Conference papers

School of Electrical and Electronic Engineering

2019

Extending Two-level Information Modeling to the Internet of Things

Paul Stacey
Technological University Dublin, paul.stacey@tudublin.ie

Damon Berry
Technological University Dublin, damon.berry@tudublin.ie

Extending Two-level Information Modeling to the Internet of Things

Paul Stacey
School of Informatics & Engineering,
Technological University Dublin, Blanchardstown,
Dublin, Ireland
paul.stacey@itb.ie

Damon Berry
School of Electrical and Electronic Engineering
Technological University Dublin, Kevin Street,
Dublin, Ireland
damon.berry@dit.ie

Abstract- **Interoperability is a major challenge for the Internet of Things (IoT). The real potential of the IoT lies in facilitating largescale sharing of high-quality context-rich information through *systems-of-IoT-systems,* rather than IoT systems that operate as isolated technology silos. Real large-scale interoperability requires layers of standards, and each layer addresses different interoperability challenges. The SensorThings API data model seeks to tackle data interoperability at the data and informational layers of IoT platforms. SensorThings API is aligned to the ISO/OGC O&M data standard, and like O&M it is semi-structured. Semi-structured models allow for variance within implementations for different use-cases, which is both necessary and detrimental to systems interoperability.**

In this paper we propose that the SensorThings API data model should be defined as a set of *archetypes*, used to capture extensible domain concepts using a *two-level modeling* IoT systems design approach. Extending two-level modeling to the IoT using the SensorThings API as a base for domain concepts definition allows for a powerful framework to manage variance within systems implementation and maintaining semantic interoperability within systems-of-IoT-systems across diverse use-cases.

Keywords—Internet of Things, SensorThings API, two-level modeling, interoperability, O&M, archetypes

I. INTRODUCTION

Data heterogeneity is characterized by the many different coding formats, constraint models and storage solutions used to capture, share and persist data. Data heterogeneity is pervasive in Internet-of-Things (IoT) data infrastructures, leading to a missed opportunity for organisations and businesses to create value leveraged off the rich data sets provided by the IoT. The IoT is seen as a core component of the emerging European Data Economy. However, data interoperability is defined as one of 5 primary barriers to the data economy [1].

Since its inception, IoT systems have been designed from a singular use-case perspective, resulting in *IoT silos* [2]. More recently, large-scale projects to address *IoT inter-silo interoperability* have emerged. Notable examples are the H2020 funded projects Interoperability of Heterogeneous IoT Platforms (INTER-IoT) [2] and the Federated Interoperable Semantic IoT/Cloud Testbeds and Applications (FIESTA) project [3]. These projects seek to leverage existing semantic technologies such as ontologies to enhance interoperability among disparate IoT systems. These projects currently exist in a standards vacuum, where internationally agreed IoT data standards, at the level of the International Standards Organization (ISO), are only beginning to emerge. Therefore, existing ontologies such as the Semantic Sensor Network Ontology (SSNO) [4] and conceptual information models such as OGC/ISO/DIS 19156 Observations & Measurements (O&M) [5] are being reused within an IoT context. However, the IoT domain is unique in its complexity and requirement for horizontal integration. This in turn presents many barriers to building consensus among a diverse community of stake holders in any IoT system design.

A. Conceptual Modeling Challenges

Gahegan & Pike [6] describe in detail the challenges of conceptual modeling within complex domains. The fundamental challenge of arriving at a consensus described by Gahegan & Pike and relating to this work is:

> The world is changing, so concepts must either adapt accordingly or become obsolete. We as individuals and groups are also constantly changing, so our needs, goals understanding and experience - i.e. our bases for constructing concepts - are also in flux. [6, p 731]

The SensorThings API [7] defined by the Open Geospatial Consortium (OGC) seeks to define a rich framework to achieve horizontal IoT systems integration. The SensorThings API is still evolving but its approach is gaining wide acceptance within the IoT community [8]. In this paper we examine the SensorThings API data model against Gahegan & Pike's challenges. We propose the adoption of a sophisticated information modeling approach known as *two-level modeling* [9] within the SensorThings API framework to meet these challenges. We show how the SensorThings API data model can be redefined as a set of extensible information artefacts known as *archetypes* and discuss how this approach can address some of the main IoT interoperability challenges.

This paper is organized as follows. Section II presents a brief introduction to both the SensorThings API and Two-level Modeling. Section III describes the tools and methods used within this study. Section IV presents the outcomes of our model mapping and the resulting archetype model of the SensorThings API data model. In section V we discuss the implications of this work and conclude with a proposal for future work.
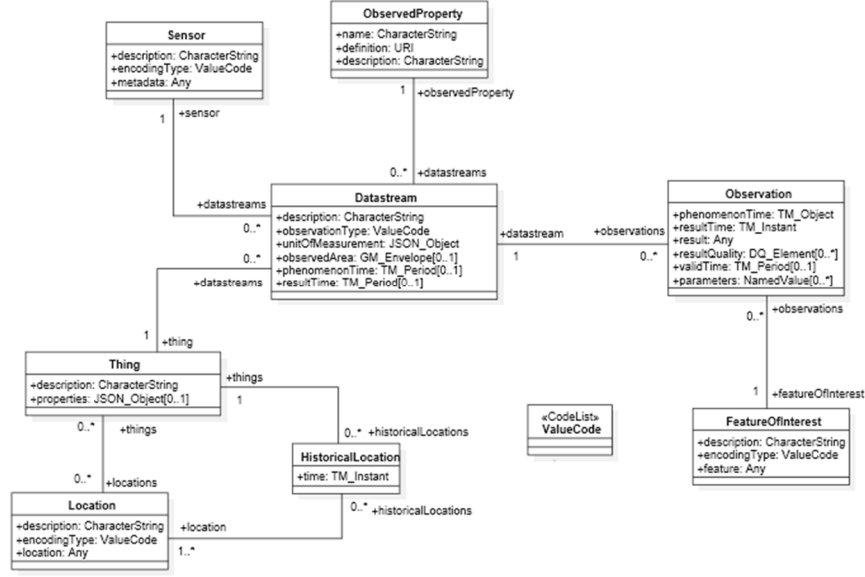
Fig. 1. SensorThings API data model [7].

II. BACKGROUND

A. SensorThings API

The OGC SensorThings API is divided into two main parts, the *sensing part* and the *tasking part*. The tasking part is the subject of future work within the OGC. This study is concerned with the more mature, sensing part.

The SensorThings API follows a rich set of principles, conventions and protocols, specifically aimed at resource constrained sensing devices. For example, the API defines a RESTful [10] based standard to enable CRUD (create, read, update, delete) based interactions for the requesting and reporting of sensed data, similar to OGC's Sensor Observation Service [11]. The sensing part also defines a data model that is based on the ISO/OGC O&M data model. The alignment with O&M can be seen in the entities defined within its data model, specifically *Observation* and *FeatureOfInterest*. In addition, the following entities are also defined: *Thing*, *Locations*, *HistoricalLocations*, *DataStream* & *Sensor* (Fig 1).

Much like O&M, the SensorThings API data model enables syntactic interoperability between heterogeneous IoT systems. Semantic interoperability is however limited. Semantic integration goes beyond combining data points solely based on syntactic representation. Typically, ontological bindings - within datasets - are used to record the meaning of the captured data. There are an increasing number of ontologies available within the IoT domain that can be used to enable semantic interoperability [12].

Syntactic based data models, and ontologies alone do not fully tackle the challenges highlighted by Gahegan & Pike (see Section I). In [13], a previous work by the authors, proposed the adoption of a two-level modeling approach leveraged on top of O&M to solve the short comings of relying solely on static object-oriented data models and ontologies. Using two-level modeling a third artefact is defined (which combines with both the data model & ontology). A brief overview of two-level modeling is described below.

B. Two-level Modeling

Typically, most information systems are designed using a singular static data model. This data model may be captured using an object-oriented, entity-relationship or indeed an alternative modeling approach, but in any case, these approaches will use a singular static model. In this *single-level* approach, domain knowledge concepts are interwound with informational concepts and programmatically committed to the information system.

As Gahegan & Pike note, in complex domains concepts are in flux. Consequently, we can say data models are subject to constant evolution - as the associated domain concepts are constantly evolving. Hard coded concepts soon become obsolete, as they no longer represent the current domain knowledge [9]. Interoperability suffers over time as heterogeneous information systems begin to emerge, all representing different implementations of the domain data and with no clear mechanism for integration of information objects [9].

To ensure longevity, many standards avoid overly defining data models and standards. However, this approach results in the production of abstract models that need to be specialized for individual use-cases. Although implementing systems will adhere to the abstract standardized data model, the particulars of the implementations are not standardized and therefore inhibit interoperability.

A two-level information modeling system design approach defines two levels, or models. The *reference model* and the *knowledge model* (Fig 1).

- The reference model contains non-volatile concepts, or classes with an abstract meaning that are not subject to change over time. These classes are hard-coded into the system software.
- The knowledge model captures the concepts that will undergo evolution over time. This model is not hard-coded into the system software but rather it evolves through consensus-based curation using cloud-based tools and is processed at runtime. Following a process of engagement between domain experts, concepts are captured in an Archetype Model using archetypes. Archetypes act as a problem specific constraint model on the underlying reference model.
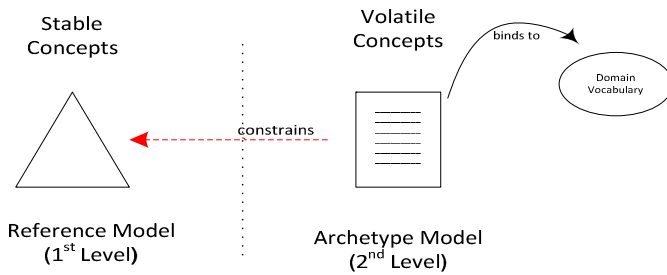

Fig. 2. The archetype model is a set of constraint statements against the underlying stable reference model. The archetype model evolves over time.

Archetypes are a set of constraint statements that are normally captured using the Archetype Definition Language (ADL) [14]. Archetypes are developed by a community of supporting domain experts and may be further specialized by these communities for particular use-cases in different jurisdictions. Systems generate information instances at run-time using *operational templates* (OPT) that adhere to the archetype model and the underlying reference model. For a more thorough overview of two-level modeling techniques the reader is directed towards [9] and [13]

C. Extending Two-Level Modeling to the IoT

When extending two-level models to enable data gathering in a new domain, beyond health, the authors recommend the following technical tasks (listed below) are performed [13].

1. Develop a generalised identity model for the domain.
2. Develop functioning binding to relevant terminologies.
3. Adopt, adapt or develop a suitable reference model.
4. Develop two-level information communication and processing for resource constrained devices.
5. Form a suitable community of supporters.
6. Develop consensus-based domain archetypes.

Previous work by the authors has considered different parts of this process. In this work, we focus on (3) & (6) above, with a view to building the required community of supporters within the IoT community, and thus further the work required in the remaining work items listed above.

Firstly, we must consider the following question: within a two-level modeling approach, does the SensorThings API meet the requirements of a reference model?

D. SensorThings API as a Reference Model

Initially it would appear that the SensorThings API data model could serve as an appropriate reference model to underpin a two-level modeling approach within IoT systems. To assess whether this is the case, we must firstly define the characteristics of a reference model.

Reference models should only capture the stable concepts within a domain, at the principles level within a multi-level ontological space (Fig 3). In [13] the authors have previously examined the O&M standard's suitability for two-level modeling. It was concluded that O&M lies just above the principles ontological level but given the maturity and wide acceptance of O&M within the community and its adoption within the INSPIRE Directive [15] it is pragmatic to choose O&M to underpin archetype definitions.

The SensorThings API extends O&M further, beyond the principles level. Here we conclude that concepts such as *DataStream* are in fact upper level *organizational concepts* within the IoT domain, and so should be defined within the archetype model and not within a reference model. This topic is dealt with in more detail in the remainder of the paper.
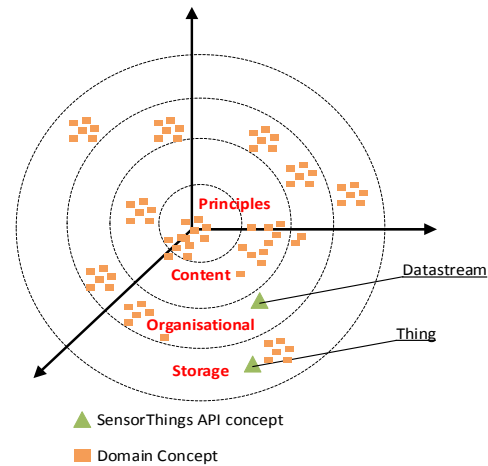
III. TOOLS & METHODS

Fig. 3. SensorThings API Ontological levels. It can be seen that SensorThings API concepts lies within the content, organizational and storage levels in a multi-level knowledge space.

Two-level modeling has matured within the health domain over a 20 year period. For health applications there are a rich set of development tools and methodologies and a large community of supporters. However, these tools and methods do not, in many
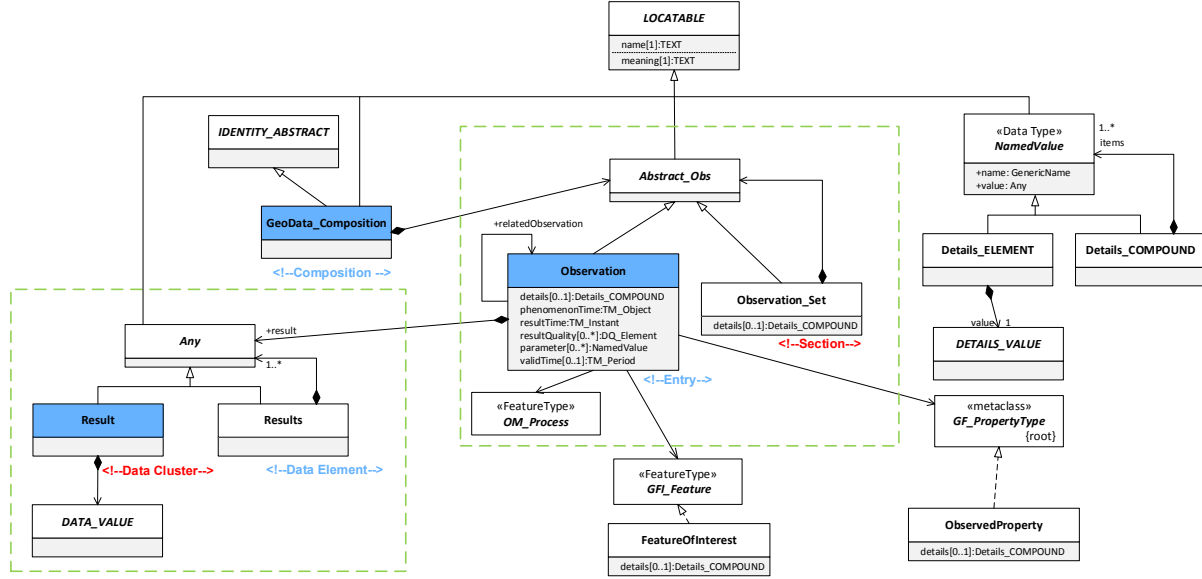
Fig. 4. An augmented Observations and Measurements model [13]. This model serves as the reference model for the two-level modeling approach. Compound/element patterns (highlighted in green) are necessary for two-level modeling.

instances directly translate to other domains. Below we describe the methodologies adapted from health practitioners and tools used within this study. Firstly, we use a multilevel ontological knowledge space to examine the concepts/entities defined within the SensorThings API data model. Next, we present an augmented O&M model, developed by the authors [13] to underpin two-level modeling within geo-spatial applications. Lastly in this section the modeling methodology and modeling tools used within this study are described.

A. SensorThings API Concepts

A reference model should only capture stable concepts i.e. at the principles ontological level. These stable concepts should be true for all instances and all usage contexts. Principles level Ontologies include DolceUltra Lite Upper Level Ontology[1]. Using this framework, IoT domain concepts derived from the SensorThings API are mapped onto a multi-level knowledge space (Fig 3).

B. O&M based reference Model

The rationale for choosing O&M as the basis for an appropriate reference model is described in [13]. In order to facilitate the definition of an archetype model, any reference model must contain *compound/element patterns*. This allows the creation of recursive aggregation of domain specific concept objects from the non-volatile concepts captured within the reference model. The reference model shown in Fig 4 is essentially the existing O&M standard augmented with these patterns at the point where extensibility is deemed necessary.

C. Archetype Modeling Methodology

Moner et al. [16] define an Archetype Modeling Methodology (AMM). The main phases of the methodology are briefly described below.

1) Phase 1 - Analysis
In this phase, the scope of the modeling is defined, initial domain concepts are discovered. Also, initial information elements are captured.

2) Phase 2 - Design
During the design phase, information structuring takes place along with constraint definitions.

3) Phase 3 - Development
Development consists of: archetype structure development, terminology binding and template structure development.

4) Phase 4 - Validation
Validation consists of a review of both the developed archetypes and associated templates.

5) Phase 5 - Publication
Validated archetypes and templates are published within the appropriate community repositories.

In this study a paper-based process is used for Phases 1- 3 described above. Once archetype structures were developed, the LinkEHR editor [17] was used to redefine the SensorThings API data model as a set of base archetypes.

---

[1] see http://www.loa.istc.cnr.it/old/DOLCE.html

TABLE I.   SENSORTHINGS API CONCEPT MAPPING TABLE

SensorThings API	Definition	Augmented O&M Base	Comments
Thing	A representation of some physical or virtual entity, equipped with one or more sensors. Sensor Platform [7]	COMPOSITION – GeoData_Composition	Thing is a domain concept that is a specialization of the reference model concept GeoData_Compostion
Datastream	A concept that groups Observations [7]	SECTION – Observation_set	Datastream is a domain concept that is a specialization of the reference model concept Observation_set
Sensor	The procedure used in the observation [7]	OM_Process	Sensor is a constraint on the empty O&M class OM_Process.
Location	A representation of the Thing's location [7]	Details_COMPOUND	Geodata_Composition contains an attribute "details" of type Details_COMPOUND which is an aggregation of Details_ELEMENT.
Observation	Act of measuring or otherwise determining the value of a property [5]	Observation	Semantically equivalent
FeatureOfInterest	The focus of the observation	FeatureOfInterest	Semantically equivalent
ObservedProperty	The property observed of the feature of interest	ObservedProperty	Semantically equivalent

IV. RESULTS

A. Concept Mapping

Table 1 shows a summary of the mapping of SensorThings API data model concepts to the augmented O&M reference model in Fig 4. It was found that each SensorThings API concept can be characterized as a constrained version of a reference model concept. For example, we can see that *Thing* is a constrained *storage* level concept (referred to as a COMPOSITION) and *DataStream* is a further constraining of the organisational concept *Observation_set* (referred to as a SECTION, see also Fig 3). Therefore, the concepts *Thing & DataStream* can be encoded as archetypes or constraints of reference model classes GeoData_Composition and Observation_set. *Sensor* is the procedure used in the measuring or otherwise observing a property of the feature of interest. It is in fact a constraint on the reference model concept *OM_Process*.

B. SensorThings as an Archetype Model

The resulting SensorThings API archetype model is made up of numerous resulting archetype definitions, defined using the LinkEHR tool (Fig 5).

Fig. 5. Using the LinkEHR multi-reference model editor, an XSD representation of Fig 4 is used to define the SensorThings API archetype model. Here the concept Thing is a set of constraint statements on the reference model concept GeoData_COMPOSITION.

The archetype model is defined against the OGC SensorThings API standard sensing entities definitions. For example *Thing* has the relationship constraints of one-to-many with Datastream, i.e. a Thing may have 1..* Datastreams. Defining this using LinkEHR is achieved by creating an archetype slot (of type reference model concept Observation_set) to the archetype definition of *Datastream* (Fig 6). Archetype slots of a particular reference model type resolve to an archetype which constrains that reference model type. LinkEHR ensures the data structures of the underlying reference model are adhered to as the archetype model is defined.
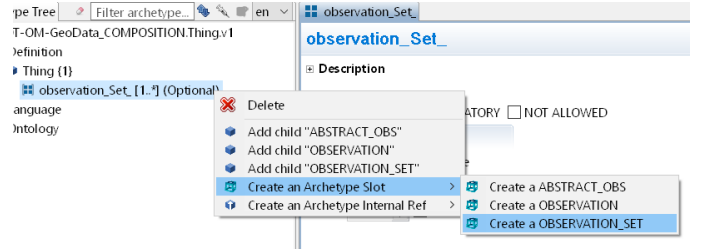

Fig. 6. Here LinkEHR is used to define the constraint: Thing may have 1..* Datastreams. Datastream is a reference model type Observation_set, and here we create an *archetype_slot* to plug in an archetype of type Observation_set. When constructing archetypes, we are bound by the underlying reference model.

Listing 1 shows a snippet of the resulting ADL representation of Thing as a simple constraint statement against GeoData_Composition.

```
Geo_Data_Document [at0000]
    occurrences matches {1..1} --Dublin City
GeoData_COMPOSITION [at0001] occurrences matches {0..*}
                                              --Thing
["at0001"] = <text = <"Slot to GeoData_COMPOSITION">
      description = <""> comment = <"This node was
solves to {TPOT-OM-GeoData_COMPOSITION.Thing.v1}">
               >
```

Listing 1. This ADL Snippet shows how using two-level modeling SensorThings & O&M can be transformed from a model of reality to a model of documentation. In this case concept [at0000] describes a subject of documentation. In this scenario this is "Dublin City". Dublin City may have zero-to-many *Things*. Concept [at0001] represents the SensorThings API concept *Thing*, which can be further spcceialised to "air quality monitoring device" for example.

V. Discussion & Conclusions

To ensure IoT domain-based data streams are truly interoperable, meta-data must be semantically rich enough for IoT systems to automatically bind disparate data streams. The SensorThings API data model provides a rich framework for achieving horizontal integration of IoT silos, enabling IoT systems-of-systems to be realised. However, the abstract nature of the SensorThings API data model means system developers must make local decisions about how to encode data structures for individual use-cases.

This study has shown that two level modeling can be extended to the IoT domain through the mapping of the SensorThings API to appropriate data patterns within an augmented O&M based data model, and consequently encoding the SensorThings API data model as a set of extensible informational artefacts (archetypes, or an archetype model).

Once mapped, modeled and published, these artefacts can enable a two-level modeling community of supporters to develop and grow within the IoT domain. Communities can agree on further specialization of the SensorThings API archetype model for individual IoT use cases and again publish these to be used within the community or to enable systems to semantically integrate through rich querying made possible by the semantically rich data sets. Querying is based on the three artefacts that define the two-level modeling approach: *reference model*, *archetypes* and *ontological bindings*.

This approach has implications for the current implementation of SensorThings API. Mapping concepts to either the reference model or the archetype model ultimately determines the access API. To future-proof systems, the access API should ideally only implement reference model concepts. The wider ramifications of this would require further evaluation, while engaging domain practitioners in further standardization work.

To further evaluate the applicability of this approach for individual use cases the authors propose that several pilot studies should be undertaken using the SensorThings API archetype model as the basis for concept definition and system implementation. Previous ocean observing use-cases [18] [19] undertaken by the authors can act as a reference for IoT pilot studies using this approach. These examples have shown how the two-level modeling approach can allow managed extensibility for individual use-cases using archetype models developed on top of an augmented O&M reference model.

References

[1] European Commission, "Study on emerging issues of data ownership, interoperability, (re-)usability and access to data, and liability", 2018. [Online]. Available: https://ec.europa.eu/digital-single-market/en/internet-of-things. [Accessed: 10-Dec-2018].

[2] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective", *Journal of Network and Computer Applications*, 2017, 81, pp.111-124.

[3] M. Serrano et al, "Cross-domain interoperability using federated interoperable semantic IoT/Cloud testbeds and applications: The FIESTA-IoT approach," *Building the Future Internet through FIRE 2016 FIRE Book: A Research and Experiment based Approach,* River Publishers, 2017.

[4] M. Compton *et al*, "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 17, pp. 25-32, 2012.

[5] ISO 19156:2011 Geographic information – Observations and measurements. 2011. doi:10.13140/2.1.1142.3042.

[6] M. Gahegan and W. Pike, "A situated knowledge representation of geographical information," *Transactions in GIS,* vol. 10, *(5),* pp. 727-749, 2006.

[7] S. Liang, C. Huang and T. Khalafbeigi, "OGC SensorThings API-Part 1: Sensing," *OGC® Implementation Standard.Http://docs.Opengeospatial.org/is/15-078r6/15-078r6.Html,* 2016.

[8] A. Kotsev *et al*, "Extending INSPIRE to the Internet of Things through SensorThings API," *Geosciences,* vol. 8, *(6),* pp. 221, 2018.

[9] T. Beale, "Archetypes: Constraint-based domain models for future-proof information systems," in OOPSLA 2002 Workshop on Behavioral Semantics, 2002.

[10] R. T. Fielding and R. N. Taylor, "Principled design of the modern Web architecture," ACM Transactions on Internet Technology (TOIT), vol. 2, (2), pp. 115-150, 2002. doi: 10.1145/514183.514185.

[11] A. Bröring, C. Stasch and J. Echterhoff, "OGC sensor observation service interface standard," *Open Geospatial Consortium Interface Standard,* pp. 12-006, 2012.

[12] G. Bajaj *et al*, "A study of existing Ontologies in the IoT-domain," *ArXiv Preprint arXiv:1707.00112,* 2017.

[13] P. Stacey and D. Berry, "Towards a Digital Earth: using archetypes to enable knowledge interoperability within geo-observational sensor systems design", *Journal of Earth Science Informatics, Springer*, 2018. doi:10.1007/s12145-018-0340-z.

[14] T. Beale, and S. Heard, "Archetype Definition Language". The openEHR Foundation, London, .2007.

[15] I. Directive, "Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE)," Published in the Official Journal on the 25th April 2007.

[16] D. Moner, J. A. Maldonado and M. Robles, "Archetype modeling methodology," *J. Biomed. Inform.,* vol. 79, pp. 71-81, 2018. doi: 10.1016/j.jbi.2018.02.003.

[17] J. A. Maldonado *et al*, "LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics," *Int. J. Med. Inf.,* vol. 78, *(8),* pp. 559-570, 2009. doi: 10.1016/j.ijmedinf.2009.03.006.

[18] P. Stacey and D. Berry, "Design and implementation of an archetype based interoperable knowledge eco-system for data buoys", *OCEANS 2017* - Aberdeen, Aberdeen, 2017, pp. 1-9. doi: 10.1109/OCEANSE.2017.8084936.

[19] P. Stacey and D. Berry, "Interoperable Ocean Observing using Archetypes: A use-case based evaluation", *OCEANS 2018* – Charleston SC, USA, 2018. doi: 10.1109/OCEANS.2018.8604834.