Implementing a Flood Management Digital Twin Using the Dutch Land Registry and Mapping Agency’s Knowledge Graph

Kristen Phan
Faculty of Electrical Engineering, Mathematics, and Computer Science
University of Twente
Enschede, Netherlands
g.t.h.phan@student.utwente.nl

Ruben van Zenden
Faculty of Electrical Engineering, Mathematics, and Computer Science
University of Twente
Enschede, Netherlands
r.vanzenden@student.utwente.nl

Vibha Ravindra
Faculty of Electrical Engineering, Mathematics, and Computer Science
University of Twente
Enschede, Netherlands
v.ravindra@student.utwente.nl

***Abstract*—Kadaster is a Dutch Land Registry and Mapping Agency. Since 2019, Kadaster has been constructing its knowledge graph, referred to as Kadaster Knowledge Graph (KKG) with the objective of expanding KKG into a knowledge base for a Digital Twin (DT) of the living environment and enabling public data to be widely used for public goods and services. In this paper, we propose a use case for KKG, a flood management DT that provides significant value to the flood management practice for the Netherlands and makes use of KKG as its data source. Additionally, we discuss how the use case can be implemented from the data, KKG’s architecture, and flood simulation DT tool standpoint.**

Keywords—Dutch Land Registry and Mapping Agency, Kadaster knowledge graph, knowledge graph, flood management, digital twin, FAIR data train, INSPIRE data models, foundational data model, Ontological Design Patterns (ODPs), Ontology Quality Methodology, Linked Stream Data, distributed machine learning

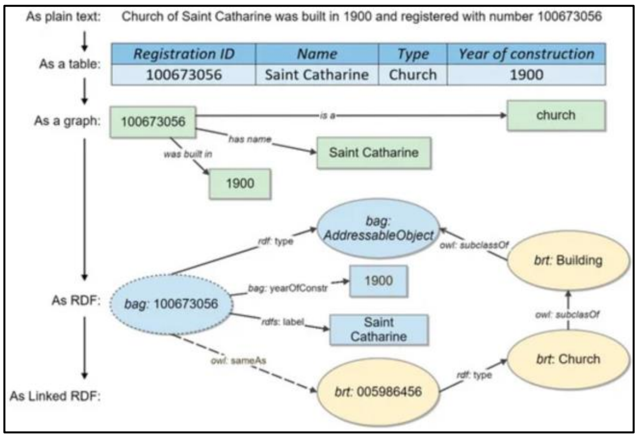
I. Introduction

In 2012 Google officially coined the term “Knowledge Graph” (KG), a data representation model where data is stored in an RDF graph with nodes and edges and mapped to different ontologies which serve as the schema of the data model and define the objects within the KG, their attributes and relations to other objects [1].

Key benefits of a KG include 1) centralizing disparate data sources into a single source and a single format i.e., Linked Data (LD) and 2) allowing machines to automatically interpret the semantics of the data, traverse the KG, and retrieve the data it is interested in using a querying language called SPARQL.

Fig. 1 is an example of data represented in an RDF graph that defines the church of Saint Catharine (an object) and its attributes such as type, registration ID, name, and year of construction [2].

Note that there are two different objects in two different datasets called BAG (Basisregistraties Adressen en Gebouwen) and BRT (Basisregistratie Topografie), representing the church of Saint Catharine: *bag:100673056* and *brt:005986456*. These two objects are related via a relation called *owl:sameAs* in the OWL ontology to illustrate that they refer to the same real-life object i.e., the church of Saint Catharine.



As plain text: Church of Saint Catharine was built in 1900 and registered with number 100673056

Registration ID	Name	Type	Year of construction
100673056	Saint Catharine	Church	1900

Fig. 1. Example of RDF data [2].

Kadaster has been a front-runner in publishing its data as LD. In 2019, Kadaster took the next step of constructing a KG, called Kadaster Knowledge Graph (KKG) that continuously expands and integrates new data not only from within Kadaster but also from external sources. The vision is to expand KKG into a knowledge base for a Digital Twin (DT) of the living environment and to enable public data to be widely used for public goods and services.

With that vision in mind, we propose a use case for KKG where a flood simulation DT can be built on top of KKG for the purpose of flood monitoring and management in the Netherlands.

In section II, we elaborate on the use case including the problem it aims to solve, the high-level design of the proposed solution, and the data procurement and integration strategies.

In section III, we discuss how the use case can be implemented from the architecture standpoint, analyzing specifically the baseline and target architecture of the KKG, and a flood simulation DT tool.

II. Use Case: Food Management Digital Twin

A. *Problem Statement*

Climate change is one of the biggest concerns for the Netherlands. Almost half of the nation sits below sea level and is gradually sinking. The Netherlands has implemented several flood defenses such as dike reinforcements to mitigate the risk of flooding. “If the water comes in, from the rivers or the sea, we can evacuate maybe 15 out of 100 people, so evacuation isn’t an option. We can escape only into high buildings. We have no choice. We must learn to live with water” (Kimmelman et al, 2017). Furthermore, Rotterdam has constructed several facilities such as underground parking garages and basketball courts that turn into emergency reservoirs during a heavy storm. Excess water can be later pumped out of the reservoirs and into the sewage system once the storm has passed to avoid sewage system overflow.

However, leaning to live with water in the long term will require a more robust flood management system that leverages both static and real-time data to enable swift and continuous flood response and mitigation planning.

B. *Solution: A Digital Twin for Flood Management*

To help the Netherlands manage flood risk, a DT can be constructed and used as a flood management system. A DT is defined as a set of information constructs that digitally represent a real-world object or system [3]. A DT helps with decision-making by the usage of simulations, machine learning, and reasoning. In this context, the DT can run simulations where the flood water levels can dynamically be altered to identify the most impacted areas and optimal locations for new emergency reservoirs to be built. Additionally, the DT can enable real-time flood monitoring and flood response planning.

Due to the wide availability of information on the city, Amsterdam is chosen as the testbed for this DT.

The design of the DT in this paper originates from the DT being implemented for the city of Dublin as outlined in the paper titled “A Digital Twin Smart City for Citizen Feedback” [4]. A smart city DT consists of six layers with their own purpose:

1. **Terrain**: Captures basic landscape information, mostly the soil map of the city.
2. **Buildings**: Captures building information.
3. **Infrastructure**: Adds the infrastructure that surrounds the buildings such as public transportation systems, highways, power lines.
4. **Mobility**: Adds mobility data such as crowd density, parking data, traffic data.
5. **Digital layer/ smart city**: Collects data from the city such as weather sensor data.
6. **Virtual layer/ digital twin**: Uses data from the previous layers to perform flood simulations and real-time monitoring.

C. *Data Procurement & Integration Strategies*

In this section, we identify existing data sources to populate the six layers of the flood management DT – see Table 1. Note that as the scope of the use case gets refined, data sources might need to be added to/ removed from this table.

Furthermore, the table addresses two additional topics: 1) data procurement: open/ closed data and 2) data integration: INSPIRE format.

Layer	**Data Required**	**Open/ Closed Data**	**Available INSPIRE Format**
Terrain	Kadaster Basisregistratie Ondergrond (BRO) soil map	Open	Data theme: Soil
Buildings	Kadaster 3D Basisvoorziening	Open	Data theme: Buildings, Addresses, Coordinate reference systems

Infrastructure	Kadaster Basisregistratie Grootschalige Topografie (BGT);
	Kadaster Basisregistratie Topografie (BRT);
	RIONED Stedelijk Water (Riolering)
		Open	Data theme: Geographical names, Hydrography, Utility and governmental services – sewer network

Mobility	RDW open data parking (Dienst Wegverkeer);
	Netherlands Mobility Panel (MPN) data;
	Druktebeeld Amsterdam data
		Open except for MPN data	Data theme: Transport networks

Digital layer	RWS waterdata en waterberichtgeving (Rijkswaterstaat); KNMI weather data (Koninklijk Nederlands Meteorologisch Instituut)	Open	Data theme: atmospheric conditions, meteorological geographical features
Virtual layer	N/A	N/A	N/A

Table 1. Data required for flood management DT use case.

1) Data Procurement: FAIR Data Train for Closed Data

As shown in Table 1, some data sources are open to public use while others are closed due to the sensitive nature of the data.

Examples of closed data are MPN mobility data and possibly sewage pipe sensor data if implemented in the future as they pertain to the Netherlands’ critical public infrastructure. Note that sewage systems are currently managed by individual municipalities and do not appear to have real-time sensor data to the authors’ knowledge at the time of writing. Therefore, sewage sensor data is not included in Table 1.

While it is easy to obtain and integrate open data sources into KKG, handling closed data might be challenging. The implementation of a distributed data analytics architecture called FAIR data train is key to tackling this challenge.

First, we will explain the basics of FAIR data train. Second, we will illustrate how FAIR data train can add value to KKG and the flood management DT.

FAIR data train is an emerging concept in the health care and agriculture sector. The idea is that instead of performing analytical tasks on a centralized data repository, which might be impossible due to sensitive data being safeguarded in closed data sources, analytical tasks are performed in a distributed manner right at the data sources and later aggregated for summary statistics [5].

The FAIR train data architecture consists of three components [5]:

Station (curator) – managed by data owner who chooses to expose specific data via a standardized interface and offers a secure environment with authentication and authorization procedure for the execution of analytical tasks, acts as a FAIR data point; uniquely identifiable.

Train (consumer) – composed by data user to perform a specific analytic task on the data provided by different data stations; uniquely identifiable and reusable.

Handler (track) – acts as a gateway between Trains and Stations; directs different Trains to the appropriate Stations; aggregates results from multiple Stations and sends them back to the Trains; tracks all communication between Trains and Stations for traceability and auditing purposes.

In the context of the flood management DT use case, Kadaster can set up a *Handler* and compose a number of *Trains* to perform analytics on closed data provided by different *Stations* managed by the respective data owners.

Suppose sewage pipe sensor data is available yet remains in closed repositories managed by individual municipalities. Kadaster and the municipalities can co-develop use cases for the sensor data and design standardized interfaces for the *Station* managed by the municipalities. That way Kadaster can "send" *Trains* to the *Stations* via the Kadaster's *Handler* to perform analytical tasks.

For example, a *Train* can be composed to analyze the sensor data and returns summary statistics such as the percentage of sewage system overflow in a pre-defined neighborhood. This statistic can be added to KKG and eventually the flood management DT to support the delivery of warnings to residents in said neighborhood of potential health hazards due to sewage overflow.

Furthermore, this statistic can be relayed back to individual municipalities for their own actions such as upgrading the sewage system in a particularly vulnerable neighborhood.

2) *Data Integration: INSPIRE Data Format*

Environmental issues often have cross-border causes and consequences. For example, the Rhine River runs through the Netherlands, Germany, Luxembourg, Switzerland, and Austria, making the river flow rates measured at different river segments by different countries valuable data for formulating and coordinating flood response plans among these countries.

Therefore, it is important that countries are able to easily exchange spatial and geographical information about their environment.

In 2017, Europe introduced INSPIRE (Infrastructure for Spatial InfoRmation in Europe) to standardize the models in which European countries represent their spatial and geographical data with the intention of making the data more accessible and interoperable. As of 2021, there are 34 spatial data themes in the INSPIRE Directive such as soil, atmospheric conditions, and natural risk zones. Each data theme contains relevant standardized data models [6].

In Table 1, we map the data required for the use case with the available INSPIRE data format. At the time of writing, it appears that only the BRO dataset has several of its subsets in an INSPIRE-compliant format. We recommend Kadaster and their data partners transform their not-yet-compliant data into an INSPIRE-compliant format using the appropriate models.

If there is no available INSPIRE data model for a specific data source, Kadaster and its partners can further extend INSPIRE data models using the INSPIRE extension methodology from Wetransform (http://inspire-extensions.wetransform.to/). The extended models can be submitted to Wetransform for approval and larger adoption.

III. Use case implementation

Now that the use case and the data requirements have been defined, we will discuss the implementation of the use case from the architecture standpoint.

First, we examine KKG's baseline architecture. Second, we propose a set of high-level requirements and design considerations for KKG's target architecture. Finally, we discuss a flood management DT tool that ingests data from the KKG and performs the actual simulations and real-time monitoring.

A. *Kadaster Knowledge Graph's Baseline Architecture*

Fig. 2 illustrates the KKG's baseline architecture. Starting at the lowest layer "Data" in purple, this is where the data is ingested from multiple data sources under the steward of different data owners, some of which are part of Kadaster and some are from external organizations.

Oftentimes, the data is encoded in the GML format, an extension of UML for geographical data. This data is not LD yet and is dispersed across multiple data silos. It is worth noting that datasets which have been loaded into the KKG thus far are static, time-independent data such as the Base Register of Addresses and Buildings or known as Basisregistratie adressen en gebouwen in Dutch (BAG). Although the data are periodically updated, they are not real-time data like those coming from sensors.

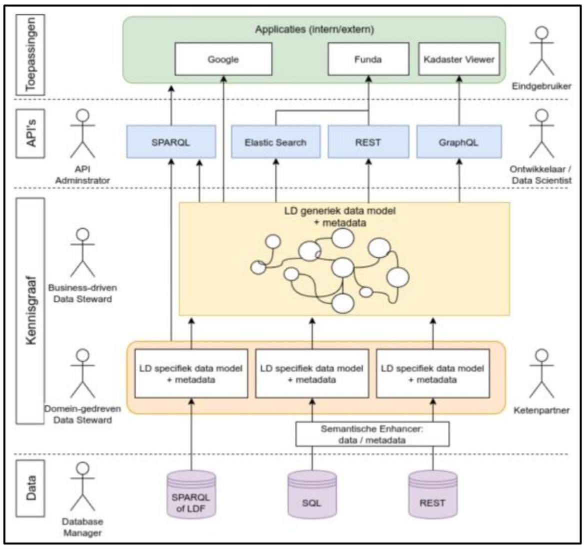

Toepassingen
Applicaties (intern/extern)
Google
Funda
Kadaster Viewer
Eindgebruiker
API's
API Adminstrator
SPARQL
Elastic Search
REST
GraphQL
Ontwikkelaar / Data Scientist
Kennisgraaf
LD generiek data model + metadata
Business-driven Data Steward
LD specifiek data model + metadata
LD specifiek data model + metadata
LD specifiek data model + metadata
Domein-gedreven Data Steward
Ketenpartner
Semantische Enhancer: data / metadata
Data
Database Manager
SPARQL of LDF
SQL
REST

Fig. 2. KKG’s baseline architecture [7].

The data then undergoes an ETL and mapping process managed in Apache Airflow and primarily driven by a so-called *Enhancer* that converts source data into LD format (the orange layer) as shown in Fig. 3. Another goal of this transformation process is to preserve the original model of the source data (i.e., having the ontology for each dataset closely mirror the original data model which is driven by legal definitions) as much as possible so that the data owners can easily recognize the transformed LD format of their data and reduce governance issue. The LD is then stored in a triple store called TriplyDB that exposes the data via APIs including SPARQL, ElasticSearch, Linked Data Fragments (LDF), REST [7].

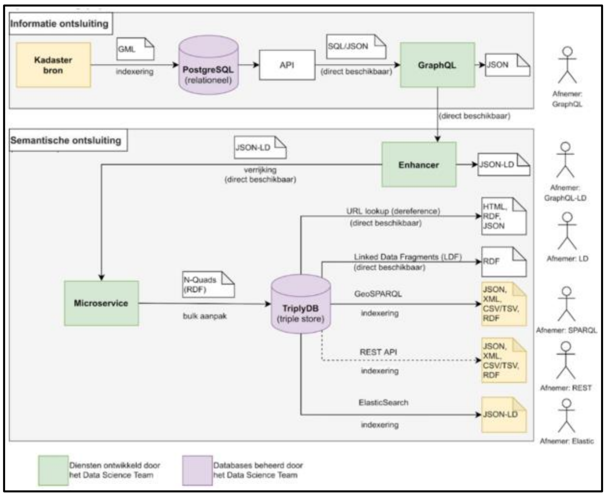


Fig. 3. KKG’s data publication pipeline [7].

So far, we have successfully transformed source data from GML format into LD format. However, we encounter the problem of having LD silos. There are two ways to overcome this problem.

First, we can develop linksets to interconnect different LD datasets. Linksets can be understood as a tool to form point-to-point connections between different data objects in different datasets. As cited in a paper by Ronzhin, two linksets exist as of 2019 [2]. The downside of this approach is that it does not take away the complexity in the data models of the datasets being linked.

Enter the second approach - build a KG that links different RDF datasets as shown in the yellow layer. A KG can be understood as a massive RDF graph using a number of ontologies as its schema (schema.org in the case of the KKG). The Kadaster team is working on linking up more and more datasets for its KG. A list of all datasets included in the KKG can be found in https://data.labs.kadaster.nl/kadaster/knowledge-graph, and a visualization of the KKG’s foundational data model based on schema.org can be found https://kadaster.wvr.io/sdo-target-model/attributes. The primary advantages of modeling the KKG using schema.org are:

- It removes the complexity of different ontologies used in modeling different datasets and, therefore, simplify the queries to be made by end-users
- It allows external data to be linked up with data within the KKG thanks to the wide user base of schema.org

However, schema.org alone is unlikely to be able to support KKG’s foundational data model. For example, schema.org is not suitable for modeling IoT devices (e.g., sensors) which is critical for the integration of real-time data from IoT devices to KKG.

Given Kadaster’s vision of extending the KKG as a knowledge base for a DT of the living environment, more and more data from different domains will be integrated into the KKG. To tackle this challenge, in later sections, we propose a systematic approach of utilizing Ontology Design Patterns (ODPs) and Ontology Quality Methodology to facilitate the development of the KKG’s foundation data model.

B. *Kadaster Knowledge Graph’s Target Architecture*

In this section, we first analyze the key requirements for the target architecture to support the use case and then address these requirements individually.

1) High-Level Requirements

Below is a list of requirements for the KKG’s target architecture:

R1: KKG will serve as the data source for the flood simulation DT – see section III, C “Flood Management DT Tool: Unity” of this paper.

R2: KKG’s foundational data model should allow for flexible representation of different ontologies used for different datasets of different domains (e.g., building, energy consumption, flood monitoring) and of static vs. real-time nature – see section III, B, 2 “Foundation Data Model” of this paper.

R3: According to the baseline architecture of KKG, the existing ETL and mapping procedures are designed to transform data from GML format to RDF format [7]. Depending on the datasets required for the use cases, the ETL and mapping procedure must be able to handle the data types (e.g., relational databases, tabular data) of the datasets of interest and transform the source data to RDF format – see section III, B, 3 “ETL and Mapping Procedures” of this paper.

R4: Strategies for capturing, storing, and querying real-time data must be devised to suit the needs of the specific use cases that require real-time data – see section III, B, 4 “Linked Stream Data and Query Processing Engines” of this paper.

2) Foundational Data Model

In this section, we propose a systematic approach to design the KKG’s foundational data model.

First, Kadaster needs to identify relevant ODPs to develop the KKG’s conceptual model. Once finished, the Kadaster team can use the Ontology Quality Methodology to evaluate different ontology candidates and use the selected ontologies to transform the KKG’s conceptual model into the KKG’s foundational data model.

a) Ontology Design Patterns (ODPs)

In a 2018 paper titled “Ontological Representation of Smart City Data: From Devices to Cities”, the authors proposed 7 ODPs to “materialize the domains and requirements that a smart city should represent" as summarized in Table 2 [8].

ODP	Description
Administrative area	Represents places delineated for jurisdiction purposes of a particular government (e.g., city, district, neighborhood, etc.)
City object	Represents all objects that can be contained by a city (e.g., devices, buildings, transport means, etc.)
Topology	Represents all things that can have a spatial extension (e.g., roads, train stations, commercial premises, etc.)
Event	Describes activities performed in a city during a specified period of time (e.g., concerts, exhibitions, races, etc.)
Key performance indicator	Represents the measured values, according to a method, in order to monitor the performance of a city (e.g., noise pollution level, air quality index, recycling rate, etc.)
Public service	Involves all services provided by public administrations and organizations (e.g., waste management, public parking, water quality control, etc.)
Observations/ measurements	Represents all measured values related to a particular property of any feature of interest (e.g., noise levels, weather conditions, air quality, etc.)

Table 2. Ontology design patterns [8].

Fig. 4 is a graphical representation of the City Object ODD. Details on other ODPs can be found in the cited paper [8].

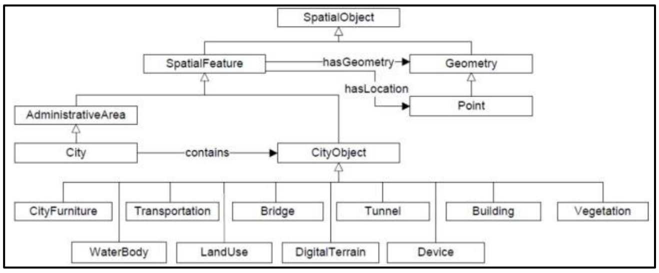


Fig. 4. City Object ODP [8].

To demonstrate the use of ODPs in the context of KKG and the flood simulation DT use case, Fig. 5 is a mockup of the conceptual model for KKG designed with some of the datasets identified for the use case (see section II, C "Data Procurement & Integration Strategies" of this paper) and the administrative area, city object, topology, event, and observations/ measurements ODPs.

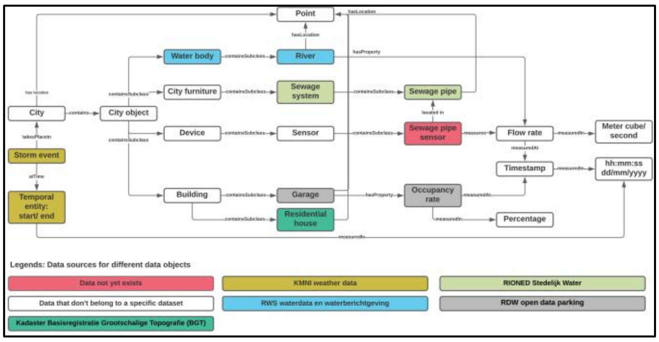


Fig. 5. Example of KKG's conceptual model designed with ODPs for the flood management DT use case.

In Fig. 5. from left to right, the city of Amsterdam, the use case testbed, contains a number of city objects such as water body (e.g., river), city furniture (e.g., sewage system), device (e.g., sewage pipe sensor), and building (e.g., residential house). The conceptual model is designed to capture data objects which can be either real-life objects e.g., a river, or the attributes and relationships between different real-life objects e.g., a river (i.e., an object) has a flow rate (i.e., an attribute) at a point in time and is located in a region (i.e., another object).

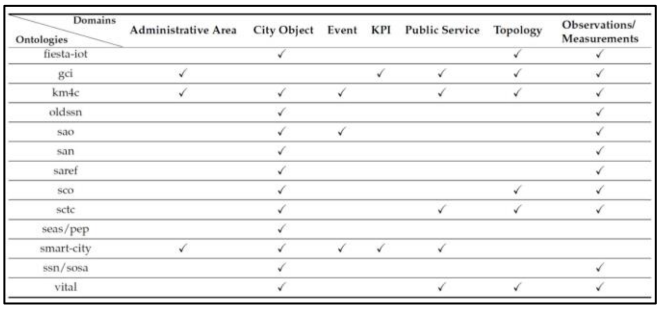
Within the flood management DT, users can use RWS water data to monitor ***river flow rate*** and assess flood risk and damage risk to bridges in different regions. Water can be a cruel force, inflicting serious damage on whatever stands in its way as demonstrated in the tragic event that happened in Schuld, Germany where multiple bridges were destroyed when fatal floods swept through the region in July of 2021 [9].

Additionally, users can monitor the capacity of the sewage system to "load balance" the flow of excess water from busy pipes to slow pipes. Note that real-time ***sewage pipe sensors*** are not yet implemented in the Netherlands but can be easily integrated into KKG once they become available thanks to the extensibility of the conceptual model.

Furthermore, users can monitor ***garage occupancy*** to determine evacuation actions so that the garages can be converted into emergency reservoirs. Last but not least, flood mitigation plans can be devised for specific ***residential areas*** based on the historical data on storm and flood impact.

The above narrative is a simple illustration of how different data sources can be integrated into a KG and serve as a knowledge base of a smart city DT for flood management purposes. The scope of the use case will of course evolve over time, making it critical that KKG uses the appropriate ODPs when building and iterating their conceptual model.

Another useful point of reference is that in the same paper titled "Ontological Representation of Smart City Data: From Devices to Cities", the authors surveyed 13 different smart city-related ontologies in terms of their coverage of ODPs [8]. The more ODPs a smart city ontology covers, the more expressive it is – see Fig. 6.



Domains / Ontologies	Administrative Area	City Object	Event	KPI	Public Service	Topology	Observations/ Measurements
fiesta-iot		✓				✓	✓
gci	✓			✓	✓	✓	✓
km4c	✓	✓	✓		✓	✓	✓
oldssn		✓					✓
sao		✓	✓				✓
san		✓					✓
saref		✓					✓
sco		✓				✓	✓
sctc		✓			✓	✓	✓
seas/pep		✓					
smart-city	✓	✓	✓	✓	✓		
ssn/sosa		✓					✓
vital		✓			✓	✓	✓

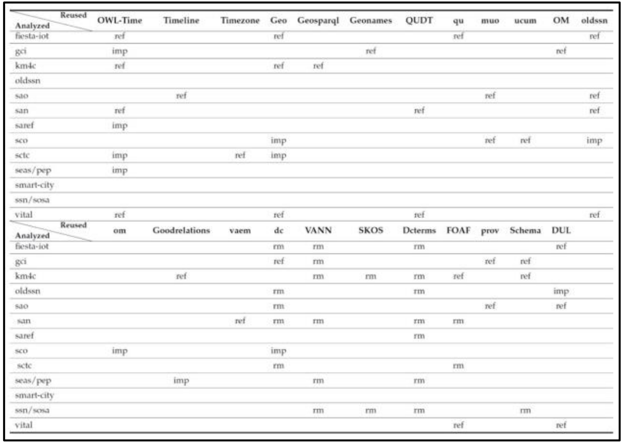
Fig. 6. Survey of 13 smart-city ontologies by their coverage of ODPs [8].

b) Ontology Quality Methodology

Once Kadaster has finished the KKG's conceptual model, the next step is to identify appropriate ontologies to realize the conceptual model.

As an additional source of reference for Kadaster, in the same paper titled "Ontological Representation of

Smart City Data: From Devices to Cities", the authors cataloged the ontologies which were reused/ referenced by the 13 smart city ontologies as shown in Fig. 7.



Reused Analyzed	OWL-Time	Timeline	Timezone	Geo	Geosparql	Geonames	QUDT	qu	muo	ucum	OM	oldssn
fiesta-iot	ref			ref				ref				ref
gci	imp					ref					ref	
km4c	ref			ref	ref							
oldssn												
sao		ref							ref			ref
san	ref						ref					ref
saref	imp											
sco				imp					ref	ref		imp
sctc	imp		ref	imp								
seas/pep	imp											
smart-city												
ssn/sosa												
vital	ref			ref			ref					ref
Reused Analyzed	**om**	**Goodrelations**	**vaem**	**dc**	**VANN**	**SKOS**	**Dcterms**	**FOAF**	**prov**	**Schema**	**DUL**	
fiesta-iot				rm	rm		rm				ref	
gci				ref	rm				ref	ref		
km4c		ref			rm	rm	rm	ref		ref		
oldssn				rm			rm				imp	
sao				rm					ref		ref	
san			ref	rm	rm		rm	rm				
saref							rm					
sco	imp			imp								
sctc				rm				rm				
seas/pep		imp			rm		rm					
smart-city												
ssn/sosa					rm	rm	rm			rm		
vital								ref			ref	

Fig. 7. Ontologies reused (ref = referenced, imp = imported, rm = referenced as metadata) by smart city ontologies [8]

When searching for appropriate ontologies to realize KKG's foundational data model from the conceptual model, Kadaster can use the Ontology Quality Methodology proposed by Gyrard, Zimmermann and Sheth and summarized in Table 3 to evaluate ontology candidates [10].

Criteria	Description	Validation Tools
Serialization	Support the OWL ontology format because it is a W3C recommendation	Apache Jena
Syntactic validation	Necessary during the compilation to load the ontology. It is an important step for the ontology quality methodology since all ontologies must be proceeded by tools	Triple Checker, OWL Manchester
Interlinking	Enhance interoperability, integration, and browsing among ontologies	LogMap
Documentation	Enhance the understandability of the ontology	Parrot, LODE
Visualization	Ease the learning phase by providing a fast understanding of the ontology which encourages the re-usability of the ontology	WebVOWL
Availability	Sharing the ontology code and documentation on the web would encourage ontology reuse	N/A
Discoverability	Improve the dissemination of ontologies in ontology catalogs and semantic search engines	Linked Open Vocabularies (LOV), Vapour
Ontology design	Detect numerous ontology pitfalls	Oops

Table 3. Ontology quality methodology [10].

3) *ETL and Mapping Procedures*

As stated in Requirement 3 of the KKG's target architecture, as the number of data sources to be integrated into the KKG grows, the more challenging it will get to manage the ETL and mapping procedure for these data sources. As a refresher, the ETL and mapping procedures aim at transforming source data into LD and subsequently integrate LD into KKG as explained in section III, A "Kadaster Knowledge Graph's Baseline Architecture" of this paper.

In addition to Kadaster's existing LD publication pipeline shown in Fig. 3, implementing a separate ETL and mapping procedure (i.e., mapping data to different ontologies) for each of the data sources in Apache Airflow will allow Kadaster to review any future changes to the data source (new data format, new data attributes, etc.) and modify the ETL and mapping procedure accordingly.

4) *Linked Stream Data and Query Processing Engines*

For the purpose of flood monitoring and management, it's critical to integrate real-time data to the KKG such as reading from weather sensors. The key distinction between static data and real-time data is that real-time data is time-sensitive. As a result, KKG must be extended to represent stream data, called Linked Stream Data (LSD).

Fig. 8 illustrates how real-time measurements of a temperature sensor and a light sensor in a living room can be linked to the sensors themselves. "The upper layer provides the context for the stream data elements in the lower layer. The appealing nature of the stream data represented in this form is that the processing engine does not need to know the schema of the incoming stream sources upfront to be able to correlate or aggregate them" [11].

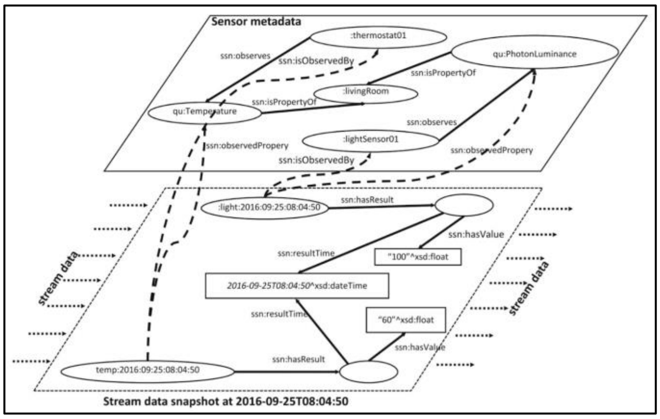


Fig. 8. Example of Linked Stream Data [11].

As real-time data is being integrated into KKG, the foundational data model must be designed to capture the data. Additional research is required in two areas:

First, Kadaster needs to determine how to best capture and store real-time data as the value of real-time data is likely to diminish as time goes on. For example, in order to evaluate the air quality at a public school, the latest readings from air sensors are most relevant and not readings from a few years back. However, archiving real-time data might provide value when the data is used for analytics and prediction purposes.

Second, Kadaster needs to implement an LSD processing engine in addition to the existing SPARQL engine. In order to process LSD, an extended version of SPARQL is needed that allows continuous queries (i.e., the queries are registered once and then continuously evaluated overtime against the changing dataset) over LSD and LD. These extended versions of SPARQL are also known as LSD processing systems. In the paper titled "Linked Stream Data Processing", the authors provide an extensive and comparative analysis of these systems, their design choices and solutions to address the different issues [12]. Kadaster can use this as a reference to determine which LSD processing system is most feasible for them.

C. *Flood Management DT Tool*

Once KKG's target architecture has satisfied requirements 2, 3, 4 outlined in section III, B, 1 "High-Level Requirements" of this paper and relevant data have been integrated into KKG, the next step is to load the data from KKG to a flood management DT tool. One option is to use a 3D modeling tool called Unity (https://unity.com/). Another option is to leverage Amsterdam's existing 3D modeling platform (https://3d.amsterdam.nl/) as Amsterdam is chosen as the testbed for the DT. The exact data loading procedure can be formulated with the chosen tool vendor, Unity or else.

Subsequently, it is critical that relevant stakeholders are involved in the design of the DT so that flood management capabilities within the DT are tailored to end user needs.

According to a 2012 paper titled "Flood Risk and Water Management in the Netherlands: A 2012 Update," key participants in flood management in the Netherlands include [13]:

1. **Royal Netherlands Meteorological Institute (KNMI):** Sends weather forecasts to the RWS and water boards.
2. **Rijkswaterstaat (RWS):** The National Water Authority; manages large lakes and the North Sea, dunes, dams, riverbanks, large navigation locks, and storm surge barriers.
3. **Twenty-five water boards:** The regional water authorities; manage drainage and irrigation channels and minor waterways; own and maintain 360 sewage treatment plants.
4. **Municipalities:** Coordinate with safety regions and water boards in carrying out flood responses at the local level.
5. **Twenty-five safety regions:** Provide emergency services and public health services.

It is likely that the participants mentioned above already have their own flood management procedures and systems in place. However, building a flood management DT using data from KKG provides a unique opportunity to integrate static and real-time data from open and closed sources into a single platform for optimal flood management decision making.

Key features of the flood management DT can fall into three broad categories to accommodate different stakeholders:

1. **Flood susceptibility modeling:** Utilized by RWS, water boards, and municipalities to model different flood scenarios and assess flood risk of different areas, which then guides the development of new flood defenses such as emergency reservoirs or the reinforcement of existing ones.
2. **Real-time flood monitoring and prediction:** Utilized by KNMI, water boards, municipalities, and safety regions to monitor flood in real-time and produce real-time flood prediction.
3. **Flood response planning:** Utilized by municipalities and safety regions to coordinate flood responses in response to real-time flood updates and predictions.

As shown in Table 1 "Data required for flood management DT use case", most of identified data sources are open for public use. However, as IoT devices continue to proliferate, more valuable data will be unlocked for flood management uses; for example, sewage pipe sensor data might be soon available as a Smart City Water System is being tested by i-Sago, 2M Engineering and other partners in the province of Noord-Brabant [14].

As more IoT data become available, a key challenge is data access constraints. For example, sewage pipe sensor data are likely to be the custody of individual municipalities which manage their own sewage systems and potentially remain in closed repositories since they pertain to national critical infrastructure.

To address this challenge, we discuss FAIR data train, a distributed data analytics architecture in section II, C, 1 "Data Procurement: FAIR Data Train for Closed Data."

Another option is to deploy distributed machine learning (ML), a practice of training multiple ML models locally on vertically or horizontally distributed data and subsequently combine the local predictions of the individual models by voting or averaging into a global prediction. Data are horizontally distributed when "subsets of instances are stored at different sites", and data are vertically distributed when "subsets of attributes of instances are stored at different sites" [15].

Depending on whether the newly available, closed data are vertically or horizontally distributed, appropriate distributed learning approaches can be examined and tested with the objective of building robust ***flood susceptibility modeling*** and ***real-time flood monitoring and prediction***, both of which serve as an input for ***flood response planning*** as part of the flood management DT.

Last but not least, as the features of the flood management DT evolve, data might be added to/ removed from the KKG, triggering a cascading effect on KKG's foundational data model, ETL and mapping procedures, and data loading procedure into the flood management DT platform.

IV. Conclusion & Next Steps

This paper is set out to accomplish two goals: 1) define a use case for KKG and 2) propose a number of architectural considerations for the realization of the defined use case.

In section II, we discuss a use case for the KKG: a flood simulation DT that makes use of the KKG as its data source. Additionally, data sources required for the use case and data procurement and integration strategies (including the topics of FAIR data train and INSPIRE data models) are discussed.

In section III, we analyze the baseline architecture of the KKG along with its strengths and weaknesses and then discuss a series of topics that assist the Kadaster team in implementing the use case: high-level requirements of the KKG's target architecture, the KKG's foundation data model, ETL and mapping procedures, and LSD and LSD processing engines. Last but not least, we discuss a potential flood simulation tool that can ingest data from the KKG and perform the actual flood simulations and monitoring.

This paper calls for a number of future works:

- Continue identifying new data sources that are relevant to flood management purposes – see Table 1 "Data required for flood management DT use case".
- Transform data sources into INSPIRE data formats for wider exchange of spatial and geographical data among European countries and enabling cross-national cooperation for flood management purposes.
- Develop use cases for closed datasets with data owners and establish a FAIR data train architecture to enable distributed data analytics on the closed datasets.
- Develop a conceptual model of KKG using ODPs and evaluate ontology candidates using the Ontology Quality Methodology to transform the conceptual model into the foundational data model of KKG.
- Develop and maintain an independent ETL and mapping procedure for each data source.
- Devise a set of strategies on how real-time data can be captured, stored, and queried in KKG using appropriate Linked Stream Data processing engines.
- Engage appropriate stakeholders and develop a minimum viable product of the flood management DT given the data and ML techniques available.
- Select a DT vendor. Develop and maintain a flood management DT.

ACKNOWLEDGMENT

The team sincerely thanks Dr. Erwin Folmer and Lexi Rowland from Kadaster for the opportunity of working on this project and for their continued support.

REFERENCES

[1] Singhal, A. (2012, May 16). *Introducing the Knowledge Graph: things, not strings*. Google. https://blog.google/products/search/introducing-knowledge-graph-things-not/

[2] Ronzhin, S., Folmer, E., Maria, P., Brattinga, M., Beek, W., Lemmens, R., & van't Veer, R. (2019). Kadaster knowledge graph: Beyond the fifth star of open data. Information (Switzerland), 10(10), [310]. https://doi.org/10.3390/info10100310

[3] Grieves, Michael W. and J. Vickers. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems." (2017)

[4] White, G., Zink, A., Codecá, L., & Clarke, S. (2021). A digital twin smart city for citizen feedback. Cities, 110, 103064. https://doi.org/10.1016/j.cities.2020.103064

[5] Oya Beyan, Ananya Choudhury, Johan van Soest, Oliver Kohlbacher, Lukas Zimmermann, Holger Stenzhorn, Md. Rezaul Karim, Michel Dumontier, Stefan Decker, Luiz Olavo Bonino da Silva Santos, Andre Dekker; Distributed Analytics on Sensitive Medical Data: The Personal Health Train. Data Intelligence 2020; 2 (1-2): 96–107. doi: https://doi.org/10.1162/dint_a_00032

[6] INSPIRE. (n.d.). *Data Specifications | INSPIRE*. Inspire Knowledge Base. Retrieved August 2, 2021, from https://inspire.ec.europa.eu/data-specifications/2892

[7] Kadaster Labs. (n.d.). *Aanpak en Architectuur van de (Kadaster) Knowledge Graph*. https://labs.kadaster.nl/cases/architectuur-integraal-bevragen

[8] Espinoza-Arias P, Poveda-Villalón M, García-Castro R, Corcho O. Ontological Representation of Smart City Data: From Devices to Cities. Applied Sciences. 2019; 9(1):32. https://doi.org/10.3390/app9010032

[9] Euronews. (2021, July 16). Homes and bridges destroyed after fatal floods sweep Germany. https://www.euronews.com/2021/07/16/homes-and-bridges-destroyed-after-fatal-floods-sweep-germany

[10] A. Gyrard, A. Zimmermann and A. Sheth, "Building IoT-Based Applications for Smart Cities: How Can Ontology Catalogs Help?," in IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3978-3990, Oct. 2018, doi: 10.1109/JIOT.2018.2854278.

[11] Sakr S., Wylot M., Mutharaju R., Le Phuoc D., Fundulaki I. (2018) Processing of RDF Stream Data. In: Linked Data. Springer, Cham. https://doi.org/10.1007/978-3-319-73515-3_5

[12] Le-Phuoc D., Xavier Parreira J., Hauswirth M. (2012) Linked Stream Data Processing. In: Eiter T., Krennwallner T. (eds) Reasoning Web. Semantic Technologies for Advanced Query Answering. Reasoning Web 2012. Lecture Notes in Computer Science, vol 7487. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33158-9_7

[13] Slomp, Robert. (2012). Flood Risk and Water Management in the Netherlands: A 2012 Update.

[14] Hoofdgebouw KVL | Van een verlaten plek is Leerfabriek KVL nu een levendige en unieke hotspot voor Oisterwijk en omgeving. (2020, December 23). Hoofdgebouwkvl.Nl. https://hoofdgebouwkvl.nl/nieuw-detail/66

[15] Peteiro-Barral, Diego & Guijarro-Berdiñas, Bertha. (2012). A survey of methods for distributed machine learning. Progress in Artificial Intelligence. 2. 10.1007/s13748-012-0035-5.