CEUR-WS.org/Vol-3824/paper10.pdf

Defining Semantics for Digital Twins of Façade Component Testing Facilities

Calin Boje[1*], Nico Mack[1], Sylvain Kubicki[1], Antoine Dugué[2], Pascale Brassier[2]

[1] *Luxembourg Institute of Science and Technology, 5 Avenue des Haut-Fourneaux, L-4362, Esch/Alzette, Grand Duchy of Luxembourg*
[2] *NOBATEK/INEF4, 67 rue de Mirambeau, 64600, Anglet, France*

Abstract

There are numerous ontologies and data models to guide the development and instantiation of digital twins. The way these are defined depends greatly on the use case of the application. Within this article we explore a case study on the Open Source, Open Access, Open Data Building Envelope Testbench facilities, their context and application in industry and how a digital twin can be instantiated on specific semantic layers. The study shows an analysis of existing tools and the ontologies used, in this specific context. The semantic challenge comes in conceptualizing the digital twin for the testing facility itself (static in nature) and the temporary façade elements which are being tested (dynamic in nature), along with their respective sensing infrastructures. This challenge is explained and discussed through the prism of available ontologies, their mapping and interactions to facilitate several use cases. These use cases are intended to capture and delimitate the context of each individual façade element test, to help deliver transparency and more convenience when building client-side applications. Several examples on querying the proposed schema are shown using GraphQL, under the current architecture in place, which consists of a GraphDB backed with Apollo Federation and BEMServer as a data provider. This technical implementation is intended to facilitate modularity of testing, and API transparency to client-side applications, targeted at eventual users of the testing facility digital twin instance. The challenges and limitations of our approach are highlighted and discussed.

Keywords

Digital twin, Testing facility, Façade element, Graph federation, Ontology mapping

1. Introduction

The testing of novel façade technologies in the EU is becoming more streamlined for small and medium enterprises, thanks to the development of several O3BET testing facilities, as part of the METABUILDING LABS[2] project. O3BET stands for Open Source, Open Access and Open Data Building Envelope Testbench, which assumes a 1:1 scale testing of façade elements in real conditions, with constant monitoring of physical parameters to measure the testing conditions and evaluate the real performances of the elements. The Digital Twin (DT) paradigm for this use case is a logical step forward, which was documented in [1]. Semantic web technologies are

---

LDAC 2024: 12th Linked Data in Architecture and Construction Workshop, June 13–14, 2024, Bochum, Germany

[*] Corresponding author.

[†] These authors contributed equally.

calin.boje@list.lu (C. Boje); nico.mack@list.lu (N. Mack); sylvain.kubicki@list.lu (S. Kubicki);
0000-0002-5150-9355 (C. Boje); 0000-0003-2985-0378 (S. Kubicki);

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[2] https://metabuilding-labs.eu/

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

a key step to integrate the various sources of information, as well as provide the means of describe the context of tests. Similar DT approaches were investigated by several recent studies, but with different use cases, most notably looking at smart buildings using a modular graph integration [2], or a fault detection use case for building DT monitoring [3]. A logical pattern emerges, with the inclusion of the Building Information Model (BIM) in various formats (sometimes undergoing several transformations), the representation of the sensor network, monitoring events, alerts.

The particularities of the O3BET combine a smart facility or building with several integral parts (façade elements) being temporary, as these are dynamically changed (mounted for testing and unmounted at the end) across the building lifecycle. Thus, it's not just the building components which are usually static in nature (or permanent) that change at small intervals (from weeks to months), but also the sensing infrastructure, which needs to be adapted to each testing requirement. The O3BET-DT is a strong candidate for advanced DT services, where simulation and prediction models bring extra value to the testing experience [1], [4]. This would enable a more consolidated testing approach, which can deliver more transparency to clients, as well as streamlined testing results, context comparison and meta-analysis of testing results.

Within this paper we introduce a preliminary ontology schema which would facilitate O3BET use cases, by combining known ontologies within the built environment. The initial testing developments were carried out using GraphDB as a back-end data integration provider in native Resource Description Framework (RDF), BEMServer[3] as a data provider for the sensing infrastructure in place, with a GraphQL endpoint for client applications. The METABUILDING LABS O3BET network accounts for the fact that each testing facility could be operated by different actors, which in turn could use different sensing infrastructure and software systems along the way. At the same time, a uniform way of delivering the testing is needed which is the key application of the O3BET-DT, where semantics play a vital role in information aggregation and contextualization. These aspects place restrictions on the O3BET-DT system architecture, which we argue can be overcome using a semantic web approach. The key outcomes of this research paper are the ontology models emulation to the real-world use cases, with their benefits and limitations. The modelling rationale is presented and discussed along with examples.

The paper is structured as follows: background is covered on key ontologies used, their interactions in section 2. Section 3 provides an overview of the development methodology of the system, and the modular graph approach. Section 4 shows several examples on querying the system with specific use cases. Finally, the benefits, limitations and future work are highlighted in the final section.

2. The context and available ontologies

2.1. Ontology modelling and references

We adopted a typical ontology development methodology, following steps from NeON [5], combined with an agile development approach in practice. The goal was not to develop a new

---

[3] https://www.bemserver.org/

ontology from scratch, but rather to identify existing reference ontologies and map and connect them. The primary use case is to represent a functional O3BET-DT aggregation of concepts and data, which was done iteratively through expert workshops, as part of the ongoing project. The outcome of these workshops is outlined in section 2.2. The ontology development is a work in progress and will undergo more iterations after initial tests within a deployed system using mock data, and in production during live tests.

As a first step, several reference ontologies from the built environment and the Internet of Things (IoT) were considered: the Building Topology Ontology (BOT)[6] for the building spatial representation; the Industry Foundation Classes (IFC) schema version 4.3 with its OWL representation (IfcOwl)[7]; the Smart Applications REFerence Ontology (SAREF)[8] which is focused on IoT representation, and its extension for the building domain, SAREF4BLDG [9]; the Semantic Sensor Network Ontology focused on defining sensors and their observations, which also includes the SOSA (Sensor, Observation, Sample, and Actuator) for its elementary classes and properties [10]. As a secondary scope, we considered the PROV ontology [11], which models the provenance of things on the internet, and NORIA which is used for representing network infrastructures, incidents and maintenance operations on networks [12]. Support ontologies such as PROPS are also incorporated to deal with properties of objects, units, etc., which are already aligned as part of the Linked Building Data Converter [13]. The vocabularies and structures of the aforementioned ontologies were compared in order to fill the required data representations, with the selected ontologies shown in Table 1, and their interactions are described in section 2.3.

2.2. Context: testing facility and dynamic façade elements

The paradigm of the DT using semantic web has been extensively studied recently, with examples such as [14], [15] for city and city district levels, [2], [16] for building monitoring, or [3] for alert and fault management. [3] proposes machine learning to tag data streams, making sensor data "more informative" on Heating, Ventilation and Air Conditioning (HVAC) systems in buildings. They recommend BRICK[4] and Haystack[5] to achieve this. However, each context is different in terms of ontological modelling choices, due to very specific use cases. This is the case when a highly specialized DT application is preferred, as opposed to a generic one.

To achieve an O3BET-DT implementation, we build upon previously defined specifications, and the available technical capabilities provided by our tools, such as the BEMServer. We can discern requirements categorized across several domain categories:

1) Procedural – what is the process behind the testing of façade elements?
    a. Campaign – a testing process, which is limited in time. A campaign is continuous and should not be interrupted or interfered with during the testing duration.
    b. Procedure – a campaign is characterized by a specific testing procedure.
    c. Actors - in simplistic terms, the facility is controlled by O3BET managers, whilst the tests are conducted for different clients; this can be expanded to

---

[4] https://brickschema.org/

[5] https://project-haystack.org/doc/docHaystack/Ontology

include actors involved in testing procurement, quality assurance, etc; Each client is entitled to access to the context and data of his/her testing cell, but not the others'.

    d. Modular testing - once a test is finalised, the façade element is demounted, and replaced with a different element for the next test. This implies the end of one campaign, and the beginning of another.

2) Spatial – what is the facility spatial structure and division during testing?
    a. Facility - The testing facility is a small-sized building, with several identical thermally insulated testing cells to enable rigorous behavioural comparison.
    b. Cells - A cell is a self-contained space, insulated from the facility and the other cells.
    c. Guard Zones - A control zone between the cells is defined, to ensure control over the testing conditions for each cell.
    d. Façades - A cell provides the testing means for one façade element for a pre-defined period of time.
3) Equipment and sensing – what are the components set in place for testing?
    a. Facility sensors – the testing facility, its spaces and cells are monitored by sensors.
    b. Cell actuators (optional) – a testing cell can include specialised equipment to control indoor air conditions (temperature, humidity) using actuators.
    c. Dynamic façade sensors – the tested element sensors are specific to each procedure, and will be installed for each campaign.
    d. Dynamic façade elements - the structure, shape and material composition of facades change with each campaign.
4) Measurements and physical properties
    a. Raw data - sensed data is stored locally on site, but also streamed to external cloud systems for post-processing and analysis.
    b. Data processing – raw data is checked, cleaned and pre-processed with specialized tools and algorithms.
    c. Physical properties - the façade element properties can be calculated based on measurement conditions.
    d. Fault detection – reading anomalies (e.g. measurements acquisition stopped, failure of sensors, out of bounds values etc.) are identified during the data processing stage, which are registered and reported.
5) Virtual system boundaries – what are the different contexts to be maintained and represented virtually?
    a. Facility DT – the testing facility DT is similar to a building DT, with specific monitoring in place, but no inhabitants, it's life cycle undergoes systematic structural reconfiguration.
    b. Façade DT - a tested element DT context is defined and influenced by its parent facility.
    c. Inter-DT interactions - The O3BET facility DT is considered to interact with the façade element DT.
    d. Campaign archiving – the element, its sensors measurements across the entire context of the testing process should be safely stored and made available on request for future work.

e. Notification system – the fault detection on the measurement process should react and notify in real-time to ensure quick curative interventions for testing fault diagnosis.

2.3. Ontologies: scope analysis

The overall preferred approach is to define an ontology schema suitable to describe and use this context within a software system architecture, which needs to aggregate several concepts, data sources, in different formats. The source format needs to be integrated and served to dedicated applications via a data federation methodology.

From the categories of context described in section 2.2, we identified we considered several domain ontologies (enumerated in section 2.1). Table 1 shows the initial set of ontologies selected at the time of writing, still a work in progress. The ontology design intent is to rely as much as possible on existing ontologies, which are maintained and in use, as well as those which already have defined mappings with other domains. Where a mapping was not possible, additional ad-hoc concepts were added, as highlighted in Figure 1. We also show equivalents for certain concepts, from other ontologies which were considered, for the core concepts, as a reference.

Table 1
List of selected ontologies for the O3BET-DT and their requirements coverage

Prefix	Namespace	Coverage
bot	https://w3c-lbd-cg.github.io/bot/	Spatial, Element
core	https://saref.etsi.org/core/	Sensing, Equipment, Measurements
ifc	http://standards.buildingsmart.org/IFC/DEV/IFC4_3/OWL#	Element, Components, Properties,
prov	https://www.w3.org/TR/prov-o/	Process, Actors
noria	https://w3id.org/noria/ontology/	Diagnosis, Fault Detection
props	http://www.w3id.org/opm#	Properties

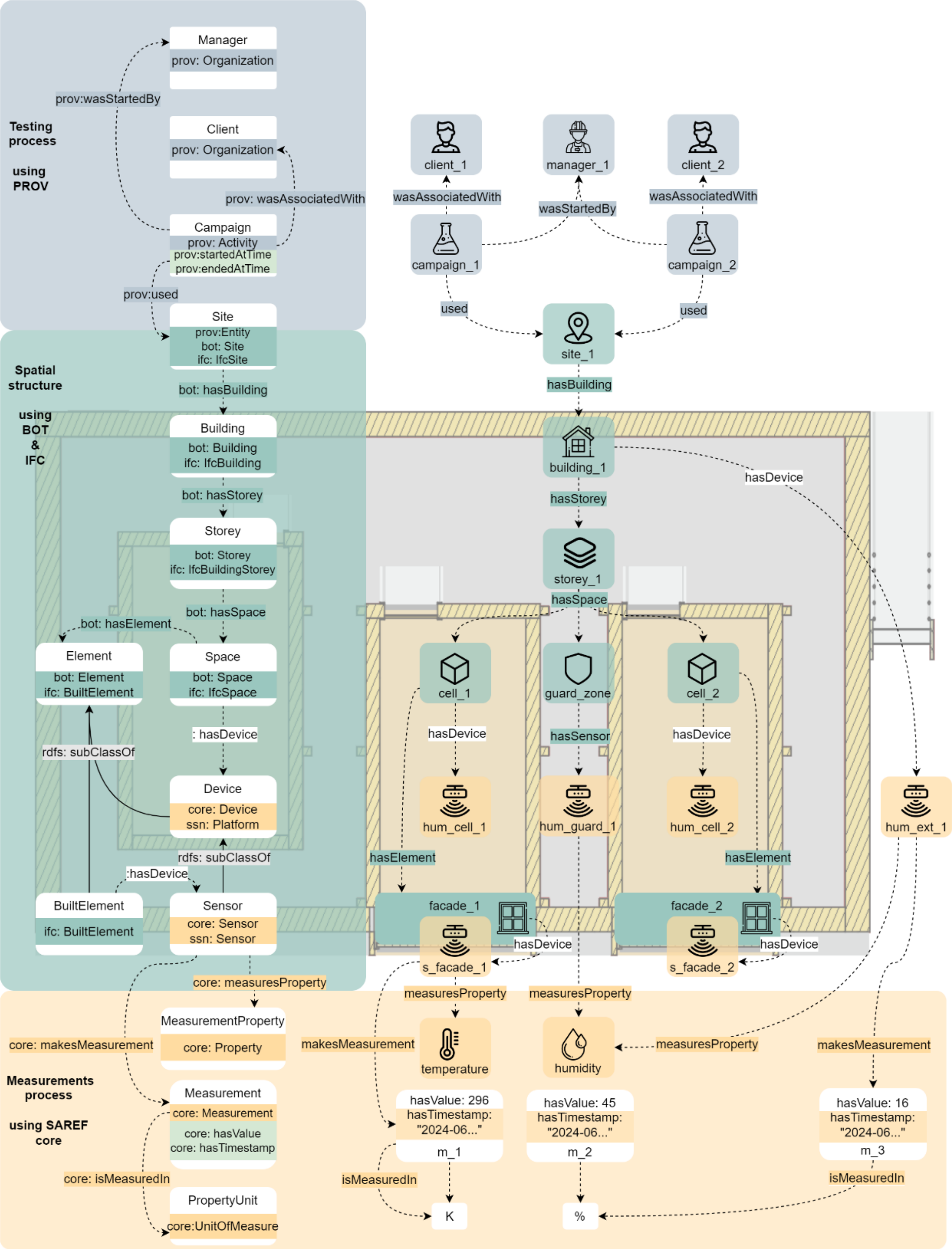


Figure 1: Schema mappings between selected ontologies and their application on the O3BET-DT use case.

In Figure 1we show the typical configuration and represent core concepts, from the testing process (using PROV), the facility (spatial structure using BOT), the sensing infrastructure in place with measurement concepts (using SAREF). For example, the *bot: Element* is a good generalization of building components, but a way to explicitly distinguish between any device (sensor or actuator in our case) and another built element type is necessary. Thus, we adopted the *ifc:BuiltElement* class, as per the new specification of IFC4.3 schema, as a subclass of *bot:Element* (as mentioned in the BOT-IFC alignment module). The *saref:Sensor* (subclass of saref:Device) is also a subclass of bot:Element. We also consider the case where a built element has a sensor attached, in the case of the façade element.

The SSN/SOSA ontology was also considered. However, SAREF adopts a perspective closer to IoT, by considering devices which suits the use case of a DT better, in this case. The SAREF4BLDG extension is not needed, and BOT provides a more flexible representation of the spatial structure, to switch between sensing elements/spaces or the building context more easily.

The PROV ontology is used to generically describe the origin of data in interactions with actors, as a consequence of activities. The actors can range from software systems to people and organizations. The key concept we adopt from PROV is the *prov:Activity*, which represents a time determined action or event, which can adequately represent the concept of a testing campaign. Thus, as shown in Figure 1, the O3BET owner is an actor able to commence the testing via *prov:wasStartedBy*, which then provides access to each client's own campaign via the *prov:wasAttributedTo* property. Additionally, the activity being characterized by a time interval, we are able to restrict the entire context (sensor measurements) for this one interval when data is federated from external timeseries.

The IFC schema is considered here as a reference via IfcOwl, but not functionally within the RDF datasets. To avoid exporting out-of-scope component properties, we filter the IFC model and export a subset, with identifier mappings in place. For example, the IFC Globally Unique Identifier plays a key role in mapping, which allows us to identify the BIM sensors in 3D space, or which can facilitate an enrichment of the BIM at a later stage, when new properties about the façade element are computed thanks to testing campaigns. Thus, avoiding unnecessary triples which would otherwise clog up the data pipelines, following a more modular approach, similar to [2]. The properties of the building elements are considered aligned with the PROPS ontology.

Another important aspect of O3BET-DT is fault management. [3] show examples on how to add annotations to measurements, in case these are suspected as faults. However, there is a gap in ontologies that deal with fault management in the IoT domain. Although research in the area are prevalent, with several examples for sensors in smart homes [17], and most notably HVAC systems [18] [3]. However, many of these ontologies are not available or maintained, and not connected to the typical IoT domain ontologies, like SAREF/SSN. The unconventional way of using BRICK schema, shown by [3], does not use a specific class for fault or anomaly detection, but rather a generic annotation class. This makes it undistinguishable from other annotation types. The more recent NORIA ontology deals with anomaly detection in ICT systems, in a generic way, outlined for an anomaly detection and root cause analysis use case [19]. It reuses BOT classes, such as *bot:Element* and *bot:Space*, which makes it convenient to incorporate and align with other domain ontologies identified for this use case. NORIA describes concepts such as *noria:Event*, *noria:EventRecord* and *noria:TroubleTicket*, to keep track of the issues and related concepts. These are considered to be managed collaboratively within a complex system of actors and resources, and it describes well what a DT system might encapsulate. Its key alignment with BOT is through using the *noria:Resource* class, a subclass of *bot:Element*, which represents things such as *saref:Device*, that can be attributed problems via *noria:EventRecord* (which can include a message), which can be audited and issued a *noria:TroubleTicket*. This in turn would be solved by an external actor (such as the O3BET manager). This is useful in keeping track and providing transparency on issues encountered during the testing process. The application of this ontology is more generic, ranging from devices fault management collaboratively between actors and organizations, to cyber-security use cases. Thus, we can only envisage in reusing a small part of it for the time being. The focus of this article is on the O3BET physical configuration and federation of measurement data itself.

3. O3BET-DT system testing

Within this section, we briefly describe the tools used to develop and deploy a system which uses the schema specified in the previous section, and more importantly, the cunning use of graphs to modularize different aspects and share common concepts and ontology individuals across testing campaigns. We also introduce the concept of archiving the context of the tests, when the façade element DT is put offline.

3.1. System setup

Many data aggregation of IoT with BIM exist, as reviewed by [20]. A common approach for semantic web applications is to define a modular graph structure, based on the data source, and connect them, as demonstrated by [2]. A building components graph is coupled with the same buildings' IoT network graph, and references to retrieve measurements from a dedicated database. It is possible with existing ontologies, such as SAREF or SSN/SOSA to write knowledge graphs at a very high level of detail. However, in practice it's hard to control the source data format for the O3BET use case, and data extraction and reformatting to RDF triples or large sensor data might incur lag, which is not ideal for DT applications. A data federation approach is preferred to avoid data transformation from timeseries into RDF. This approach has become common practice when linking data, where the identifiers of sensors in dedicated databases provide independent access points to the data, and a secondary query is set in place to retrieve data points at specific times, as shown by [2], [3], [20] to name a few examples.

Our current testing setup uses an Ontotext GraphDB with Apollo Federation in place[6]. The GraphDB hosts the merged ontology schema (shown in section 2) and provides a GraphQL to the client-side application for visualization of the data together with the BIM model. The GraphQL endpoint is transparent to the client application, and one unified schema is served. The Apollo Federation fetches the sensor data in the background from another external dedicated system – the BEMServer.

---

[6] https://www.ontotext.com/products/ontotext-platform/

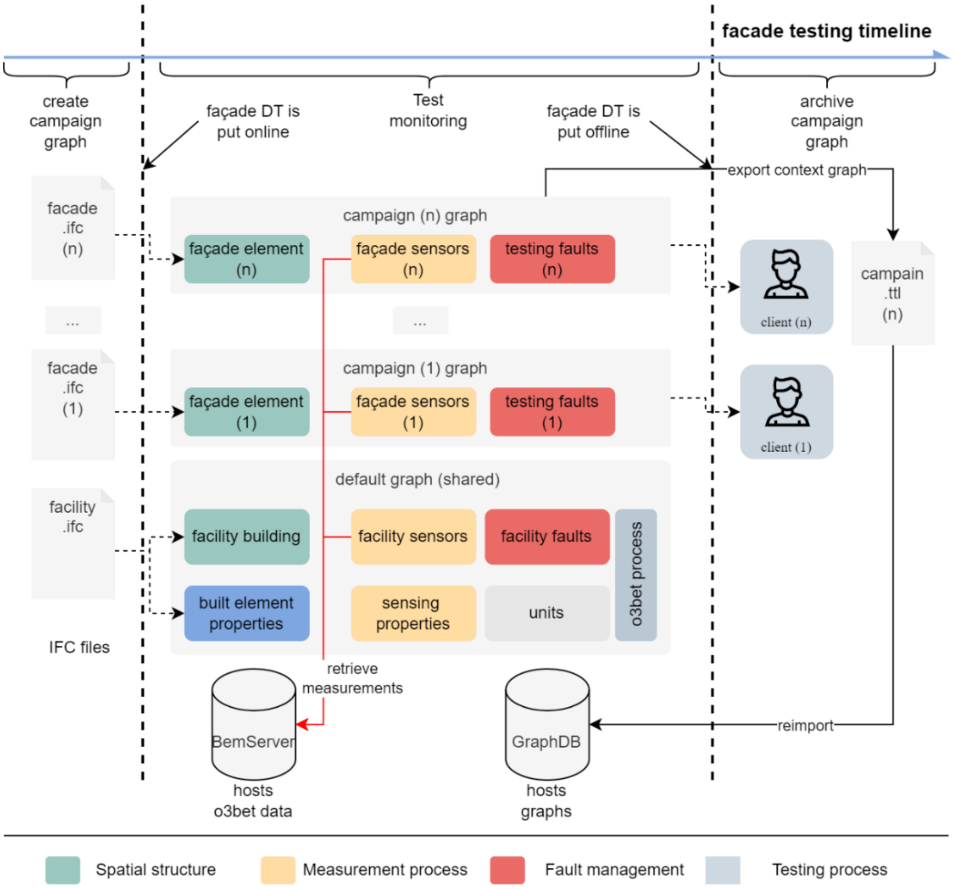


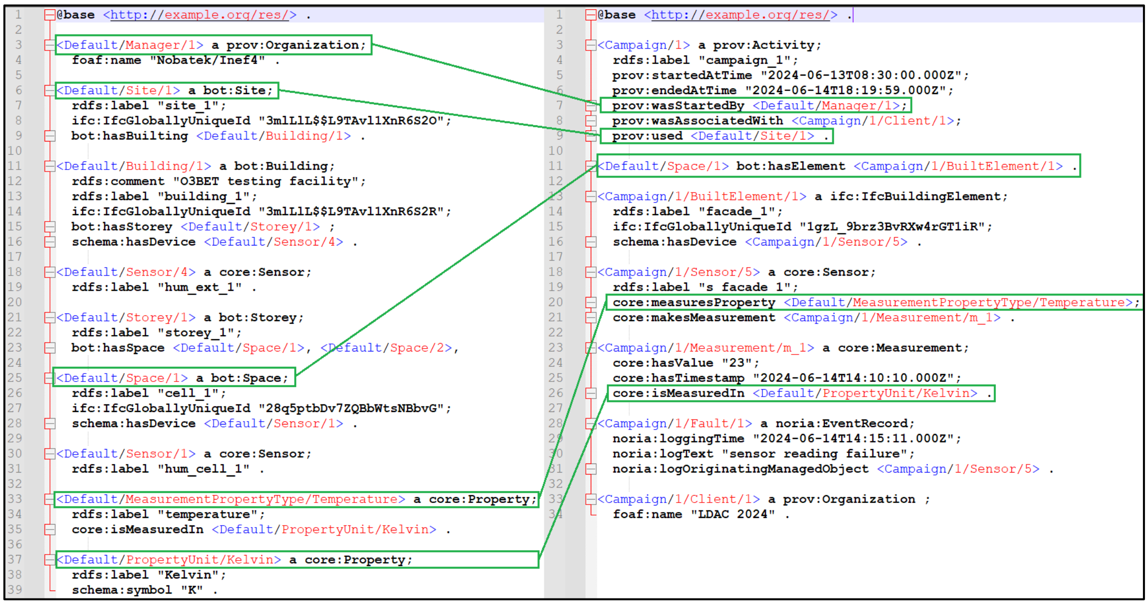
Figure 2: Data flows within the test environment, with federation, import and export per testing campaign.

3.2. Modularization of graphs

As suggested in Figure 2, we define the context of each testing campaign into a single graph. The sensor data, and BIM files however, are external sources, fetched on request. Each campaign represents a self-contained graph, with its unique Uniform Resource Identifier (URI). This provides a modular approach to testing, which serves multiple purposes: (1) the functional requirement to deliver a data set about the campaign context at the end of the tests; (2) a common data template which describes the context and (3) enables methods of comparisons between campaigns. To facilitate this process across the testing life cycle, we consider that each campaign context is created from BIM files and additional information; data is gathered during the testing period, and then they are archived at the end of the test, as shown in Figure 2. We consider that during this process, the testing facility DT (the building) remains online, whilst the façade element DT (the component) can go over several life cycles. For example, once a new campaign starts, the tested element is demounted from the building, and the sensor readings are stopped for that respective cell and component. Then, a new test is being prepared. This happens on a per cell basis. According to Figure 2, we keep the context of the facility in the default graph, whilst each campaign graph is created separately, with links to common elements, sensors, properties and units for example. At the end of the test life cycle, we can construct the self-contained dataset for delivery to the client, as RDF or other semantic web data formats. A fully constructed graph (instances, connections and sensor readings) can be delivered, or a structured package of combined graphs can be delivered to clients, such as in the methodology proposed by [21].

4. Sample queries on O3BET context

4.1. Campaign retrieval



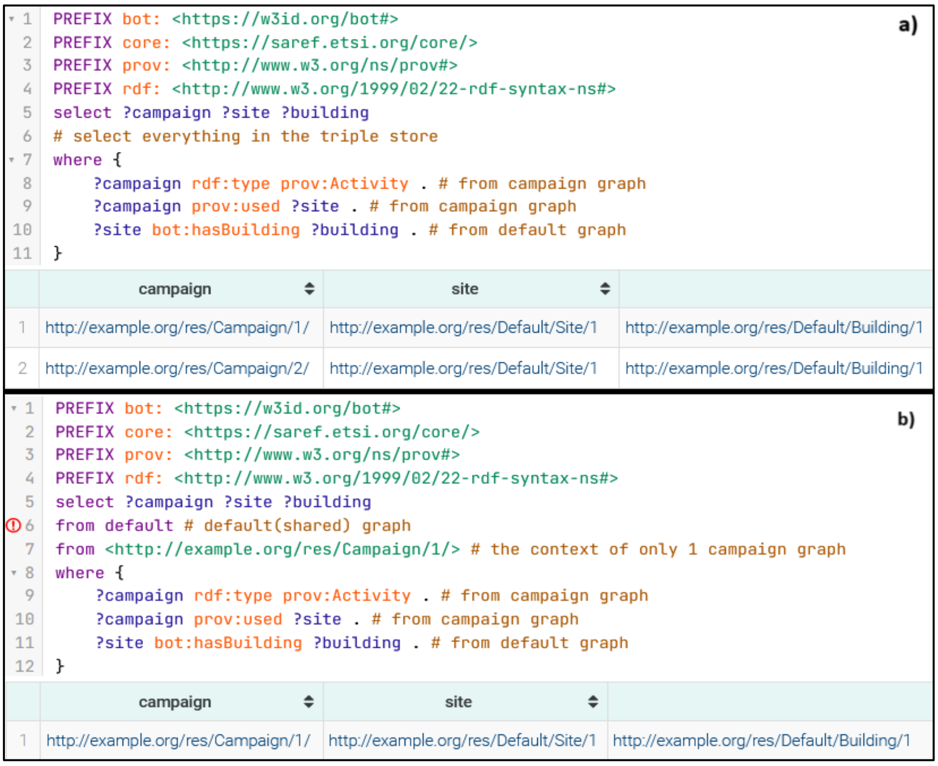
```
@base <http://example.org/res/> .

<Default/Manager/1> a prov:Organization;
  foaf:name "Nobatek/Inef4" .

<Default/Site/1> a bot:Site;
  rdfs:label "site_1";
  ifc:IfcGloballyUniqueId "3mlLlL$$L9TAvl1XnR6S2O";
  bot:hasBuilting <Default/Building/1> .

<Default/Building/1> a bot:Building;
  rdfs:comment "O3BET testing facility";
  rdfs:label "building_1";
  ifc:IfcGloballyUniqueId "3mlLlL$$L9TAvl1XnR6S2R";
  bot:hasStorey <Default/Storey/1> ;
  schema:hasDevice <Default/Sensor/4> .

<Default/Sensor/4> a core:Sensor;
  rdfs:label "hum_ext_1" .

<Default/Storey/1> a bot:Storey;
  rdfs:label "storey_1";
  bot:hasSpace <Default/Space/1>, <Default/Space/2>,

<Default/Space/1> a bot:Space;
  rdfs:label "cell_1";
  ifc:IfcGloballyUniqueId "28q5ptbDv7ZQBbWtsNBbvG";
  schema:hasDevice <Default/Sensor/1> .

<Default/Sensor/1> a core:Sensor;
  rdfs:label "hum_cell_1" .

<Default/MeasurementPropertyType/Temperature> a core:Property;
  rdfs:label "temperature";
  core:isMeasuredIn <Default/PropertyUnit/Kelvin> .

<Default/PropertyUnit/Kelvin> a core:Property;
  rdfs:label "Kelvin";
  schema:symbol "K" .
```

```
@base <http://example.org/res/> .

<Campaign/1> a prov:Activity;
  rdfs:label "campaign_1";
  prov:startedAtTime "2024-06-13T08:30:00.000Z";
  prov:endedAtTime "2024-06-14T18:19:59.000Z";
  prov:wasStartedBy <Default/Manager/1>;
  prov:wasAssociatedWith <Campaign/1/Client/1>;
  prov:used <Default/Site/1> .

<Default/Space/1> bot:hasElement <Campaign/1/BuiltElement/1> .

<Campaign/1/BuiltElement/1> a ifc:IfcBuildingElement;
  rdfs:label "facade_1";
  ifc:IfcGloballyUniqueId "1gzL_9brz3BvRXw4rGT1iR";
  schema:hasDevice <Campaign/1/Sensor/5> .

<Campaign/1/Sensor/5> a core:Sensor;
  rdfs:label "s facade 1";
  core:measuresProperty <Default/MeasurementPropertyType/Temperature>;
  core:makesMeasurement <Campaign/1/Measurement/m_1> .

<Campaign/1/Measurement/m_1> a core:Measurement;
  core:hasValue "23";
  core:hasTimestamp "2024-06-14T14:10:10.000Z";
  core:isMeasuredIn <Default/PropertyUnit/Kelvin> .

<Campaign/1/Fault/1> a noria:EventRecord;
  noria:loggingTime "2024-06-14T14:15:11.000Z";
  noria:logText "sensor reading failure";
  noria:logOriginatingManagedObject <Campaign/1/Sensor/5> .

<Campaign/1/Client/1> a prov:Organization ;
  foaf:name "LDAC 2024" .
```

Figure 3: Sample TTL statements of the default graph (left) which is shared, and linked statements in the testing campaign graph (right) outlined in green.

The interaction between the different graphs is shown as a sample in Figure 3. The separation of scopes from Figure 2 is followed. The linking statements are explicit in the campaign set of graphs, which refer to various concepts from the default graph, such as the building elements, spatial structure, common sensing infrastructure, etc. This avoids redundancy of statements, and allows a separation of datasets from the start. Normally, measurement data is still hosted externally and retrieved via Apollo Federation, on demand. We demonstrate how a campaign is retrieved from its respective graph, in union with the default graph for setting the scope, in Figure 4(a) without specifying context, and (b) with restricting the context to the key graphs. Thus, querying data about a single campaign is restricted to its own scope. In the case where we want to compare measurements across campaigns, and compare different elements, we can expand the selection of named graphs using SPARQL.

4.2. Comparison of values



a)

```
PREFIX bot: <https://w3id.org/bot#>
PREFIX core: <https://saref.etsi.org/core/>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select ?campaign ?site ?building
# select everything in the triple store
where {
    ?campaign rdf:type prov:Activity . # from campaign graph
    ?campaign prov:used ?site . # from campaign graph
    ?site bot:hasBuilding ?building . # from default graph
}
```

	campaign	site	
1	http://example.org/res/Campaign/1/	http://example.org/res/Default/Site/1	http://example.org/res/Default/Building/1
2	http://example.org/res/Campaign/2/	http://example.org/res/Default/Site/1	http://example.org/res/Default/Building/1

b)

```
PREFIX bot: <https://w3id.org/bot#>
PREFIX core: <https://saref.etsi.org/core/>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select ?campaign ?site ?building
from default # default(shared) graph
from <http://example.org/res/Campaign/1/> # the context of only 1 campaign graph
where {
    ?campaign rdf:type prov:Activity . # from campaign graph
    ?campaign prov:used ?site . # from campaign graph
    ?site bot:hasBuilding ?building . # from default graph
}
```

	campaign	site	
1	http://example.org/res/Campaign/1/	http://example.org/res/Default/Site/1	http://example.org/res/Default/Building/1

Figure 4: Sample SPARQL on retrieving campaigns (a) unspecified, (b) named graphs.

```
query SampleQuery($graph: [String!]) {
  sensor(from: $graph) {
    rdfs_label
    makesMeasurement {
      isMeasuredIn {
        id
      }
      hasValue
      hasTimeStamp
    }
  }
}
```

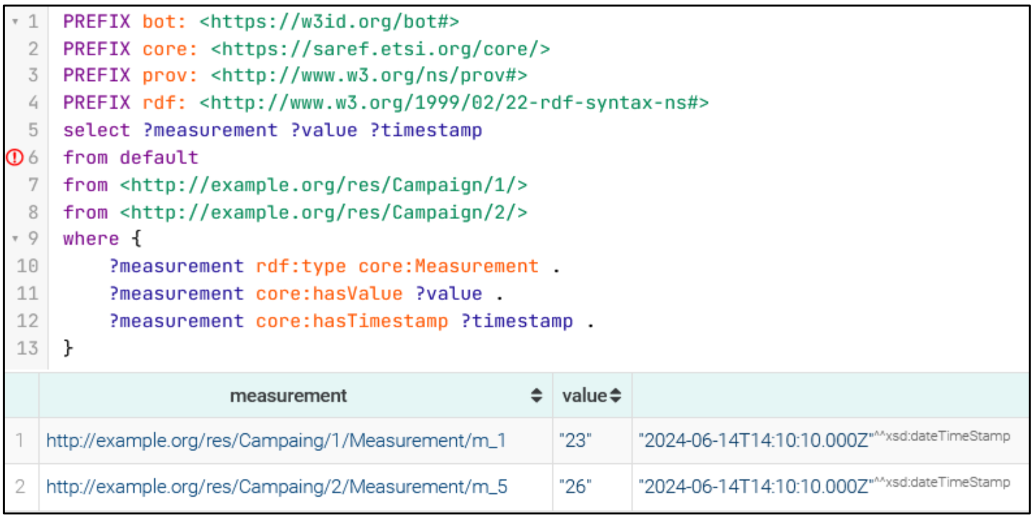
QUERY VARIABLES

```
{
  "graph": "http://example.org/res/Campaign/2/"
}
```

```
{
  "data": {
    "sensor": [
      {
        "rdfs_label": "s_facade_2",
        "makesMeasurement": [
          {
            "isMeasuredIn": {
              "id": "http://example.org/res/Default/PropertyUnit/Kelvin"
            },
            "hasValue": "26",
            "hasTimeStamp": "2024-06-14T14:10:10Z"
          }
        ]
      }
    ]
  }
}
```

Figure 5: GraphQL on finding a sensor and its measurements within the context scope of a single campaign.

On the client-side application, such as a web-service where the instance of the façade element DT is visualized, we aggregate the data around the campaign graph, which we retrieve using a GraphQL end-point, as shown in Figure 5. The advantage lies in more flexible data structures on the client side. GraphQL, however, only allows the inclusion of one named graph in a request for a concept at a time, which can bring certain limitations. This is shown in Figure 5 where we restrict the scope to the campaign_1 graph, via a variable binding. If we want to scope in on two distinct campaigns (and their respective graphs) we would need to run two separate queries using GraphQL. The same can be done in SPARQL using one request, as there is no limit on the named graphs, as shown in Figure 6, where we retrieve the sample measurements with a scope on several named graphs.



```
PREFIX bot: <https://w3id.org/bot#>
PREFIX core: <https://saref.etsi.org/core/>
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
select ?measurement ?value ?timestamp
from default
from <http://example.org/res/Campaign/1/>
from <http://example.org/res/Campaign/2/>
where {
    ?measurement rdf:type core:Measurement .
    ?measurement core:hasValue ?value .
    ?measurement core:hasTimestamp ?timestamp .
}
```

	measurement	value	
1	http://example.org/res/Campaing/1/Measurement/m_1	"23"	"2024-06-14T14:10:10.000Z"^^xsd:dateTimeStamp
2	http://example.org/res/Campaing/2/Measurement/m_5	"26"	"2024-06-14T14:10:10.000Z"^^xsd:dateTimeStamp

Figure 6: SPARQL on finding a sensors and readings from multiple scopes for range comparison.

5. Conclusions and future work

The O3BET process leverages on the DT concept and promises a more open way to test façade components, by providing access and transparency of the process to small and medium enterprises which innovate and want to test new technologies and evaluate their performances in real conditions. The implementation of such a concept needs to consider several requirements and boundaries in terms of tools, data, and semantic models which can represent the entire process.

The proposed semantic model for an O3BET-DT reuses several domain ontologies (BOT, SAREF, PROV, NORIA) to achieve data aggregation for testing of façade elements within a testing facility. It tries to meet several requirements (functional, technical, procedural) and to deliver interactions between the several DTs involved (facility, elements) using modular graphs. The case of accelerated life cycle of components in tandem to the building DT life cycle presents several challenges across the process, some of which were explained and demonstrated above.

The segregation of contexts using modular graphs is somehow in contradiction with a semantic web and linked data paradigm, as we intentionally keep each context in different graphs, but we reuse resources wherever possible to avoid redundancy and compare the testing campaigns. This presents challenges in terms of correctly constructing queries to navigate across graphs in SPARQL or GraphQL, as was shown above.

The contribution to the design and construction industry provided by O3BET generically, is a streamlined way to test and develop innovative façade components, while the DT underpinned by semantics shows the interactions between a building and its components as individual DTs with different scopes, and how they can be monitored and managed across the life cycle (e.g. in renovation cases).

Future work will focus on deployment of our test system into production and integration with site real-time data, where the limits of the system, scalability, and the semantic model would be tested and improved as part of the project. In terms of security, we will integrate these functionalities into other larger digital platforms, as part of the project network, where access rights and authentication will be managed by dedicated services.

Acknowledgements

This paper is published as part of the METABUILDING LABS project, which received funding from the European Union's Horizon 2020 research and innovation program under grant agreement N°953193. The sole responsibility for the content of this document lies entirely with the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] C. Boje *et al.*, "Digital Twin systems for building façade elements testing," Jul. 2023. doi: 10.35490/EC3.2023.240.

[2] I. Fatokun *et al.*, "Modular Knowledge integration for Smart Building Digital Twins," 2023. [Online]. Available: http://ceur-ws.org

[3] X. Xie, J. Merino, N. Moretti, P. Pauwels, J. Y. Chang, and A. Parlikad, "Digital twin enabled fault detection and diagnosis process for building HVAC systems," *Autom Constr*, vol. 146, Feb. 2023, doi: 10.1016/j.autcon.2022.104695.

[4] C. Boje, S. Kubicki, A. Guerriero, Y. Rezgui, and A. Zarli, "Digital twins for the built environment," in *Buildings and Semantics*, London: CRC Press, 2022, pp. 179–199. doi: 10.1201/9781003204381-10.

[5] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology framework: Ascenario-based methodology for ontologydevelopment," *Appl Ontol*, vol. 10, no. 2, pp. 107–145, Sep. 2015, doi: 10.3233/AO-150145.

[6] M. H. Rasmussen, P. Pauwels, M. Lefrançois, G. F. Schneider, C. A. Hviid, and J. Karshøj, "Recent changes in the Building Topology Ontology," *5th Linked Data in Architecture and Construction Workshop (LDAC 2017)*, no. October, 2017, doi: 10.13140/RG.2.2.32365.28647.

[7] J. Beetz, J. van Leeuwen, and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 23, no. 1, pp. 89–101, Feb. 2009, doi: 10.1017/S0890060409000122.

[8] R. García-Castro, M. Lefrançois, M. Poveda-Villalón, and L. Daniele, "The ETSI SAREF Ontology for Smart Applications: A Long Path of Development and Evolution," in *Energy Smart Appliances*, Wiley, 2023, pp. 183–215. doi: 10.1002/9781119899457.ch7.

[9] P.-V. Maria and G.-C. Raul, "Extending the SAREF ontology for building devices and topology," in *6th Linked Data in Architecture and Construction Workshop (LDAC), CEUR Workshop Proceedings*, 2018, pp. 16–23.

[10] M. Compton *et al.*, "The SSN ontology of the W3C semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, Dec. 2012, doi: 10.1016/j.websem.2012.05.003.

[11] P. Groth and L. Moreau, "PROV-overview," *W3C Working Group Note*, vol. 1135, pp. 881–906, 2013.

[12] L. Tailhardat, Y. Chabot, and R. Troncy, "NORIA-O: an Ontology for Anomaly Detection and Incident Management in ICT Systems," *Semantic Web Journal*, 2022.

[13] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, and R. Klein, "The IFC to Linked Building Data Converter - Current Status," in *6th Linked Data in Architecture and Construction Workshop (LDAC), CEUR Workshop Proceedings*, 2018.

[14] S. R. De Meij, A. J. A. Donkers, D. Yang, and M. Klepper, "Making Urban Energy Use More Intelligible Using Semantic Digital Twins," 2022. [Online]. Available: http://ceur-ws.org

[15] C. Hoare, T. B. M. Alqazzaz, U. Ali, S. Hu, and J. O'donnell, "Development of a National Scale Digital Twin for Domestic Building Stock," 2023. [Online]. Available: http://ceur-ws.org

[16] Z. Chevallier, B. Finance, and B. C. Boulakia, "A reference architecture for smart building digital twin," in *CEUR Workshop Proceedings*, 2020, pp. 1–12.

[17] A. Mallak, C. Weber, M. Fathi, A. Behravan, and R. Obermaisser, "A Graph-Based Sensor Fault Detection and Diagnosis for Demand-Controlled Ventilation Systems Extracted from a Semantic Ontology," in *INES 2018 - IEEE 22nd International Conference on Intelligent Engineering Systems, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 000377–000382. doi: 10.1109/INES.2018.8523895.

[18] N. Moretti, X. Xie, J. Merino, J. Brazauskas, and A. K. Parlikad, "An openbim approach to iot integration with incomplete as-built data," *Applied Sciences (Switzerland)*, vol. 10, no. 22, pp. 1–17, Nov. 2020, doi: 10.3390/app10228287.

[19] E. Pardo, D. Espes, and P. Le-Parc, "A Framework for Anomaly Diagnosis in Smart Homes Based on Ontology," *Procedia Comput Sci*, vol. 83, pp. 545–552, 2016, doi: 10.1016/j.procs.2016.04.255.

[20] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends," *Automation in Construction*, vol. 101. Elsevier B.V., pp. 127–139, May 01, 2019. doi: 10.1016/j.autcon.2019.01.020.

[21] A. Borrmann, J. Abualdenien, and T. Krijnen, "Information containers providing deep linkage of drawings and BIM models," *Proceedings of the CIB W78 Conference*, pp. 832–835, 2021.