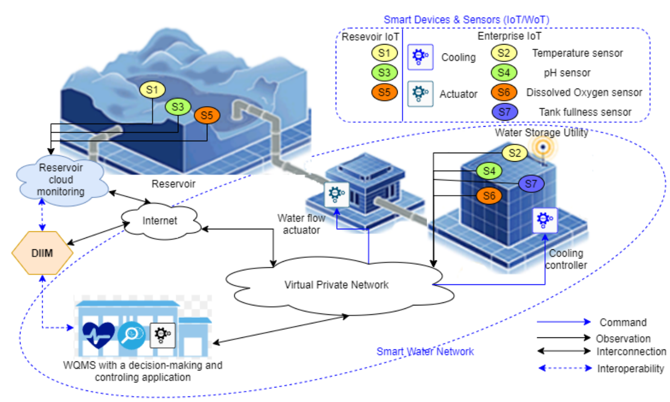

Fig. 4. A case study for interoperability of IoT-enabled WQMSs

Figure 4 illustrates a potential interoperability scenario for smart sensing in the water domain. Here, a network of IoT/WoT (smart sensors and devices) is constructed through the interconnection of several smart devices to support the pervasive and ubiquitous functionality of a WQMS. On one side, we see cloud-based water quality monitoring of the reservoir through S_1, S_2, and S_3 sensors for temperature, pH, and dissolved oxygen, respectively. On the other side, we see an enterprise that uses lake water as a drinking water resource to produce drinking water bottles. The enterprise has built up a SWN by interconnecting the enterprise IoT/WoT and WQMS in a Virtual Private Network (VPN) to monitor and manage the quality of the stored water. The smart sensors (S_2, S_4, S_6, and S_7) observe temperature, pH, dissolved oxygen and tank fullness and send the data to the enterprise WQMS for centralized remote monitoring and management. The decision-making application of WQMS makes decisions based on enterprise logic. It issues commands for the smart devices (cooling controller and actuator) to keep the stored lake water under ideal conditions, e.g., water temperature in a range of 2-15 °C, pH value in 6-9 logarithmic units, dissolved oxygen concentration in 80-120 mg/L and tank fullness in 700000 and 900000 m^3. Table I lists the rules computed by the decision-making system when sensors send the observation data to the WQMS. Consequently, the controller dispatches operational commands to operate the cooling and actuator utilities.

TABLE I
ENTERPRISE LOGIC RULES TO MANAGE THE STORED WATER

Property	Rule	Utility	Command
Temperature	if $S_2 value$ > 15	water cooling	on
Temperature	if $S_2 value$ <= 2	water cooling	off
Water volume	if $S_7 value$ <= 700000	actuator	open
Water volume	if $S_7 value$ >= 900000	actuator	close

Considering the case when the temperature of the fetched water exceeds 15 °C, the cooling system at the water storage utility must cool down the water. This operation will result in electricity consumption and raise the production cost. Alternatively, as shown in Table II, the stored water can also be cooled by opening the actuator if the reservoir water temperature is less than 15 °C and there is still storage capacity in the water tank. However, the decision-making application of WQMS must understand and evaluate the reservoir sensor data to make real-time decisions. Therefore, the syntactic and semantic interoperability [25] of the data collected by the IoT must be enabled regardless of the data serialization format and lexical label name of the observed data.

TABLE II
EXTENDED LOGIC RULE TO MANAGE WATER TEMPERATURE

Property	Rule	Utility	Command
Temperature	if $S_2 value$ > 15 AND if $S_1 value$ < 15 AND if $S_7 value$ < 80000	actuator	open

Table III displays the seven modelled sensors as an indicative interoperability scenario of the enterprise WQMS with reservoir cloud monitoring. S_1, S_3 and S_5 sensors are from a Chinese manufacturer, and they use JSON data format to serialize and use the terms *temp*, *ph* and *DO* to label their observed data. S_2 sensor is from an American manufacturer, and it uses RDF data format to serialize and uses term *Temperature* to label its observation. S_4, S_6, and S_7 sensors are from a European manufacturer, and they use XML data format to serialize and use the terms *pH* and *DissolvedOxygen* to label the observed data. The decision-making system of the enterprise WQMS follows the Semantic Web approach. Therefore, it expects the data to be well defined in RDF/XML format and uses terms that the Australian Government Linked Data Working Group defines for the marine water quality observations in a water quality ontology [26]. Additionally, the Enterprise Water Quality Monitoring (EWQM) ontology (re)uses the concepts (terms) from the Simple Knowledge Organization System (SKOS) data model and units of measurement from Quantities, Units, Dimensions, Data Types (QUDT) ontology. In contrast to the enterprise application, none of the IoT for reservoir refers to any ontology or data model standard. Although their data is publicly accessible from the cloud in JSON and CSV formats. This situation poses challenges of syntactic and semantic heterogeneity that the enterprise must address if it wants to use the data of reservoir sensors in its WQMS.

TABLE III
REPRESENTATION OF THE MODELLED SENSORS

Sensor	Sensor observation property	Serialization format	Label name of observed property	Unit of observed property
(IoT)	***(Data context)***	***(Data syntax)***	***(Data semantic)***	
S_1	Temperature	JSON	temp	°F
S_2	Temperature	RDF/XML	Temperature	°C
S_3	pH	JSON	ph	
S_4	pH	RDF/XML	pH	logarithmic units
S_5	Dissolved Oxygen	JSON	DO	ppm
S_6	Dissolved Oxygen	RDF/XML	DissolvedOxygen	mg/L
S_7	Water volume	RDF/XML	WaterVolume	m^3

C. DIIM's application in a case study

We describe the DIIM application procedure while enabling interoperability in the previously outlined case study of a water bottling enterprise and a water reservoir. Table IV displays an indicative setup of DIIM for the given case study.

TABLE IV
DIIM'S INDICATIVE PARAMETER SETUP

Parameter	Input
application domain	water quality monitoring
IoT data subscriber	enterprise WQMS
IoT data publisher	reservoir cloud monitoring application
domain-specific ontologies	SAREF, Geo, Time, QUDT, GeoRSS
application-specific ontologies	DIIM, IoT data ontology, EWQM
subscriber/publisher profiles	WQMS-profile, reservoir cloud-profile
domain-specific standard	WaterML2
IoT collect data	data collected from publisher
Parameter	**Output**
IoT push data	syntactically and semantically harmonized IoT data for subscriber
IoT ontology	IoT ontology with collected data and annotations as references to the terms of other ontologies

Based on the setup, DIIM will execute the following operational activities:

1) **Property subscription**: DIIM's operational activity starts when enterprise WQMS subscribes for the water

quality property (Temperature in °C) at a particular location and in the RDF/XML format at a specified endpoint. DIIM creates a subscriber profile of the subscription requested by the WQMS.

2) **TD lookup**: DIIM uses its DIIM ontology for water quality description to semantically match the TD for the subscribed water quality property in the IoT cloud. The lookup services keep on searching unless an IoT is found. Then, DIIM creates a publisher profile of the matched IoT and links it to the subscriber profile that matches the semantic search.
3) **Data collection**: The data collector starts collecting the meta and measurement data based on the publisher profile. An exemplary input to DIIM as collected IoT data in JSON format is revealed below. The fetched data is stored in collected IoT database.


```
### DIIM Input: a snippet of IoT data in JSON format ###
Meta data: "Name":"S1","Description":"The sensor measures water
temperature in Fahrenheit", "serial":"00-14-22-01-23-45",
"model":"BFG9000", "mac":"50:8c:b1:77:e8:e6",
"latitude":51.75543,"longitude":-1.03248
Measurement data: [{"4baa-a2ff-8741efad4e63": {"temp":[
{"timestamp":"2021-08-09T17:01:28.796Z","values":{"value":20}},
{"timestamp":"2021-08-09T17:01:38.792Z","values":{"value":24}},
... ]}}]
```


4) **Data analysis**: The collected data is analysed, and flags are set in the next step. Since the profiles of WQMS (subscriber) and reservoir (publisher) do not match, DIIM sets flags for serialization message content format RDF/XML, property name (term) harmonization and property value conversion.
5) **Semantic harmonization**: Since a flag for the property name and value conversion is set, the data objects of the collected data are relabeled, and its value is converted according to the subscriber profile. As shown in Table V, DIIM will map the terms of reservoir and WQMS to DIIM, WQMS ontology, and domain-specific ontologies. DIIM will create an IoT ontology for the reservoir and annotate its terms with the matched terms of other ontologies. DIIM will convert the temperature value from Fahrenheit to Celsius and store it in the IoT ontology.

TABLE V
ONTOLOGICAL ALIGNMENT OF TERMS AMONG RESERVOIR IOT, DIIM KB, AND ENTERPRISE WQMS

Terms of			
Reservoir	**DIIM Ontologies**	**WQMS**	**WQMS Ontology**
S_1	saref:Temperature sensor	S_2	ewqm:Sensor
temp	saref:Temperature	Temperature	qudt:water_temperature
f	saref:Temperature unit	C	qudt:unit
water	saref:Water	Water	ewqm:object
latitude	geo:latitude	Location	georss:point
longitude	geo:longitude	Location	georss:point
timestamp	time:dateTimeStamp	dateTime	ewqm:dateTime
values,value	saref:value	value	ewqm:value
Name	rdfs:Literal	name	ewqm:name
Description	rdfs:Comment	description	ewqm:description

The DIIM generated IoT ontology populated with collected data and annotated with references to the terms of other domain-specific ontologies is as follows:


```
<!—- DIIM Output: IoT ontology with examples of data and annotations —->
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:XMLS="http://www.w3.org/2001/XMLSchema#"
xmlns:iotonto="http://www.iot4win.co.uk/diim/iotonto/"
xmlns:saref="http://uri.etsi.org/m2m/saref#"
xmlns:ewqm="http://waterenterprise.com/wqm/ewqm#"
xmlns:georss="http://www.georss.org/georss/"
xmlns:qudt="http://qudt.org/2.1/schema/qudt">
<owl:Class rdf:about="iotonto:#Observation"/>
<owl:Class rdf:about="iotonto:#Sensor"/>
<owl:ObjectProperty rdf:about="iotonto:#hasObservation">
<rdfs:subPropertyOf rdf:resource="owl:#topObjectProperty"/>
<rdfs:domain rdf:resource="iotonto:#Sensor"/>
<rdfs:range rdf:resource="iotonto:#Observation"/>
</owl:ObjectProperty>
<iotonto:Sensor rdfs:label="saref:Temperature sensor; ewqm:Sensor">
<iotonto:name rdf:datatype="XMLS:string">S1</iotonto:name>
<iotonto:description rdf:datatype="XMLSchema:string">The sensor measures
water temperature in Fahrenheit</iotonto:description>
<iotonto:latitude rdf:datatype="XMLS:float">51.75543</iotonto:latitude>
<iotonto:longitude rdf:datatype="XMLS:float">-1.03248</iotonto:longitude>
<iotonto:mac rdf:datatype="XMLS:string">50:8c:b1:77:e8:e6</iotonto:mac>
<iotonto:serial rdf:datatype="XMLS:string">00-14-22-01-23-45</iotonto:serial>
<iotonto:model rdf:datatype="XMLS:string">BFG9000</iotonto:model>
<iotonto:measurementUnit
rdf:datatype="XMLS:string">f</iotonto:measurementUnit>
<iotonto:observeredObject
rdf:datatype="XMLS:string">water</iotonto:observeredObject>
</iotonto:Sensor>
<iotonto:ObservationCollection>
<iotonto:id rdf:datatype="XMLS:string">4baa-a2ff-8741efad4e63</iotonto:id>
<iotonto:property rdf:datatype="XMLS:string">temp</iotonto:property>
<iotonto:hasObservation rdf:parseType="Collection">
<iotonto:Observation> <iotonto:timestamp
rdf:datatype="XMLS:string">2021-08-09T17:01:28.796Z</iotonto:timestamp>
<iotonto:values> <rdf:Seq> <rdf:li>20</rdf:li> </rdf:Seq> </iotonto:values>
</iotonto:Observation>
</iotonto:hasObservation> </iotonto:ObservationCollection>
<owl:Axiom>
<owl:annotatedSource rdf:resource="iotonto:#4baa-a2ff-8741efad4e63"/>
<owl:annotatedProperty rdf:resource="iotonto:#property"/>
<owl:annotatedTarget rdf:datatype="XMLS:string">temp</owl:annotatedTarget>
<rdfs:label rdf:datatype="XMLS:string">qudt:water_temperature</rdfs:label>
<rdfs:label rdf:datatype="XMLS:string">saref:Temperature</rdfs:label>
</owl:Axiom> </rdf:RDF>
```


6) **Syntactic harmonization**: The WaterML2 time-series standard is first adopted for the push IoT data. Then, an OWL translator serializes the data in RDF/XML format.
7) **Data push**: After validation, the Data distributor pushes the syntactically and semantically harmonized publisher data in the WaterML2 time-series standard and semantics of EWQM ontology to subscriber endpoint.

In summary, DIIM's steps to enable interoperability in the use-case are: (i) DIIM takes subscription parameters from enterprise WQMS and IoT data from reservoir cloud as inputs. (ii) DIIM analyses the collected data and transforms the collected data in an OWL/RDF expressed semantic model. (iii) Ontology alignment tools align the newly generated semantic model with the domain-specific ontologies and the ontology of enterprise WQMS. DIIM records all found matches in the semantic model as annotations. (iv) Finally, OWL/RDF translator adopts the WaterML2 standard for time-series data and transforms the semantic model in enterprise WQMS's required format. Then DIIM uses the terms from enterprise WQMS's ontology to relabel the data. Since DIIM has also aligned the semantic model of the reservoir IoT data to domain-specific ontologies, therefore, the reservoir IoT data also becomes interoperable to all those applications which support these domain-specific ontologies.

V. Conclusion

This work presents a novel method to address the syntactic and semantic interoperability challenges of IoT-enabled SWNs. *DIIM's syntactic interoperability approach* overcomes the serialization format issues during the parsing of the IoT data by data-consuming applications by applying MDA methods for data format translation. Additionally, DIIM adopts domain-specific standards, e.g. WaterML2, to represent water-related data in time-series before delivering it to the consumer application. *DIIM's semantic interoperability approach* harmonizes the semantic models of IoT and a data-consuming application by aligning their ontologies to each other. Suppose an IoT application neither uses an existing ontology nor builds one ontology for its data. In that case, DIIM creates a semantic model in OWL from the available IoT data while finding the alignment of its terms to the terms of the domain-specific ontologies and (re)-annotate the IoT semantic model. With this method, DIIM enables interoperability between IoT and a SWN application and enhances the interoperability to the next level by adopting domain-specific ontologies and standards. Because, after alignment of IoT's semantic model to domain-specific ontologies, the IoT data becomes interoperable for all those applications that use these domain-specific ontologies. In the motivation scenario, DIIM acts as a mediator in the water domain while enabling the data-based interoperability between an IoT platform and a SWN application. However, any other discipline can use this approach to enable interoperability, where utilization of IoT data is beneficial.

Acknowledgment

European Union's Horizon 2020 supports this research and innovation programme under the Marie Skłodowska-Curie Innovative Training Networks (ITN) IoT4Win - Internet of Things for Smart Water Innovative Network (765921).

References

[1] Sensus, "Water 20/20: Bringing smart water networks into focus," 2019. [Online]. Available: https://smartwaternetworks.com/

[2] C. Turnitsa, "Extending the levels of conceptual interoperability model," in *Proceedings IEEE summer computer simulation conference, IEEE CS Press*, 2005.

[3] B. Brodaric, N. Booth, E. Boisvert, and J. Lucido, "Groundwater data network interoperability," *Journal of Hydroinformatics*, vol. 18, no. 2, pp. 210–225, 10 2015. [Online]. Available: https://doi.org/10.2166/hydro.2015.242

[4] I. Standardization, "Iso/iec 7498-1: 1994 information technology–open systems interconnection–basic reference model: The basic model," *International Standard ISOIEC*, vol. 74981, p. 59, 1996.

[5] E. Wassermann and A. Fay, "Interoperability rules for heterogenous multi-agent systems: Levels of conceptual interoperability model applied for multi-agent systems," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 89–95.

[6] S. Howell, Y. Rezgui, and T. Beach, "Automation in Construction Integrating building and urban semantics to empower smart water solutions," *Automation in Construction*, vol. 81, pp. 434–448, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.autcon.2017.02.004

[7] IEEE, "Ieee guide for smart grid interoperability of energy technology and information technology operation with the electric power system (eps), end-use applications, and loads," *IEEE Std 2030-2011*, pp. 1–126, 2011.

[8] W3C, "Semantic web," 2015. [Online]. Available: http://www.w3.org/standards/semanticweb/

[9] A. Groß, C. Pruski, and E. Rahm, "Evolution of biomedical ontologies and mappings: overview of recent approaches," *Computational and structural biotechnology journal*, vol. 14, pp. 333–340, 2016.

[10] N. Kalatzis, G. Routis, Y. Marinellis, M. Avgeris, I. Roussaki, S. Papavassiliou, and M. Anagnostou, "Semantic interoperability for IoT platforms in support of decision making: An experiment on early wildfire detection," *Sensors (Switzerland)*, vol. 19, no. 3, 2 2019.

[11] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, D. Anicic, A. Bröring, V. Kulkarni, K. Fysarakis, and G. Spanoudakis, "The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018, pp. 1–7.

[12] A. Bröring, S. Schmid, C. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, and E. Teniente, "Enabling iot ecosystems through platform interoperability," *IEEE Software*, vol. 34, no. 1, pp. 54–61, 2017.

[13] G. Anzaldi Varas, W. Wu, A. Abecker, E. RUBIÓN, A. Corchero, A. Hussain, and M. QUENZER, "Integration of water supply distribution systems by using interoperable standards to make effective decisions," in *11th International Conference on Hydroinformatics, HIC 2014*, 08 2014.

[14] A. S. Rao and M. P. Georggeff, "Decision Procedures for BDI Logics," *Journal of Logic and Computation*, vol. 8, no. 3, pp. 293–343, 06 1998. [Online]. Available: https://doi.org/10.1093/logcom/8.3.293

[15] M. Winikoff, L. Padgham, J. Harland, and J. Thangarajah, "Declarative and procedural goals in intelligent agent systems," *International Conference on Principles of Knowledge Representation and Reasoning 22-25 April 2005*, 2002.

[16] S. Howell, Y. Rezgui, and T. Beach, "Water utility decision support through the semantic web of things," *Environmental Modelling & Software*, vol. 102, pp. 94 – 114, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364815216307551

[17] G. Consortium, "Gene ontology," 01 2020. [Online]. Available: http://geneontology.org/

[18] A. Maedche and S. Staab, "Semi-automatic engineering of ontologies from text," in *Proceedings of the 12th international conference on software engineering and knowledge engineering*. Citeseer, 2000, pp. 231–239.

[19] A. Gómez-Pérez, M. Fernandez-Lopez, and O. Corcho, *Ontological Engineering*. Springer-Verlag London, 2004.

[20] S. Izza, "Integration of industrial information systems: from syntactic to semantic integration approaches," *Enterprise Information Systems*, vol. 3, no. 1, pp. 1–57, 2009. [Online]. Available: https://doi.org/10.1080/17517570802521163

[21] S. K. Howell, Y. Rezgui, T. Beach, W. Zhao, J. Terlet, and H. Li, "Smart water system interoperability: Integrating data and analytics for demand optimized management through semantics," *ICCCBE*, pp. 1–9, 2016.

[22] K. Schwab, *The fourth industrial revolution*. Currency, 2017.

[23] H. D. W. g. OGC, "Waterml2," 2014, accessed 23 Jan 2021. [Online]. Available: http://waterml2.org/

[24] W3C, "Convertertordf," accessed 20 Oct 2021. [Online]. Available: https://www.w3.org/wiki/ConverterToRdf

[25] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 796–809, 2019.

[26] A. G. L. D. W. Group, "Water quality data archive," 2016, accessed 6 July 2021. [Online]. Available: http://environment.data.gov.au/