A Hypercat–enabled Semantic Internet of Things Data Hub

Ilias Tachmazidis¹, Sotiris Batsakis¹, John Davies², Alistair Duke², Mauro Vallati¹, Grigoris Antoniou¹, and Sandra Stincic Clarke²

¹University of Huddersfield, UK
²British Telecommunications, Ipswich, UK
Outline

- Motivation
- BT Hypercat Data Hub
- BT Hypercat Ontology
- Data Translation
- BT SPARQL Endpoint
- Federated Querying
- Use Cases
- Future Directions
Smart city is based on the use of technology in order to improve:

- efficiency
- effectiveness
- city services
- quality of the inhabitants' lives

Successful smart city solutions require the collection and maintenance of relevant sensor data (IoT data)
Eight industry-led projects were funded by Innovate UK to deliver IoT 'clusters'
The **BT Hypercat Data Hub** was part of the Internet of Things Ecosystem Demonstrator programme
Hypercat was developed, which is a standard for representing and exposing Internet of Things data hub catalogues
IoT / smart city projects include:

- MK:Smart (http://www.mksmart.org), based on the BT Data Hub that is Hypercat-enabled but not semantically enriched
**Motivation (4/4)**

- Smart city solutions need to combine data from various sources and be interoperable
  - Adding semantics to data is the best way of ensuring interoperability and improving data quality through enrichment
- The aim of this work was to add semantics to the BT Data Hub
BT Hypercat Data Hub

- BT SPARQL Endpoint
- SPARQL to SQL
- RESTful API (XML/JSON)
- Hypercat Catalogue (JSON)
- Relational DB (sensors)
- Relational DB (events)
- Relational DB (locations)
- LOD Cloud
- Ontology

Part 2: BT Hypercat Data Hub
BT Hypercat Ontology (1/4)
BT Hypercat Ontology (2/4)

- Hypercat Core URI:

- BT Hypercat URI:
Feed: a source of sensor readings, events or locations

Subclasses of Feed:
- SensorFeed: a source of sensor readings
- EventFeed: a source of events
- LocationFeed: a source of locations
A Feed contains classes of type of Datastream or Location

**DataStream**: a stream of sensor readings or events

Subclasses of **DataStream**:
- **SensorStream**: a stream of sensor readings
- **EventStream**: a stream of events
RDF triple is provided within a single line, in N-Triples format, namely:
\[
\langle \text{subject} \rangle \ \langle \text{predicate} \rangle \ \langle \text{object} \rangle .
\]

The URI of a SensorFeed is generated as:
\[
\langle \text{http://api.bt-hypercat.com/sensors/feeds/feedID} \rangle
\]
An RDF triple providing the type of a SensorFeed:

<http://api.bt-hypercat.com/sensors/feeds/feedID>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://portal.bt-hypercat.com/ontologies/bt-hypercat#SensorFeed>
Dynamic translation of SPARQL queries into SQL, using Ontop

Implicit information is extracted from the ontology through reasoning (RDFS, OWL 2 QL)

Semantically richer information is extracted compared to the knowledge that is stored in the relational database

URI prefixes:
- bt-sensors: http://api.bt-hypercat.com/sensors/
- bt-hypercat: http://portal.bt-hypercat.com/ontologies/bt-hypercat#
Data Translation – SPARQL to SQL (2/2)

- **Mapping ID:** a unique id for a given mapping
- **Target (Triple Template):** RDF triple pattern to be generated (SQL variables are given in braces, such as {feed.id})
- **Source (SQL Query):** SQL query to be submitted to the relational database

<table>
<thead>
<tr>
<th>Mapping ID</th>
<th>mapping:SensorFeed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target (Triple Template)</strong></td>
<td>bt-sensors:feeds/{feed.id} a bt-hypercat:SensorFeed</td>
</tr>
<tr>
<td><strong>Source (SQL Query)</strong></td>
<td>SELECT feed.id FROM feed</td>
</tr>
</tbody>
</table>
BT SPARQL Endpoint (1/5)

SPARQL to SPARQL (Jena)

SPARQL to SQL (Ontop)

Relational DB (sensors)

SPARQL to SQL (Ontop)

Relational DB (events)
SPARQL to SQL: supports the translation of SPARQL queries to relational databases, using Ontop

SPARQL to SPARQL: queries internally all available SPARQL to SQL endpoints, using Apache Jena

End users submit queries to the SPARQL to SPARQL endpoint
BT SPARQL Query Editor

Query Text

```
SELECT * WHERE { ?s ?p ?o . } LIMIT 50
```

Results Format:

- HTML
- XML
- JSON
- CSV
- TSV

Run query
BT SPARQL Endpoint (4/5)

- BT SPARQL Query Editor: User Interface that allows graphic submission of queries
- Results are returned in five formats: HTML, XML, JSON, CSV and TSV
The following query retrieves classes of type `Datastream` (and its subclasses, namely `SensorStream` and `EventStream`):

```sparql
PREFIX hypercat: <http://portal.bt-hypercat.com/ontologies/bt-hypercat#>
SELECT DISTINCT ?s
WHERE{ ?s a hypercat:Datastream . }
```
Federated SPARQL supports federated queries that combine the BT SPARQL Endpoint with the LOD cloud, using Apache Jena.
External SPARQL endpoints that are part of the LOD and could be combined with the BT SPARQL Endpoint:

- DBPedia
- FactForge
- OpenUpLabs
- European Environment Agency
Use Cases – The SimplifAI Project (1/2)

- Aimed at urban traffic management and control
- Targeting a higher level of data integration, while capturing and exploiting real-time and historical urban data sources
- Traffic management utilised the semantically enriched data
- Strategies in real-time were enabled to solve challenges caused by exceptional or unexpected conditions
Raw data was taken from a large number of transport and environment sources, and integrated into the BT Hypercat Data Hub.

Automated Planning was able to alleviate traffic congestion caused by exceptional circumstances.

Simulations showed that on average, the area is de-congested 20% faster, and tail-pipe emissions are reduced by 2.5%.
Use Cases – City Concierge (1/3)

- CityVerve is a Manchester, UK based IoT Demonstrator project
- City Concierge is one of CityVerve’s use cases
- City Concierge aims to increase uptake of walking and cycling as a preferred travel mode in Greater Manchester
- City Concierge aims to develop a city user interface for the city region, integrating transportation and visitor services
The scope of the use case includes:

- improvements in the way people navigate around the city
- a digital solution in conjunction with physical wayfinding assets

The BT Hypercat Data Hub provides the required infrastructure and functionality in order to enable the City Concierge
Use Cases – City Concierge (3/3)
Future Work

- Further semantic enrichment of the implemented system
- Support for GeoSPARQL queries
- Spatiotemporal reasoning over stored data
Thank You!