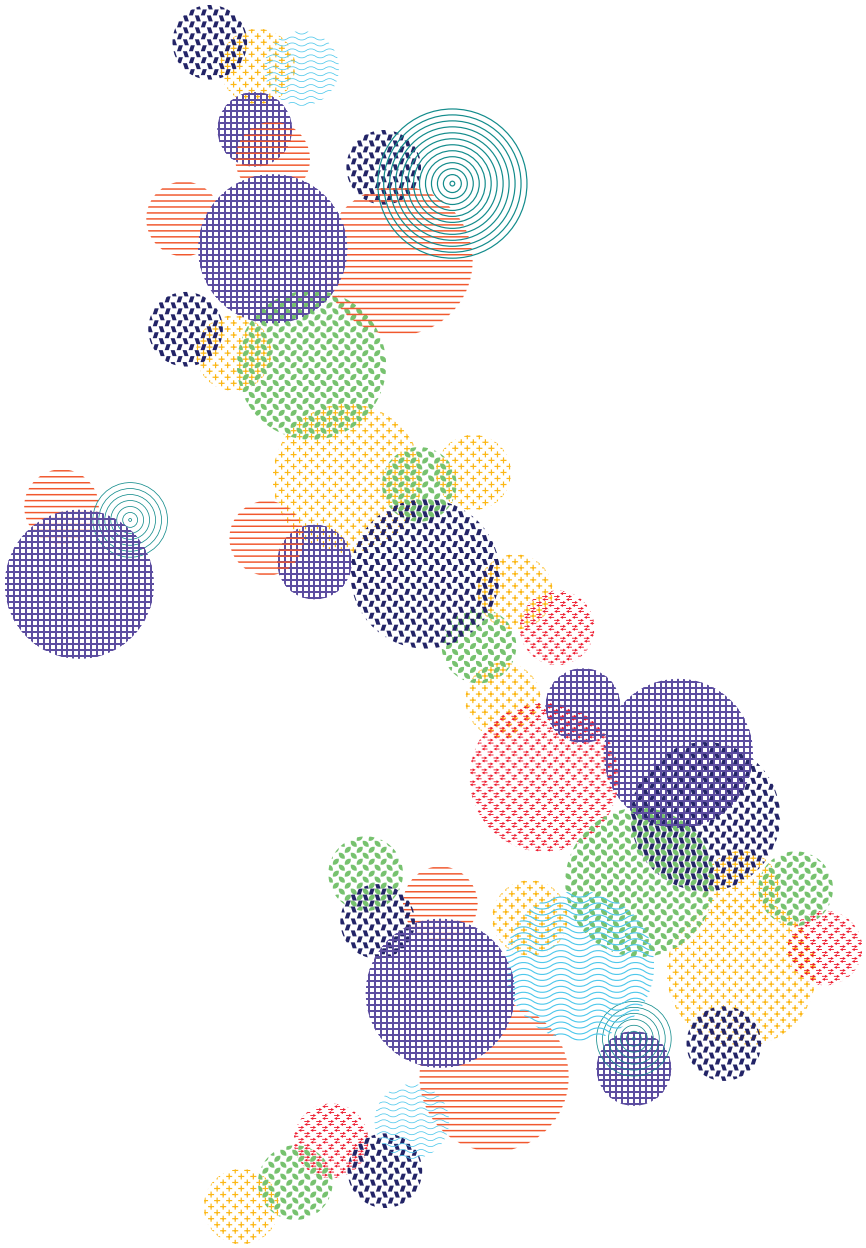
NATIONAL DIGITAL TWIN PROGRAMME

CReDo
Climate Resilience Demonstrator

CReDo Technical Report 1:
Building a cross-sector digital twin

February 2022



The Climate Resilience Demonstrator, CReDo, is a climate change adaptation digital twin demonstrator project developed by the National Digital Twin programme to improve resilience across infrastructure networks.

CReDo is a pioneering project to develop, for the first time in the UK, a digital twin across infrastructure networks to provide a practical example of how connected-data and greater access to the right information can improve climate adaptation and resilience. CReDo is the pilot project for the National Digital Twin programme demonstrating how it is possible to connect up datasets across organisations and deliver both private and public good.

Enabled by funding from UKRI, The University of Cambridge and Connected Places Catapult, CReDo looks specifically at the impact of extreme weather, in particular flooding, on energy, water and telecoms networks. CReDo brings together asset datasets, flood datasets, asset failure models and a system impact model to provide insights into infrastructure interdependencies and how they would be impacted under future climate change flooding scenarios. The vision for the CReDo digital twin is to enable asset owners, regulators and policymakers to collaborate using the CReDo digital twin to make decisions which maximise resilience across the infrastructure system rather than from a single sector point of view.

CReDo's purpose is two-fold:

1. To demonstrate the benefits of using connected digital twins to increase resilience and enable climate change adaptation and mitigation.
2. To demonstrate how principled information management enables digital twins and datasets to be connected in a scalable way as part of the development of the information management framework (IMF).[1]

This first phase of CReDo running over the period April 2021 to March 2022 has focused on delivering a minimum viable product to bring the datasets together to offer insight into infrastructure interdependencies and system impact. Separate technical papers have been produced to describe each stage of the project so far:

CReDo Technical Paper 1: Building a cross sector digital twin

CReDo Technical Paper 2: Generating flood data

CReDo Technical Paper 3: Assessing asset failure

CReDo Technical Paper 4: Understanding infrastructure interdependencies and system impact

CReDo Technical Paper 5: CReDo and the Information Management Framework

The technical papers are nested under the CReDo Overview report, and all CReDo reports and related materials can be found on the Digital Twin Hub, https://digitaltwinhub.co.uk/projects/credo.

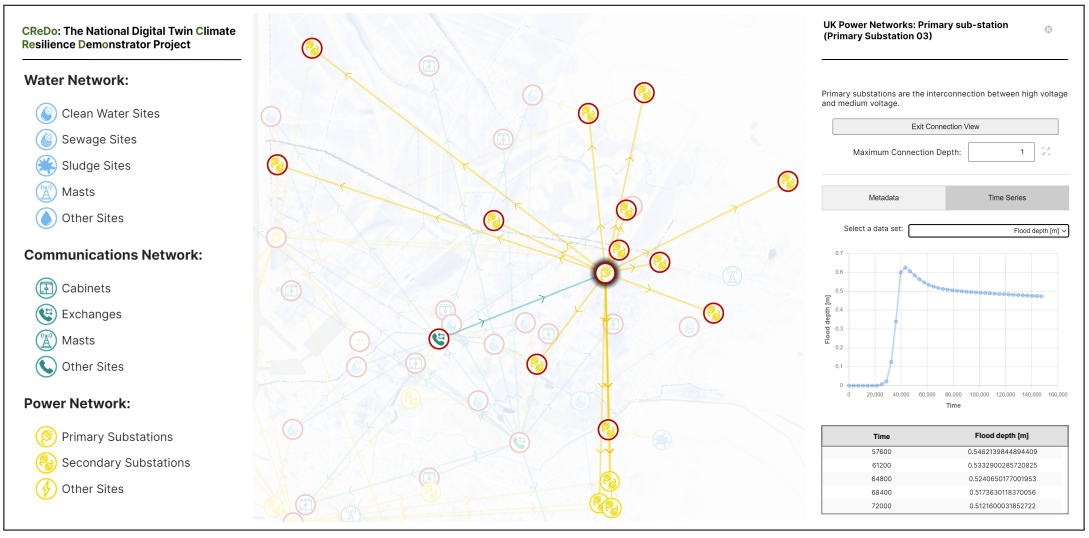
[1] IMF - DT Hub Community (digitaltwinhub.co.uk)

Contents

Summary

CReDo aims to demonstrate how the National Digital Twin programme could use connected digital twins to increase climate resilience. This first phase of the project investigates how to implement a digital twin to share data across sectors to investigate the impact of extreme weather, in particular flooding, on energy, water and telecoms networks. The current digital twin integrates flood simulations for different climate change scenarios with descriptions of the energy, water and telecoms networks, and models the interdependence of the infrastructure to describe the resilience of the combined network.

CMCL Innovations were engaged by the Centre for Digital Built Britain (CDBB) and the Connected Places Catapult (CPC) as part of CReDo to develop a digital twin of assets from Anglian Water, BT and UK Power Networks. The digital twin combines a description of the logical connectivity between the assets with flood data to resolve the effect of floods on individual assets and the corresponding cascade of effects across the combined network. It demonstrates how to achieve basic interoperability between data from different sectors, and how this data might be combined with flood data for different climate scenarios to begin to explore the resilience of the combined network and identify vulnerabilities to support strategic decision making and capital planning.



The first phase of the digital twin and an accompanying visualisation were implemented on DAFNI, the *Data & Analytics Facility for National Infrastructure*. This report describes the use and technical implementation of the current digital twin. Recommendations are made for how it could be extended to improve its ability to support decision making, and how the approach could be scaled up by the National Digital Twin programme.

All asset data in this report are synthetic, designed to be representative of the asset data used in the digital twin.

Findings

- The first phase of a digital twin was implemented to enable interoperability between data shared by the energy, water and telecoms sectors.
- The current digital twin combines the results of flood simulations and data about assets from each sector with models that describe the effect of flooding on the assets.
- The sharing of data and the ability to visualise the connectivity of assets was valuable and led to the correction of anomalies that could not be seen when looking at any single network.
- The current digital twin represents data using simple hierarchical ontologies. This enabled abstraction and, together with the ability to access the data as a knowledge graph, was fundamental to achieving interoperability and describing the dependencies between assets.
- An inconsistency was present in one ontology. The issue was overcome with a conversation and a few lines of extra code, and did not inhibit the functionality of the digital twin. The ontologies could be improved at a later date to address the issue more robustly.
- Although the data were accessed as a knowledge graph, some were able to be hosted using relational databases, allowing the use of established technology for each type of data.
- Future work should consider how to represent scenario-specific information in the digital twin, and how to simulate events efficiently whilst still providing wide-scale data coverage.
- The level of detail represented in a digital twin should be driven by the needs of use cases. In the case of the current digital twin, it should be extended to include whatever is necessary to describe the resilience of the combined network to support decision making.
- Future work should consider the extension of the digital twin to other domains and sectors, and should include an assessment of the value of the increase in resilience that could be achieved as a result of being able to assess the combined network using shared data.

1 Introduction

This report documents the technical implementation of the first phase of the CReDo digital twin. The digital twin integrates a description of assets from the energy, water and telecoms networks with the output from flood simulations for different climate change scenarios. It resolves the effect of floods on individual assets and the corresponding cascade of effects across the combined network.

CMCL Innovations were engaged by the Centre for Digital Built Britain (CDBB) and the Connected Places Catapult (CPC) as part of CReDo to develop a digital twin to describe the interdependencies between:

- Anglian Water's water and sewerage assets.
- BT's communication assets.
- UK Power Networks' power network assets.

The digital twin uses a knowledge graph to combine a description of the assets with data from flood simulations. The knowledge graph uses ontologies to represent information as a directed graph. The nodes of the graph express data as instances of concepts that describe the type, operational state and location of each asset. The edges of the graph express relationships between nodes. Figure 1 illustrates the idea. This data structure is exploited to describe the connectivity of the assets and achieve the required interoperability between data from different sources.

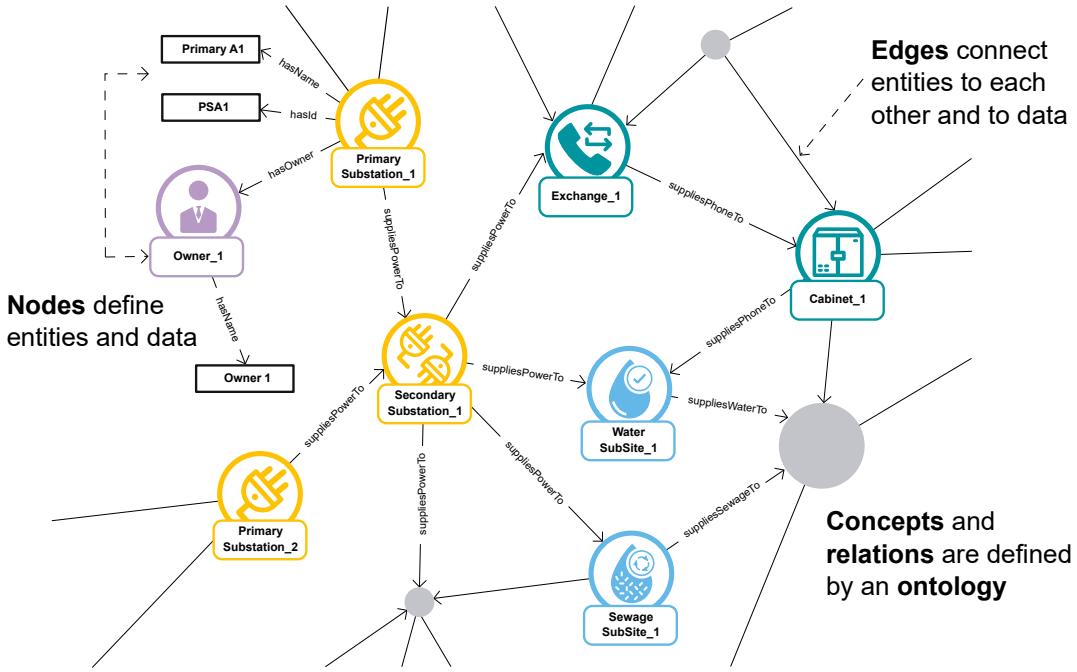


Figure 1: Snippet of a knowledge graph showing the logical connectivity of the assets in the CReDo digital twin.

A demonstrator of the digital twin was developed on DAFNI, the *Data & Analytics Facility for National Infrastructure* [1]. The software and a public-facing synthetic data developed as part of CReDo are published under a permissive open-source licence so that asset owners and third parties can use the demonstrator, as well as test and develop the ideas using their own data.

The focus of this first phase of CReDo was on developing an approach for how to implement a digital twin to enable interoperability between asset data from different sectors, and how to interface with data describing floods and models describing the effect of the floods. A number of activities ran in parallel:

i Development of the first phase of a digital twin that combines data describing the assets with models describing the failure of the assets to allow exploration of how different flood scenarios affect the combined network.
ii Development of flood simulations for different climate scenarios.
iii An expert elicitation process to develop an approach to implementing '*individual asset failure models*' that describe how the operational state of a given asset is affected by a flood.
iv An operational research process to develop '*system-wide impact models*' that describe how the effects of a flood cascade through a network of assets.
v An evaluation of the expected benefits of CReDo.

Future phases of CReDo should consider how to scale up the approach and align it with the Information Management Framework being developed by the National Digital Twin programme.

The **purpose of this report** is to document the implementation of the first phase of the digital twin as part of activity (i). Activities (ii)–(v) are documented elsewhere [2–5]. The work described in this report was performed by CMCL Innovations as part of the CReDo project between September 2021 and February 2022. The report is structured as follows: Section 2 provides an overview of how components of the digital twin interact with each other. Section 3 provides technical details of the components of the digital twin. Section 4 explains how to use the digital twin. Section 5 discusses lessons learned and makes recommendations with respect to how the digital twin could be extended to improve its ability to support decision making.

2 Overview

This section describes the organisation of the current CReDo digital twin and discusses some of the main architectural considerations.

The CReDo digital twin uses a knowledge graph to represent data describing assets from the energy, water and telecoms networks. The knowledge graph includes information about the type, operational state and location of each asset, and the physical and logical connectivity of the assets. The knowledge of the connectivity is used to resolve the cascade of effects caused by a failure in any of the networks. The digital twin is accompanied by a visualisation that displays assets from each network and allows exploration of detailed information about each asset, its connectivity and operational state.

Figure 2 shows a schematic of the digital twin workflow. The workflow starts by ingesting asset and flood data into a knowledge graph. The flood simulations [see 2] report information at a set of discrete time points, all of which are processed during the initial data ingestion. The workflow initialises other aspects of the knowledge graph including the representation of the dependencies between assets. The workflow enters a time loop. At each iteration:

- The knowledge graph is updated with the flood depth at each asset.
- Data is extracted from the knowledge graph for processing by individual asset failure models that describe how the flood affects the operational state of a given asset [see 3].
- A knowledge-graph-based information cascade model is applied to propagate the changes from the individual asset failure models back to the knowledge graph.
- Data is extracted from the knowledge graph for processing by system-wide impact models [see 4]. The models describe the cascade of effects throughout each individual asset network and then through the combined network.
- The information cascade model is re-applied to propagate the changes from the system-wide impact models back to the knowledge graph.

Finally, the workflow exits the loop and extracts data from the knowledge graph for visualisation.

The workflow was implemented on DAFNI [1]. The workflow combines individual asset failure and system-wide impact models in addition to a separate information cascade model. The individual asset failure and system-wide impact models are explicitly incorporated as separate models on DAFNI. This makes it straightforward to swap between different versions of these models. As the names suggest, the individual asset failure models act on individual assets, while the system-wide impact models act across the individual and combined networks, but not necessarily across the entire network. The information cascade model provides a complementary means of resolving the cascade of effects across the network, and an efficient default option in the absence of other models or while more specific models are developed. The individual asset failure, system-wide impact and information cascade approaches could be integrated at a later date if desired.

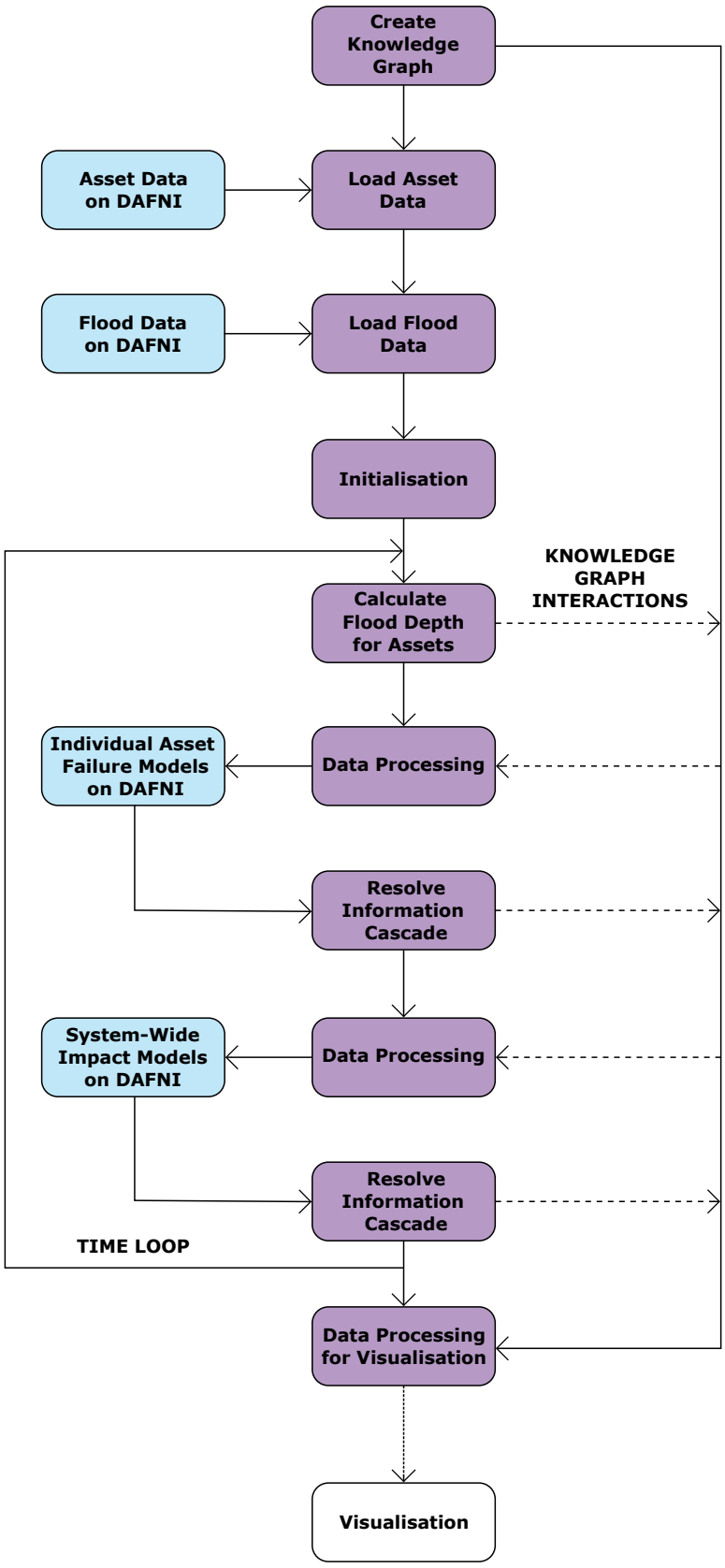


Figure 2: Current CReDo digital twin workflow.

The decision to use a knowledge graph was informed by a proposal from the team developing the current digital twin for how to implement connected digital twins that support the interoperability of distributed data across sectors, and ensure that data are connected, discoverable and queryable via a uniform interface [6]. The knowledge graph fulfils these requirements and provides a convenient way to represent arbitrarily structured data that lends itself to describing the connectivity between assets.

In this first phase of development, it was chosen to limit the lifetime of the knowledge graph to the duration of the workflow. This provided a pragmatic way to simplify the process of protecting the data provided by the asset owners whilst experimenting with the digital twin. However, the digital twin must be recreated by reprocessing the data each time the workflow is run. This approach should be refined to produce a persistent digital twin in future phases of CReDo.

Technical details of the components in the central column of Figure 2 (purple and white boxes) are described in Section 3. The other components (blue boxes) are described elsewhere [2–4].

3 Technical details

This section provides technical details of the components of the current CReDo digital twin and describes how they work. Readers who are only interested in using the digital twin can skip this section.

The software and synthetic data developed as part of CReDo are published under a permissive open-source licence. Similarly, the software dependencies are publicly available under sufficiently permissive, and mostly open-source, licences. See Appendix A for details.

3.1 Ontologies

The data in the digital twin are represented as instances of ontological classes in a knowledge graph. The knowledge graph expresses the data as a directed graph, where the nodes of the graph are concepts or their instances (*i.e.* data items) and the edges of the graph are links between related concepts or instances. The starting point for creating the knowledge graph is to define a type of schema, known as an ontology, that defines classes, object properties and data properties expressing facts about and a semantic model of the domain of interest. Object properties link an instance of a class (the domain) to an instance of a class (the range).[2] Data properties link an instance of a class (the domain) to a data element (the range).[3] Object properties and data properties may both be structured hierarchically.

The current digital twin uses two types of ontology:

- A core asset ontology to define core concepts that apply throughout the digital twin.
- Sub-domain ontologies to define concepts relevant to specific asset classes, in this case the energy, water and telecoms networks.

The ontologies are hierarchical, where the sub-domain ontologies inherit from and extend the concepts and relations defined by the core asset ontology. By first intent, the queries acting on the digital twin are defined in terms of the core asset ontology, using inheritance relations to retrieve data about individual assets from the sub-domain ontologies. This is important because it allows the sub-domain ontologies to change while minimising disruption to the core business logic of the digital twin. This was particularly advantageous during the development process and will make it simpler to extend the digital twin in the future.

[2] *e.g.* <Asset> (domain) <hasOwner> (object property) <AssetOwner> (range).
[3] *e.g.* <Asset> (domain) <hasName> (data property) <string> (range).

3.1.1 Core asset ontology

The core asset ontology defines the core concepts that apply throughout the digital twin. The main components of the ontology are shown schematically in Figure 3. The filled (purple) boxes represent classes, the hollow (white) boxes represent data objects, and the arrows represent object or data properties, depending on whether they point to a class or data object.

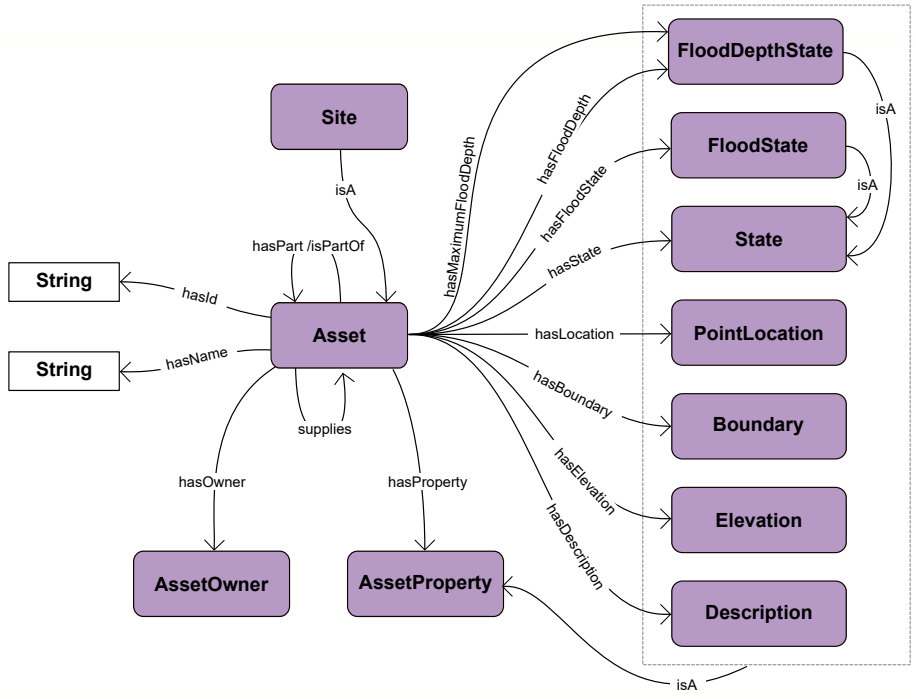


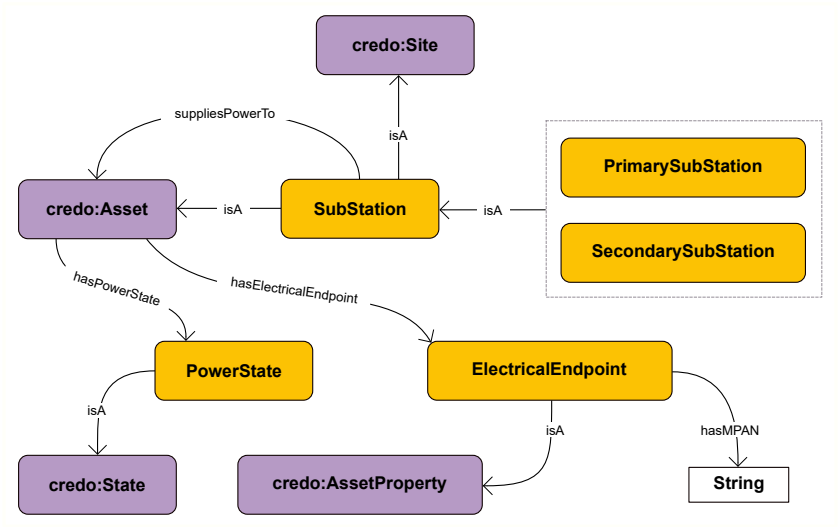
Figure 3: Core asset ontology (simplified).

The central concept of the core asset ontology is the *Asset* class. The class supports *hasPart* and *isPartOf* properties that describe physical part-hood relationships between assets, and a *supplies* property that provides a base property for describing operational dependencies between assets. The *Asset* class also supports *AssetProperty* classes that describe geospatial and operational information about an asset. The *State* concept provides a base class from which more specific classes can be defined to represent the operational state of an asset. The *FloodState* and *FloodDepthState* classes are two examples. The *FloodState* represents whether an asset is flooded (true or false). The *FloodDepthState* represents the current flood depth at an asset and the maximum flood depth that an asset can withstand before it is considered to be flooded. Finally, a *Site* class was added to support the visualisation (which currently only shows assets down to a site level).

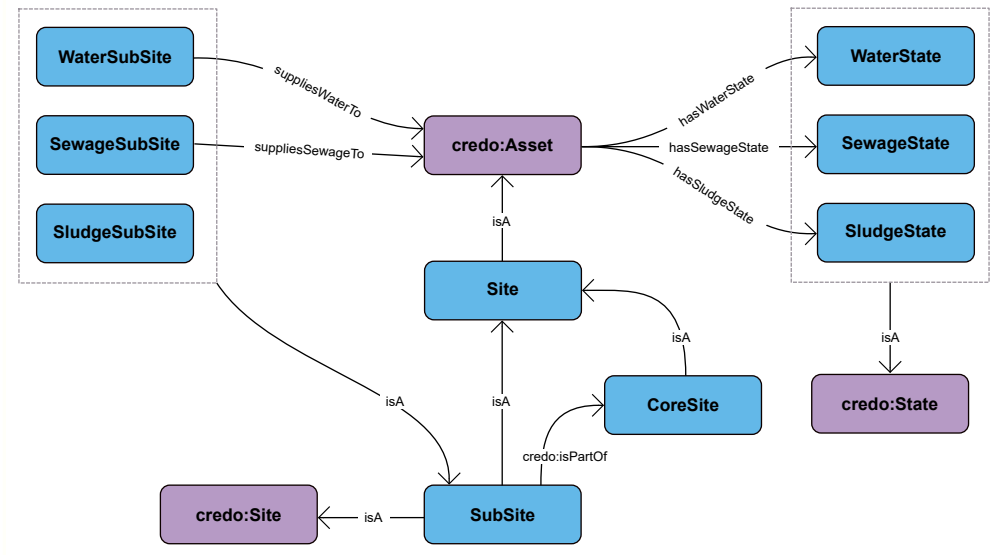
The core asset ontology was sufficient to describe all asset data included in the current digital twin, yet was kept as simple as possible whilst experimenting in this first phase of the project. Future phases of CReDo should consider how to align this approach with the Information Management Framework being developed by the National Digital Twin programme.

3.1.2 Sub-domain ontologies

The sub-domain ontologies define the concepts that are necessary to describe each asset network. The main components of each sub-domain ontology are shown schematically in Figure 4. The purple boxes represent classes from the core asset ontology, showing the inheritance relations used to extend the core asset ontology. The other boxes represent classes used to describe the energy, water and telecoms networks respectively.

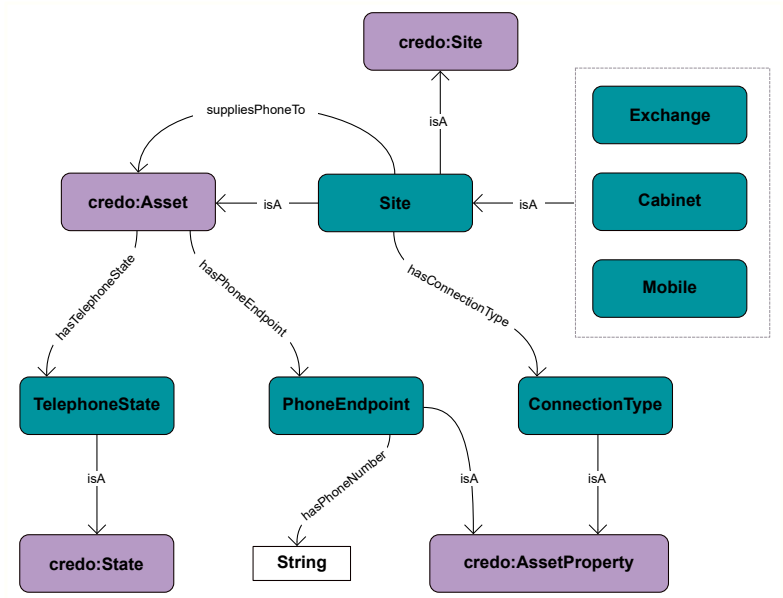


(a) Energy network.



(b) Water network.

Figure 4: Sub-domain ontologies (simplified), part 1.



(c) Telecoms network

Figure 4: Sub-domain ontologies (simplified), part 2.

The sub-domain ontologies introduce specialisations of the core asset ontology *Site* class:

- In the case of the energy network, this is the *SubStation* class, which is further divided into *PrimarySubstation* and *SecondarySubstation* classes.
- In the case of the water network, this also includes a water-specific *Site* and *SubSite* class, which is further divided according to whether the site deals with water, sewage or sludge.
- In the case of the telecoms network, this includes a telecoms-specific *Site* class, which is further divided into *Exchange*, *Cabinet* and *Mobile* classes.

In addition, the sub-domain ontologies introduce specialisations of the core asset *supplies* property: *suppliesPowerTo*, *suppliesWaterTo*, *suppliesSewageTo* and *suppliesPhoneTo*. These provide the basis for describing the interdependencies of the energy, water and telecoms networks.

The sub-domain ontologies also introduce specialisations of the core asset ontology *State* class. The current digital twin uses:

- *PowerState* to describe whether mains power is available at an asset.
- *WaterState* to describe whether clean water is available at an asset.
- *SewageState* to describe whether sewage is able to flow at an asset.
- *TelephoneState* to describe whether landline connectivity is available at an asset.

This approach is readily extensible and could be modified to include, for example, the availability of mobile voice and data signals or the state of backup power from batteries and generators, the state of charge of the batteries and the fuel available for the generators. This provides the basis for providing a fine-grained description of the operational state of an asset and is critical for the information cascade model described in Section 3.5.

The inconsistency in the use of 'phone' and 'telephone' in Figure 4c is noted. This is typical of the sort of issue that hinders interoperability. In future iterations of the digital twin, the ontologies should be modified to disambiguate the terminology and align it with that used by the asset owners, for example to disambiguate landline and mobile connectivity, voice and data services. The ontologies could also be mapped to the Foundation Data Model [7] that is anticipated to arise as part of the Information Management Framework [8] being developed by the National Digital Twin programme. This would help establish consistency with other datasets in the National Digital Twin, for example, by providing a clear definition of what is meant by 'asset' to facilitate the future expansion of the digital twin.

3.2 Knowledge graph hosting

The data in the digital twin are represented using a knowledge graph. This means that ontologies are used to define the possible classes and properties that can be represented in the knowledge graph, and that data are represented as Resource Description Framework (RDF) [9] triples and can be queried via SPARQL operations [10]. This provides an extensible data structure that is well suited to describing the connectivity between assets.

In the current digital twin, the knowledge graph is hosted using a combination of relational database (RDB), ontology-based data access (OBDA) and graph database solutions. Figure 5 shows the overall architecture. The sections below describe the major components in more detail.

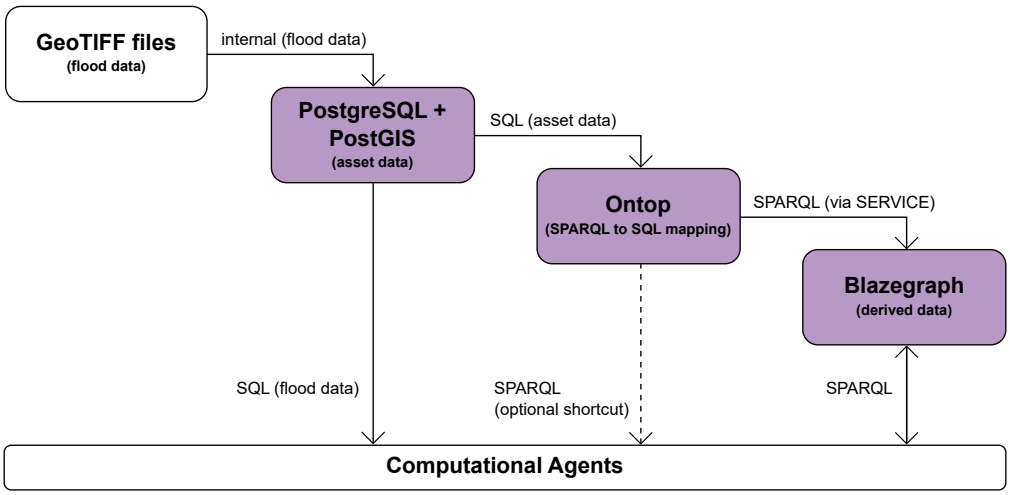


Figure 5: Knowledge graph architecture in the current digital twin.

3.2.1 Relational database and ontology-based data access

The data used by the digital twin that are inherently tabular or geospatial are hosted using a PostgreSQL [11] database with the PostGIS extension [12] to provide geospatial functionality. An Ontop server [13] is used to provide ontology-based data access to the relational database via a SPARQL endpoint, so that data from the RDB can be queried as per any other knowledge

graph. The Ontop server is configured via a file that maps RDF triples to SQL queries. Ontop can accept maps written in the standard RDB to RDF Mapping Language (R2RML) [14] format. However, the current digital twin uses Ontop's own format, which is more human readable and can be trivially converted to R2RML.

Mapping 1 shows an example of an Ontop mapping that maps RDF triples to SQL queries. The mapping is divided into two sections: target and source. The target provides a template for internationalised resource identifiers (IRIs) that identify the resources in the knowledge graph and the RDF triples that define the resources. The terms within the curly brackets are variables, the values of which are retrieved from the RDB when they form part of an SPARQL query sent to the Ontop server. The source defines the SQL query that is used to retrieve the variables. Ontop is able to analyse the mappings so that only the parts of the SQL query required to answer the SPARQL query are actually sent to the RDB.

```
mappingId   example-site

target      example:asset/{id} a example:ClassOfAsset ;
              :hasId {id}^^xsd:string

source      SELECT 'Shortcode' AS id FROM
```

Mapping 1: Example snippet of an Ontop mapping file (pretty printed).

The use of Ontop simplifies the process of ingesting data. Standard software is used to upload data supplied in standard formats to the RDB, and the knowledge graph can be restructured by updating a mapping file that is written in a standardised format. This was advantageous because it made it easy to update the knowledge graph as the underlying ontologies evolved during the development of the digital twin. The alternative would have been to write custom code to convert data into RDF triples and upload them to a graph database at each iteration, which would be error prone and could take significant time to execute.

One reason for choosing an OBDA-based solution is that it naturally aligns with the format of the asset data (which had been exported from RDBs to supply it to CReDo), and could potentially sit on top of such RDBs to share selected data as part of a National Digital Twin. The Ontop mappings also provide access to the geospatial data handling capability of PostGIS, which is superior to the native geospatial functionality available via current graph databases.

The use of Ontop could be extended to allow Digital Terrain Model (DTM) data, stored in the RDB, to be accessed via the knowledge graph, for example to describe the elevation of assets. The same approach could be used for flood data, although the time-dependent aspect of the flood data would require more work.

Finally, it should be noted that Ontop is limited to providing read-only access to the underlying data. In one sense this provides an advantage because it prevents unintentional modification of data. However, it also presents a disadvantage because some data (for example the operational state of assets) must be updated for the digital twin to function.

3.2.2 Graph database

A Blazegraph [15] graph database hosts the remaining data used by the digital twin. This consists of the data that require read and write access such as the state of the assets, metadata supporting the resolution of the information cascade through the knowledge graph (see Section 3.5) and the class definitions for the ontologies. The graph database is additionally used to manage access to time-dependent data using a time series client, developed to support the implementation of digital twins using a dynamic knowledge graph [6] as part of the World Avatar research project [16].

Blazegraph natively offers a SPARQL endpoint to query and update data and supports the standard query federation mechanism via an explicit `SERVICE` keyword.

3.2.3 Query federation

The digital twin uses federated queries to retrieve data from the OBDA and graph databases via a single SPARQL endpoint. This provides a step towards distributed hosting, where there exists a single uniform interface to access data published by different entities on different computer systems. There are several options for how to implement federation.

The current digital twin uses the standard federated query functionality [17] provided by Blazegraph, including the `SERVICE` SPARQL keyword to specify explicitly the sub-queries that should be sent to the Ontop endpoint. See Section 3.6 for an example. This is computationally efficient but requires knowledge of where things are stored, which is undesirable.

An alternative might be to use Teiid [18], which provides the functionality to perform (federated) queries over RDBs, files, and RESTful and generic HTTP endpoints. Teiid could be used to federate SQL queries over multiple RDBs so that asset data could be stored in separate locations, yet still accessed via Ontop. Another option might be to use the FedX [19] server that is provided as part of the RDF4J [20] library. This method would provide a virtual SPARQL endpoint that automatically routed each part of a query to the appropriate constituent endpoint(s). Further research would be needed to explore possible authentication methods beyond the defaults.

3.3 Data ingestion

The data incorporated into the digital twin were supplied in a variety of formats. The data from the asset owners had been exported from RDBs, so were mostly in form of tabular data in addition to some shapefiles [21] and raster (*i.e.* pixel-based) data. The floods were described by raster data, with different rasters for different time points and different climate scenarios.

3.3.1 Asset data

The tabular data describing the assets were provided in the form of Excel and comma separated value (CSV) files. The data were pre-processed using a Portable Operating System Interface (POSIX) shell script. The required transformations were specified in a configuration file for each input file. The pre-processing was minor and typically:

- Removed quotation marks from around descriptions.
- Removed units from quantities.
- Removed assets that were missing information required by the Ontop mappings, such that Ontop would have been unable to add the asset to the knowledge graph. Examples include missing 'site code', 'category name' and 'substation no.' data.
- Added entries for 'out of area' sites to provide a target (in the digital twin) for things that were referenced from, but not included in the set of asset data.

The pre-processed data were exported to CSV files and uploaded into the PostgreSQL RDB using csvkit [22] and set up using the PostgreSQL client tool [11]. Once in the RDB, the asset data were able to be queried as part of the knowledge graph used by the digital twin via the SPARQL endpoint provided by the Ontop server. This approach maximised the use of standard Linux software for text processing and components of csvkit [22] for uploading data. Full details of the coverage of the digital twin, missing data and assumptions are given in Appendix B.

3.3.2 Flood data

The flood depth data were uploaded from inlined GeoTIFF files to the PostgreSQL RDB using a tool supplied with the PostGIS client [12]. Where required, file format conversions were performed using GDAL [23]. This ensured that all the geometric shapes and raster data were in the correct format and the required auxiliary information was correctly generated in the RDB.

The data ingestion was tested with flood data from two sources:

- Data from HiPIMS, the *High-Performance Integrated Hydrodynamic Modelling System* for urban flood simulations. HiPIMS outputs data in an ASCII raster format[4] defined by Esri [24]. The model describes the temporal spread of a flood and separate output files are provided for each output time. The ASCII raster files are converted to GeoTIFF files using GDAL.
- Data from the Environment Agency (EA) that describe tidal floods for different probabilistic scenarios (*i.e.* 1 in 20, 1 in 200 and 1 in 1000-year events) [25, 26]. The data are provided as TIFF image and world file pairs[5] and are converted to GeoTIFF files using GDAL.

The data for all time steps (or all scenarios in the case of the EA data) are loaded before the time loop in the workflow in Figure 2. The name of the original files is stored alongside the raster data in the RDB. When the flood data are queried from within the time loop, the filename is used to infer the time step of the flood simulation. This approach mirrors the practice of other software, including GeoServer [27], which is used to serve the flood data to the visualisation (see Section 3.7). One consequence of this approach is that loading the EA data in one go causes the probabilistic EA scenarios to be treated as time steps. This was convenient for testing purposes, but it would be more correct to treat each EA scenario separately in the future.

A region of interest polygon is uploaded in addition to the flood data. This was included for the purpose of visualising the region of interest for the flood simulations.

[4] See https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/esri-ascii-raster-format.htm#GUID-D0420D89-9419-4910-8B4F-B8BF7B8B4EC3.
[5] See https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/world-files-for-raster-datasets.htm.

The digital twin uses a state updater agent to calculate the flood depth for each asset at iteration of the time loop shown in Figure 2. The agent uses an SQL query to retrieve the data for flood depth between previous and current iteration of the time loop (so this potentially retrieves data from multiple flood time steps). The flood depth relative to ground level for each asset is stored in the knowledge graph as a time series using the World Avatar time series client [6, 16].

3.4 Synthetic data

A set of synthetic asset data was created to support dissemination of CReDo whilst respecting the confidentiality of the data provided by the asset owners. The synthetic data describe similar asset types, with similar interdependencies and similar connections to the real data. The assets in the synthetic data are located in the same region that was studied in the flood simulations, however the data are completely synthetic regarding the location, distribution, density, connectivity and complexity of the asset network. These synthetic data enable a representative, yet simplified, demonstration of the digital twin using real flood data and asset impact models without disclosing sensitive information.

The synthetic data help support the narrative of the need for connected digital twins by providing an example of the impact of flood cascading across multiple networks. The synthetic data enable the re-use of the flood data and asset impact models generated during the CReDo project without identifying the location of any real assets, and provide the opportunity to use the visualisation of the CReDo digital twin to generate screenshots and videos for use in public promotional and dissemination materials. Additionally, the synthetic data will help third parties instantiate sample digital twins and test the integration of their own data, either in self-led research or in workshops/collaborative exploration with CReDo project partners.

The synthetic data are provided as spreadsheets that use the same file names and column headers as the data provided by the asset owners, and can be ingested without changing any code.

3.4.1 Region

The synthetic data are bounded by a longitude and latitude range, defined as per the World Geodetic System 1984 (WGS 84, also known as EPSG 4326) coordinate reference system.

3.4.2 Number and location of synthetic assets

The synthetic data contain fewer assets than the real data to simplify demonstrations of the digital twin. The asset locations were generated randomly within the synthetic region, subject to the following constraints. Assets were forbidden from being in unrealistic locations, such as in bodies of water. Assets were required to be distributed across the synthetic region, as opposed to being densely populated in a sub-region. Table 1 summarises the number of assets in each network.

Network	Asset Type	Number
Energy	Primary substations	3
	Secondary substations	30
Water	Sewage sites	10
	Water sites	2
	Sludge sites	1
Telecoms	Exchanges	3
	Mobile masts	3
	Cabinets	10

Table 1: Number of assets in the synthetic data.

3.4.3 Connectivity of synthetic assets

The synthetic data follows the trends seen in the connectivity of the real data. Primary substations provide power to secondary substations with a many-to-one relationship. Some secondary substations have an additional fallback connection to a second primary substation. The sewage, water and sludge networks remain separate from each other. The water networks form directed graphs that are typically, but not always, acyclic. The sewage network ends near a river and flows downhill. The majority of water assets receive power from secondary substations. The telecoms network consists of exchanges providing landline connectivity to cabinets and other exchanges with many-to-one relationships. The exchanges receive power from secondary substations and provide landline connectivity to some primary substations and some assets from the water network. Mobile masts exist but are not connected to anything in the synthetic data.

The real data did not provide information about the elevation of assets, so the decision was made to locate the synthetic assets at ground level. To maintain consistency, the same definition of ground level was used as for the flood data.

3.5 Information cascade

The CReDo digital twin is able to resolve the cascade of effects caused by the failure of assets. This is a key requirement for modelling how scenarios are coupled across sector boundaries and affect the combined network of assets. The knowledge graph that forms the basis of the digital twin provides a flexible data structure that is used to describe the dependencies between assets. In addition, the knowledge graph is used to encode relationships that link each individual asset to models that describe the behaviour of the asset, in this case in response to a flood.

The cascade of effects is resolved by an information cascade model that works in conjunction with the individual asset failure and system-wide impact models. These approaches provide a complementary means of resolving the cascade. The information cascade model acts across the entire knowledge graph and takes advantage of an experimental derived information framework that is being investigated as part of the World Avatar research project [16] to support the implementation of digital twins using a dynamic knowledge graph [6].

This section describes how the knowledge graph represents dependencies between assets and how this is used by the information cascade model. The individual asset failure and system-wide impact models were developed separately and are described in separate reports [3, 4].

3.5.1 Representation of dependencies

Information describing the dependencies between assets, and hence the specialisations of the *State* class introduced by the sub-domain ontologies (see Section 3.1), is elicited from the asset data, for example by matching the meter point administration numbers (MPANs) between assets supplying and receiving electrical power. The knowledge graph represents these dependencies using a *Derivation* class. Figure 6 shows an example. Whether Asset A has power depends on whether or not it is flooded. This affects whether Asset B has power, which depends on both whether Asset A is able to supply power and whether Asset B is flooded.

The instances of *Derivation* form a graph of dependencies between the states of the assets via the *belongsTo* (dependent state) and *isDerivedFrom* (dependencies) properties. Additional classes and properties (not shown) are used to specify the model that describes the relationship between the dependent state and dependencies of each *Derivation*. This provides the basis of the logic that is required to resolve the information cascade.

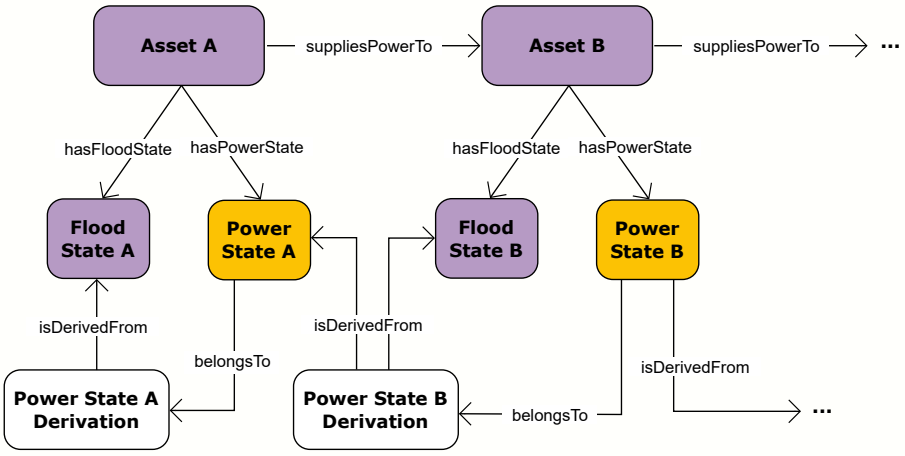


Figure 6: Representation of dependencies between assets (simplified).

3.5.2 Cascade algorithm

The knowledge graph encodes the information required by the information cascade model to update the state of each asset: what the state depends on and how to update it. In this sense, the information cascade is embedded in the fabric of the knowledge graph because it is entirely controlled by the information represented in the knowledge graph.

The information cascade is triggered when the state of an asset is queried from the knowledge graph via the derived information framework arising from the World Avatar research project [6,

16]. The cascade algorithm traverses the knowledge graph using a depth-first search to identify everything that the state depends on. When the search reaches a state with no further dependencies, the algorithm traverses back along the branch. At each step back along the branch, it compares the time stamp of the current state with its dependencies. If a state is out of date, the algorithm updates it using an update agent (see Section 3.5.3) specified as part of the instance of the *Derivation* class for the state in the knowledge graph. In this manner, everything in the dependency tree is updated as needed and in sequence as part of the response to the query.

Figure 7 shows an example. In this case, the information cascade is triggered by querying Telephone State A. The cascade algorithm traverses each branch of the dependency tree for Telephone State A, checking whether the current item is out of date compared to its inputs. The algorithm finds that this is the case for Flood State C, which is out of date compared to Flood Depth State C. The algorithm calls the update agents specified for each state in the knowledge graph to update Flood State C, followed by Power State C and Power State A, and then Power State B and Telephone State B, and finally Telephone State A as the algorithm traverses back up each branch of the dependency tree.

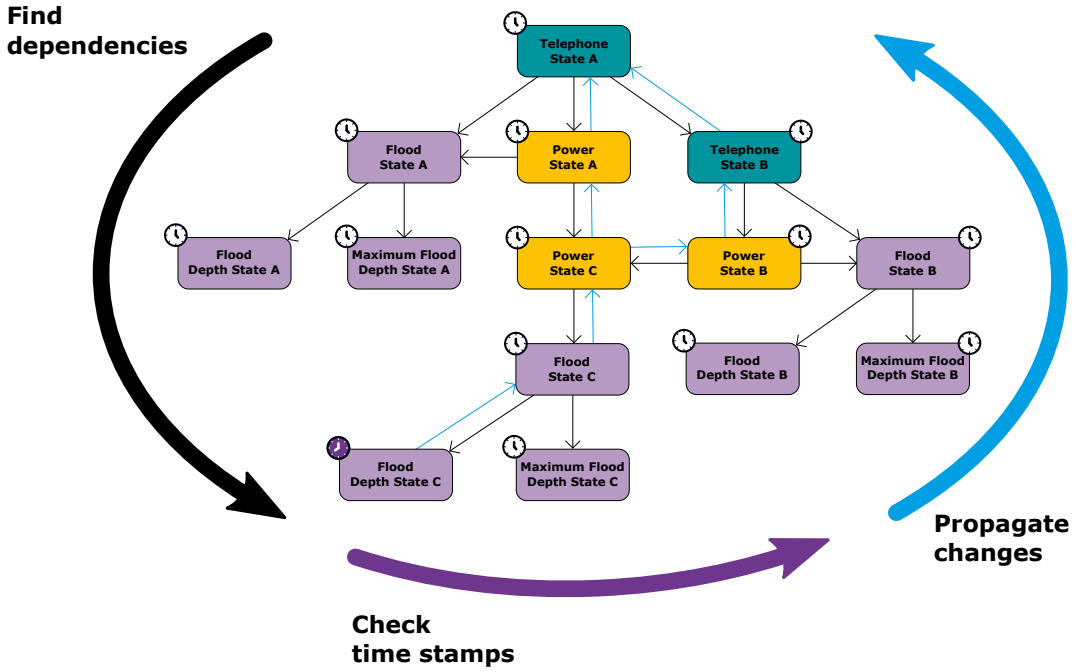


Figure 7: Example information cascade.

This approach seeks to ensure that information is up to date at the point of accessing it, and provides scalability by localising the cascade of updates to include only the minimum necessary operations. The approach has been demonstrated to work when the dependencies form acyclic graphs. Research is required to consider how the approach might deal with cyclic dependencies and cases where multiple models (*e.g.* the individual asset failure and system-wide impact models) affect the state of an asset. Future research should also formally assess the computational efficiency of the approach.

In the current digital twin, the individual asset failure and system-wide impact models provide additional input to the cascade via files that contain a list of values to be inserted into the knowledge graph. The current digital twin forces a full update of the network after inserting the results from each model. The forced update reassesses the flood state of each asset, which denotes whether the flood depth at an asset is greater than the 'maximum flood depth' that the asset can withstand. The 'maximum flood depth' is currently arbitrarily and uniformly set to 0.6 m relative to ground level. This needs to be revisited and replaced with something more suitable in the future. An expert elicitation process was conducted in parallel to the development of the current digital twin to investigate possible approaches [3].

It should be noted that cyclic graphs do exist in the connectivity of assets, for example in the water network. While these parts of the network are ignored by the information cascade model, they are described by the system-wide impact models, so are included in the current digital twin. In this sense, the information cascade and system-wide impact models complement each other, with the information cascade model providing blanket coverage across the entire network, and the system-wide impact models providing the ability to handle cyclic graphs.

3.5.3 Update logic for individual assets

The information cascade model uses an update agent to update the state of individual assets. The logic applied by the update agent used in the current digital twin is summarised in Table 2.

State	Dependencies	Update logic
current *FloodDepthState* (curFDS)	user input	N/A
maximum *FloodDepthState* (maxFDS)	user input	N/A
FloodState (FS)	curFDS, maxFDS	curFDS $>$ maxFDS
PowerState (PS)	FS, $(\mathrm{PS}_1, \mathrm{PS}_2, \cdots)$	$\neg\,\mathrm{FS} \wedge \left(\bigvee_{i=1}^{n} \mathrm{PS}_i\right)$
TelephoneState (TS)	FS, (PS), $(\mathrm{TS}_1, \mathrm{TS}_2, \cdots)$	$\neg\,\mathrm{FS} \wedge \mathrm{PS} \wedge \left(\bigvee_{i=1}^{n} \mathrm{TS}_i\right)$
WaterState (WS)	FS, PS, $(\mathrm{WS}_1, \mathrm{WS}_2, \cdots)$	$\neg\,\mathrm{FS} \wedge \mathrm{PS} \wedge \left(\bigvee_{i=1}^{n} \mathrm{WS}_i\right)$
SewageState (SS)	FS, PS, $(\mathrm{SS}_1, \mathrm{SS}_2, \cdots)$	$\neg\,\mathrm{FS} \wedge \mathrm{PS} \wedge \left(\bigvee_{i=1}^{n} \mathrm{SS}_i\right)$

Table 2: Logic used to update asset states. Dependencies without subscripts belong to the current asset. Dependencies in round brackets are optional. Dependencies with subscripts belong to other assets.

The dependencies listed in Table 2 fall into three categories:

- Dependencies belonging to the current asset are shown with no subscripts.
- Optional dependencies are shown in round brackets.
- Dependencies belonging to the other assets are shown with subscripts.

The telephone state, for example, always depends on the flood state of the current asset. It may

also depend on the power state of the current asset (Exchanges and Fibre Cabinets have a power supply, Legacy Cabinets do not). It may also depend on the telephone connection supplied to it by other assets ($TS_1, TS_2, \cdots$). The update logic is written so that it provides the condition for the Boolean states to be true. In the event that optional states are not present, the respective terms are omitted. A telephone state is therefore true if an asset is not flooded, $\neg$ FS, and it is supplied with power (optional), $\wedge$PS, and any incoming telephone connection (optional) is operational, $\wedge\,(\bigvee_{i=1}^{n} TS_i)$. The update logic is currently hard coded within the update agent, but could ultimately be specified within the knowledge graph.

In the current digital twin, the update agent acts in addition to the individual asset failure and system-wide impact models. This was advantageous in this first phase of the project because it provided a default option while the other models were developed. In the future, the digital twin could be modified so that the information cascade triggers the individual asset failure and/or system-wide impact models instead of the current update agent. This would offer the potential to combine the computational efficiency and scalability offered by localising update operations with the use of detailed models to describe the behaviour of critical assets, alongside more generic models to describe the behaviour of other assets. In the meantime, a control is provided in the current digital twin to allow the choice to experiment with using the individual asset failure and system-wide impact models without invoking the update agent.

3.6 Data processing

Data are extracted from the digital twin at three points in the workflow in Figure 2. At each point, data are queried from the knowledge graph and written to a set of output files. This process acts as an interface between the knowledge graph and the individual asset failure and system-wide impact models, and has made it easier to develop the models and knowledge graph in parallel. The output files are also used to provide data to the visualisation (see Section 3.7).

The data processing code executes a sequence of SPARQL queries to obtain information from the knowledge graph. The queries are constructed purely on the basis of the core asset ontology described in Section 3.1. This use of the core asset ontology provides a level of abstraction, enabling new sub-domain ontologies to be added without the need to change the data processing.

Query 1 shows an example that retrieves all assets, along with their name and ID. The last line returns the sub-classes of the generic CReDo asset type defined in the core asset ontology. These are passed as part of a sub-query to the Ontop server using the SERVICE keyword to retrieve data about the assets. A similar technique is used for queries about asset properties and connections.

```
SELECT ?asset ?assetClass ?name ?id
WHERE { SERVICE <ontop SPARQL endpoint URL> {?asset a ?assetClass ;
    CReDo:hasId ?id .
OPTIONAL { ?asset CReDo:hasName ?name . }}
?assetClass <http://www.w3.org/2000/01/rdf-schema#subClassOf>* CReDo:Asset .}
```

Query 1: Example query to retrieve all assets from the knowledge graph.

The following summarises the data processing workflow. Each step involves a separate query:

- Query all assets along with their IDs and names (see Query 1).
- Query coordinates of sites. (*Site* is a sub-class of *Asset*, see Figure 3).
- Query connections between sites.
- Query part-hood relationships between sites.
- Query properties of assets. This includes assets identified via the part-hood relationships.
- Query states of sites, *e.g.* telephone and power states.

Once the queries and analysis are complete, the data are written to files that provide a snapshot of the information in the knowledge graph. The current digital twin adopts JavaScript Object Notation (JSON) and Geographic JSON (GeoJSON) formats for the output because they are standard formats for web applications and are easy to process by the other models.

The current data processing outputs information from the entire knowledge graph. This was useful during the development of the digital twin because it ensured that the individual asset failure and system-wide impact models had access to the full dataset. However, it will become impractical as the digital twin grows. In the future, it will be important to develop method(s) to limit the scope of the data processing, for example geospatial constraints or constraints that limit the types of asset that are considered. The use of files to transfer data also presents disadvantages. It is possibly slow compared to other methods. More importantly, it negates the efficiencies that might be offered by exploiting the structure of the knowledge graph. It would also hinder the use of the visualisation to support operational decisions. Alternative approaches should be considered.

3.7 Visualisation

A browser-based visualisation was developed to help demonstrate the benefits of sharing data and enabling interoperability. By first intent, asset owners and third parties will be able to integrate the capabilities provided by the digital twin and visualisation into their own systems.

The visualisation presents a map that allows the assets, connections between assets, and cascade of failures extracted from the digital twin to be viewed in relation to the real-world. The data are presented in selectable layers to provide the ability to focus on desired aspects of the network. Individual items can be selected to view more detailed metadata and time series data. Additional controls facilitate the exploration of the connections to and from individual assets.

3.7.1 Implementation

Figure 5 shows the architecture of the visualisation. The major components are described below.

As a web-based application, the majority of the visualisation is implemented in JavaScript (JS), along with some Hypertext Markup Language (HTML) and Cascading Style Sheet (CSS) files. The JS natively handles the JSON and GeoJSON files provided by the data processing. A web server is required to make the visualisation available for viewing in a modern web browser, so an Apache HTTP server [28] is used to host a folder containing the visualisation files.

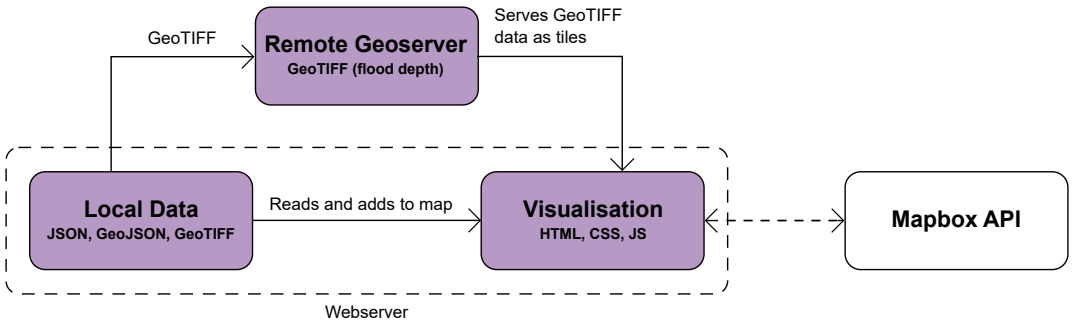


Figure 8: Architecture of the visualisation.

A number of mapping libraries are available to facilitate interactive online maps (Google Maps [29], Leaflet [30], OpenLayers [31] *etc.*). The current visualisation uses the Mapbox [32] library because it contains a large number of customisation options, supports all required data formats, has in-depth documentation, and appears to offer a lower barrier-to-entry compared to other tools. MapBox provides access to its API via access token. The visualisation currently uses a free token that permits up to 50,000 API requests per month. A premium plan must be purchased for usage levels above this.

Mapbox, like many mapping libraries, requires geospatial data to be encoded using the WGS 84 coordinate reference system. Where required, coordinate transformations are performed as part of the initial data ingestion into the knowledge graph by taking advantage of the ability of Ontop to access the geospatial data handling capability provided by PostGIS.

An instance of Geoserver [27] is used to serve the flood depth data to the visualisation. Geoserver is able to split the raw raster images that contain the flood depth into tiles, so that only the portion of the data relating to the region shown on the map needs to be sent to the client. The ChartJS [33] library is used to display the time series data associated with assets. The library provides a number of different chart options, each with a variety of customisation options and is accompanied by detailed documentation and a variety of examples.

3.7.2 Extensibility

The visualisation is designed to be extensible. In addition to objects with point locations (*e.g.* sites), the visualisation is able to support the representation of 1D (*e.g.* pipes and cables), 2D (*e.g.* geographic regions, areas of interest) and 3D (*e.g.* buildings) objects. New datasets can be added simply by adding the corresponding geospatial and metadata to the GeoJSON and JSON files used by the visualisation. The visualisation is also able to display raster data. This ability is used to display the flood data and could be extended to include other data in the future, for example maps of signal strength around mobile masts. Note however that additional work, over and above extending the visualisation, would be required to allow raster data to be queried and included in the individual asset failure, system-wide impact and information cascade models, for example to assess the impact of a power outage on mobile coverage at an asset.

The following are suggestions for features that could improve the usability of the visualisation:

- **Site lookup.** Add controls to search for and find sites via their name or unique identifier. This would make it easier to investigate the state of a particular site.
- **State explanations.** Add features that help explain why an asset has acquired a particular state. This would help understand the cause-and-effect relationships in a given scenario. Further work would be required to determine how best to do this.
- **Schematic view.** Add a schematic view (*e.g.* London Underground map) to display the logical connectivity of assets. This would make it easier to understand the dependencies in the network. However, such schematics could be very busy (initial attempts to view the connectivity by naively plotting everything were overwhelming, and therefore not very useful), so some thought would be required to determine how best to do this. Perhaps by restricting the schematic to show only the connectivity of an asset or selected set of assets?

Some specific technical improvements are also recommended:

- **Lazy loading.** The current visualisation loads everything from the GeoJSON and JSON files when it is initialised. This was advantageous during the development of the digital twin because it has the benefit of simplicity. However, it makes the visualisation slow to load and will become impractical as the digital twin grows. A lazy loading solution would delay loading resources until they are needed, making the visualisation feel more responsive.
- **Interactive granularity.** The visualisation will become increasingly busy as data is added to the digital twin. A more granular description of assets is likely to be required to support better modelling of scenarios, yet may be unhelpful when trying to explore the data visually. The ability for the visualisation to adapt its behaviour based on whatever is visible at the time is anticipated to be helpful. So, for example, the connectivity between a substation and an asset would be shown in terms of the connectivity between individual pylons if the visibility of the pylons was enabled. If the pylons were subsequently hidden, the visualisation would revert to showing the logical connectivity between the substation and the asset.
- **Live view.** The current visualisation uses post-processed data. Consideration should be given to developing a 'live view' that shows real-time content from the digital twin. Whether this is useful will depend on how the use cases for the digital twin evolve.

3.8 Implementation on DAFNI

3.8.1 DAFNI Workflow

DAFNI [1] provides a graphical platform [34] to create, manage and execute user-defined *workflows*. Workflows consist of a number of *models*, the behaviour of which can be controlled using input parameters. Models can also be given access to data that is stored securely on the National Infrastructure Database (NID) [35]. The DAFNI platform allows users to upload their own models and create workflow templates, which can include optional steps to publish new data to the NID and/or examine workflow results using one of the built-in types of visualisation.

The technology underlying the workflow framework on DAFNI is Argo [36] – an extension of the Kubernetes [37] container orchestration software. The workflow in Figure 2 was implemented in Argo by creating Podman [38] containers for the components of the workflow and running them as Argo *tasks*. Dependencies can be specified for each task, ensuring that the containers run in the required sequence. Asset data, flood data and other configuration files for the knowledge graph are mounted to fixed file system locations inside the relevant containers when the workflow is executed. This means that different versions of the files can easily be swapped in and out to test, for example, how different flood scenarios affect asset availability. The behaviour of each component can also be controlled at the workflow level by setting input parameters that get passed down to individual tasks.

Figure 9 shows the containers (purple boxes) included in the Argo workflow template, the input data (blue boxes) and parameters they require, and the outputs they produce (white boxes). The containers that host the knowledge graph must be created and persist while the rest of the workflow runs to serve requests to add, retrieve and update data. This is achieved by instructing Argo to run them in *daemon* mode, meaning that they continue to run in the background while the other parts of the workflow execute. Argo terminates the workflow when all other (non-daemon) tasks have completed. The template includes a 'readiness probe' which checks that each knowledge graph container is ready to receive requests before proceeding with the data upload stage. The workflow outputs processed flood data and a number of JSON and GeoJSON files that provide the input to the visualisation. When executed on DAFNI, these output files will be published to the NID, ready for retrieval by the visualisation at some later time.

Deployment of the Argo workflow on DAFNI and work to add DAFNI support for the Argo features used by the digital twin is being undertaken by DAFNI in collaboration with CMCL Innovations.

3.8.2 Visualisation

The CReDo digital twin visualisation is designed to run as a standalone web application, accessed through a browser. To run the visualisation via the DAFNI platform, it will need to retrieve and display the workflow output from the NID data store. Existing DAFNI visualisations use Keycloak [39] to obtain an authentication token that provides access to the NID; efforts are underway by the DAFNI team, in collaboration with CMCL Innovations to add similar functionality to the CReDo visualisation.

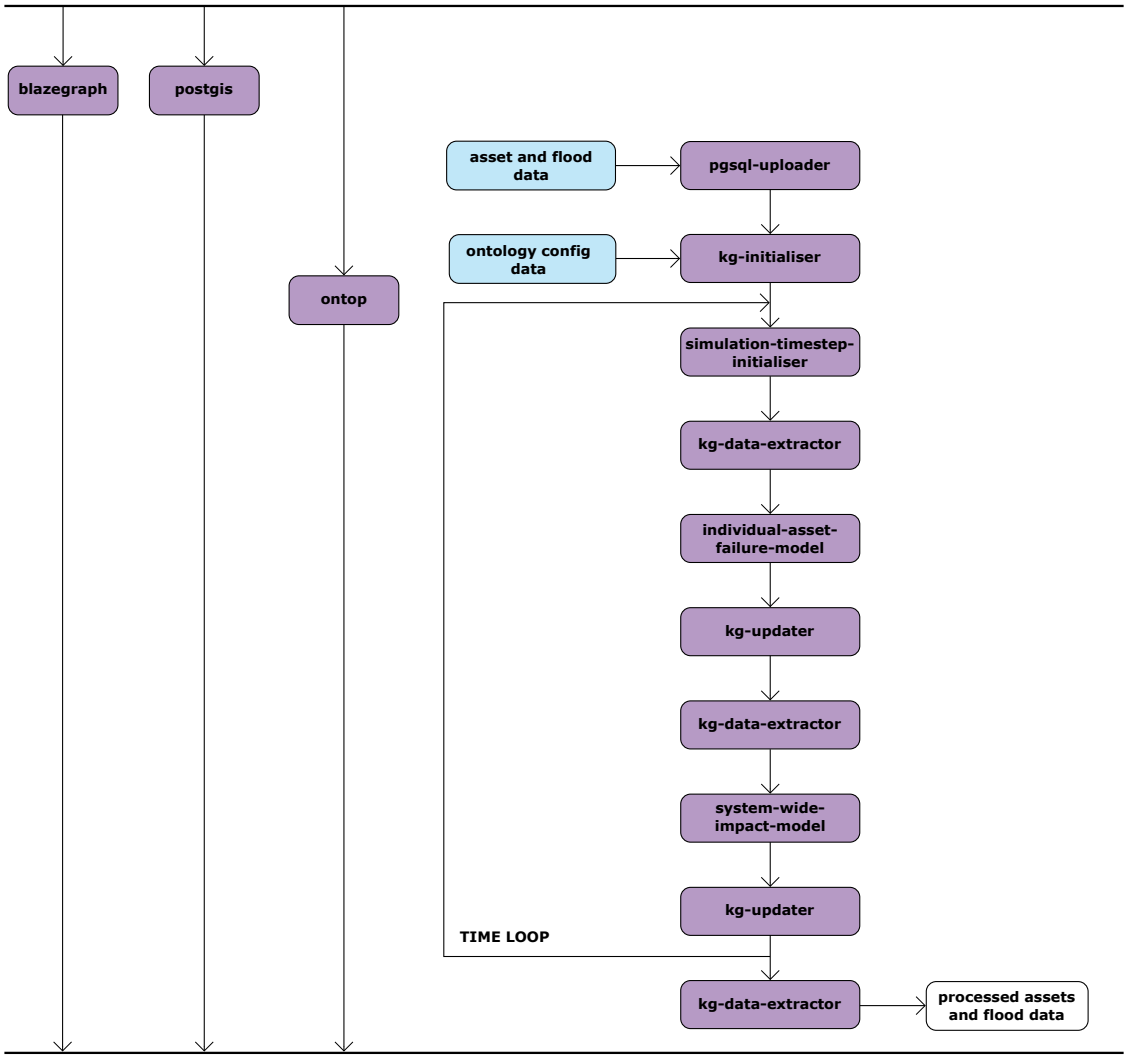


Figure 9: Argo implementation of the CReDo digital twin workflow. The purple boxes show the Podman containers incorporated into the workflow. The blue boxes show input data. The white boxes show output data.

4 How to use the digital twin

This section explains how to run the CReDo digital twin, and how to use the visualisation to explore the effect of floods for different climate scenarios.

4.1 Running the digital twin

The full deployment of the CReDo digital twin on DAFNI requires new functionality to be added to DAFNI. The DAFNI team have started the necessary work, but for now the digital twin can only be executed by directly interacting with the underlying Argo workflow software.

By default, the CReDo digital twin will use synthetic data (see Section 3.4). To use other data, or switch between flood data corresponding to different climate scenarios, users can search the DAFNI data catalogue [40] for the appropriate data set, note the unique ID and version ID, and supply them in their request to run the workflow. The results of the workflow will be published to a new dataset in the NID, ready to be loaded into the visualisation.

4.2 Visualising data from the digital twin

The DAFNI team are working in collaboration with CMCL Innovations to deploy the visualisation so that it can be accessed via the graphical interface provided by the DAFNI platform [34].

Figure 10 shows the basic layout of the visualisation. It consists of a geographical map with associated controls to explore the effect of floods resulting from different climate scenarios on the cascade of failures across the combined network of assets. The controls are situated on the left of the visualisation. They present options that allow the user to adjust the view, style and content of the visualisation. These options (from top to bottom) are detailed below:

- **Camera.** The camera controls provide default options to reset the view of the map. The view can also be adjusted manually: Left-click on the map and move the mouse to pan the view. Right-click on the map and move the mouse tilt and rotate the view. Use the scroll wheel on the mouse to zoom. The ability to enable a Depth of Field (DoF) filter that improves depth perception by fading out distant objects is also provided.
- **Terrain.** The terrain controls provide options to change the look and feel of the underlying map. An option to enable 3D terrain is also provided.
- **Layers.** The layer controls provide the ability to toggle the visibility of items on the map, including the flood depth, and different types of assets and connections. An option to disable geographical place names is also provided to help declutter the map.
- The **scenario** and **timestep** controls provide the ability to change the scenario (*i.e.* combination of flood and asset data) and time step (*i.e.* time step of the flood data) selection. Changing these settings will change the data shown on the map.

- The **latitude** and **longitude** of the mouse cursor is shown for convenience.

A collapsible sidebar is present on the right of the visualisation. The sidebar displays introductory text, a legend, and links to more detailed information. When something is selected by clicking on it on the map, the sidebar shows detailed information about the selected item.

Figure 10: Layout of the visualisation.

Figure 11 shows different time points from a simulation of a plausible worst case 1 in 100-year storm in 2070, given emission scenario RCP8.5 [2]. The visualisation shows the location of assets (or more strictly, assets that are sites) from the energy, water and telecoms networks overlaying the flood data. Different icons denote different types of asset. Connections between assets may also be shown beyond a minimum zoom level. Assets of the same type that are in close proximity are clustered and shown using a single icon, with the number in the icon indicating the number of assets in the cluster. Increasing the zoom level reveals the individual sites. A red ring around an icon indicates an asset that has failed (for any reason). Clusters of assets in which at least one asset has failed will also feature a red ring.

Assets and clusters can be selected by clicking on them on the map. If a cluster is selected, the sidebar will present a drop-down list of the assets within the cluster to allow the selection of individual assets. Once an asset is selected, the sidebar will display detailed information about the asset, including the name of the asset, metadata about the asset, metadata about assets connected to it, and the time history of the state of the asset. A control (not shown) next to the name of the asset will pan and zoom the map to show the location of the asset. Example data from the sidebar is shown in Figure 12.

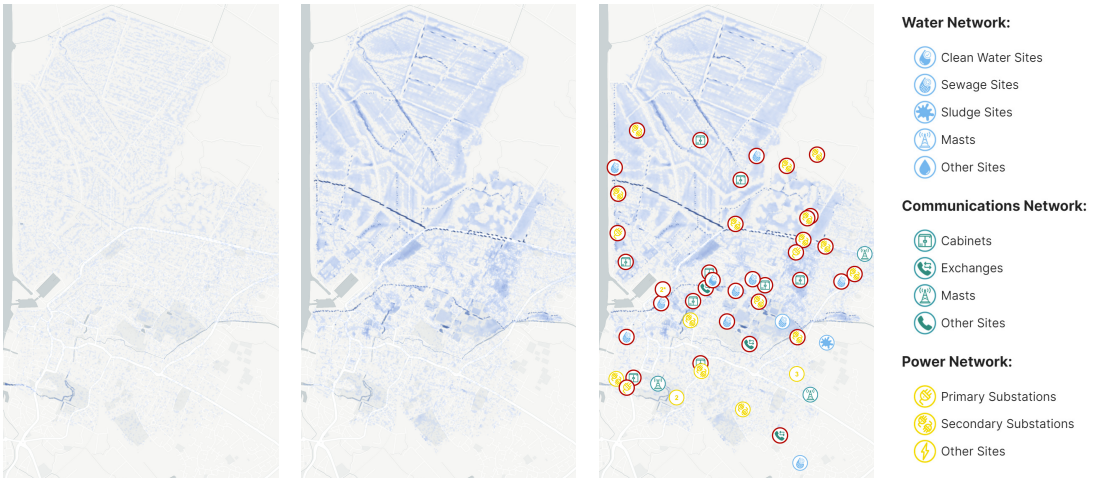
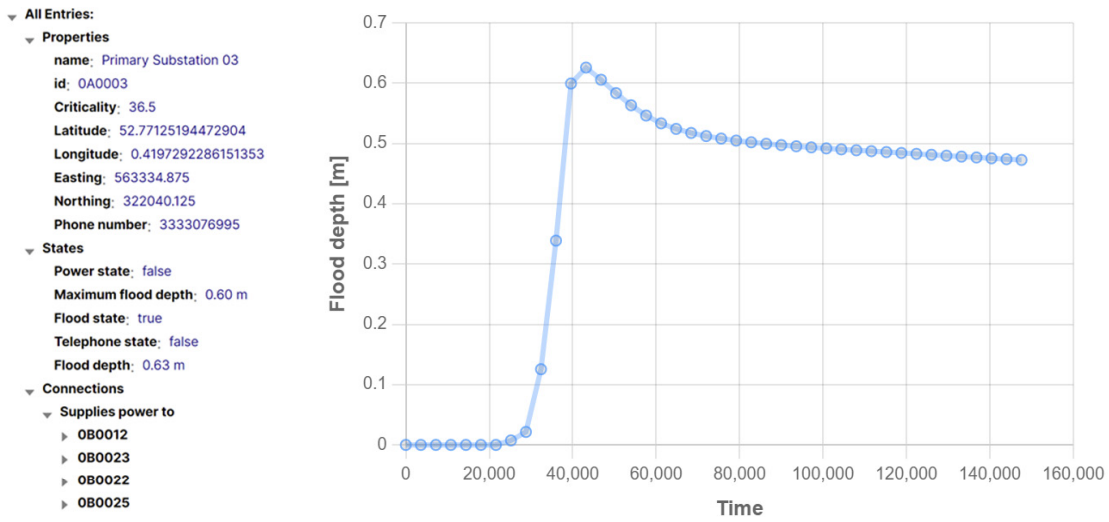


Figure 11: Visualisation of flood data (left to right, time points from a simulation of a plausible worst case 1 in 100-year storm in 2070, given emission scenario RCP8.5 [2]), overlaid with synthetic asset data (third panel from the left) and the legend from the sidebar. Failed assets are indicated by red circles around the asset markers.



(a) Metadata about an asset. *(b) History of the flood depth.*

Figure 12: Visualisation of asset properties (for a synthetic asset).

Once an asset has been selected, the sidebar also provides a 'View Direct Connections' control (not shown) that changes the visualisation to show a focused view of the selected asset and its direct connections (up to a configurable depth). The focused view makes it easy to explore the connectivity of assets.

Figure 13 shows an example of the focused view. The panel on the left shows assets in the vicinity of a flood, with a dark ring indicating the selected asset. The panel on the right shows a focused view of the direct connections to the selected asset. The selected asset is a secondary substation that is not directly compromised by the flood. The direction of the arrows on the connections show that it is supplied by two primary substations, indicating the existence of a fallback supply route and therefore suggesting some resilience in the network. However, both primary substations are compromised by the flood, so the secondary substation is also compromised. The outgoing connections from the secondary substation are to a telephone exchange and two sewage sites, which are therefore also compromised.

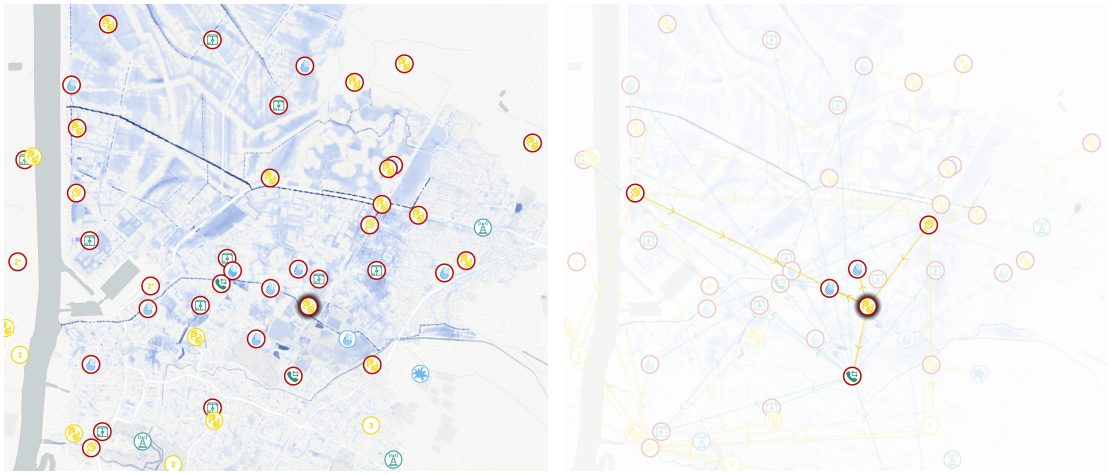


Figure 13: Visualisation of assets and connections. Synthetic assets in the vicinity of a flood (left panel). The dark ring around the asset in the centre of the image indicates that it has been selected. Focused view of the selected asset and its direct connections (right panel).

It is clear that most of the assets in Figure 13 have failed, including assets from each of the energy, water and telecoms networks and including assets outside the direct flood zone. This demonstrates how the CReDo digital twin might be used to describe the resilience of the combined network and illustrates what can be achieved by sharing data across sectors.

5 Recommendations

The CReDo digital twin demonstrates the use of a knowledge graph to implement a digital twin that shares data across sectors to investigate the impact of extreme weather, in particular flooding, on energy, water and telecoms networks. This section summarises what has been achieved, discusses lessons learned and makes recommendations for how the digital twin might be extended in the future.

A digital twin and an accompanying visualisation have been developed to demonstrate how to increase climate resilience by sharing models and data across sectors. The digital twin uses a knowledge graph to integrate data from flood simulations and a description of the dependencies between assets from the energy, water and telecoms networks with models that simulate the response of the combined network of assets to a flood. The digital twin was developed using data describing:

- Anglian Water's water and sewerage assets.
- BT's communication assets.
- UK Power Networks' power network assets.

The browser-based geospatial visualisation of the digital twin allows the assets, logical connectivity of the network and operational state of the assets to be explored. The visualisation is overlaid by data from flood simulations. The data are presented in selectable layers to provide the ability to focus on different aspects of the network. Assets can be selected to view metadata about the asset and the time history of the operational state of the asset throughout the flood. Additional controls facilitate the exploration of the connections to and from individual assets.

The digital twin demonstrates how to achieve basic interoperability between data from different sectors, and how this might be used in conjunction with flood simulations for different climate scenarios to explore the resilience of the combined network and identify vulnerabilities to support strategic decision-making. The digital twin is designed to be extensible. It should be straightforward to extend the breadth of coverage to include assets from other sectors and other regions of the country, and the depth of coverage to include a more granular description of assets and their operational state. Such extensions will be necessary to enable the digital twin to properly address extreme weather scenarios, and to extend the digital twin to other use cases.

A public demonstrator of the digital twin and visualisation is to be made available on DAFNI, the *Data & Analytics Facility for National Infrastructure* [1]. The corresponding software and a synthetic dataset are published under permissive licences so that asset owners and third parties can test the ideas on their own data and contribute to future developments.

5.1 Lessons learned

The sharing of data, and even just the ability to visualise the connectivity of assets across the networks was valuable to the asset owners. On a number of occasions during the development of the digital twin, this led to the identification and correction of anomalies that improved data quality and that would not have been spotted if looking at any single network in isolation.

The technical approach in the current work is based on an idea for how to use a dynamic knowledge graph to implement connected digital twins [6]. The current digital twin uses a knowledge graph that represents data using hierarchical ontologies. This enabled abstraction, such that the core business logic of the digital twin was able to be defined in terms of a core asset ontology. This was fundamental to achieving interoperability and makes it simpler to extend the digital twin. It was also advantageous because it allowed sub-domain ontologies that inherited from the core asset ontology to evolve without disrupting the business logic while the digital twin was developed.

The digital twin was able to function as intended, despite an inconsistency in the use of 'phone' and 'telephone' in one of the sub-domain ontologies. In this case, a conversation and a few extra lines of code were sufficient to overcome the issue. However, this is an example of the type of issue that would hinder the scale up of digital twins if there had to be many such conversations. This simultaneously demonstrates that progress can be made with a pragmatic approach to ontologies and the value of consistency. The problem could also have been solved more robustly by extending the ontologies to include sufficient relations to clarify the situation. In the future, the concepts and structure of the ontologies used by the current digital twin could also be aligned with hierarchical ontologies arising from the Information Management Framework [8]. This would aid consistency and interoperability between ontologies.

Knowledge graphs represent data using ontologies, and in this sense, are a natural extension of the use of ontologies. The knowledge graph made it easy to make use of the abstraction provided by the ontologies in the current digital twin. It was well suited to representing arbitrarily structured data such as the dependencies between assets, in addition to metadata such as what something *is*, what model describes its behaviour, and how to access and use the model. Data from knowledge graphs is able to be queried via a uniform interface provided by SPARQL (a W3C standard) [10, 17]. There is no need to use different methods to query different types of data. Furthermore, there is no concept of 'tables' in a knowledge graph, so there is no need to know 'where' data are stored. For all these reasons, knowledge graphs suit the progression of digitalisation and may reduce barriers to scale up.

Although it is advantageous to *access* the data as a knowledge graph, it is not necessary to *store* it as a knowledge graph. How data is stored should be determined by the type of data, with the most appropriate method used in each case, and with an additional preference to avoid unnecessary processing. This was demonstrated by the current digital twin, where the asset and flood data were hosted with minimal processing using conventional relational databases, but served and accessed as a knowledge graph using an ontology-based data access solution.

The National Digital Twin requires a distributed architecture. Ontology-based data access offers the possibility to map existing data to ontologies arising from the Information Management

Framework [8]. This could potentially be combined with the federation of endpoints to enable the National Digital Twin to create a knowledge graph that connects to (read-only) data at source.

The architecture of the National Digital Twin also needs to support scenario-specific information, in addition to information describing the status quo. In the context of a future CReDo digital twin, a (read-only) connection to data at source might be used to describe the current state of each asset network, while scenario-specific data would need to be stored elsewhere, for example the changes in the network in response to a given flood scenario.

Digital twins should describe both the *state* and *behaviour* of things. The current digital twin uses the knowledge graph to demonstrate the possibility of associating each individual asset (technically the operational state of each asset) with a model that describes its behaviour. While research would be required to refine this approach, it offers the potential for an architecture that allows digital twins to grow organically. In the case of the current digital twin, for example, assets could be added and associated with a model without the need to change the internals of the model. Alternatively, it would be possible to associate critical assets with new models that provided more specialised descriptions of their behaviour, as appropriate.

5.2 Extension of the digital twin

It is recommended that future developments extend the coverage of the digital twin to include other assets and other regions of the country. The choice of what scenarios to consider should drive the choice of what data to include. The locations of wooden electricity pylons, for example, would need to be included to describe many scenarios that affect electrical networks. Likewise, the ability to use raster data describing signal strength around mobile masts would need to be developed to describe the effect of power outages on mobile telecoms. A sufficiently granular description of the assets would also make it possible to change the switching state of the energy, water or telecoms networks in the digital twin. This would open the door to automating the use of the digital twin to evaluate how to create a more resilient network, and in the fullness of time, assessing the benefits of the increased resilience in terms of saving money and saving lives. Some aspects of this could be straightforward. Others might require some research.

The choice of how to represent the state of an asset should be given more detailed consideration and should also be driven by the choice of what scenarios to consider. In the current digital twin, the status of a site is either ‘working’ or ‘not working’. In reality, a site may often be somewhere in-between. The inclusion of backup power from batteries and generators, and a description of the state of charge of the batteries or the fuel available for the generator (both examples of non-Boolean states) are a case in point, and are both missing from the current digital twin. Developments in this direction should consider how to represent whether something is partially functional or partially flooded. It is anticipated that developing a more granular description of assets will go some way to addressing this issue. Future developments should link assets to models (and/or update logic) that describe the change in the state of the asset. Finally, the specification of the ‘maximum flood depth’ that an asset can withstand, and the mechanism used to trigger a full cascade should be replaced with something more suitable.

The current digital twin forces the cascade of effects to be resolved across the entire asset network. This simplicity was useful during the development process, but it will not scale. Future work should consider how to localise the operations performed by digital twins to enable them to simulate events efficiently whilst still providing wide-scale coverage. This will not be as simple as just bounding operations because of the possibility of dependencies on things outside the bounds that remain important in the scenario. An attempt was made to investigate how to localise operations using the connectivity of assets in the current digital twin. How to do this efficiently remains an important question for the scale up of digital twins. Further research is recommended.

The visualisation should be extended as the digital twin grows. Features to search for specific sites, explore and visualise connections, and understand cause-and-effect relationships in a scenario would add immediate value. Likewise, it will become increasingly useful to show abstract logical connections based on what is visible at the time, so for example, the connections between a substation, electricity pylons and an asset, would be displayed as a logical connection directly between the substation and asset if the pylons were hidden in the visualisation. Finally, options should be considered for how to avoid the visualisation slowing down as more data is added, and how to show 'live' (as opposed to post-processed) data to support operational use cases.

Future developments should engage additional partners and asset owners, for example, local, regional and national distributors, regulators and agencies. The involvement of the Environment Agency would naturally align with the consideration of extreme weather events, and would facilitate access to data from environmental sensors. The road network is an important component of many scenarios, for example is a road passable so that someone might reach an asset? The involvement of the relevant Highways Authority might therefore be helpful. The involvement of Ordnance Survey would facilitate the integration of useful data. Roads and buildings are a case in point.

This first phase of CReDo has developed a digital twin to demonstrate how to achieve interoperability between data shared by the energy, water and telecoms networks to explore the resilience of the combined network to floods. However, the digital twin is capable of doing much more. It could, for example, describe the response of the combined network to other types of event. Examples could include planned outages, changes in the configuration of the networks or other weather events. Future work should consider the extension of the digital twin to other use cases, and should quantify the value of the increase in resilience that could be achieved as a result of being able to share data. A preliminary assessment was performed for the current digital twin and is reported separately [5].

Nomenclature

CDBB	Centre for Digital Built Britain
CPC	Connected Places Catapult
CReDo	Climate Resilience Demonstrator
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DAFNI	Data & Analytics Facility for National Infrastructure
DTM	Digital Terrain Model
EA	Environment Agency
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
HiPIMS	High-Performance Integrated Hydrodynamic Modelling System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IRI	Internationalised Resource Identifier
JSON	JavaScript Object Notation
JS	JavaScript
MPAN	Meter Point Administration Number
NID	National Infrastructure Database
OBDA	Ontology-Based Data Access
POSIX	Portable Operating System Interface
R2RML	RDB to RDF Mapping Language
RDB	Relational Database
RDF	Resource Description Framework
REST	Representational State Transfer
RML	RDF Mapping Language
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language

TIFF	Tagged Image File Format
W3C	World Wide Web Consortium
WGS	World Geodetic System

References

[1] *DAFNI, Data & Analytics Facility for National Infrastructure*, https://dafni.ac.uk, 2021.

[2] J. Salter and G. Shaddick, *CReDo Technical Paper 2: Generating flood data*, Centre for Digital Built Britain (CDBB), 2022.

[3] C. J. Dent, B. Mawdsley, J. Q. Smith and K. Wilson, *CReDo Technical Paper 3: Assessing asset vulnerability*, Centre for Digital Built Britain (CDBB), 2022.

[4] L. Schewe and M. Reyes-Salazar, *CReDo Technical Paper 4: Understanding infrastructure interdependencies and system impact*, Centre for Digital Built Britain (CDBB), 2022.

[5] F. Bondiolotti, D. Popov and M. Guijon, *CReDo benefits report*, Centre for Digital Built Britain (CDBB), 2022.

[6] J. Akroyd, S. Mosbach, A. Bhave and M. Kraft, "Universal Digital Twin – A Dynamic Knowledge Graph," *Data-Centric Engineering*, vol. 2, e14, 2021. DOI: 10.1017/dce.2021.10.

[7] C. Partridge *et al.*, *A Survey of Top-Level Ontologies – to inform the ontological choices for a Foundation Data Model*, Centre for Digital Built Britain (CDBB), 2020. DOI: 10.17863/cam.58311.

[8] J. Hetherington and M. West, *The pathway towards an Information Management Framework - A 'Commons' for Digital Built Britain*, Centre for Digital Built Britain (CDBB), 2020. DOI: 10.17863/cam.52659.

[9] G. Klyne and J. J. Carroll, *Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004*, World Wide Web Consortium (W3C), http://www.w3.org/TR/2004/REC-rdf-concepts-20040210, 2004.

[10] *SPARQL 1.1 Overview, W3C Recommendation*, https://www.w3.org/TR/sparql11-overview, 2013.

[11] *PostgreSQL*, https://www.postgresql.org, 2021.

[12] *PostGIS*, https://postgis.net, 2021.

[13] G. Xiao *et al.*, "The virtual knowledge graph system ontop," in *The Semantic Web – ISWC 2020*, J. Z. Pan *et al.*, Eds., Springer International Publishing, 2020, pp. 259–277, ISBN: 978-3-030-62466-8.

[14] *R2RML: RDB to RDF Mapping Language, W3C Recommendation*, https://www.w3.org/TR/r2rml, 2012.

[15] *Blazegraph*, https://blazegraph.com, source code available at https://github.com/blazegraph/database/wiki/About_Blazegraph, 2021.

[16] *The World Avatar*, http://theworldavatar.com, source code available at https://github.com/cambridge-cares/TheWorldAvatar, 2021.

[17] *SPARQL 1.1 Federated Query, W3C Recommendation*, https://www.w3.org/TR/sparql11-federated-query, 2013.

[18] Red Hat, *Teiid: Cloud-native data virtualization*, https://teiid.io, 2021.

[19] Eclipse Foundation, *Federation With FedX*, https://rdf4j.org/documentation/programming/federation, 2020.

[20] Eclipse Foundation, *Eclipse rdf4j*, https://rdf4j.org, 2021.
[21] ESRI, *Shapefile Technical Description*, https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf, 1998.
[22] *csvkit 1.0.6*, https://csvkit.readthedocs.io/en/latest, 2016.
[23] *GDAL*, https://gdal.org, 2021.
[24] *Esri GIS Mapping Software*, https://www.esri.com, 2021.
[25] JBA Consulting, "East Anglian Coastal Modelling: Final Summary Report," *Environment Agency*, 2019.
[26] JBA Consulting, "East Anglian Coastal Modelling: Model development report," *Environment Agency*, 2019.
[27] *Geoserver*, http://geoserver.org, 2021.
[28] *Apache HTTP server*, https://httpd.apache.org, 2021.
[29] *Google Maps Platform*, https://developers.google.com/maps, 2021.
[30] *Leaflet*, https://leafletjs.com, 2021.
[31] *OpenLayers*, https://openlayers.org, 2021.
[32] *Mapbox*, https://www.mapbox.com, 2021.
[33] *ChartJS*, https://www.chartjs.org, 2021.
[34] *DAFNI Platform*, https://facility.secure.dafni.rl.ac.uk, 2021.
[35] *National Infrastructure Database*, https://dafni.ac.uk/the-national-infrastructure-database-nid, 2021.
[36] *Argo*, https://argoproj.github.io, 2021.
[37] *Kubernetes*, https://kubernetes.io, 2021.
[38] *Podman*, https://podman.io, 2021.
[39] *Keycloak*, https://www.keycloak.org, 2021.
[40] *DAFNI Data Catalogue*, https://facility.secure.dafni.rl.ac.uk/data, 2021.

All links were accessed in Dec 2021 unless stated otherwise.

Appendices

A Source code

The code developed by CMCL Innovations and the synthetic data developed by the Connected Places Catapult on behalf of CReDo are published under a permissive open-source licence and are held under version control at the Science & Technology Facilities Council (https://gitlab.stfc.ac.uk/credo/base-repo/).

The code depends on several container images and library functions developed in collaboration with the Computational Modelling Group (https://como.ceb.cam.ac.uk/) at the University of Cambridge and the Cambridge Centre for Advanced Research and Education in Singapore (CARES) (https://www.cares.cam.ac.uk/). These dependencies are published under a permissive open-source licence and are available via The World Avatar package repositories on GitHub (https://github.com/cambridge-cares/TheWorldAvatar). All other dependencies can be resolved via Maven Central (https://search.maven.org/), PyPI (https://pypi.org/), and Docker Hub (https://hub.docker.com/).

B Data coverage

B.1 Included data

Water network

- Clean water, sewage and sludge sub-sites (IDs, names, locations, MPANs, phone numbers).
- Plant item part-hood hierarchy (which assets contain which other assets, including the asset descriptions).
- Clean and waste water connections between sub-sites (based on the core site connections specified by the raw data).

Energy network

- Primary substations (IDs, names, locations, phone numbers).
- Secondary substations (IDs, names, locations).
- Power connections between primary and secondary substations (including fallback options).
- Power connections between secondary substations and water network sub-sites.
- Power connections between secondary substations and communications network sites (exchanges and fibre cabinets).

Telecoms network

- Exchanges (IDs, names, locations, MPANs).
- Fibre cabinets (IDs, names, locations, MPANs).
- Mobile masts (IDs, names, locations).
- Connections between exchanges and landline telephones (not yet via street cabinets).
- Connections between exchanges and fibre cabinets.
- Phone connections to substations.
- Phone connections to water network sub-sites.

Asset states

- Flood state, flood depth and maximum flood depth (see the assumptions in Section B.3).
- Availability of mains power.
- Availability of landline telephone connectivity.
- Availability of mains water (only between water network sites).
- Availability of sewage connection (only between water network sites).

B.2 Data not included

This section lists data that are not included in the current digital twin. Such data could be included in subsequent iterations.

Water Network

- Distribution zones for water supply.
- Granularity of connectivity at the sub-site level and below.

Energy Network

- Power connections to mobile masts.

Telecoms Network

- Locations of legacy cabinets.
- Outgoing connections from fibre cabinets.
- Mobile signal coverage provided by mobile masts.
- Connections between telephone exchanges and mobile masts.
- MPANs of the mobile masts and therefore the connectivity to the electrical grid.

B.3 Assumptions and simplifications

This section lists assumptions and simplifications made in the treatment of the data in the current digital twin.

Common

- All assets have a 'maximum flood depth', which is the depth of water at which an asset fails.
- The maximum flood depths use assumed values that need to be replaced with real data.
- All connections specified as being directly between major assets, with intermediate assets such as pipe junctions, telegraph poles and electrical pylons ignored, even when data are available.

Water network

- Ignored the presence of backup power from batteries and generators.
- The data grouped sub-sites geographically into core sites, which are named after the primary sub-site in the group. The raw data specified water and sewage connections for core sites, while the rest of the data (*e.g.* locations and type of asset) were provided for the sub-sites and the assets within them. To reconcile this, it was decided to simplify the digital twin by assigning connections to the sub-site with the same name as the core site. An undesirable consequence of this choice was that it left some sub-sites unconnected. The possibility

of manually connecting the sub-sites was considered, but rejected both because of the desire for an automated solution and because of the time-consuming nature of the task. The options of assuming that all sub-sites have the same connections as their core site, or of explicitly adding new assets (with aggregated locations) to represent the core sites were also considered, but not chosen because they made the digital twin more complex without substantially improving it.

Energy network

- Power supply connections mapped by matching MPANs from assets to MPANs supplied by substations.
- The connectivity between primary and secondary substations was generated using heuristics rather than known connections.

Telecoms network

- Ignored the presence of backup power from batteries and generators.
- Ignored incoming connections to mobile masts due to lack of data.
- Ignored mobile coverage.

Acknowledgements

Lead Author

Jethro Akroyd

Contributors

Amit Bhave
George Brownbridge
Elliot Christou
Michael Hillman
Markus Hofmeister
Markus Kraft
Jiawei Lai
Kok Foong
Lee Sebastian Mosbach
Daniel Nurkowski
Owen Parry

The Centre for Digital Built Britain at the University of Cambridge's National Digital Twin programme is funded by the Department for Business, Energy and Industrial Strategy via UK Research and Innovation.

Akroyd, J (2022). CReDo Technical Report 1: Building a cross-sector digital twin.
https://doi.org/10.17863/CAM.81779