A Contextual Ontology for Distributed Urban Data Management

Cathal Hoare
School of Materials and Mechanical Engineering and UCD Energy Institute, University College Dublin, Ireland
Sergio Pinheiro
BAM Ireland, Dublin, Ireland
Shushan Hu
School of Computer Science and Information Engineering, Hubei University, Wuhan, China
James O'Donnell
School of Materials and Mechanical Engineering and UCD Energy Institute, University College Dublin, Ireland

The evolution of ICT in the construction domain has yielded a variety of heterogeneous data sources. While bespoke approaches have been developed to explore data merging for a variety of purposes, few have explored how to develop a multi-purpose information organisation that can be reconfigured on a per project basis. This paper describes an approach that, using a lightweight central server, is used to investigate the effectiveness of loose federations of information sources that together serve the information needs of a project. The central server provides both a common context through which relationships between the information sources can be expressed and a data register to enable information discovery. The paper describes the creation of an ontology to capture this context, and a software architecture to support its use. The efficacy of the approach is illustrated through describing the use of the server for marshalling data used in a renovation project.

1. Introduction

While society faces the significant twin challenges of housing its growing population and addressing climate change, it has, in recent times, been provided with new sources of information - for example, arising from widespread adoption of sensors, or arising from digitisation of city management - to help tackle these crises (Howard et al 2012). The confluence of advances in computing power, sensor technology and machine learning have helped create ever more accurate models that, in turn, inform decision support systems, helping policy makers and others to make better use of limited resources (Reinhart & Davila 2016). However, these advances have not achieved their full potential; in part, this is because of the challenges of managing and integrating these sources of information in order to maximise their combined utility (Curry et al 2013).

Management of this information, in a building and construction, context is complicated by several factors including the longitudinal, dynamic and heterogeneous nature of buildings. Furthermore, digitisation of this information has a high cost while existing in a fragmented value chain (ECTP 2019). The information sources that describe urban areas are deeply domain specific and not designed for interoperability; they are expressed in many formats and produced by diverse software systems. Because these information sources are usually implemented to support one facet of building construction or management, singly, they do not provide value to a broader audience; new utility is created when several of these sources are integrated (Li Y et al 2013), resulting in a rich view of an urban area that can inform decision support systems. Facilitating data exchange and interoperability between systems that use these sources must be carefully managed, especially in projects where development resources are limited, where data is sensitive for

commercial or legal reasons and uses of data over a project's lifespan are not anticipated at its inception (Rezgui et al 2011).

This paper, expands on the work of Hoare et al (Hoare et al 2019) by examining how a loose federation of data sources can serve the information needs of a project. It is shown that the approach is flexible, requiring limited technical effort to integrate data sources for a particular project, allows data sources to be repurposed for other projects, and distributes the cost of data management across the project's participants. In order to investigate the approach, a light weight central server was implemented to:

1. Serve a common context across the project, and so express equivalence and containment relationships between a set of entities expressed by separate different data sources; other information in the sources can be queried through their relationship to these entities. This is used to overcome data exchange problems arising format and representational issues.
2. Provide a data source directory to enable information discovery, and inform project participants as they formulate queries to the system.

The server runs as a cloud service. It is implemented using services and protocols from the semantic web and linked-data domains. The RDF framework is used to define a central context. This captures a series of relationships - defined using OWL - between entities represented in project participants' information repositories - specifically, *contained-in*, *contained-by* and *equivalent-to*. These relationships, agreed by project participants as part of the approaches methodology, serve as keys across which joins between information sources can be made, negating differences in format and structure. These keys can then be exploited through the use of sub-queries in the SPARQL query language. Sub-queries are executed using SPARQL's federated query functionality.

Information sources are not stored on the server. Rather, they remain under the control of the data creator. Information sources are added to a data registry that is part of the server; this records using the DCAT ontology. This arrangement serves two purposes. First, the data creator maintains control on their data, exposing only that which is required by the project and so helping manage commercial and regulatory requirements. Secondly, maintenance of the data and supporting technology remains the responsibility of its owner; this distributes cost of maintenance across the project, and simplifies data access between a project's information contributors. Where the structure of the data changes or a new information sources is added to the project, only the affected SPARQL queries must change. While this would require some technical expertise, the effort is substancially less than if code changes were required, as would be the case were access provided through an API.

The server has been used in several projects. In the case of NewTrend[*], the approach was applied post-hoc to represent information used to serve simulation software that informed decision support for urban renovation. In this case, a multi-scale multi-model data model was established. District areas were represented through CityGML served using the CityDB CityGML tool and custom data representations developed as part of the project, while builds were accessed through IFC managed on per building BIM systems. Data from each source was maintained on their original servers, and served by querying each sources through an agent that queried the source and translated its output to RDF. The core context created a representation of the urban area creating equivalence relationships between CityGML and IFC building objects. A second project, ESIPP[†], used the server to manage a different set of data for a different purpose. In this case, the relationships between the electricity transmission network, distribution network and individual buildings were captured using the model at a national scale. The diverse data sources, were then queried by Ireland's Economic and Social Research Institute (ESRI)[‡] as a single information network to determine the effect green technologies on the electricity network. In this paper, the former use case will serve as an example of the server's use.

The remainder of the paper is organised as follows. In the next section, we briefly examine relevant literature that informs the development of the work described here. Section 3 describes the development of the core spine ontology and provides a technical description of the server's implementation. Section 4 provides a description of an application of the methodology/framework to support acoustic simulations. The paper concludes with a summary

[*] http://newtrend-project.eu
[†] https://esipp.ie
[‡] https://www.esri.ie

of the paper's contribution and a discussion on future work to be conducted as part of the project.

2. Literature Review

Facilitating interoperability of multi-format, multi-scale information sources is a vexing and persistent issue (Abbasabadi & Ashayeri 2019). Construction has proven to be no different, where diverse uses, narrow margins, a lack of cross enterprise standards, and the formations of task specific data silos has created both horizontal (e.g. between functional areas in projects) and vertical (e.g. between individual building representations and neighbourhoods) barriers to data exchange (ECTP 2019). A range of solutions developed inline with software engineering advancements have been developed, including the replacement of monolithic applications, the use of software APIs and data management techniques including semantic and linked-data approaches (Rezgui et al 2011). Even so, many solutions in the literature suggest applications that repackage data and inject them into a purpose specific data management schema (Li Y et al 2013). While valuable cross-domain information may be contained by an expression of that information, the form in which it is expressed will make it difficult to consume for other uses. Not only is it difficult to exchange the information at a process and systems level because of its specificity, the information may also require domain expertise and manipulation that is beyond the scope of members of either community (Pauwels et al 2017).

Broadly, two approaches have evolved for providing data interoperability. These include the 'muiltimodel' and 'Linked Building Data' approaches. The multimodel (MM) approach (Scherer & Schapke 2011), typified by the Mefisto project, creates an overarching model by creating links between disparate model element; this collection of links can be used to query across the entire collection of models. While there is no prescriptive approach to this type of implementation, semantic web and linked data approaches provide a medium that is ideally suited to the approach. The Linked Building Data (LBD) approach explicitly uses semantic web/linked-data approaches to express the data and semantics of AEC related data [§]. Several similar approaches have been reported. IFCOwl [¶] sought technologies such as RDF and OWL to express the IFC schema semantically. Similarly, COINS (Van Nederveen et al 2010) uses semantic technologies that uses geometry as a central reference model (or context) to organise construction oriented data.

The multimodel approach can be lightweight, relatively easy to implement and simple to maintain as it manages the links between data models and not the entire dataset itself. Management of data is generally managed by systems specifically designed to manage those formats. These systems are also likely to be familiar to the data controller for those formats. However, its simplicity – particularly the lack of a central context – makes the evolution of the data schema somewhat unstructured and potentially prone to complexity, making it difficult to maintain over time. Purely semantic approaches are deeply structured. However, issues persist around expressing data contained in format specific systems in these semantic formats. Of greater concern, adoption of these approaches requires expertise in the area of the semantic web and its formats.

Methods and architectures for integrating cross-domain, heterogeneous data sources using semantic technologies have been a rich avenue of investigation. Semantic approaches have been found to be useful for tackling issues of scalability, as well as catering for situations where the selection of data sources is dynamic (Boje et al 2020). Distributed or federated approaches have also been found to be useful. Mendes De Farias et al (**?**) presented an exploration of the advantages of distributing semantic data sources. It was found that the approach is appropriate for integrating heterogeneous data sources, proving both scalable and flexible.

Recent regulatory and social issues have brought the issue of data access to the fore. Federation is also found to offer benefits in overcoming these. Several research manifestos have been devised to guide the evolution of the semantic web. For instance, (Bernstein et al 2016) has suggested that topics around *lightweight semantics* is one of several key research directions. We embrace this concept in this work by creating a lightweight ontology that can be used to unite other more domain specific ontologies. This not only ensures greater utility for the ontology but also ensures that adopting the ontology is low cost in terms of expertise, systems and work required. In addition, this work seeks to embrace the goals of the EU's Ethics Guidelines for Trustworthy AI

[§] see https://w3c-lbd-cg.github.io/lbd/
[¶] see https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/

(EU 2019) by remaining cognisant of *'realworld'* concerns such as trust and regulatory compliance. By ensuring that data controllers can maintain control of data on their own servers, their users' confidence will not be undermined by disclosures about data sharing. Service providers are able to control what information will be exposed to a project. Furthermore, service providers will be encouraged to participate more widely in projects as they will be able to maintain control over their compliance with regulatory demands such as the EU's GDPR requirements[‖].

Several works have sought to implement federated architectures that are underpinned by semantic techniques. Rasusen et al (Rasmussen et al 2017) presented work on creating task specific knowledge graphs for the AEC domain. The work is underpinned by an ontology that admits data integration and admits parameterised queries. The approach, while in concept, is similar to the work presented here, focuses on a narrower domain and does not support as broad a range of sources as the work presented here. Niknam et al (Niknam et al 2017) propose a modular approach to modelling building information. A core shared ontology that must underpin all ontologies used in the system is used to provide a common structure. SPARQL queries are used to investigate and reason about the data. However, the ontology is limited to building information, and so cannot cater for a wider range of information sources.

This paper proposes that context presents an opportunity for data organisation. All information exists in context – that is a set of physical or conceptual statuses that give meaning to data (Debes et al 2005). it is proposed that the relationships between physical entities, for example, buildings or urban areas provide a context for their associated data. While providing a broad context, it also provides a context that is common to many information sources. Furthermore, when data is diverse and covers a broad range of domains, users must be aided to help them perform information discovery. the remainder of this paper will present and demonstrate a semantic approach and associated server for achieving this.

3. Methodology

The DDIM is composed of three separate processes, illustrated in Figure 1. The first takes multiple information sources and mines

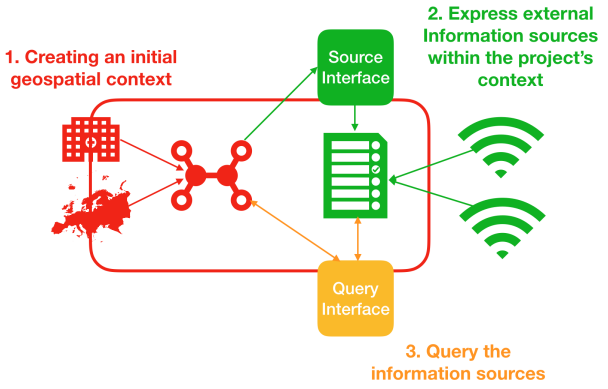

Figure 1. Typical Structure of a data management system managing heterogeneous multi-scale inforamtion

them to populate a contextual model that describes both hierarchical and equivalence relationships between these sources. This creates a contextual spine that, in effect, creates a single information source that allows reasoning across scales and negates the impact of format and other data exchange issues.

The second process makes this model available to project participants. When they publish information, it is expressed in an appropriate context taken from the context model. While this information is hosted on the contributors' own server in RDF format, details about the sources are recorded in a DCAT registery on the DDIM server.

The third and final process is query formulation by project participants interested in accessing the information. These queries are submitted to the DDIM server as SPARQL queries. The creator of these queries can query the context model to understand the structure of the context for the project, identifying both classes[**] and individual instances of interest. They can also query the DCAT registry, using it to access the information sources' schema to understand their structure. Informed, they can formulate federated SPARQL queries to satisfy their information needs. The remainder of this section will describe the development of the context ontology and the implementation of the lightweight server that hosts both the ontology and a data registry to enable data discovery. The server uses linked-data to represent information. A brief summary of this technology is provided next.

[‖] see https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/principles-gdpr$_e n$

[**] *class* in this case refers to the object oriented concept

3.1. The Context Ontology

Linked data is a set of design principles for sharing machine-readable data on the Web [††]. It is made up of named data that can be exchanged and reasoned about. Data can be expressed statically as RDF or dynamically embedded in the format. This data is named using a unique Uniform Resource Identity (URI) that allows it to be found and shared across the web. The data can contain links to other named data sets and meta data about the relationship between these linked datasets. Given a rich enough network of data, a computer algorithm can reason about entities based on the contexts provided by their related data.

While the ability to reason about distributed data is in itself sufficient motivation to embrace the concept of linked data, the schema has further advantages. Because the data is distributed, maintenance is managed by each organisation and its impact on any one participant is reduced; only expertise about data related to an organisation's expertise need be maintained, while this simple structure allows new data to be easily integrated. Furthermore, the data exposed need only be the subset that is relevant and permissible for an organisation's remit. While there is an overhead in re-expressing data in RDF format, the process can be largely automated and supported by mature software tools.

Data Catalogue (DCAT) is an RDF vocabulary that allows reasoning about online data catalogues (W3C 2014). Its use promotes discoverability of data sources and provides a mechanism to promulgate the data source's schema. The vocabulary also provides metadata that can be used to reason about a dataset's provenance. The vocabulary provides a rich set of attributes that can provide context about a dataset and its versions.

This arrangement means that the server manages a reference to some dataset and does not maintain a copy of the data (reducing data duplication and management overheads) as well as ensuring that the dataset itself remains under the control of its owner.

3.2. Development of Common Context Ontology

The NeOn methodology (Suárez-Figueria 2010) was used to develop the context ontology used by the DDIM. The method emphasises the development of ontology networks made up of relatively simple knowledge representations; ideally, these are reused where possible. The methodology develops an ontology over five steps, including requirements definition, terms extraction, ontology conceptualisation, search and selection and implementation.

3.2.1. Ontology Requirements Definition

This phase of development seeks to identify the purpose and scope of the ontology, identify intended end-users, identify uses and list the functional and non-functional requirements.

3.2.2. Purpose, scope and implementation

The purpose of the core ontology is to represent common entities between different information formats that are used to describe a collection of buildings at various scales from district to building components. Once common entities are represented, they can be used as keys in order to form mappings between the schema and admit queries across different formats. Since the combination of different formats represent data at different levels of detail, it can be said to provide a context into which other information sources can be placed and accessed. Finally, since information in different formats can evolve at various rates, the ontology also captures versions of data represented, and so provides a mechanism to synchronise across entity versions. The ontology has a scope that extends through various levels of detail over a district area. The ontology was implemented in RDF and OWL.

3.2.3. Intended End-users

The intended end-users are the technical team that would commission and run the DDIM to support a project and other collaborating teams that would contribute information placed into the context provided by the server. Indirectly, other project stakeholders such as decision makers and planners would make use of the output of simulations and analysis resulting from the use of the ontology.

3.2.4. Intended Uses

The main intended uses of the ontology are to:

- Provide an ontology that expresses common entities across different formats used to represent information about buildings in a district area;

[††] see https://www.w3.org/standards/semanticweb/data

- Provide a context in which other ontologies can be placed, and so create a complex ontology network that describes information about the area, and;
- Support simulations and other decision support processes related to a project.

3.2.5. Identifying functional and non-functional requirements and term extraction

An information system is ultimately defined by its ability to support the activities of its users. The Competency Question (CQ) technique (Gruniger & Fox 1995) is used to identify what actions must be carried out by agents using a system, and the constraints that they must operate under. When expressed within the scope and purpose of the ontology, these questions form a set of functional requirements that inform later steps of the design methodology. For the Core Spine Ontology, some examples of competency questions are:

- What stakeholders will be involved in creating a network of data for a district space?
- What levels of detail will they need to access information at (urban, building, building element, etc)?
- What formats will they use to represent these different levels of information?

The different CQs were categorised by type, for instance, questions about mapping across formats were grouped into the mapping domain. Each CQ was categorised into one of three groups including entity representation, mapping elements and versioning.

The CQs contain concepts that should be managed by the ontology. The extracted terms are *information sources, context levels (data scales), entity, bridging entity, version, agent, building part, context, extension*

3.2.6. Ontology Conceptualisation

The initial conceptualisation of the Core Spine Ontology was drafted to better understand the main classes and the relationships between them. This version of the model was derived from the terms extracted from the functional requirements and the purpose and scope statement. The mapping and entity representation domains were merged since they are closely related; this results in two domains, one to support versioning and one to create a context for the district.

The coarse structure of the ontology states that all entities have either a contained in, contained by or same as relationship. All entities are sub-classes of the entity object, which captures the entity's data format; therefore a bridge or mapping between formats can be formed where two entities have a same as relationship. Not only must entities be expressing information about the same real-world object, they must also be synchronised in terms of version information; currently this is based on information that is manually managed as part of DDIM admin process.

The context component seeks to provide an expression of physical contexts and describes the relationship between these contexts. The model provides unified views of a project's information at various scales; these scales can be traversed, allowing typical functions such as aggregations of information to provide a high level overview. In the case of the NewTrend project, the context hierarchy (determined by the project's needs) included a view of an urban area that contained buildings and building components such as stories and spaces. In the ESIPP project, the hierarchy represented Ireland's transmission network, represented as a hierarchy of transmission nodes, small areas and individual buildings.

When a information provider publishes information, it is expressed in terms of one of these contexts. For example, in the ESIPP project, information about individual transmission nodes are indexed using the unique key of the transmission node represented in the central context model. Similarly, information about a class of entity can also be expressed; for example, in the ESIPP project, archetype information is associated with the building type expressed in the central context model. In all instances, this information is hosted by the data provider. Such a scheme allows new sources of information to be added to the project; the sole requirement is that the information is indexed using a key contained by the central context model.

Each entity is associated with a set of version information that captures change events about an entity through representing a version id, time stamp for time the change event occurred and the agent that made the change.

3.2.7. Ontology Selection

The fourth phase of the NeOn methodology calls for identification of existing ontologies that can be reused in order to reduce the

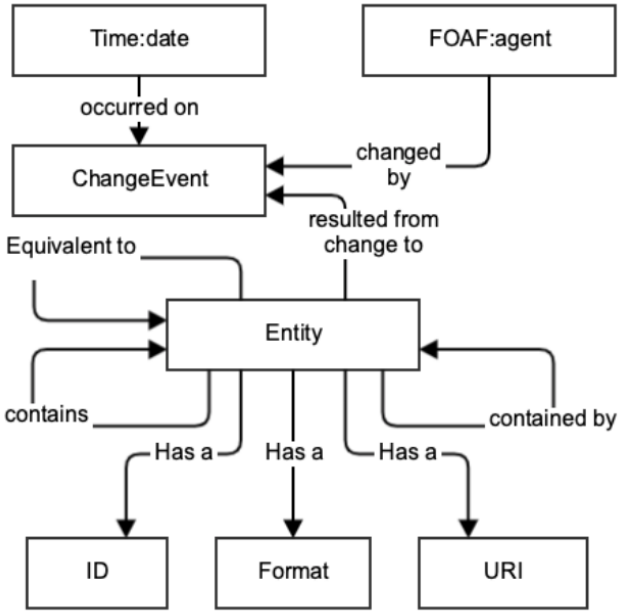

Figure 2. The final representation of the Core Spine Ontology

cost of development. Given the focused nature of the core spine ontology, no applicable third party work was identified. Of course, data sources attached to the schema may use existing ontologies.

The final part of the NeOn Methodology calls for implementation of the ontology.

3.2.8. Defining the Core Spine

The final version of the core spine provided an implementation broadly in accordance with that described in the conceptualisation phase; it is shown in Figure 2. The ontology is deliberately sparse to allow for the adoption of several different source formats.

3.3. Implementing the Server

The DDIM consists of three key functional areas (shown in Figure 3.):

- The context model descibed in the previous section;
- A series of interfaces to allow client software to query (through RESTful or SPARQL based interfaces) and to interact with the notifications system;
- A DCAT compliant data catalogue;

The DDIM server provides a semantic context for building related data, placing it in time and space. The resulting context can be used by clients to query the associated information. The server can be queried through either RESTful APIs or SPARQL query interfaces. A common context is defined and used to populate the cotext model described previously. This context is agreed between organisations participating in the project.

In addition to configuring the server, information providers must make their information available through an interface that supports SPARQL queries. A variety of techniques have been used to achieve this - including publishing information through triple stores such as Apache Jena or GraphDB, adding software agents that express information queried from relational databases as RDF. This information must be expressed using a context represented in the central context model. This requirement has proven to be easily met by data providers, as the context is agreed at a project's kickoff, allowing existing keys used in a data source to be included in the central context model.

Two modes of interaction with the server occur. The first, related to commissioning the server so that it can support a project. This step initialises the central context, supports publishing of data sources and registration of these sources on a data register hosted by the server. Thereafter, the second mode of interaction is prevalent (though new sources can be commissioned at any time), when a client seeks to access the data, they will first query the catalogue for the available RDF schema, and using these, formulate SPARQL queries by interrogating the context model and the information sources described in the catalogue.

Organisations can include their data sources in a project by expressing it in terms of the context model for the project. Once the information has been published, it must be made discoverable. This is done by adding a data entry to the DDIM' data catalogue. This entry is in turn exposed using the DCAT vocabulary. The vocabulary is expressive, allowing details including the title, version, data of publication and modification, data use licence URL and many more attributes to be expressed. This entry is discovered by connecting clients. The structure of the data is provided, and can be used to formulate a SPARQL query.

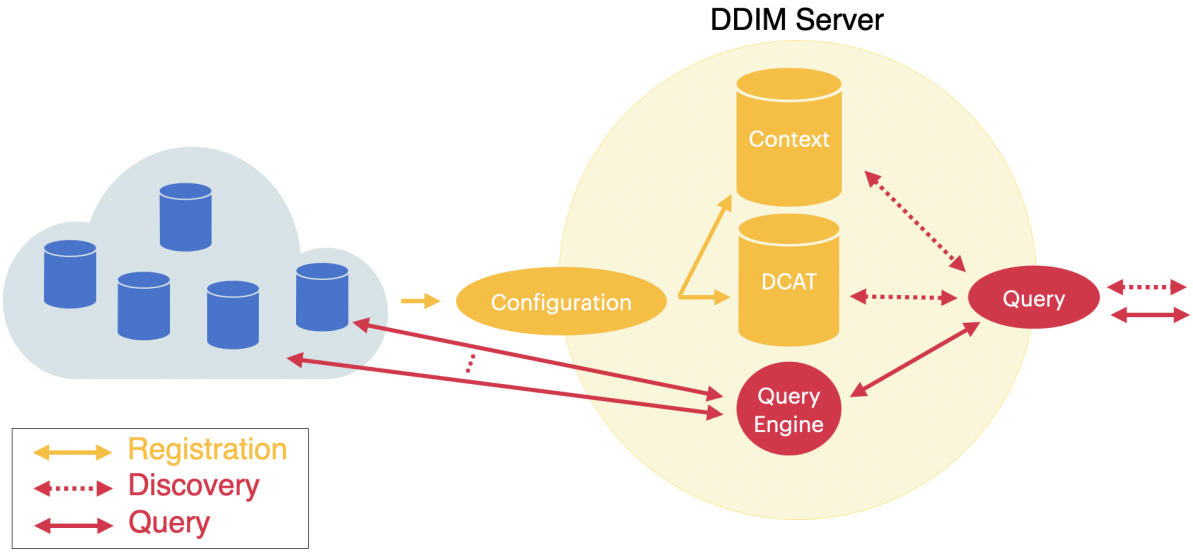

Figure 3. DDIM Server Architecture showing (i) setup steps in yellow (ii) information discovery in dashed red and (iii) information query in solid red

Data access is further controlled through the DDIM by adding access control for artefacts listed in the catalogue. DDIM user accounts are grouped, and access privileges are determined by these.

The central server uses GraphDB as a triple store that contains both the context model and a custom implementation of the DCAT ontology. The server is accessed through a web based graphical user interface that supports both comissioning of the server and subsequent data interrogation. This requires little technical expertise beyond a knowledge of the SPARQL query language. Commissioning of data sources requires technical expertise; it has been found in both the case of the NewTrend data and ESIPP project that this was a discrete task that lasted several days for each source.

4. Scenario of Usage

This section describes the use of the server in a real world scenario. Having previously developed a multi-model central data store for a large collaborative Horizon 2020 project called NewTrend(Maile et al 2018), the approach described here was applied to provide the same services. The overall process used to commission the server will be described.

The project in question sought to support decision support for district area scale renovation by providing energy, thermal and acoustic simulations at both district and individual building levels. A distributed architecture implemented a data collection framework, a simulation *hub* (that contained a suite of simulators) and a results interface that allowed for interpretation of results. All of these components connected to a centralised multi-model data store. The store supported consumption of information from project stakeholders; these sources included urban models in CityGML format, building models in IFC and custom information formats to support the data collection framework. The simulation hub was served this information in a form that it could consume, and wrote the results back to the server where they could be queried by the results interface.

While the approach was successful, several issues arose, including issues around centralising commercially sensitive information, cost of maintenance, and ability to extend the services provided by the server. Because the entire data store was centralised, extensions to support new data requirements was especially burdensome - especially as knock on effects on other components connecting to the server had to be avoided.

This work consisted of five distinct steps:

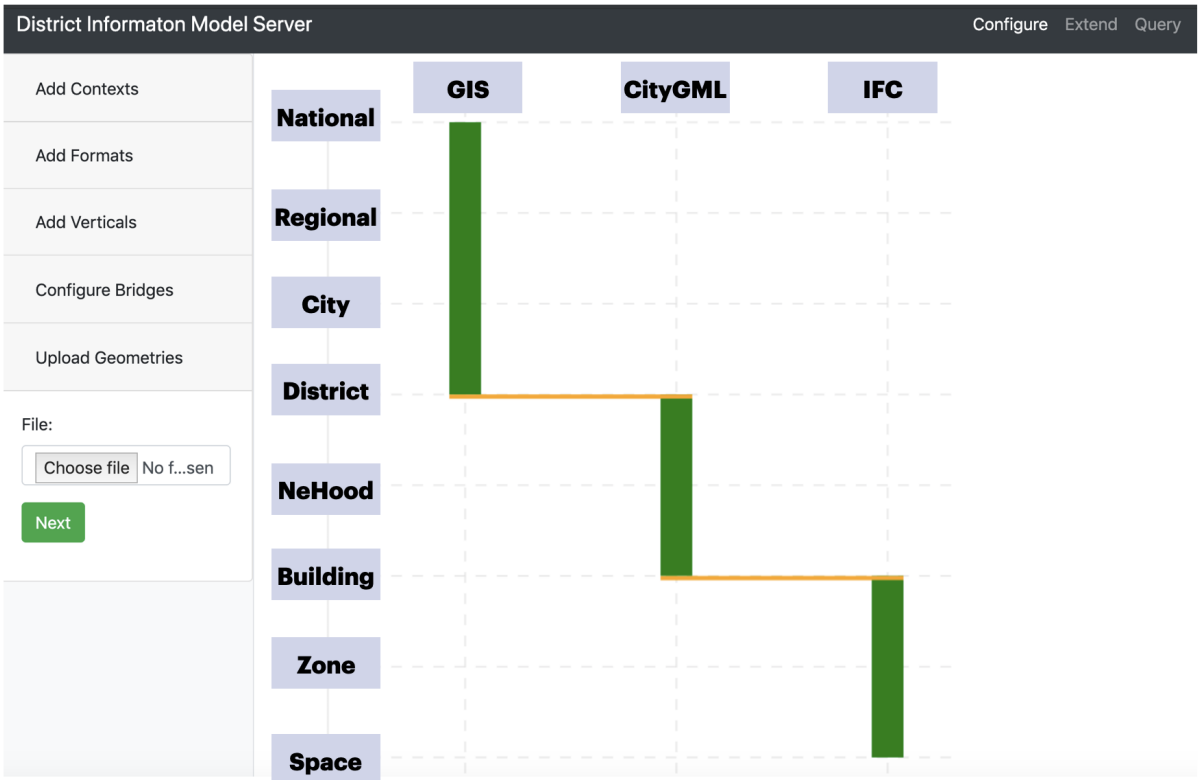

Figure 4. Screenshot of the context definition interface showing context definition created for the NewTrend project's Data

- Setting up and commissioning the server;
- Deciding on the structure of the context;
- Defining the central context model;
- Addition of data source;
- Formulation of queries by project participants.

In the context of the project, the server was set up and administered by a system administrator. While the server can be run on a networked server, extensive support is provided in order to utilise cloud based computing. A virtual machine image is available that unpacks and sets up an Amazon Web Services (http://aws.amazon.com/) instance through the Bitnami service (http://www.bitnami.com). The owner of the service then uses a web browser to complete some initial steps in the setup, including providing a project description and giving other users permissions to interact with the server. The use of cloud services also reduces the overhead required for maintenance as periodical updates are made available by Amazon and Bitnami. Backup and other features are scheduled through a web based graphical interface provided by Bitnami. Typically, an AWS instance to support this project was run for an average of $43 per month, while the Bitnami service subscription cost $47 per month.

Independent of the server's commissioning, a context structure was agreed. This informs the context model creation process. For the NewTrend project, the context hierarchy is as follows:

$$NationalData \rightarrow RegionalData \rightarrow DistrictData \rightarrow BuildingData \rightarrow SpaceData$$

National and regional data was expressed in a GIS based format, while district information was formatted as CityGML and building and space data was expressed in the IFC format. The administrator used a graphical interface, shown in Figure 4 that is served by the server to express the context relationships and populate the context model. This is done in the following steps:

- A list of context levels are defined. In this case, eight were chosen. These appear vertically on the left hand side of the interface.

```
#filename: testareatobuiling.ttl

@prefix dc: <http://127.0.0.1/cathalhoare/TestMallow#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/1999/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/1999/07/owl#> .
@prefix owl2: <http://www.w3.org/2006/12/owl2#> .

dc:contains rdf:type owl:ObjectProperty ,
        owl:TransitiveProperty ,
        owl2:IrreflexiveProperty .

dc:iscontainedby rdf:type owl:ObjectProperty ,
        owl:TransitiveProperty ,
        owl2:IrreflexiveProperty .

dc:isequivilantto rdf:type owl:ObjectProperty ,
        owl:TransitiveProperty ,
        owl2:IrreflexiveProperty .

dc:t1 rdf:type dc:entity ;
        dc:id 220 ;
        dc:format "CityGML" ;
        dc:type "area" ;
        dc:context "area" ;
        dc:contains dc:b7174 ;
        dc:contains dc:b9995 ;
        dc:contains dc:b335 ;
        dc:contains dc:b7254 ;
        ....
```

Listing 1. An fragment of the output produced by the context creation process

- A similar list of information sources is provided. For this project, four sources were used. While int his example, the formats form a contiguous set, this is by no means common.
- A classes that represent physical entities contained by each format are identified. For example, IFCBuildings and IFCSpaces in IFC. One is selected for the extremity of each source (classes can also be selected for intermediate context levels as required). These selections are used to define the contained-in and contained-by relationships in the context model.
- Relationships between endpoints of format lines are then defined. A rule that defines equivalence between formats is defined by selecting one or more attributes contained in each class and defining an equivalence between them. In this case, buildings we defined as equivalent when both the CityGML Building class' address and the address contained by the IFC Building object were found to be equivalent. These relationships inform the equal-to relationships in the context model.

While this task was carried out by the administrator, other project participants' information was published. A variety of techniques were used to achieve this - including publishing information through triple stores such as GraphDB in the case of custom data collected as the project, adding a software agents that express information queried from relational databases as RDF in the case of CityGML and use of a custom software framework that translated queries into low level API calls to query instances of the OpenBIM Server in the case of IFC. This information was expressed as RDF and indexed using class instances that participants had agreed

```
PREFIX a: <http://127.0.0.4/testserv/testaccoustic#>
PREFIX dc: <http://127.0.0.1/testserv/TestMallow#>

select ?buildingarchetype ?strategicnoisemappingvalue where {

    #Get test area
    ?t dc:context "area" .
    ?t dc:id 220 .

    #Get the buildings in the area
    ?t dc:contains* ?buildinginstance .
    ?buildinginstance dc:type ?buildtype
        filter(?buildtype = "Building") .
    ?buildinginstance dc:id ?buildingid .

    SERVICE <repository:accoustic> {
       ?strategicnoisemappingvalue a:aid ?buildingid .
    }
}
```

Listing 2. An example SPARQL query submitted to the project server to retrieve acoustic data for all buildings

would define the context model. Each source was registered on the DCAT registry. This included information about the source's URI and documentation that described the organisation of the data source.

Once the sources that contain context information are made available, the administrator triggered a software program that create an RDF representation of the physical entities in the project. This is stored in the server's triple store and made available for querying. And example of the output, in Turtle format is shown in Listing 1. Here, after the preamble, an example of an entity, an area, is presented. Its format is CityGML, and it contains several buildings. The file contains all instances of relationships between entities as defined by the rules defined by the administrator in the context definition process. For brevity, the file is truncated; these outputs are expansive; in the case of the national grid project, the context definition alone ran to 345MB, detailing the relationships between 154 transmission nodes, c. 16,500 small areas and c. 2 million houses.

The server was now considered commissioned and can be used to access information. In order to formulate a query, a user examines both the contextual model, to understand the context in which information is placed, and the registry, to learn what sources are available, and by examining DCAT entries of interest, their structure and location. The user then formulates a SPARQL query, which is federated across available sources. The query is submitted to the server and results are returned.

Listing 2 demonstrates a query to the server. In this case a project participant seeks to discover the *strategicnoisemappingvalue* for each building in a district. The acoustic information is contained on separate server that is contributed to the project by a technical partner.

The data seeker initially consults the context model to understand what contexts have to be accessed and the DCAT register to understand what relevant data sources exist. Having identified that querying a specific area will, in this instance, satisfy their need, they formulate a query to isolate all buildings in the area. The query continues by making a federated query to the acoustic service to retrieve acoustic information for all buildings identified in the context model. In this case, since GraphDB is used as the triple store, the federated query is submitted through a service configured on the server.

5. Conclusions and Future Work

This paper described an approach that, using a lightweight central server, is used to investigate the effectiveness of loose federations of information sources that together serve the information needs of a project. The central server provides both a common context through which relationships between the information sources can be expressed and a data register to enable information discovery. The creation of an ontology to capture this context was described along with a software architecture to support its use. The efficacy of the approach was illustrated by describing the use of the context model and server for marshalling data used in the NewTrend project.

The overall architecture of the server continues to evolve. Approaches such as peer-to-peer are being investigated in order to establish the optimal deployment solution in terms of technical simplicity, scalability and cost. Improved approaches for adding new sources are also being investigated; in particular, these seek to mitigate the task of adding new information sources.

6. Practical Relevance of the work

The work described in this paper is relevant to others undertaking a project where disparate information sources must be combined in order to inform simulation and other decision support systems. The contributions of this paper will, individually or together, assist new projects that must combine data sources while operating with limited technical and funding resources.

7. Ackowledgements

This publication has emanated from research supported by a research grant from Science Foundation Ireland (SFI) under the SFI Strategic Partnership Programme Grant number SFI/15/SPP/E3125. The opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Science Foundation Ireland.

REFERENCES

[Abbasabadi & Ashayeri 2019] Abbasabadi N and Ashayeri JKM (2019) *Urban energy user modelling methods and tools: A review and outlook.* Energy and Building 161.

[Bell & Kolbe 2017] Bell C and Kolbe TH (2017) *CityGML and the Streets of New York - A Proposal for Detailed Street Space Modelling.* ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. IV-4/W5. 9-16.

[Bernstein et al 2016] Bernstein A and Hendler J and Noy N (2016) *A New Look at the Semantic Web.* Communications of the ACM. 59(9): 35-37.

[Boje et al 2020] Boje C, Guerriero A, Kubicki S and Rezgui (2020) *Towards a semantic Construction Digital Twin: Directions for future research.* Automation in Construction 114

[Rezgui et al 2011] Rezgui Y and Boddy S and Wetherill M and Cooper G (2011) *Past, present and future of information and knowledge sharing in the construction industry: Towards semantic service-based e-construction?* Computer-Aided Design 43(5)502-515.

[Curry et al 2013] Curry E and O'Donnell J and Corry E and Hasan S and Keane M (2013) *Linking building data in the cloud: Integrating cross-domain building data using linked data* Advanced Engineering Informatics. 27(2):206-219.

[Debes et al 2005] Debes M and Lewandowska A and Seitz J (2005) *Definition and Implementation of Context.* Proceedings of the 2nd Workshop on Positioning, Navigation and Communication. Hanover, Germany.

[De Farias et al 2016] Mendes De Farias T, Tarcisio and Roxin, Ana and Nicolle, Christophe (2016) *A Federated Approach for Interoperating AEC/FM Ontologies.* LDAC2016 - 4th Linked Data in Architecture and Construction Workshop. Madrid, Spain.

[ECTP 2019] ECTP (The European Construction Technology Platform) (2019) *Artificial Intelligence – Research and Innovation Needs – Manufacturing, Energy Intensive Industries, Bio-based Industries and Construction.*

[EU 2019] EU (High-Level Expert Group on Artificial Intelligence) (2019) *Ethics Guidelines for Trustworthy AI.*

[Gruniger & Fox 1995] Gruniger M and Fox MS (1995) *Methodology for the design and evaluation of ontologies.* Workshop Notes of ICJAI-95. Montreal, Canada. 1-19.

[Hoare et al 2019] Hoare-C, Ali-U and O'Donnell-J(2019)*Dynamic District Information Server: On the Use of W3C Linked Data Standards to Unify Construction Data.* 2019 European Conference on Computing in Construction Chania, Crete.

[Howard et al 2012] Howard B, Parshall L and Thompson S et al (2012) *Spatial distribution of urban building energy consumption by end use.* Energy and Buildings. 45:141-151.

[ISO 2018] ISO (2018) ISO 16739-1:2018 Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries — Part 1: Data schema.

[Li Y et al 2013] Li Y and García-Castro R and Mihindukulasooriya N and O'Donnell J and Vega-Sánchez S (2013) *Enhancing energy management at district and building levels via an EM-KPI ontology* Automation in Construction. 99:152-167.

[Maile et al 2018] Maile T and Orova M and Ntimos D and Stjelja D and MacSweeney R and Barbagelata M and Asens P and Bazzan E (2018) *DELIVERABLE D6.3: Application of the Methodology and Tool*

[Niknam et al 2017] Niknam M, Karshenas Saeed (2017). *A shared ontology approach to semantic representation of BIM data.* Automation in Construction. 80. 22-36.

[Pauwels et al 2017] Pauwels P, Zhang S and Lee Y (2017) *Semantic web technologies in AEC industry: A literature overview.* Automation in Construction. 73:145-165.

[Rasmussen et al 2017] Rasmussen M, Lefrançois M, Pauwels P, Hviid C, Karlshøj J(2017) *Managing interrelated project information in AEC Knowledge Graphs.* Automation in Construction. 108.

[Reinhart & Davila 2016] Reinhart CF, Davila CC (2016) *Urban building energy modelling – A review of a nascent field.* Building and Environment. 97: 196-202.

[Scherer & Schapke 2011] Scherer RJ and Schapke SE (2011) *A distributed multi-model-based Management Information System for simulation and decision-making on construction projects.* Advanced Engineering Informatics. 25:582-599.

[Suárez-Figueria 2010] Suárez-Figueria MC (2010) *Neon Methodology for Building Ontology Networks: Specification, Scheduling, Reuse.* Universidad Politécnica de Madrid.

[Van Nederveen et al 2010] Van Nederveen S, Beheshti R and Willems P (2010) *Building information modelling in the Netherlands: A status report.* Proceedings of the 18th CIB World Building Congress. Salford, UK.

[W3C 2014] W3C (2014) *Data Cataloge Vocabulary (DCAT). https://www.w3.org/TR/vocab-dcat/.*