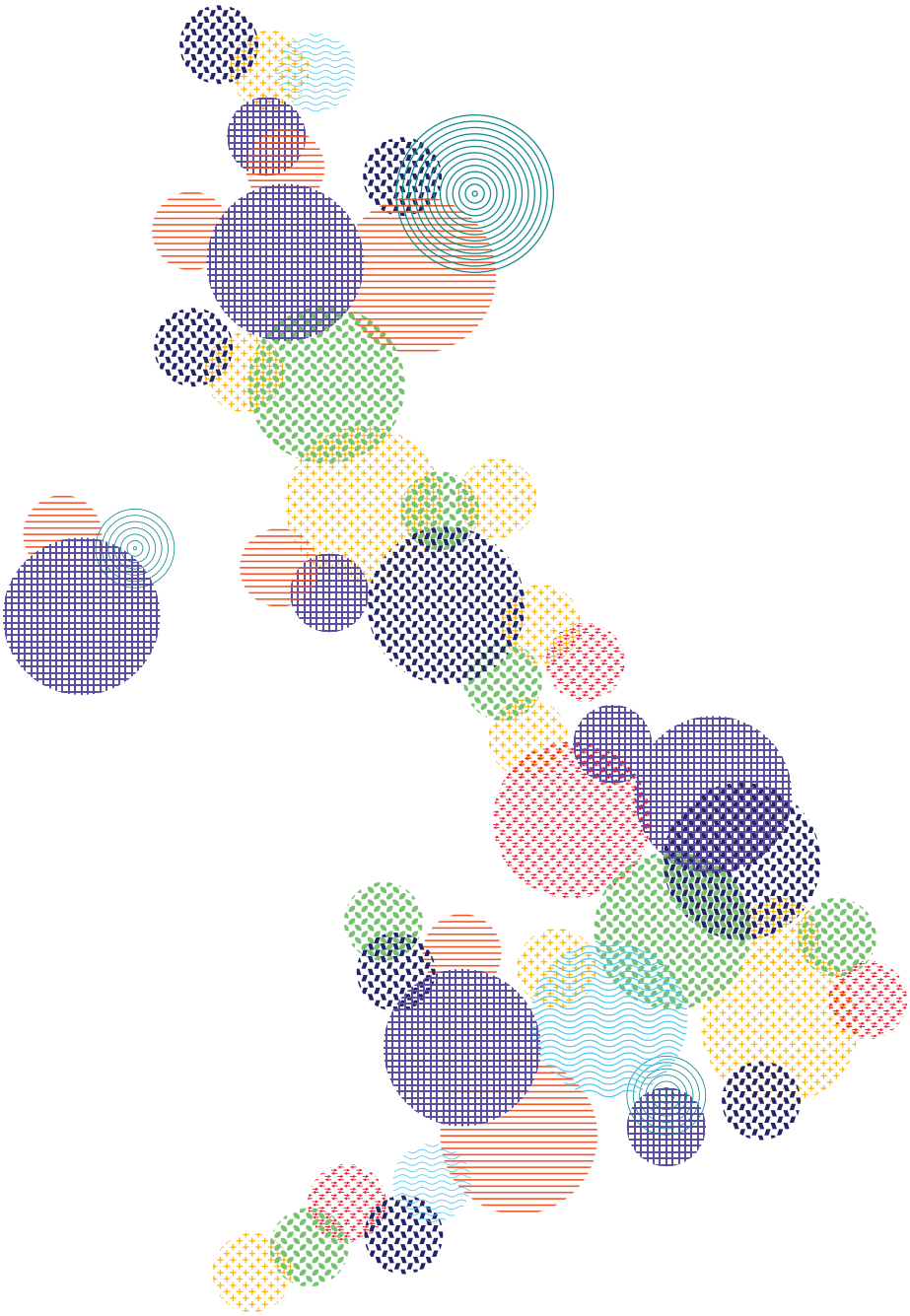
NATIONAL DIGITAL TWIN PROGRAMME

CReDo
Climate Resilience Demonstrator

Implementation of the CReDo digital twin

January 2022



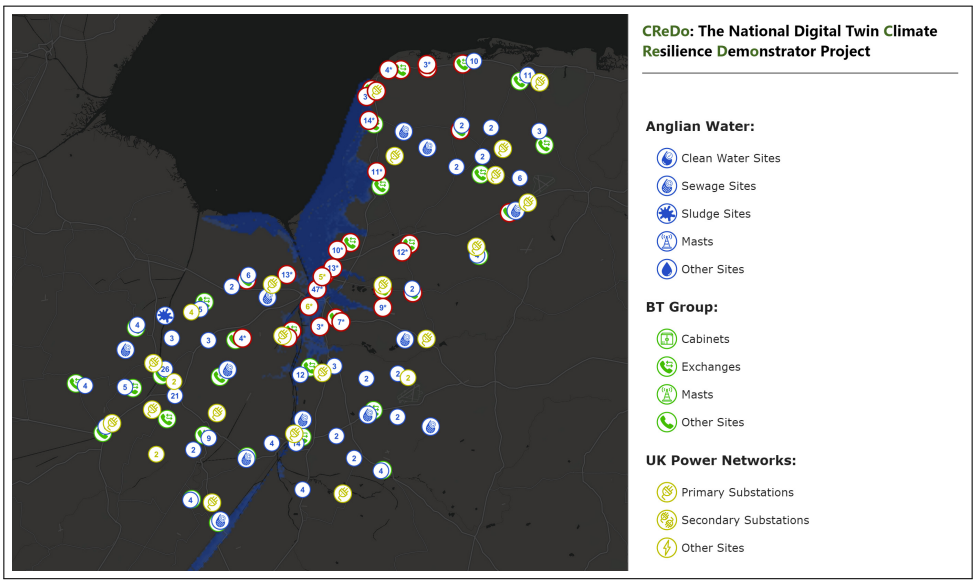
Contents

Summary

CReDo is the flagship project of the National Digital Twin programme in the UK. The CReDo digital twin demonstrates how to increase climate resilience using a knowledge graph to share data across sectors. The digital twin integrates flood simulations for different climate change scenarios with descriptions the energy, water and telecoms networks. The digital twin models the inter-dependence of the infrastructure to enable it to describe the resilience of the combined network.

by CPC and CDBB

CMCL Innovations were engaged 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 inter-dependencies between the assets with flood data to resolve the effect of the flood on individual assets and the corresponding cascade of effects across the combined network.



The digital twin was implemented on DAFNI, the *Data & Analytics Facility for National Infrastructure*. The technical approach is based on an idea proposed by CMCL Innovations, the University of Cambridge and the Cambridge Centre for Advanced Research and Education in Singapore (CARES) for how to use a dynamic knowledge graph to implement a Universal Digital Twin.

connected DT

This report describes the technical implementation and use of the CReDo digital twin. Recommendations are made with respect to (i) how the digital twin could be extended to support decisions in operations and capital planning, and real-time response to extreme weather events, and (ii) lessons for the National Digital Twin programme.

This report includes material that is internal to CReDo. Such text is formatted like this.

Findings

1 Introduction

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

by CPC and CDBB

CMCL Innovations were engaged as part of CReDo to develop a digital twin to describe the inter-dependencies 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 the description of the assets with data from flood simulations, and with models describing the effect of the flood on individual assets, on each individual network and on the combined network.

There were also several constraints. A version of the digital twin for use by the asset owners was required to be delivered on DAFNI, the *Data & Analytics Facility for National Infrastructure* [1]. It was agreed that the data supplied by the asset owners would be kept confidential by securing it on (and not allowing it to leave) DAFNI. Software development was to take place on a virtual machine hosted by DAFNI. Finally, the software developed as part of CReDo would be published under a permissive open-source licence so that asset owners and third parties could subsequently test the ideas on their own data. As a consequence, it was necessary to create synthetic data to support public-facing as well as asset-owner facing versions of the digital twin.

The **purpose of this report** is to document the implementation of the CReDo digital twin. The work described in the report was performed by CMCL Innovations as part of the CReDo project between September and December 2021. 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 support decisions in operations and capital planning, and real-time response to extreme weather events.

2 Overview

This section describes the organisation of the 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 data include information about the type, location and operational state of each asset, and the physical and logical connections between assets. The digital twin uses knowledge of the connectivity to resolve the system-wide cascade of effects caused by a failure in any of the networks. The digital twin is accompanied by a visualisation that displays top-level assets from each network. Clicking on an asset enables the exploration of detailed information about the asset, its connectivity and operational state via a side panel.

The decision to use a knowledge graph was informed by a proposal for how to implement a Universal Digital Twin [2] that supports the interoperability of distributed data across sectors, and ensures that the data are connected, discoverable and queryable via a uniform interface. This approach provides a convenient way to represent arbitrarily structured data and lends itself to describing the connectivity between assets.

Figure 1 shows a schematic of the CReDo digital twin workflow. The workflow starts by ingesting asset and flood data into a knowledge graph. The flood data 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 workflow updates the knowledge graph with the flood depth at each asset.
- The workflow extracts data from the knowledge graph for processing by individual asset effect models that describe how the flood effects the operational state of each asset.
- The workflow applies a knowledge-graph based information cascade model to resolve the effect on the combined network and propagates the changes back to the knowledge graph.
- The workflow extracts data from the knowledge graph, this time for processing by system-wide impact models. The models describe the cascade of effects throughout each individual network and then through the combined network.
- The workflow re-applies the knowledge-graph based information cascade model and propagates the changes back to the knowledge graph.

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

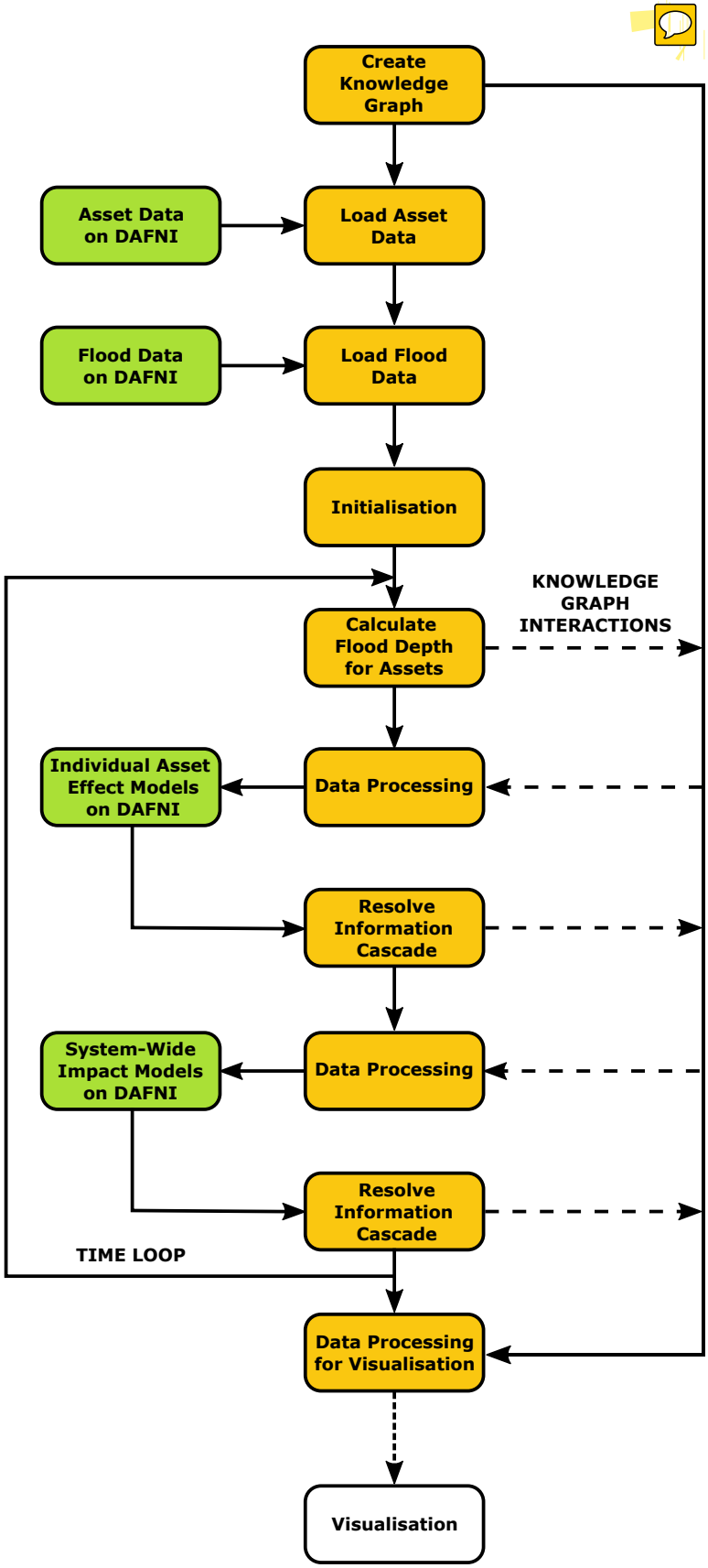


Figure 1: Schematic of the CReDo digital twin workflow.

The workflow was implemented on DAFNI [1]. This provides a convenient method to interact with the data from the flood simulations and the individual asset effect and system-wide impact models, all of which are also hosted on DAFNI. However, this choice also involves compromises. The knowledge graph only persists for the duration of the workflow. This means that the digital twin must be recreated by reloading the data each time the workflow is run. This is inconsistent with the vision of the National Digital Twin programme, where it is implicit that the digital twin should persist. However, it was a pragmatic way to simplify the process of protecting the data provided by the asset owners whilst implementing the digital twin.

The technical details of the implementation of the components in the central column of Figure 1 (orange and white boxes) are described in section 3. The choice to use the individual asset effect and system-wide impact models in addition to the information cascade model is discussed. The other components (green boxes) were developed separately and are described elsewhere.

3 Technical details

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

The code developed by CMCL Innovations and the synthetic data developed by the Connected Places Catapult on behalf of the CReDo project 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 CreDo 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).[1] Data properties links an instance of a class (the domain) to a data element (the range).[2] Object properties and data properties may be structured hierarchically.

The CreDo digital twin uses two types of ontology:

- A top-level ontology is used to define core concepts that apply throughout the digital twin.
- Domain ontologies are used to define concepts relevant to specific asset classes, in this case the energy, water and telecoms networks.

The ontologies are hierarchical, where the domain ontologies inherit from and extend the concepts and relations defined by the top-level ontology. By first intent, the queries acting on the digital twin are defined in terms of the top-level ontology, using the inheritance relations to retrieve data about individual assets from the domain ontologies. This is important because it allows the 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.

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

3.11 Top-level ontology

The top-level ontology defines the core concepts that apply throughout the digital twin. The main components of the ontology are shown schematically in Figure 2. The filled (orange) 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.

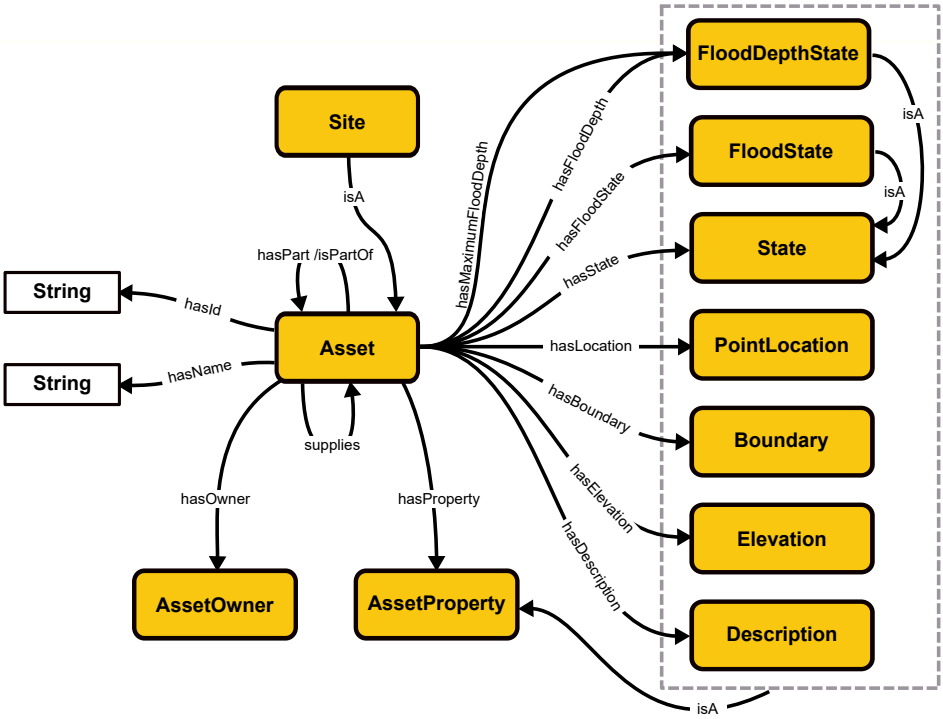


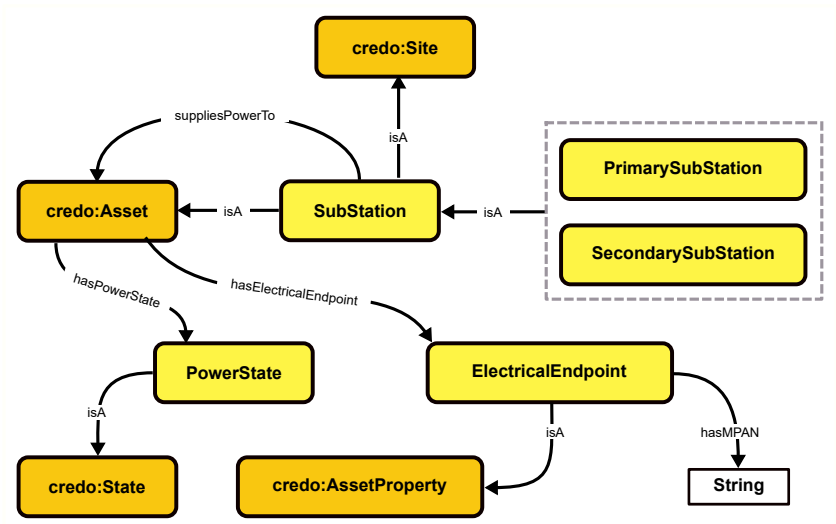
Figure 2: Top-level ontology (simplified).

The central concept of the top-level ontology is the *Asset* class. The class supports *hasPart* and *isPartOf* properties that describe the physical part-hood relationships between assets, and a *supplies* property that provides a base property for describing the 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. Lastly, a *Site* class was added to support the visualisation (which currently only shows assets down to a site level).

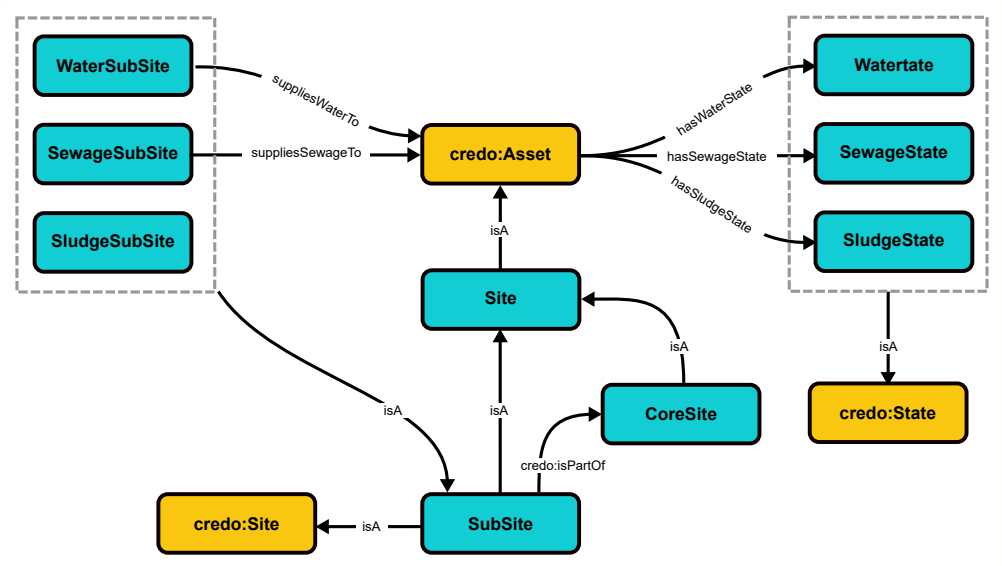
The top-level ontology was sufficient to describe all asset data included in the CReDo digital twin. The choice to use a minimal ontology was pragmatic and motivated both by the desire to keep things as simple as possible.

3.12 Domain ontologies

The domain ontologies define the concepts that are necessary to describe each asset network. The main components of each domain ontology are shown schematically in Figure 3. The filled yellow, cyan and blue boxes represent classes used to describe the energy, water and telecoms networks respectively. The filled orange boxes represent classes from the top-level ontology, showing the inheritance relations that were used to extend the top-level ontology.

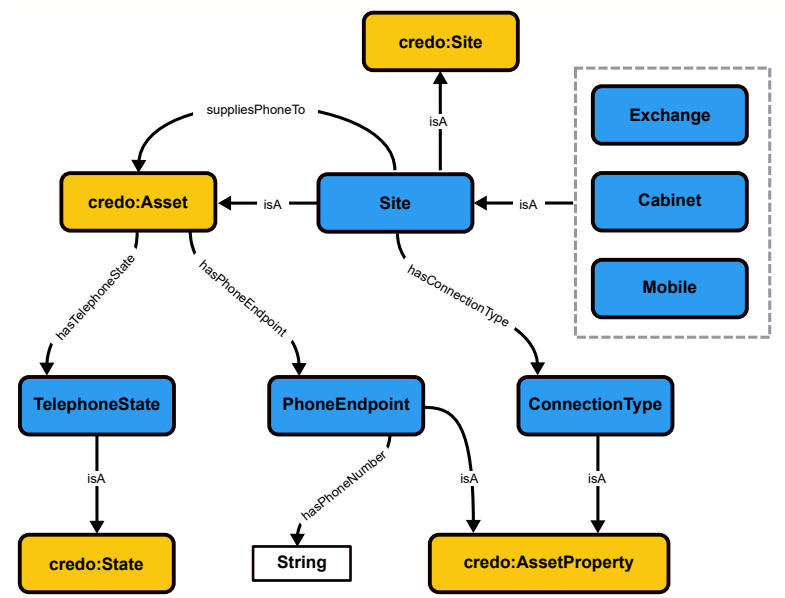


(a) Energy network.



(b) Water network.

Figure 3: Domain ontologies (simplified), part 1.



(c) Telecoms network

Figure 3: Domain ontologies (simplified), part 2.

The domain ontologies introduce specialisations of the top-level *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 what is being processed (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 domain ontologies introduce specialisations of the top-level *supplies* property: *suppliesPowerTo*, *suppliesWaterTo*, *suppliesSewageTo* and *suppliesPhoneTo*. These provide the basis for describing the inter-dependencies of the energy, water and telecoms networks.

The domain ontologies also introduce specialisations of the top-level *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 communication is available at an asset.

This approach is readily extensible and could be modified to include, for example, the state of backup power from batteries and generators, the state of charge of the batteries and the fuel available for the generators, or the availability of mobile voice and data signals. 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 the use of ‘phone’ and ‘telephone’ in Figure 3c is noted. This is typical of the sort of inconsistency problem that inhibits 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, and to adopt the recommendations for grounded ontologies that are anticipated to arise from the work of the National Digital Twin programme to develop a Foundation Data Model [3] as part of an Information Management Framework [4].

3.2 Knowledge graph hosting

The digital twin is implemented 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) [5] triples and can be queried via SPARQL operations [6]. This provides an extensible data structure that is well suited to describing the connectivity between assets.

3.21 Relational database and ontology-based data access

The data used by the digital twin that are inherently tabular or geospatial in nature are hosted using a PostgreSQL [7] relational database (RDB) with the PostGIS extension [8] to provide geospatial functionality. An Ontop server [9] is used to provide ontology-based data access (OBDA) to the RDB 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) [10] format. However, the CReDo 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 will be 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 those variables. Ontop is able to analyze the mappings so that only the parts of the SQL query required to answer the SPARQL query are actually sent to the RDB.

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

```
mappingId   example-site

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

source      SELECT 'Shortcode' AS id FROM
```

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 standardized 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 can 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) raster 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 in order for the digital twin to function.

3.22 Graph database

A Blazegraph [11] graph database hosts the remaining data used by the digital twin. The data consist of the property and class definitions for the ontologies, data that require read and write access such as the state of the assets, and meta-data supporting the resolution of the information cascade through the knowledge graph (see Section 3.5). 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 [2] as part of the World Avatar research project [12].

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

3.23 Query federation

The digital twin uses federated queries to retrieve data from the OBDA and graph databases via a single SPARQL endpoint. There are several options for how to perform such federated queries.

The current digital twin uses the standard federated query functionality [13] provided by Blazegraph. This uses the `SERVICE` SPARQL keyword to specify explicitly the sub-queries that should be sent to the Ontop endpoint. This is efficient but the requirement of knowing where things are stored is less than optimal. An alternative might to use the FedX [14] server that is provided as part of the RDF4J [15] library. This method would provide a virtual SPARQL endpoint that automat-

ically routed each part of a query to the appropriate constituent endpoint(s). Teiid [16] 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. Further research would be needed to explore possible authentication methods beyond the defaults.

3.3 Data ingestion

The data incorporated into the digital twin was supplied in a variety of formats. The data from the asset owners had been exported from RDBs, so was mostly in form of tabular data in addition to some shapefiles [17] and raster data. The floods were described by raster data, with different rasters for different time points and different climate scenarios.

3.31 Asset data

The tabular data describing the assets was provided in the form of Excel and CSV files. The data were pre-processed using a 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 [18] and set up using the PostgreSQL client tool [7]. 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 [18] for uploading data. Full details of the coverage of the digital twin, missing data and assumptions are given in Appendix B.

3.32 Flood data

The flood depth data are uploaded from inlined GeoTIFF files to the PostgreSQL RDB using a tool supplied with the PostGIS client [8]. Where required, file format conversions are performed using GDAL [19]. This ensures that all of the geometric shapes and raster data are in the correct format and that the required auxiliary information is correctly generated in the RDB.

The data ingestion has been tested with flood data from two sources:

- Data from HiPIMS, the *High-Performance Integrated Hydrodynamic Modelling System*. HiP-

IMS outputs data in an ASCII raster format[3] defined by Esri [20]. The model is capable of describing 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 describes floods for different probabilistic scenarios (*i.e.* 1 in 20 year, 1 in 200 year and 1 in 1000 year events). The data are provided as TIFF image and world file pairs[4] 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 able to be loaded in one go. The name of the original files is also stored alongside the raster data in the RDB. When the flood data is queried, the filename is used to infer the time step of the flood simulation. This approach mirrors the practice of other software, including GeoServer [21], 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 is able to be displayed in the visualisation and was included for the purpose of showing the region of interest for the flood simulations.

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 1. 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 for each asset is stored in the knowledge graph as a time series using the World Avatar time series client [12].

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 inter-dependencies 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. This synthetic data enables a representative, yet simplified, demonstration of the CReDo digital twin in the context of 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 a sample

[3] 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.

[4] See https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/world-files-for-raster-datasets.htm.

digital twins and test the integration of their own data (*e.g.* Highways Agency), 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.

3.41 Region

The synthetic data are bounded by a longitude and latitude range, (0.37468, 0.43573) and (52.74937, 52.78721) respectively, defined as per the World Geodetic System 1984 (WGS 84, also known as EPSG 4326) coordinate reference system.

3.42 Location

The synthetic data contains fewer assets than the real data to avoid unnecessary complexity during demonstrations of the digital twin. Table 1 summarises the number of assets in each network. 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.

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

Table 1: Number of assets in the synthetic data.

3.43 Connectivity

The synthetic data follows the general trends seen in the connectivity in 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 acyclic. The sewage network ends near a river and follows a downhill topography. The majority of water assets are provided power by secondary substations.

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, ground level was defined using the flood data, which was provided relative to ground level.

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 failure 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 naturally lends itself to describing the interdependencies between assets. Furthermore, the flexibility of the knowledge graph also makes it possible to encode information to link individual assets to models that describe the behaviour of the asset (or set of assets), in this case in response to a flood. This approach enables the knowledge graph to combine multiple models, for example models to describe the effect on individual assets and separate models to describe the effect on networks of assets.

The current digital twin combines individual asset effect and system-wide impact models in addition to a separate information cascade model. The individual asset effect and system-wide impact models are explicitly incorporated in the workflow in Figure 1 and exist as separate models on the DAFNI platform. This makes it straightforward to swap different versions of these models in and out of the digital twin. As the names suggest, the individual asset effect model acts 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 acts across the entire knowledge graph. It provides a complementary means of resolving the cascade of effects across the network, and a default option in the absence of other models and/or while more specific models are developed. The implementation of the information cascade model takes advantage of an experimental derived information framework that is being investigated as part of the World Avatar research project [12] to support the implementation of digital twins using a dynamic knowledge graph [2].

This section describes the implementation of the information cascade model. The individual asset effect and system-wide impact models were developed separately and are described elsewhere.

3.51 Representation of dependencies

Information describing the dependencies between assets, and hence the specialisations of the *State* class introduced by the 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 4 shows an example. Whether Asset A has power depends 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 class 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 resolve the information cascade through the knowledge graph.

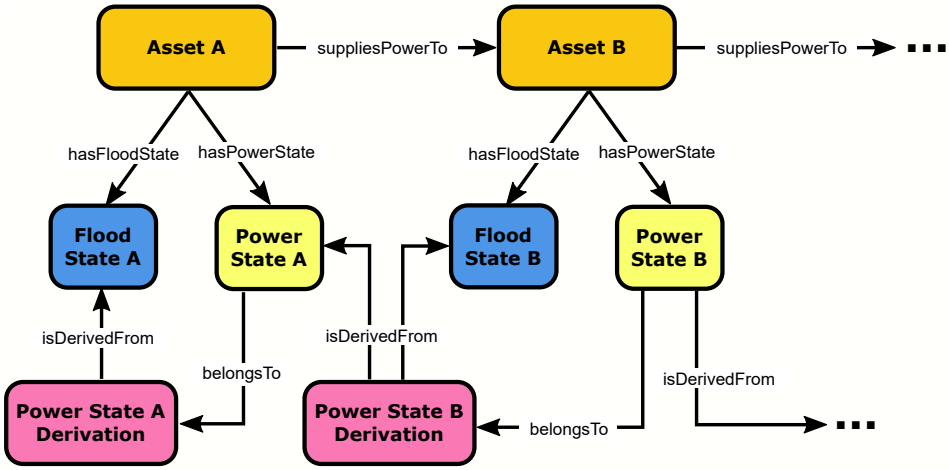


Figure 4: Representation of dependencies between assets (simplified).

This approach seeks to minimise the number of operations that are required to cascade changes through the knowledge graph. The approach has been shown to work when the dependencies form acyclic graphs. Research is required to consider how the approach might deal with circular dependencies and cases where several models (*e.g.* the individual asset effect and system-wide impact models) affect the state of an asset. Future research should also formally assess the computational efficiency of the approach.

Note that acyclic graphs do exist in the connectivity of the 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 in the current digital twin. In this sense, the information cascade and system-wide impact models are seen as complementary approaches, each with advantages and disadvantages.

3.52 Update logic

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 model is embedded in the fabric of the knowledge graph because it is entirely controlled by the information represented in the knowledge graph.

An update agent is used by the information cascade model to update the state of individual assets. The update logic applied by the agent is summarised in Table 2. The dependencies listed in the table 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 ($\mathrm{TS}_1, \mathrm{TS}_2, \cdots$). 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 from the update logic. 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, $\wedge\,(\bigvee_{i=1}^{n} \mathrm{TS}_i)$. The update logic is currently hard coded within the update agent, but could ultimately be specified within the knowledge graph.

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: Update 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 information cascade is triggered by the update operations, which recursively traverse the network of dependencies between the states of assets, updating each state before updating things the depend on the state. At each step in the cascade, the knowledge graph is queried to retrieve information about how to perform the update and the next state to be visited.

The individual asset effect models and system-wide impact models provide input to the cascade via files that contain a list of values that are inserted into the knowledge graph, after which a full cascade is triggered. In lieu of the availability of final individual asset effect and system-wide impact models, the current digital twin forces a full update of the network at each iteration of the time loop in Figure 1. The forced update reassess the flood state of each asset, which as per Table 2, 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. This needs to be revisited and replaced with something more suitable in the future.

3.6 Data processing

Data is extracted from the digital twin at three points in the workflow in Figure 1. At each point, data is 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 effect and system-wide impact models. The main advantage of this is abstraction: the other models do not need to know

how to access the knowledge graph directly. Lastly, the output files produced by this process are also used for visualisation purposes (see Section 3.7).

The data processing code executes a sequence of SPARQL queries to obtain information from the knowledge graph. The queries constructed purely on the basis of the top-level ontology described in Section 3.1. This use of the top-level ontology provides a level of abstraction. New domain ontologies can be added without the need to change any of the data processing queries.

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 top-level 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.

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

```
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 .}
```

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 2).
- 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.

The data processing offers an opportunity to analyse the data, for example, by counting how many things are supplied by each asset as a first attempt at providing critically scores for the assets. After 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 are easy to process by the other models.

The current data processing outputs information from the entire knoweldge graph. This was useful during the development of the digital twin because it ensured that the individual asset effect and system-wide impact models had access to the full data set. 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 and, perhaps most importantly, it negates many of the benefits that models might access if they were able to interactively traverse the data structure of the knowledge graph. Alternative approaches should be considered.

3.7 Visualisation

The assets, connections between assets, and cascade of failures extracted from the digital twin by the data processing can be viewed using a browser-based visualisation. The visualisation presents a map that allows the geospatial elements of the data 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 meta and time series data. Additional controls provide a detailed connection view to facilitate the exploration of the connections to and from individual assets.

3.71 Implementation

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 [22] is used to host a folder containing the visualisation files.

A number of mapping libraries are available to facilitate interactive online maps (Google Maps [23], Leaflet [24], OpenLayers [25] *etc.*). The current visualisation uses the Mapbox [26] 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 [21] 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 [27] 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.

3.72 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 data sets can be added simply by adding the corresponding geospatial and meta data 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. In the future, this could be extended to include other raster data, 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 effect, system-wide impact and information cascade models, for example to asses 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:

- **Exploration of connectivity.** Add controls to view the connections to and from a specific asset, and to traverse the network of connections by clicking on connected assets. This would make it easier to explore the connectivity of assets.
- **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 failure 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. 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, reducing the loading time for the visualisation.
- **Interactive granularity.** The visualisation will become increasing 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 failure 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, if the visualisation shows all the electricity pylons between a substation and an asset, the connectivity between individual pylons would be shown. 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 relies on post-processed data. Consideration should be given as to whether it would be beneficial to have a 'live view' that shows real-time content from the digital twin. Whether this is beneficial will depend on how the use cases for the digital twin evolve.

3.8 Implementation on DAFNI

3.81 DAFNI Workflow

DAFNI [1] provides a graphical platform [28] 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) [29]. 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 [30] – an extension of the Kubernetes [31] container orchestration software. The workflow in Figure 1 was implemented in Argo by creating Podman [32] 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 in order 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 5 shows the containers (orange boxes) included in the Argo workflow template file, the input data (green boxes) and parameters that they require, and the outputs that they produce (white boxes). The containers that host the knowledge graph must be created and persist while the rest of the workflow runs in order 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 also includes a 'readiness probe' for each knowledge graph container, which checks that they are 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.

DAFNI does not yet support all of the Argo features that are required by the CReDo digital twin, so the approach to date has been to work directly with Argo to develop a suitable workflow template. The Argo workflow can be executed by running Argo locally and has been implemented and tested using dummy data, and should be able to run on DAFNI with minor changes. Deployment of the Argo workflow on DAFNI and work to add DAFNI support for the Argo features used by the CReDo digital twin is being undertaken by the DAFNI team in collaboration with CMCL Innovations.

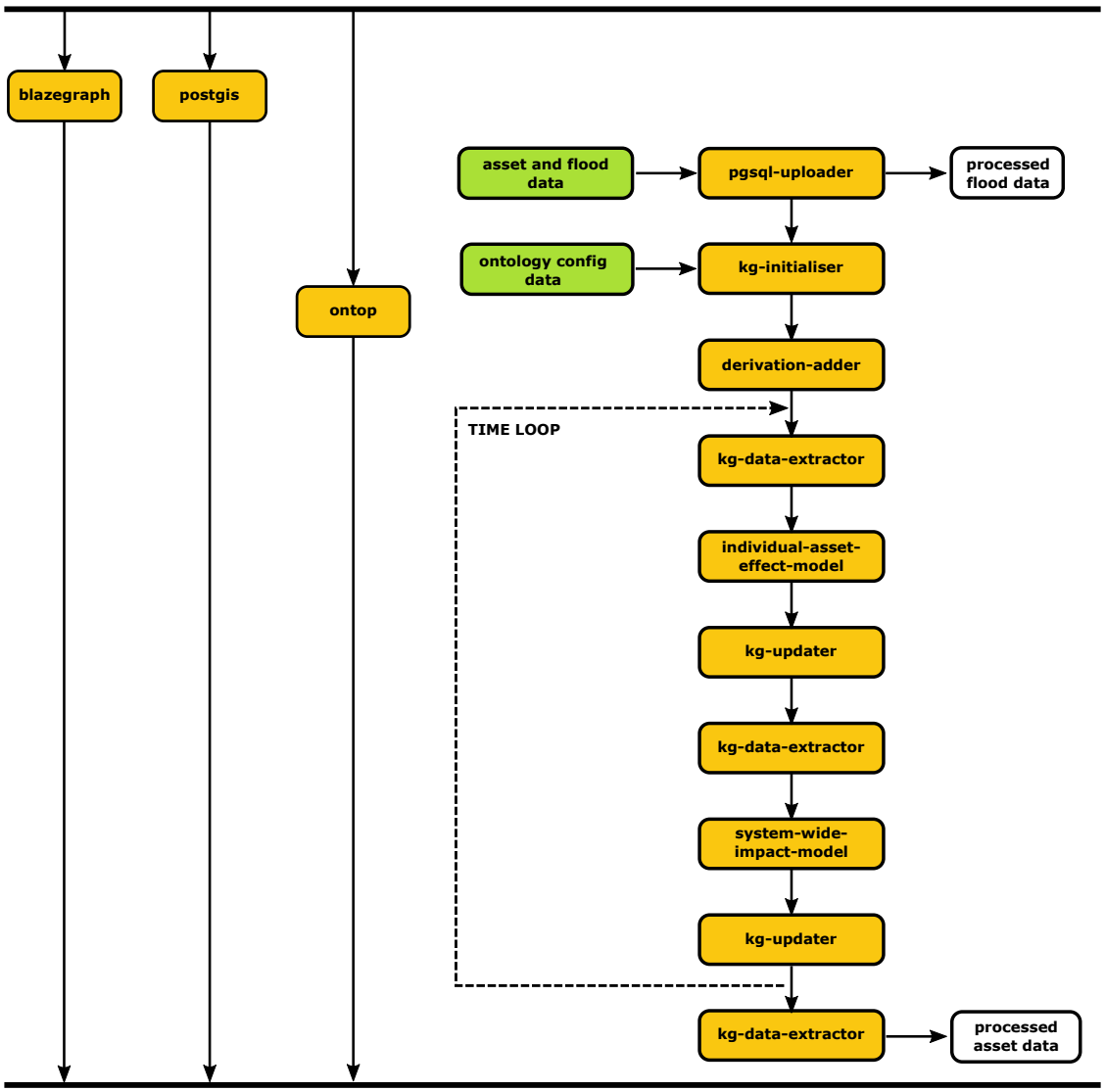


Figure 5: Argo implementation of the CReDo digital twin workflow. The orange boxes show the Podman containers incorporated into the workflow. The green boxes show input data. The white boxes show output data.

3.82 Visualisation

The CReDo digital twin visualisation is designed to run as a standalone web application, accessed through a browser. In order to run the visualisation via the DAFNI platform, it will need to retrieve and display the workflow output from the NID data store used by DAFNI. Existing DAFNI visualisations use Keycloak [33] to obtain an authentication token that provides access to NID; efforts are currently underway by the DAFNI team, in collaboration with CMCL Innovations to add similar functionality to the CReDO visualisation.

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 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.Since there is no public access to the Argo server on DAFNI, running the workflow requires liaising with DAFNI staff, who can then submit a new workflow instance with the required input parameters.

By default, the CReDo digital twin will use synthetic data (see Section 3.4). In order to use other data, or switch between flood data corresponding to different climate scenarios, users can search the DAFNI data catalogue [34] 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 data set in the NID, ready to be loaded into the visualisation.

4.2 Visualising data from the digital twin

The visualisation 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 networks of assets from different sectors.

Figure 6 shows the basic layout of the visualisation. The controls are situated on the left of the visualisation. They present a number of 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 that is shown on the map.

- The current **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.

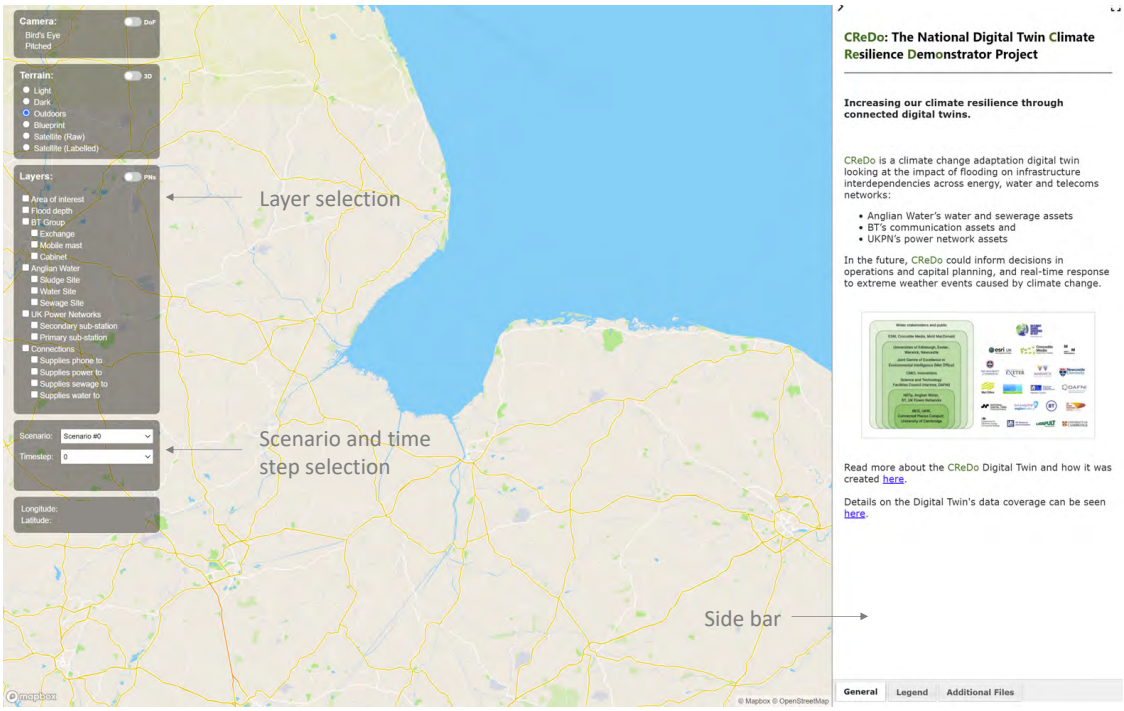


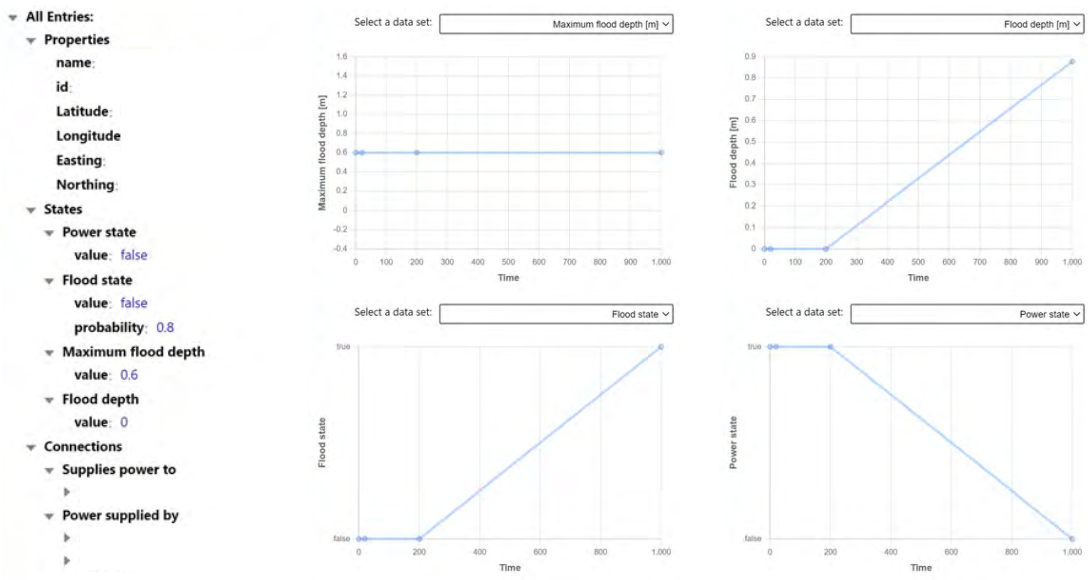
Figure 6: Layout of the visualisation. Note that some information is redacted to protect confidential data.

Figure 6 shows different EA flood scenarios selected via the timestep control.[5] 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 a number in the icon indicating the number of assets in the cluster. Increasing the zoom level will reveal 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 individual 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, sidebar will display detailed information about the asset, including the name of the asset, meta data about the asset, meta data 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 8.

[5] Note that it would be more correct to treat the EA flood data as scenarios, see page 16.

Figure 7: Visualisation of flood data (left to right, EA flood data for 1 in 20 and 1 in 1000 year events), overlaid with locations of assets (third panel from the left) and the legend from the sidebar. Failed assets are indicated by red circles around the asset markers.



(a) Meta data. *(b) History of asset states.*

Figure 8: Visualisation of asset properties. Note that some information is redacted to protect confidential data.

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 and see what depends on what.

Figure 9 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 right panel shows a focused view of the direct connections to the selected asset. The selected asset is a secondary substation. The direction of the arrows on the connections show that it is able to be 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 connection from the secondary substation is to a clean water site, which is therefore also compromised.

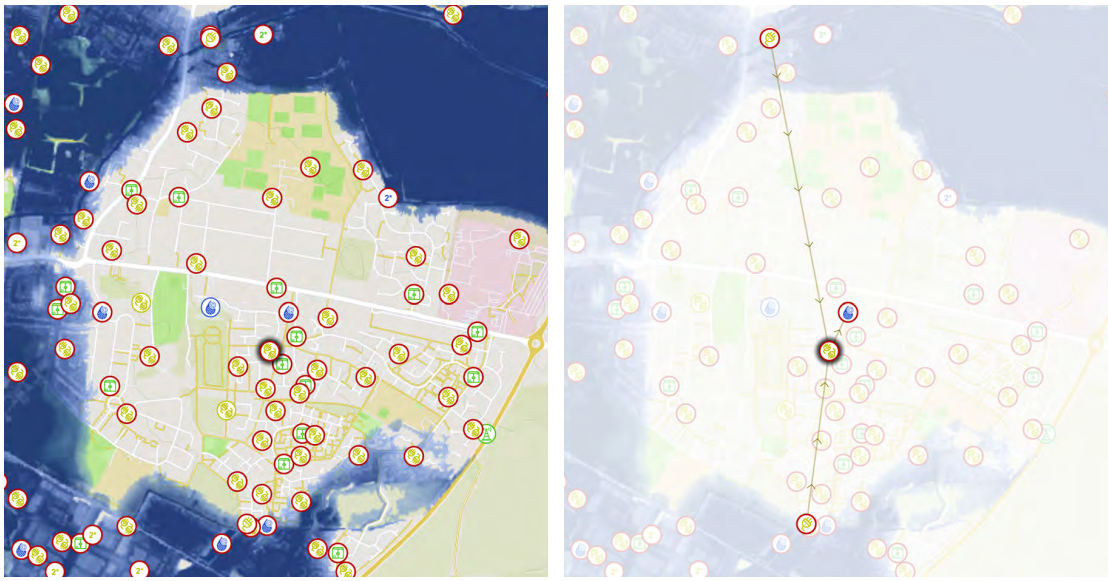


Figure 9: Visualisation of assets and connections. Assets in the vicinity of a flood (left panel). The dark ring around the asset in the centre of the image indicates that it has been selected. Focused view of the selected asset and its direct connections (right panel).

It is clear that most of the assets in Figure 9 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 create a digital twin that integrates data across sectors.This section summarises what has been achieved, discusses lessons learned and makes recommendations for future developments.

A digital twin has 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 a description of the inter-dependencies between assets from the energy, water and telecoms networks with data from flood simulations, and with models describing the effect of the flood on individual assets, on each individual network and on the combined network. The digital twin was developed using data describing:

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

By combining the digital twin with flood simulations for different climate change scenarios, it is possible to begin to explore the resilience of the combined network and identify vulnerabilities.

This report documents the technical implementation of the CReDo digital twin and explains how to use it. The digital twin and an accompanying visualisation will be made available on DAFNI, the *Data & Analytics Facility for National Infrastructure* [1]. The corresponding software and a synthetic data set are published under a permissive, mostly open-source, licences so that asset owners and third parties can test the ideas on their own data and contribute to future developments.

The technical approach is based on an idea for how to use a dynamic knowledge graph to implement a Universal Digital Twin [2]. The knowledge graph supports the interoperability of distributed data across sectors and provides a convenient way to represent and query arbitrarily structured data via a uniform interface, and lends itself to describing the connectivity between assets.

The data in the knowledge graph are represented using a set of hierarchical ontologies. A top-level ontology is used to define core concepts. Domain ontologies that inherit from and extend the top-level ontology are used to define specific concepts and relations relevant to the energy, water and telecoms networks. The queries acting on the digital twin are defined in terms of the top-level ontology, using the inheritance relations to retrieve data from the domain ontologies. This approach separates the details of the domain ontologies from the core business logic of the digital twin. This was advantageous because it allowed the domain ontologies to evolve during the development of the digital twin, and will make it simpler to extend the digital twin in the future.

An ontology-based data access solution was used to map the asset data to the ontologies. This

allowed the asset data to be accessed via the knowledge graph without the need to perform any data conversion and made it straightforward to recreate the knowledge graph as and when the ontologies changed during the development of the digital twin. Ontology-based data access potentially offers a means to map existing sources of data directly to the Foundation Data Model [3], and together with use of hierarchical ontologies, aligns with the approach taken by the Information Management Framework [4] that is being developed by the National Digital Twin programme.

The implementation of the digital twin on DAFNI. Limitations.

The current digital twin allows... Basic data interoperability very significant for AO.

It is recommended that future iterations of the digital twin...

The coverage of the digital twin and the data processing could be extended in a number of directions. The breadth of the digital twin should be extended to include other assets and to cover other regions of the country. The choice to formulate the data processing queries in terms of the top-level ontology is intended to minimise the need for disruptive changes when making this type of change. The depth of the digital twin should also be extended. A more granular description of assets and the ability to include more types of data will be required to support more detailed modelling of failure scenarios. 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.

The choice of what failure scenarios to consider should drive the choice of what additional data to include. A sufficiently granular description of the assets would also make it possible to do things like changing the switching state of the energy, water or telecoms networks in the digital twin. This would open the door to automating the process of using the digital twin to evaluate the consequences of mitigating actions recommended by models operating on the digital twin. Some aspects of this could be straightforward. Others might require some research.

The way in which the state of assets is described should be given more detailed consideration. Currently 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.

The digital twin could also be extended to incorporate other types of data that would enhance how the digital twin could be used. The inclusion of data about buildings, for example, might facilitate the reporting of different measures of the criticality of assets in terms whether they supply hospitals or schools *etc.* The inclusion of live data, for example data from sensors reporting river levels, would move the digital twin in the direction of supporting operational decisions.

The capabilities of the visualisation should be extended to facilities more ways to use the digital

twin. Features to search for specific sites, to make it easier to explore and visualise connections, and to understand the cause and effect relationships in a given failure scenario would add immediate value. As the data available via the digital twin becomes more granular, it is anticipated that it would be useful to develop the visualisation to show abstract logical connections based on whatever is visible at the time, so So for example, the connections between an 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. The visualisation should also be developed to use lazy loading to prevent it slowing down as more data is added to the digital twin. Finally, options should be considered for how to show 'live' (as opposed to post-processed) data. This would be required to support some operations-type decisions.

The digital twin describes the cascade of failures throughout combined network. CReDo has focused on floods, but the digital twin could describe the cascade of effects due to any type of failure. Future work should consider the extension of the digital twin to consider other other uses of the digital twin, including the consideration of types of failure. This would necessarily go hand in hand with the consideration of new types of data.

Future developments should seek to engage additional partners. The involvement of the Environment Agency would naturally align with the consideration of flood scenarios, and might facilitate access to data from environmental sensors. The state of the road network is an important component of many failure scenarios, for example is a road passable so that someone might reach an asset in the event of a flood? The involvement of the Highways Agency might therefore be helpful. The involvement of Ordnance Survey is also likely to be important, and would facilitate the integration of a lot of useful data. Roads and buildings are a case in point.

Nomenclature

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
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
WGS	World Geodetic System

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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 (accessed Dec 2020), 2004.

[6] *SPARQL 1.1 Overview, W3C Recommendation*, https://www.w3.org/TR/sparql11-overview (accessed Dec 2021), 2013.

[7] *PostgreSQL*, https://www.postgresql.org (accessed Dec 2021), 2021.

[8] *PostGIS*, https://postgis.net (accessed Dec 2021), 2021.

[9] 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.

[10] *R2RML: RDB to RDF Mapping Language, W3C Recommendation*, https://www.w3.org/TR/r2rml (accessed Dec 2021), 2012.

[11] *Blazegraph*, https://blazegraph.com, source code available at https://github.com/blazegraph/database/wiki/About_Blazegraph (accessed Dec 2021), 2021.

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

[13] *SPARQL 1.1 Federated Query, W3C Recommendation*, https://www.w3.org/TR/sparql11-federated-query (accessed Dec 2021), 2013.

[14] Eclipse Foundation, *Federation With FedX*, https://rdf4j.org/documentation/programming/federation (accessed Dec 2021), 2020.

[15] Eclipse Foundation, *Eclipse rdf4j*, https://rdf4j.org (accessed Dec 2021), 2021.

[16] Red Hat, *Teiid: Cloud-native data virtualization*, https://teiid.io (accessed Dec 2021), 2021.

[17] ESRI, *Shapefile Technical Description*, https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf (accessed Dec 2021), 1998.

[18] *csvkit 1.0.6*, https://csvkit.readthedocs.io/en/latest (accessed Dec 2021), 2016.

[19] *GDAL*, https://gdal.org (accessed Dec 2021), 2021.

[20] *Esri GIS Mapping Software*, https://www.esri.com (accessed Dec 2021), 2021.

[21] *Geoserver*, http://geoserver.org (accessed Dec 2021), 2021.
[22] *Apache HTTP server*, https://httpd.apache.org (accessed Dec 2021), 2021.
[23] *Google Maps Platform*, https://developers.google.com/maps (accessed Dec 2021), 2021.
[24] *Leaflet*, https://leafletjs.com (accessed Dec 2021), 2021.
[25] *OpenLayers*, https://openlayers.org (accessed Dec 2021), 2021.
[26] *Mapbox*, https://www.mapbox.com (accessed Dec 2021), 2021.
[27] *ChartJS*, https://www.chartjs.org (accessed Dec 2021), 2021.
[28] *DAFNI Platform*, https://facility.secure.dafni.rl.ac.uk (accessed Dec 2021), 2021.
[29] *National Infrastructure Database*, https://dafni.ac.uk/the-national-infrastructure-database-nid (accessed Dec 2021), 2021.
[30] *Argo*, https://argoproj.github.io (accessed Dec 2021), 2021.
[31] *Kubernetes*, https://kubernetes.io (accessed Dec 2021), 2021.
[32] *Podman*, https://podman.io (accessed Dec 2021), 2021.
[33] *Keycloak*, https://www.keycloak.org (accessed Dec 2021), 2021.
[34] *DAFNI Data Catalogue*, https://facility.secure.dafni.rl.ac.uk/data (accessed Dec 2021), 2021.

Version Control

Version	Date	Author	Status	Change Description
0.1	10 Dec 2021	Jethro Akroyd	Draft	Document created
0.2	17 Jan 2022	Jethro Akroyd	Draft	First draft

Authors and Contributors

Lead Author

Jethro Akroyd

Contributors

Amit Bhave
George Brownbridge
Elliot Christou (Connected Places Catapult)
Michael Hillman
Markus Kraft
Jiawei Lai
Kok Foong Lee
Sebastian Mosbach
Daniel Nurkowski
Owen Parry

Edited by

CMCL Innovations

www.cmclinnovations.com

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.

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 Missing data

Water Network

- Missing distribution zones for water supply.
- Missing granularity of connectivity at the sub-site level and below.

Energy Network

- Missing power connections to mobile masts.

Telecoms Network

- Missing locations of legacy cabinets.
- Missing outgoing connections from fibre cabinets.
- Missing mobile signal coverage provided by mobile masts (raw data is available but not considered at the moment).
- Missing connections between telephone exchanges and mobile masts.
- Missing MPANs of the mobile masts and therefore the connectivity to the electrical grid.

B.3 Assumptions

Common

- All assets have a 'maximum flood depth', which is the depth of water at which an asset fails.
- The maximum flood depths use assume 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 available.

Water network

- The data grouped sub-sites geographically into core sites, which were 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.

[redacted]

Energy network

- Power supply connections mapped by matching MPANs from assets to MPANs supplied by substations. Some of these are known to be incorrect in the original data, as confirmed by UKPN.
- The connectivity between primary and secondary substations was generated by UKPN using heuristics rather than known connections.

Telecoms network

- No assumptions.