

2017-6

Design and Implementation of an Archetype Based Interoperable Knowledge Eco-system for Data Buoys

Paul Stacey

Institute of Technology, Blanchardstown, paul.stacey@tudublin.ie

Damon Berry

Technological University Dublin, damon.berry@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/engscheleart>



Part of the [Computer and Systems Architecture Commons](#), and the [Other Computer Engineering Commons](#)

Recommended Citation

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

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact yvonne.desmond@tudublin.ie, arrow.admin@tudublin.ie, brian.widdis@tudublin.ie.



This work is licensed under a [Creative Commons Attribution-Noncommercial-Share Alike 3.0 License](#)

Design and Implementation of an Archetype Based Interoperable Knowledge eco-System for Data Buoys

Paul Stacey
Department of Engineering
Institute of Technology Blanchardstown
Dublin, Ireland
paul.stacey@itb.ie

Damon Berry
School of Electrical and Electronic Engineering
Dublin Institute of Technology
Dublin, Ireland
damon.berry@dit.ie

Abstract—This paper describes the ongoing work of the authors in translating two-level system design techniques used in Health Informatics to the Earth Systems Science domain. Health informaticians have developed a sophisticated two-level systems design approach for electronic health documentation over many years, and with the use of archetypes, have shown how knowledge interoperability among heterogenous systems can be achieved. Translating two-level modelling techniques to a new domain is a complex task. A proof-of-concept archetype enabled data buoy eco-system is presented. The concept of *operational templates-as-a-service* is proposed. Design recommendations and implementation experiences of re-working the proposed architecture to run on ultra-resource constrained data buoy platforms using *templates-as-service* are described.

Keywords—semantic; archetypes; data buoy systems; Digital Earth; information management; Linked Data

I. BACKGROUND

In 1998 US Vice President Al Gore set out his vision for what he termed “Digital Earth” [1]. More recently, Craglia et al. presented a renewed vision for a Digital Earth leading to 2020 [2], several grand challenges were highlighted. Among these challenges, semantic interoperability between geo-observational systems, have been highlighted as one of the key barriers to a Digital Earth.

Data buoy systems monitoring oceanographic environmental phenomena are important sources of computable data within the Digital Earth paradigm. However, data buoy systems are often built in isolation, and their data representations and associated documentation systems, where they exist, are not adequately designed for secondary use and higher order knowledge generation.

With the emergence of the Open Geospatial Consortium’s (OGC) Sensor Web Enablement (SWE) standards [3], there has been a burst of activity to progress interoperability within the Oceanic observing communities [4]. OGC’s SWE standards when employed, enable a syntactic level of interoperability between reporting systems. However, semantic interoperability remains a work in progress. In [5], Leadbetter et al. demonstrate how SWE based standards can be linked with ontologies in an oceanographic observational system to realise *Born Semantic/Connected* data buoy systems. While ontologies enhance a system’s interoperability beyond the

syntactic level, [6] argues that more advanced frameworks are needed to achieve interoperability at a knowledge level.

In-order to help realise the paradigm shift needed to realise a Digital Earth, the authors have previously proposed that techniques known as *two-level modelling*, developed in the Health Informatics domain to tackle similar problems of how data, information and knowledge concepts are modelled and managed, could applied to the Earth systems science domain [7]; specifically, data buoy platforms [8]. Recently, [9] also acknowledges the potential benefit of a two-level modelling approach to enabling interoperable knowledge sharing in oceanographic systems. To date no major program of work has fully translated an operational two-level information system outside of the health domain. Although the benefits of translating two-level modelling systems design techniques to non-health domains are potentially great, the translation effort is complex, and requires a significant investment in terms of tool development and wider adoption within the community of domain practitioners. In health, two-level modelling has evolved over a 20-year period.

Complex domains such as health and Earth systems science based sub domains’ (e.g. oceanography) knowledge is constantly evolving. Capturing volatile domain specific knowledge concepts in an observational system, such as a data buoy and supporting information management infrastructures, invariably leads to a mismatch between the needs of the domain practitioner and the concepts represented. The core issue is the inflexible representation of concepts and how they are managed over time.

Complex domains are typically composed of two categories or levels of domain concepts. Those concepts that remain stable over a long period of time, and those concepts that are prone to evolution, as the domain knowledge evolves over time.

A. Two-Level Information Systems

Information systems are typical composed of a single level or a singular information model. Beale [10] argues that single level information systems cannot support the evolution of domain concepts, as concepts are captured in a static data model. A two-level approach is proposed. In a dual-model

information system, non-volatile or stable concepts are still represented using traditional data modelling techniques and represented in the first level of a dual-model information system, this level is referred to as the *reference model*. Where a reference model exists, the challenge is then, how are the semantics of the reference model, or the volatile domain specific knowledge concepts, that have not been captured by the reference model, defined?

Beale notes that knowledge level concepts are effectively constraints on the more general, stable domain concepts (captured within the reference model) [10]. In the dual-model approach, the knowledge-level concepts are modelled as a set of constraint statements. In two-level modelling, a set of constraint statements are referred to as an *archetype*. While reference models are typically modelled by computer scientists, using object-oriented techniques; archetypes are developed by domain practitioners, using a community consensus approach.

For a system to capture, store and serve data that adheres to the community knowledge model, the archetype is used as the basis to constrain the production of the information data objects at run-time. The archetype describes the structure, and detail of instantiated records of information. At run-time, archetypes may be represented in memory in an *archetype-enabled kernel*. Archetypes are intended to be maintained using a Web based management, review, validation and publishing library system [11]. Communities of domain experts access and contribute to the archetype management system, taking part in the review and validation process. As domain knowledge evolves, community derived archetypes are also updated. As systems constrain data instances at run-time using archetypes, two-level systems avoid the creeping obsolescence inherent with single-level systems.

B. Semantic Data Buoy Systems

Data buoys can be categorized into several different classes such as surface, sub-surface, near-shore or off-shore. The term *buoy*, typically refers to the float of a *buoy system*. Buoy systems incorporate anchoring, floats and installed instrumentation [12]. Here, the term buoy system is used for a singularly deployed physical float with anchorage and instrumentation, that is both near-shore and of the type surface, with additional sub surface instrumentation.

Data buoy systems are typically technologically constrained systems, with power and deployment location dictating the buoys available computing, storage and communications ability. These resource restrictions, typically prescribe the use of somewhat impoverished methodologies to describe, transport and store resultant observational data.

Wireless sensor networks (WSNs) research, and more recently SensorWebs [13], have helped to contribute to the current fascination with the Internet of Things (IoT). The vision of an IoT, has precipitated an acceleration of

development in sensor based data reporting techniques for ultra-constrained systems. As the effort of realizing core IoT enabling technologies has reached some stability, other constrained platforms, such as data buoy systems can now benefit through the adoption of IoT related solutions.

Previously bare and primitive data descriptions produced by data buoy systems for proprietary reporting systems, can now include rich markup, capturing data provenance at the edge of the network (the data buoy system). Although many IoT enabling technologies are now deployment ready, application level based standards to capture data provenance and allow semantic interoperability are currently only emerging. Semantic methodologies and solutions do not typically deploy well on constrained systems, such as data buoy systems. Enabling “data provenance at the edge” in a standardized way would help to ensure constrained data buoy systems that directly contribute machine-readable and process-able real-time knowledge within a Digital Earth paradigm.

The remainder of this paper describes the ongoing work of the authors to develop a constrained two-level modelling solution, suitable for ocean observing. The following elements are described.

- A proof-of-concept implementation architecture of a proposed translational re-application of two-level systems design approaches from the e-health domain, to realise an archetype-based interoperable knowledge eco-system for data buoy in-situ remote sensing.
- Re-designing the architecture to allow a resource constrained data buoy to be archetype-enabled. A set of pragmatic implementation solutions are presented. The concept of run-time operational templates-as-a-service is proposed, and an implementation is presented.

II. GEO-OBSERVATIONAL ARCHETYPE-ENABLED ARCHITECTURE

In the oceanic engineering domain where in-situ marine based observatories are essential, the representation of recorded observational knowledge is difficult, as it is ever changing and evolving. Modelling, and thus enabling the recording of uncertainty is not readily possible. Traditional database design and indeed object oriented approaches assume a static understanding of entities or classes of information. Therefore, these design methodologies cannot represent the true nature of knowledge within this evolving domain. Over time the chosen data-model becomes outdated. The key challenge for oceanic information management is representation of concepts that are constantly in-flux; along with the difficult task of maintaining a consistent understanding of concepts as they are communicated to different parties.

Static data models and “bare-bones” representations of domain concepts are a core issue. Also, enrichment of data and information with interoperable and computable use-case knowledge; generated through a domain practitioner’s analysis activities, is not generally provided for. Ontologies have been proposed to achieve interoperability of captured domain concepts. However, ontologies alone do not allow for knowledge evolution in a domain in an interoperable way, arrived at by way of community consensus.

In [10], Beale comments that one of the main problems with “standard” models is that they embody no single point of view. Change is not dealt with very well and invariably implementations tend to wander from the standard model to accommodate the peculiarities of the implementation. The OGCs SWE based Observations & Measurements (O&M) standard [14] can be classed as a standard model. [15] also highlights Beale’s issue with standard models within the Ocean observing community, “the design and implementation of the Ocean Sensor Web should maintain a balance between adherence to the GEOSS, OGC-SWE standards, and the concerns of practical and efficient implementation in the ocean observation domain.”

O&M is currently an implementing under the European Union’s INSPIRE directive [16]. However, even the INSPIRE guidelines for use of O&M notes the issue of variance when using standard models, “O&M is a very generic standard, allowing for very different design patterns depending on the domain as well as the Use Cases to be supported. Application of O&M within a technical community requires that the community agree on standard content for the key slots in the model, as well as on required extensions to the base classes provided within the standard. In particular, it is necessary to have standard vocabularies”.

A. Archetypes

The term archetype is generally defined as a universally understood symbol or term. In information systems design an archetype is a set of constraints on a *reference model*. These constraints provide semantic relationships between elements based on knowledge. Using archetypes, an archetype model can be developed that formalizes the volatile knowledge concepts within the 2nd level of a two-level based information system. Archetypes allow for the necessary variability employed by domain practitioners to be managed in an interoperable.

Constraint statements can be captured using the Archetype Definition Language [17]. ADL is best suited to constraining reference models that are very generic in nature. Where a suitable reference model exists, ADL can then be used to constrain the instances of these generic classes to represent specific domain concepts. ADL was developed for the clinical domain. However, ADL can be used to define archetypes for any domain where there exists a formal reference model.

B. O&M as a reference model

The design of the information level using a reference model is well understood. Non-volatile concepts represented in the reference model can be modelled using traditional conceptual

```

...
<xs:complexType name="GeoData_Composition">
  <xs:complexContent>
    <xs:extension base="IDENTITY_ABSTRACT">
      <xs:sequence>
        <xs:element name="archetype_node_id" ... maxOccurs="1" />
        <xs:element name="name" ... />
        <xs:element name="details_Compound" ... />
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="GeoObservation_set" type="OBSERVATION_SET" />

<xs:complexType name="OBSERVATION_SET">
  <xs:complexContent>
    <xs:extension base="ABSTRACT_OBS">
      <xs:sequence>
        ...
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Listing 1. XSD snippet of augmented O&M model with compound/element patterns. The augmented O&M model serves as the reference model within the dual-model approach. The 2nd level is captured using an archetype-model which are constraint statements on the reference model.

modelling approaches. Given O&M’s growing acceptance as a standard for observational systems, O&M has been proposed as a candidate to under-pin a two-level modelling approach for data buoy systems [7] [8] [9].

To derive and extend domain specific concepts a compound/element pattern (Listing 1 and Figure 1) within the reference model is used. The creation of recursive aggregation of objects from the non-volatile concepts is a core requirement for any reference model within a dual-model system. For this work, O&M has been augmented with the compound/element pattern where the authors propose an opportunity for further constraining is needed to full-fill specific use-case needs.

```

[ {
  "id": "identity_model_ref_ID",
  "GeoData_Composition": {
    "archetype_node_Id": "TPOT-OM-
      GeoData_Composition.Weekly_Buoy_Data.v1",
    "name": "Marine_Data_Buoy_Weekly_Report",
    "details_COMPOUND": {
      "details_ELEMENT": {
        "archetype_node_Id": "[at0001]",
        "name": "buoy_location",
        "DATA_VALUE": "53.127, -11.200"
      }
    }
  },
  [{"GeoObservation_Set" : {
    "archetype_node_Id": "[at0002]",
    "name": "Buoy_Instrument_Readings",
    "meaning": "Interval Trig Buoy Multi Instrument Read",
    "details_COMPOUND": {
      "details_ELEMENT": {
        "archetype_node_Id": "[at0003]",
        "name": "triggertime",
        "DATA_VALUE": "2017-01-11T11:40:00.000Z"
      }
    }
  }],
  [{"Observation": {
    "archetype_node_Id": "[at0003]",
    "name": "observation_measure",
    "observedProperty": {
      "archetype_node_Id": "[at0004]",
      "name": "temperature"
    }
  }],
  ...
} ]

```

Listing 2. Extract from a JSON information object instance. Here an augmented O&M data model serves as the reference model, or the 1st level in a two-level model system. Knowledge concepts are captured in an archetype document. Information instances are created from the reference model to ensure interoperability between systems. Use-case specialisms defined in an archetype are bound to the instance information using at-codes. Each at-code acts as an archetype_node_id binding.

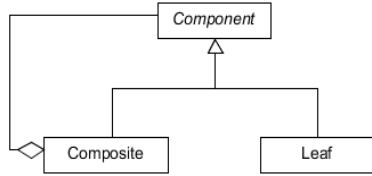


Fig 1. Compound/element pattern embedded within O&M.

C. Enabling knowledge rich information objects

Figure 2 shows an archetype enabled Earth system science based observational system eco-system. Several disparate inter-connected supporting systems are shown. Development of a library of community derived archetypes is supported by the online management system and archetypes are available through an online repository. For any specific use-case, the system builder and associated domain specialists use the necessary subset of archetypes available within the archetype library. Archetypes may be further specialized per use-case, and location, and are combined to produce a set of operational templates (OPT). The in-situ remote sensor system uses these OPTs locally to instantiate information instances, as shown in Listing 2.

Once created, information instances are transmitted to a supporting data-store for persistence. Information objects are instantiated from the reference model only. Information instances form a *directed-acyclic graph* that contain labels or bindings at various points. Bindings are in the form of *at-codes*, and relate the information instance concepts to their knowledge domain specific concept, defined within the ADL based archetype or operational template.

III. CONSTRAINED ARCHETYPE-BASED BUOY SYSTEMS

Creating knowledge rich information objects adds significant additional overhead in terms of processing, storage and transportation. Constrained data buoy systems typically do not have the resources needed to implement a typical archetype-based system deployment. Archetype methodologies have been developed for the health domain, where typically constraints on systems are not of major concern. Scaling issues can be solved through vertical and/or horizontal system scaling. This is not possible on a data buoy system.

Within an archetype-based approach three levels of information exist, *wholly-static* concepts that are captured in the reference model, *quasi-static* concepts which are agreed in each of the archetypes and dynamic data or instance information. Processing and transportation of static data within a constrained system represents wasted resource usage. By identifying static information residing within an archetype'd information instance, and removing this from the information instance, a leaner information object can be realized.

Most observational system should only need to report timestamped DATA_VALUES and the archetype to which the instance is bound; and to which it may be validated against. Presentation layer applications using data instances can later reconstitute the semantically rich information using a similar knowledge framework as shown in Figure 2.

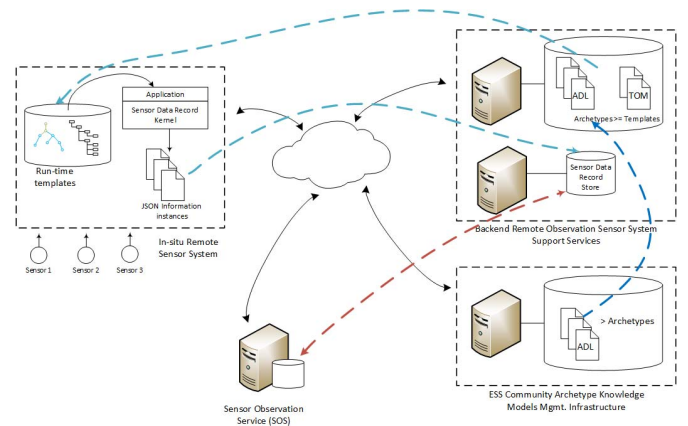


Fig. 2. Overview of a two-level model support observation sensor system architecture. The additional processing, storage and communication load has been found to be prohibitive for deployment across many data buoy platforms. Run-time templates need a kernel to run on the data buoy platform.

Archetypes follow a tree like structure derived from the compound/element patterns inherent in the underlying reference model. Data instance structures must also follow the same tree structure of the underlying reference model, and associated archetype. As a result, two-level system data instances are a specialized type of graph data. Recently recorded observational data instances may only represent a simple node within the larger complex instance data graph. Fragmentation of an information instance temporarily as a distributed graph, or federated graph, with the bulk (wholly-static and quasi-static nodes) of the information instance residing in a resource rich backend infrastructure, an observational system need only process and transport a minimal data node within the overall data graph. This approach is analogous to the Linked Data [18] approach developed in recent years to realise the semantic Web. Using Linked Data approaches, a fragmented archetype'd information instance can be hosted across a supporting knowledge eco-system.

Archetype based systems are designed to enable the creation of knowledge rich interoperable documentation of domain use-case knowledge. The creation of information usually results in appending information to a document. Distributing the information instance between the in-situ observational system and the back-end knowledge framework reduces the potential significant overhead on constrained observational systems mandated by an dual model approach.

A. Linked Data

Linked Data is an approach for exposing, sharing and connecting structured data using URIs and RDF [18]. The core principles of Linked Data provide the basic recipe for connecting data using Web technologies. Structured data (as opposed to unstructured) refers to data with a high level of organization, such as information residing within a relational database. Structured data markup is a text-based organization of data that is included in a file served from the Web. Linked Data techniques use the generic graph-data model of RDF to structure and link data within a Linked Data approach.

Linked data patterns have been used to demonstrate the Linked Data Ocean concept [19]. Linked data patterns are typically supported using RDF. RDF is an XML based syntax. XML is a powerful language for defining rules for the encoding of documents. However, XML is generally not suited to constrained observational systems, due to its verbosity and the complexity of XML parsers. The JavaScript Object Notation (JSON) is a simple standard for the exchange of hierarchically structured JavaScript objects. JSON parsing is more efficient than XML and results in smaller exchange overhead. JSON-LD [20] is a standard designed to serialize RDF using JSON. JSON-LD is a concrete RDF syntax, and so a JSON-LD document is both an RDF document and a JSON document and correspondingly represents an instance of an RDF data model.

B. Archetypes & RDF

The RDF data model is composed of atomic data entities referred to as *semantic triples*. A triple is composed of three nodes within the RDF graph, and codifies a statement about semantic data. Triples of this type are the basis for representing machine-readable knowledge. An RDF graph can be visualised as a node and directed-arc diagram in which each triple is represented as a *node-arc-node* link (*Subject - Predicate - Object*). RDF creates a graph structure to represent data. Serializations of RDF such as JSON-LD allow the markup of data instances using a structured data graph. RDF does not describe how the graph structure should be used. RDF schema (RDFs) is a schema language that allows information modelers to express the meaning of the RDF graph data. RDF and its schema extension RDFs provide support for distributed information and can be used to realize the data instance fragmentation described above. However, RDF & RDFs do not provide the same semantic modeling capabilities as a reference model with an associated constraining archetype. The Ontology Web Language (OWL) provides additional vocabulary and semantic formalisms to RDF/RDFs. For example OWL provides the `owl:Restriction` construct.

OWL provides rich semantics that forms useful support for a federated data paradigm such as Linked Data. To enable the power of a two-level information system design approach within a constrained buoy system, the authors propose the fragmentation of archetype'd data instances using a Linked Data approach. Fragmentation can be realized within the dual-model approach by employing Semantic Web technologies and techniques.

Lezcano et al. have shown how archetypes can be translated automatically into OWL, to enable a reasoning engine based on archetypes [21].

C. ADL to OWL to JSON-LD

Reference [22] provides a succinct introduction to the steps necessary to translate archetypes represented in ADL to OWL. The Artemis project [23] developed a framework to map archetypes between different standards. A syntactic transformation of (ADL-defined) archetypes into OWL format was produced. However, the Artemis framework requires a manual mapping to take place. The Poseacle project [23] also

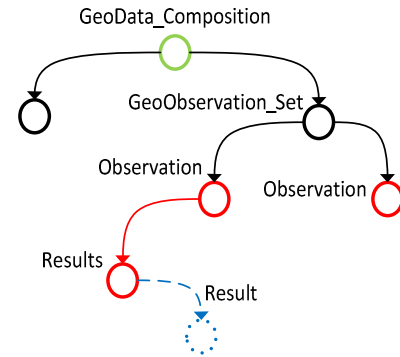


Fig 3. Archetype'd information instance graph representation. The Result node contains the dynamic information that is observed from a data buoy systems. The graph is formed using a Linked Data approach.

provides a semantic transformation of ADL archetypes into OWL.

The Born Semantic approach described in [4] uses the O&M JSON encoding OM-JSON [25] to support a Linked Data approach. The process used is to overlay OM-JSON onto JSON-LD, this allows an RDF inferred graph to be created. Here the authors propose that the ADL defined archetype, or operational template serves the function of OM-JSON proposed for Born Semantic systems in a more flexible way, while realizing the greater benefits of two-level modelling.

The archetype approach has been designed to append data to documents rather than replace or delete. This works well for the approach presented here where the JSON instance shown in Listing 2 is coerced to the JSON-LD format. The JSON-LD inferred RDF graph is composed of *tripified*-data. Triples serve as the basis upon which fragmentation of the information object can occur. Figure 3 illustrates the approach. The inferred graph in Figure 3 is made up of node-arc-node structures. Each node within the graph represents an entity, which can hold any number of attributes. In JSON an attribute is a key-value pair. A triple contextualizes a node, forming a relationship based on a predicate. For example, *Observation - has - Results*; *Results-contain-Result*. Using JSON-LD each set of key-value pairs (node) can be located on a different physical data-store, within a distributed or federated information system, similar to the “shards” concept used in MongoDB. The distributed graph data approach means that a constrained observational system, such as a data buoy must only serve the necessary key-value pairs of a *Result*, once context for that result is provided, or once the result node is *tripified*.

A. Information object fragments & JSON-LD

JSON-LD introduces the `@Context` concept, which is used to define the vocabulary binding for the data concepts used in the JSON-LD document. The context is also a set of rules for interpreting a JSON-LD document. Here the authors propose that JSON-LD context serves to enable the binding of a graph node to the information instance graph hosted on the backend supporting infrastructure. A context can be directly embedded within a JSON-LD document, or as in this case put into a separate document and referenced (shown in Listing 3 and

```

{
  "@Context" : {
    "obj_store" : "coap://tpot.arch-dev.ie/obj_store/",
    "obj_id" : {
      "@id" : "obj_store:obj_id",
      "@type" : "@id"
    },
    "at0002" : "obj_id:at0002/",
    "at0004" : "at0002:at0004/",
    "at0008" : {
      "@id" : "at0004:at0008",
      "@type" : "@id"
    },
    "DV" : {
      "@id" : "at0008:#at0009",
      "@type" : "@id"
    },
    "resultTime" : {
      "@id" : "at0008:#at0010",
      "@type" : "@id"
    }
  }
}

```

Listing 3. Extract from a JSON-LD representation. Information instance fragments are bound to archetypes/OPTs via the @Context. The @id represents the parental information instance of this observation_set. Where at0004 refers to an observation_set with the readings for at0008 (temperature) which is an observation fragment belonging to the sensor_data_record of {object_id} defined by the archetype {opt_id}. The URI fragment, denoted by the # symbol, denotes this the end of the URI path. This is defined by the last aggregation level within the reference model

Listing 4). Here the context is used to link the data instance data to the actual instance hosted on the server.

To reduce the size of the graph node, key-value pairs are represented using the at-codes defined within the archetype (Listing 3). In [26] Sundvall et al. have shown how archetype-based health record systems can be implemented through the application of a REST architecture. In the approach described, a similar methodology is employed to allow the binding of graph nodes to URIs; Listing 3 illustrates this. The context (Listing 3) defines keys (of a key-value pair) and their corresponding context within a specific data graph. JSON-LD context definitions are hosted on the backend-support services infrastructure (Figure 2). Contexts are created from OWL representations of ADL based operational templates. Contexts are exposed using the RESTful architectural approach via a URI. This allows a Result node (Figure 3) to maintain its context within the data graph.

Listing 4 shows the resulting node representation which a data buoy system must adhere too. In this simple example, a data value (DV) key has the value 10.23. This data value is bound the JSON-LD context definition to its meaning using URIs composed of at-codes. At-codes are defined within the archetype (not shown here). Table 1 shows the triple JSON-LD based representation of the value. When the data buoy system transmits the result, the supporting backend infrastructure can process JSON object (or information instance fragment) using

```

{
  "@Context" : "coap://tpot.arch-dev.ie/microcontexts/{microCtxId}",
  "obj_id" : "{sdr_object_id}",
  "@id" : "at0008",
  "DV" : "10.23",
  "resultTime" : "<time_stamp>"
}

```

Listing 4. Extract from a JSON-LD representation of Result (Figure 3).

TABLE 1. A triple representing a temperature reading.

Subject	Property	Value
coap://tpot.arch-dev.ie/obj_store/{obj_id}/at0002/at0004/at0008	coap://tpot.arch-dev.ie/obj_store/{obj_id}/at0002/at0004/at0008/#DV	10.23

the JSON-LD context. The result of the backend processing step results in an information instance shown in Listing 2.

IV. IMPLEMENTATION & EVALUATION

A. Runtime Operational Templates as a Service

A core principle of the approach presented is to enable the fragmentation of archetype-based instance data between a constrained system (data buoy) and backend supporting infrastructure and services. A RESTful architectural style has been adopted to enable the Linked Data paradigm. Fundamental to information instance creation in an archetype enabled system are operational templates, and a runtime template kernel. The federated graph approach described above requires a novel template kernel to support the creation of valid graph data nodes on the data buoy and the backend. The concept of Operational Templates as a Service (OPTaaS) has been developed to support the overall approach and facilitate interactions between a micro-kernel and the federated template kernel. Figure 4 shows the interactions between the data buoy systems and OPTaaS component. RESTful interactions within a constrained environment require a great deal of overhead, and may not be possible using traditional methods. For this work the Constrained Application Protocol (CoAP) [27] is used to support a constrained RESTful approach.

B. Constrained Application Protocol (CoAP)

To support web services running on platforms with very limited resources the IETF formed the Constrained RESTful Environments group (CoRE) [28]. CoRE has been tasked with

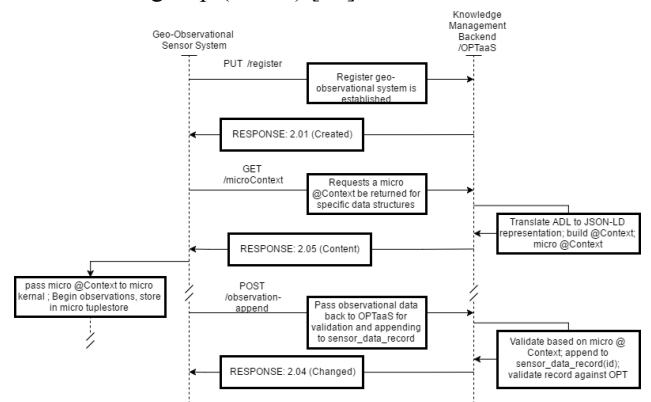


Fig. 4. RESTful interactions between a data buoy and the operational template as a service (OPTaaS). The CoAP protocol is used for message exchanges. The OPTaaS holds the runtime templates and builds a fragment template as a micro @Context. The micro @Context is cached by the observational sensor system and used to perform preliminary JSON-LD processing prior to posting to the OPTaaS web service. The OPTaaS holds a run-time template for the observational system and performs full validation of the information instance as they are received.

developing a framework for deploying web services to constrained environments, such as sensor nodes. In the CoRE framework, a network of nodes called devices interact. Devices are responsible for one or more resources, which could be a representation of sensors, actuators, and combinations of values or other information. Devices in the network can send messages to each other to request, query and publish data. As part of the overall effort to enable constrained RESTful environments, CoAP was defined.

CoAP is a specialised Web transfer protocol for use with constrained nodes and networks. CoAP provides a request/response interaction model between application endpoints. Unlike HTTP based protocols, CoAP uses UDP as its transport layer and employs a simplified re-transmission mechanism. CoAP is designed to easily interface with HTTP for integration with the Web with very low overhead and simplicity for constrained environments.

C. Evaluation System Overview

A data buoy software system architecture has been defined (Figure 5) to evaluate the described technique. The goal of the initial evaluation is to verify the methodology described, with a core requirement to reduce the size of the data instance required on the constrained system, without comprising the knowledge infrastructure. The backend remote observation support infrastructure is hosted on a server running Ubuntu Linux. The data-buoy test-rig was developed using an ARM 1GHz Cortex A8 processor based board.

A test archetype was developed for proof-of-concept implementation. An XML serialization of the O&M data model was produced. The LinkEHR editor [29] was used to constrain the information model further for the implementation. An ADL representation of the test archetype was produced, and stored within a simple archetype store. Community derived archetypes are hosted in the archetype repository in ADL format (Figure 5). Operational templates are used to further specialize archetypes for specific use-cases. For this implementation, operational template are assumed to be equivalent to the serialized archetype, i.e. no further specialization or constraining has been performed. The OPTaaS component requests the conversion of an OPT for use within the constrained data buoy test rig system. The Validation & Converter component retrieves the template from the template store in ADL format. This template is then converted to OWL format. The ADL file was translated to OWL using the technique described in [21]. The library Owl2jsonld [30] is used to produce a JSON-LD context from the OWL translation. The resulting JSON-LD representation was manually fragmented to produce a micro-context for the graph node *Result* shown in Figure 3. The micro-context store is made available via an URI.

A basic data buoy OPTaaS client application was created using node.js. A minimal backend supporting infrastructure was developed. The OPTaaS server was also implemented using node.js. The OPTaaS client and server both use the node.js based coap library node-coap [31]. A basic runtime kernel and validator was developed, based on the openEHR Java Reference Implementation [32]. The runtime template

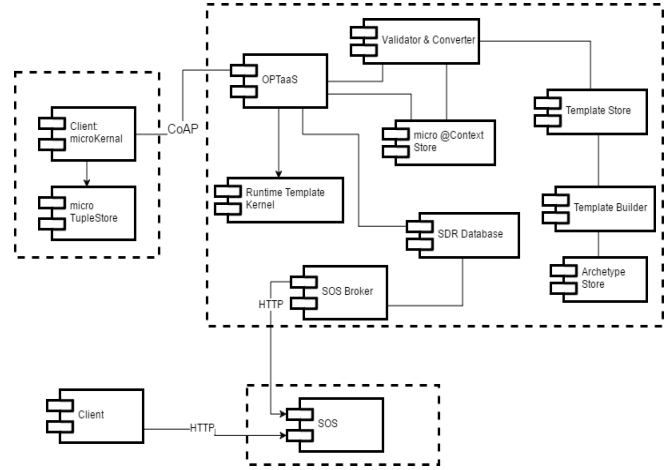


Fig 5. Evaluation software system architecture

kernel component is used by the OPTaaS to process JSON-LD observations and resolve the triples to an RDF store. The OPTaaS server interacts with the sensor data record store (SDR Datastore) via a call to localhost. Apache JENA [33] is used as the SDR Datastore.

V. DISCUSSION & FUTURE WORK

Node.js is used to implement many of the system components. This has allowed for rapid prototyping to allow evaluation of the proof-of-concept. However, it was found that a lot of manual steps had to be employed within the process. The evaluation has informed a set of work packages for future development. The openEHR Java Reference Implementation is specifically designed for openEHR archetypes. The LinkEHR editor is a multi-reference model archetype editor. LinkEHR developers have announced that LinkEHR will be made available as an open source project in the near future. For the current work, archetypes are created manually using LinkEHR, an open-source version of LinkEHR would greatly enhance the development described. The proof-of-concept work has also allowed the authors to further specify the components necessary for further constraining of the system in terms of technical constraints; these are outlined below.

D. Ultra-constrained considerations

Semantically annotating captured data at source is problematic in constrained systems. Born connected system mechanisms are computationally expensive, and in a resource-constrained environment, this may not be possible. Preliminary evaluation of the described technique has shown that semantically rich data objects can be supported using a Linked Data approach. However, our evaluation is not ideal. As Pottie observed in [34], *Every bit transmitted brings a sensor node one moment closer to death*.

The use of URIs to semantically enrich data objects can present an unacceptable overhead in some constrained environments. URI lengths are in general too long for packets in a constrained communication environment directly. The specified mmessage size for a CoAP payload should be less than 1024 bytes to avoid IP fragmentation.

Codification of URIs have been proposed to overcome this limitation. The authors are using the experience gained from the described evaluation to further constrain the technique described. The next stage of evaluation is to implement the technique on an ARM Cortex M3 based test rig, with further constraints on communications and power. For this work a Contiki OS [35] based micro-kernel is being developed.

It is noted that triples are the base of the entire RDF knowledge model. Triples can be represented using many different formats. However, many of these are suitable for constrained systems due to computational constraints and limitations on packet size. JSON-LD has been shown to be an efficient serialisation mechanism for RDF based data. However more efficient approaches exist. In [36] the authors have developed a promising approach that fulfills the following criteria: Low memory usage, small message size, type awareness, simple processing and a standardized solution. Their work uses the EXI format for RDF/XML data representation. XML interchange using EXI has been shown to be more efficient than JSON and binary JSON encodings [37].

The ongoing work to translate two-level modelling to constrained Earth system science based observational environment will adopt the concrete grammar approach described in [36] and extend it to help realise a RDF/linked data style for a federated archetype-based instance data. At the time of writing, the W3C Web of Things (WoT) Interest Group have published a WoT Current Practices draft [38], which also provides several promising approaches, useful for the work presented here.

VI. CONCLUSION

Through a proof-of-concept implementation the authors have shown how two-level modelling systems design techniques can be successfully employed in an Earth systems science domain context such as ocean observing using constrained data buoy systems. However, it was found that knowledge frameworks such as two-level modelling add additional overhead in terms of meta-data additional processing for parsing and validation. These additional overheads are prohibitive for deployment on many resource-constrained observational systems, such as a data buoy systems.

It was found that through the combination of RESTful architectures, Linked Data techniques and the implementation of the concept of run-time operational templates-as-a-service, it was possible to significantly reduce the overhead on a constrained in-situ remote observational system, such as a data buoy; while also realising the benefits of a two-level modelling eco-system.

REFERENCES

- [1] A. Gore, "The digital earth: understanding our planet in the 21st century," *Australian Surveyor*, vol. 43, pp. 89-91, 1998.
- [2] M. Craglia et al, "Digital Earth 2020: towards the vision for the next decade," *International Journal of Digital Earth*, vol. 5, pp. 4-21, 2012.
- [3] M. Botts et al, "OGC® sensor web enablement: Overview and high level architecture," in *Geo Sensor Networks Anonymous Springer*, 2008, pp. 175-190.
- [4] J. Buck and A. Leadbetter, "Born semantic: Linking data from sensors to users and balancing hardware limitations with data standards," in *EGU General Assembly Conference Abstracts*, 2015, pp. 3781.
- [5] Leadbetter, J. Buck and P. Stacey, "Practical solutions to implementing born semantic data systems," in *AGU Fall Meeting Abstracts*, 2015.
- [6] P. Diviacco et al, "Bridging semantically different paradigms in the field of marine acquisition event logging," *Earth Science Informatics*, vol. 8, pp. 135-146, 2015.
- [7] P. Stacey, D. Berry, "Applying two-level modelling to remote sensor systems design to enable future knowledge generation," in *IEEE YP Conference in Remote Sensing Abstracts*, Barcelona, 2015.
- [8] P. Stacey, D. Berry, "Towards a digital earth: Using archetypes to enable knowledge interoperability within geo-observational sensor systems design", unpublished.
- [9] P. Diviacco and A. Leadbetter, "Balancing Formalization and Representation in cross-domain data management for sustainable development," *Oceanographic and Marine Cross-Domain Data Management for Sustainable Development*, pp. 23, 2016.
- [10] T. Beale, "Archetypes: Constraint-based domain models for future-proof information systems," in *OOPSLA 2002 Workshop on Behavioral Semantics*, 2002.
- [11] D. Kalra, T. Beale and S. Heard, "The openEHR foundation," *Stud. Health Technol. Inform.*, vol. 115, pp. 153-173, 2005.
- [12] H. O. Berteaux, "Buoy Engineering," John Wiley & Sons, 1976.
- [13] K. Delin, and J. Shannon, "Sensor web: a new instrument concept." *Symposium on Integrated Optics. International Society for Optics and Photonics*, 2001.
- [14] S. Cox, "Observations and measurements," *Open Geospatial Consortium Best Practices Document. Open Geospatial Consortium*, 2006.
- [15] Y. Jiang, J. Li and Z. Guo, "Design and implementation of a prototype system of ocean sensor web," in *Wireless Sensor Network*, 2010. *IET-WSN. IET International Conference on*, 2010, pp. 21-26
- [16] INSPIRE, D2.9 Draft Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development. [Online]. Available at: http://inspire.ec.europa.eu/documents/Data_Specifications/D2.9_O_M_Guidelines_v2.0rc3.pdf (Accessed: 11 April 2017), 2013.
- [17] T. Beale, and S. Heard, *Archetype Definition Language, The openEHR Foundation, London*, 2007.
- [18] C. Bizer, T. Heath and T. Berners-Lee, "Linked data-the story so far," *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205-227, 2009.
- [19] A. Leadbetter et al, "Linked Ocean Data 2.0," *Oceanographic and Marine Cross-Domain Data Management for Sustainable Development*, pp. 69, 2016.
- [20] World Wide Web Consortium, "JSON-LD 1.0: a JSON-based serialization for linked data," 2014.
- [21] L. Lezcano, M. Sicilia and C. Rodríguez-Solano, "Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules," *J. Biomed. Inform.*, vol. 44, pp. 343-353, 2011.
- [22] O. Kilic, V. Bicer and A. Dogac, "Mapping archetypes to OWL," *Middle East Tech.Univ., Ankara, Turkey, Tech.Rep.TR-2005-3* [Online]. Available at [Http://www.Srdc.Metu.Edu.tr/webpage/publications/2005/Mapping_ArchypestoOWLTechnical.Pdf](http://www.Srdc.Metu.Edu.tr/webpage/publications/2005/Mapping_ArchypestoOWLTechnical.Pdf), (Accessed: 11 April 2017), 2005.

- [23] A. Dogac et al, "Artemis: Deploying semantically enriched Web services in the healthcare domain," *Inf Syst*, vol. 31, pp. 321-339, 2006.
- [24] J. T. Fernandez-Breis et al, "Poseacle: An ontological infrastructure for managing clinical archetypes in semantic web environments," in *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems, 2007*, pp. 2299.
- [25] S. Cox and P. Taylor, "OM-JSON-a JSON implementation of O&M," Presentation to the Open Geospatial Consortium Technical and Planning Committee Sensor Web Enablement Domain Working Group, Nottingham, UK, 2015.
- [26] E. Sundvall et al, "Applying representational state transfer (REST) architecture to archetype-based electronic health record systems," *BMC Med. Inform. Decis. Mak.*, vol. 13, pp. 57-6947-13-57, May 9, 2013.
- [27] Z. Shelby et al, "Constrained Application Protocol (CoAP), draft-ietf-core-coap-13," Orlando: The Internet Engineering Task Force-IETF, Dec, 2012.
- [28] Z. Shelby, "Constrained RESTful environments (CoRE) link format," 2012.
- [29] J. A. Maldonado et al, "LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics," *Int. J. Med. Inf.*, vol. 78, pp. 559-570, 2009.
- [30] S. Reyes, owl2json library, [Online]. Available at <http://github.com/stain/owl2jsonld/> (Accessed: 11 April 2017).
- [31] M. Collina, node-coap library, [Online]. Available at <https://github.com/mcollina/node-coap> (Accessed: 11 April 2017).
- [32] R. Chen and G. Klein, "The openEHR Java reference implementation project," *Stud. Health Technol. Inform.*, vol. 129, pp. 58, 2007.
- [33] A. Jena, "A free and open source Java framework for building Semantic Web and Linked Data applications," [Online]. Available at <http://Jena.Apache.Org> (Accessed: 11 April 2017).
- [34] G. Pottie. Casting the Wireless Sensor Net, MIT Technology Review, July 2003.
- [35] A. Dunkels, B. Gronvall and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on, 2004*, pp. 455-462.
- [36] S. Käbisch, D. Peintner and D. Anicic, "Standardized and efficient RDF encoding for constrained embedded networks," in *European Semantic Web Conference, 2015*, pp. 437-452.
- [37] B. W. Hill, Evaluation of Efficient XML Interchange (EXI) for Large Datasets and as an Alternative to Binary JSON Encodings, 2015.
- [38] Web of Things (WoT) Current Practices, Unofficial Draft, [Online]. Available at https://www.w3.org/WoT/IG/wiki/Thing_Description (Accessed: 11 April 2017), 2017.