# Towards Automatic Generation of Digital Twins: Graph-based Integration of Smart City Datasets

Sebastian Ernst[0000−0001−8983−480X], Leszek Kotulski[0000−0002−0164−0048], and Igor Wojnicki[0000−0002−4406−4992]

AGH University of Science and Technology,
Department of Applied Computer Science,
Al. Mickiewicza 30, Kraków, Poland
{ernst,kotulski,wojnicki}@agh.edu.pl

**Abstract.** This paper presents a graph-based approach to modelling and analysis of spatial (GIS) datasets supporting the deployment of *smart city* solutions. The presented approach is based on the *spatially-triggered graph transfromations* (STGT) methodology, which allows for *materialisation* of spatial relationships detected using suitable tools, as well as performing measurements and modifications of geometries. The theory is illustrated using a real-world example which concerns street lighting. It shows how an existing traffic sensor network can be used to enable dynamic dimming of lamps, which can result in significant energy usage savings. Also, network analysis is applied to broaden the coverage of such systems, even in case of sensor sparsity. The presented results have been obtained in a real-world project and are due for larger-scale validation in the near future.

**Keywords:** graph transformations · GIS · smart cities · energy saving

## 1 Introduction

Reduction of energy usage is an important goal for governments and has implications in numerous areas, including economy, sustainability, protection of the environment and assuring the safety of citizens. It has also been one of the underlying goals of the *smart city* concept.

While there are many definitions of a *smart city* [1,8], practically all of them include the use of ICT solutions based on data collection and analysis. Hence, many projects supporting the development of smart cities involved deployment of sensor networks (or integration of data registered by existing devices). Data availability is an enabler for the data to be utilised in day-to-day operations. For instance, real-time vehicle traffic intensity data can be used to dim the streetlights [7,12], as the EN 13201 standard used throughout Europe and in other parts of of the World allows for dynamic changes of the road lighting class [4].

However, a big problem lies in the *integration* of individual datasets. Defining the relationships between lamps, the areas each of them illuminates, how

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

these areas constitute a road segment and which road segments each traffic sensor applies to is a time-consuming task. Another problem arises when not all streets are covered by sensors, which may reduce the applicability of dimming and, as a result, diminish the positive impact of dimming. These issues are further elaborated in Section 2. Such challenges are addressed during establishing so-called Digital Twins [5] which are actual data-based representations of physical infrastructural objects, policies, as well as sociological or technology bound behaviors.

This paper proposes a twofold solution to such problems. First, a graph structure, the Digital Twin Graph, able to store spatial data originating from various sources is introduced. Then, graph transformations, including those triggered by spatial relations (see STGT in Section -sec. 3.1), are used to integrate the objects from separate datasets as well as to derive added value by interpreting the data in each dataset.

Building upon the example of lamp dimming based on traffic intensity, as a result of this phase, the sensors are associated with the road segments they monitor. Later, the graph model is further analysed to estimate the coverage of the existing sensors in order and expand the applicability of dynamic lighting, thus aiding the city officials in the decision-making process. This shows that the proposed approach can be used to both enable day-to-day operations and support long-term evolution of smart cities.

## 2 Motivation

As mentioned in Section 1, collection of smart city data is the first step, but actual benefits come from proper utilisation of such data. Following up on the example of dynamically-dimmed street lighting, savings can be generated by just providing appropriate control logic for existing devices. These savings are also significant, as street lighting is one of the main contributors of cities' energy bills [9]; as shown in [11], implementing such dynamic control can reduce power consumption by 34%.

A common problem with dataset integration is that often they are logically linked only by the spatial (GIS, Geographic Information System) component. For instance, having a dataset modelling streetlight locations (as points) and one modelling streets (as lines or polygons), there usually won't be an attribute relationship between them. Therefore, to determine that a lamp illuminates a given street, one must resort to analysing spatial relationships, such as the distance. This process is not straightforward, as we've shown e.g. in [3]: while it was possible to perform large-scale (Washington D.C.) lamp-to-street assignment programatically, the process involved a lot of fine-tuning and was error-prone.

During a pilot project for intelligent street lighting, jointly executed by the city of Kraków, Poland, and AGH University of Science and Technology, there were two challenges related to traffic intensity sensors:

1. the sensor locations were provided as points, without any relationship with streets,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

2. the sensor network was rather sparse, which limited the applicability of street lighting.

The first problem was solved by simply performing a manual assignment of sensors to the street segments they monitor. The second issue required more work and resulted in development of the concept of *virtual sensors*, based on the Dual Graph Grammar formalism [13]. However, the formulae used by the virtual sensors to compute the estimated traffic intensity based on the neighbouring real sensors still had to be developed by hand.

This paper presents a graph model, along with the necessary formalisms and operations, to resolve both of the aforementioned challenges. It uses publicly-available OpenStreetMap data as the basis for integration of other datasets and proposes an extensible graph model to store all of its data. It allows for automatic assignment of camera-based sensors to street segments, and by inferring the possible traffic flows, it can also estimate the coverage of neighbouring streets automatically.

## 3 Formal background

We assume that the digital twin of city will have a multi-layer graph structure. Its flat representation will be generated with the help of methods described in[6,10]. Every layer will be generated by another graph grammar, with transformations fired by a dedicated expert system considering both spatial and semantic relations.

### 3.1 Spatially-Triggered Graph Transformations

The Spatially-Triggered Graph Transformations (STGT) mechanism was first introduced in [2] and involves the usage of specialised tools to detect spatial relationships among objects in different datasets. This allows for *materialisation* of such relationships in the graph, which, in turn, enables their usage for further analysis.

In essence, the methodology uses an external tool, called the *expert system* ($\Xi$), to:

- detect spatial relationships (such as vicinity, intersection, parallelism) between geographic objects modelled by graph nodes,
- perform spatial measurements and/or transformations of existing geometries, e.g. to supplement the detected relationships with attributes (such as distance, angle, etc.) or to generate new geometries based on existing ones.

In practice, to implement the concept of STGT, we use a programmatic procedure based on spatial (GIS) analysis tools, such as a spatial database (PostGIS[1],

[1] `https://postgis.net`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

Spatialite[2]) or a spatial analysis library, such as GeoPandas[3]. The latter choice seems preferable in this case, as it is well-suited to ad-hoc analysis.

A practical example of such application can be found in Section 4.3, and the formal notations of the STGT concept and its prerequisites are presented below.

### 3.2 Notations

The Digital Twin Graph will represent the entire structure of the city data and the semantic relations among their elements, and formally it is a eight-tuple.

**Definition 1 (Digital Twin Graph).** *The DTG is an attributed graph over the set of node labels $\Sigma$ and the set of edge labels $\Gamma$, such that:*

$$DTG = (V, E, \Delta, lab_{X\ (X=V,E)}, att_{X\ (X=V,E)}, \Pi)$$

*where:*

- $V$ *is a finite, nonempty set of graph nodes identified unambiguously by some injective indexing function* $\mathrm{Index} : V \to \mathbb{N}$,
- $E \subseteq V \times V$ *is a set of edges,*
- $\Delta \subseteq V$ *is a set of nonterminal nodes.*
- $lab_V : V \to \Sigma$ *is a node labelling function,*
- $lab_E : E \to \Gamma$ *is a edge labelling function,*
- $att_X : X \to 2^{Att\_name_X \times Att\_val_X}$ $(X = V, E)$*, is a node/edge attributing function, such that for* $a \in X$ *return a set of pairs* $(an, av)$ *where* $an \in Att\_name_X$ *is an attribute name, and* $av \in Att\_val_X$ *is an attribute value.*

Comments: the set of available attributes in the attributing depends on the label of the node or edge; however, formally, the notation which is a subset of all possible name and values combinations is sufficient.

The transformations are formally defined as a part of the graph grammars notation presented below.

**Definition 2 (Graph grammar).** *A* graph grammar $\Omega$ *is a tuple:*

$$\Omega = (\Sigma, \Gamma, \Delta, \Phi, S, \Pi)$$

*where:*

- $\Sigma$ *is the set of node labels,*
- $\Delta \subset \Sigma$*, is the set of terminal node labels,*
- $\Gamma$ *is the set of edge labels,*
- $\Phi$ *is the set of transformation rules (see def:TranGra),*
- $S$ *is the starting graph,*
- $\Pi$ *is the graph grammar validation condition, that verifies the current state of the graph.*

[2] `https://www.gaia-gis.it/fossil/libspatialite/index`
[3] `https://geopandas.org/en/stable/`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

We use $\Pi$ as the validation condition of the graph grammar because sometimes it is necessary to execute a sequence of transformations and some of the intermediate states may not form a correct graph. The validation condition $\Pi$ usually is not expressed in the graph's definition, but in theory we always assume that there are no non-terminals in the final graph, so we have at least one condition:

$$\forall v \in V : lab_V(v) \in \Delta \Rightarrow \ G \ is \ a \ final \ graph$$

The non-terminal nodes are used only in temporary states where a sequence of transformations is used to move the graph from one final state to another one.

At each level, the transformations will work on a subset of $V$ and $E$, i.e. it is true that:

- for each transformation $\Omega$ $V_\Omega \subseteq V$
- for any node $x \in V - S$ exist the transformations $\Upsilon$ such that $x \in V_\Upsilon$.

**Definition 3 (Graph Transformation).** *A five tuple*

$$(L, R, roots, eval_V, eval_E) \in \Phi_\Omega$$

*is a transformation rule when:*

- *$L$, $R$ are the graphs called the left- and right-side of a transformation, respectively; each of them can be a set of weakly-connected graphs.*
- *For a given left-hand-side graph $L$ consisting of $k$ connected components, $cont$ is a set of indices, $\{i_1, i_2, \ldots, i_k\}$, such that $i_j$ is a least index (referred to as a root) in a $j$-th connected component. The subset $roots \subseteq V_L$ contains vertices with indices belonging to $cont$.*
- *$eval_X : X_R \to 2^{Att\_name_X \times Att\_val_X}$ $(X = V, E)$ is the node/edge attribute evaluation function, respectively (see $att_X$ functions in Definition [def:dsg])*
- *the graph transformation mechanism in based on the single push-out mechanism:*
  1. *the morphism $h : L \to H$ is designated (basing on $cont$ function),*
  2. *nodes of the graph $h(L - R)$ are removed from $H$,*
  3. *edges with removed nodes are removed too,*
  4. *graph $R$ is added to $H$ (nodes and edges belonging to $L \cap R$ joins the $R$ graph with $H - (L - R)$, and for them the transformations only change their attributes values)*

The $cont, eval_V$ and $eval_E$ functions are the formal representation of an interface between the graph transformations and the expert systems, representing domain-specific knowledge.

Let $\Xi$ be an expert system which both sees the current digital twin graph and represents some knowledge (e.g., it is able to point out that a luminary illuminates a given street based on their geometries), so it launches some transaction that enriches the digital twin information. Formally, it will be defined as follows.

This concept is the formal background of the spatially-triggered graph transformations (STGT) methodology.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

**Definition 4 (Transformation firing).** *For any graph $H$ and any transformation $\omega = (L, R, match, eval_V, eval_E) \in \Phi_\Omega$ the expert system fires the transaction when:*

1. *$\Xi$ designates the function $match : roots \rightarrow H$,*[4]
2. *$\Xi$ designates $eval_V, eval_E$ functions, i.e., defines the values of attributes of the graph $R$,*
3. *the transformation $\omega$ is applied to the $H$ graph.*

Let us note that the left side of the production is not a single graph $L$, but a sequence of graphs, such that each vertex with the lowest index is mapped to a corresponding vertex in the input (transformed) graph.

## 4 Solution outline

To illustrate the proposed solution using a practical example, we will guide the reader through the individual steps of graph generation and analysis. Each step of the presented method will be presented in relation to the formal base presented in sec. 3.

The structure of this section is as follows. First, in section 4.1, a graph representing 100% of the data contained in the OpenStreetMap repositories for the area of interest is generated.

The data originating from OSM forms the background (the "common denominator") for integration of datasets related directly to the *smart city* solution being implemented. In this case, the dataset contains the parameters of camera-based traffic intensity sensors. Section 4.2 describes the step of adding nodes representing these objects to the graph.

However, at this stage, the sensor nodes are not yet semantically assigned to the real-world objects they monitor, i.e. to the streets being observed by each camera. Section 4.3 discusses the application spatially-triggered graph transformations to accomplish this. A GIS analysis tool (formally, the "expert system" as described in sec. 3) is used to first estimate the area "seen" by each camera, and then to detect the streets which intersect these areas. These detected relationships are persisted as graph edges; this *materialisation* of phenomena detected in the data is one of the key characteristics of STGT.

Finally, as the sensor network is relatively sparse, it seems beneficial to *propagate* this information in order to estimate the traffic levels on unmonitored streets. Section 4.4 shows how the network structure of OpenStreetMap data can be utilised to detect areas in which the certainty of estimate traffic intensity is the lowest, making them ideal candidates for furhter extension of the sensor network.

[4] Theoretically, such a designation can be made by the graph transformation mechanism (defined in the point 3), but in practice, it leads us to the NP-complete graph isomorphism problem.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

### 4.1 OpenStreetMap graph generation

The first stage involves building a graph, containing 100% of data included in the source OpenStreetMap data.

The OpenStreetMap data model is simple, as it contains only three types of entities:

- *nodes*, which are always points with a latitude/longitude geographic location,
- *ways*, which are ordered sequences of points,
- *relations*, which are used to define relationships between other objects (of any of the three types).

Each object has a unique identifier (integer). The semantics of each object are defined by a set of key-value pairs, called *tags*. The taxonomy of keys and values is defined in OSM documentation[5].

We assume that the OSM data is available through an Overpass API[6], or in a PostgreSQL/PostGIS database created using a tool such as Osmosis[7] or Osmium[8].

The resulting graph will typically be managed by a graph database, such as Neo4J[9] or Apache AGE[10]. These solutions use the Cypher[11] language to model and query the data, therefore some examples will present the Cypher code used for experiments.

**Nodes** For each OSM node, a graph node with the label `Node` will be created. All node data is stored as graph attributes:

- the OSM identifier, as the `osm_id` attribute,
- the geometry (geographic location of the point), stored as a Well-Known Text (WKT)[12] value in the `geometry` attribute,
- all other tags and values as attributes.

The Cypher code representing an example node is provided in lst. 1, and its visual representation is in fig. 1.

**Ways** Ways are ordered sequences of nodes. Hence, for each way, a graph node with the label `Way` will be created. The attributes are modelled in the same way as for nodes (see sec. 4.1), with the distinction that the geometry is usually a WKT linestring.

[5] `https://wiki.openstreetmap.org/wiki/Map_features`
[6] `http://overpass-api.de`
[7] `https://wiki.openstreetmap.org/wiki/Osmosis`
[8] `https://wiki.openstreetmap.org/wiki/Osmium`
[9] `https://neo4j.com`
[10] `https://age.apache.org`
[11] `https://neo4j.com/developer/cypher/`
[12] `https://libgeos.org/specifications/wkt/`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

```
CREATE (n:Node {
  osm_id: 6958500807, tourism: 'museum',
  name: 'Muzeum AGH', wikidata: 'Q11786993'
});
```

Listing 1: Cypher representation of an OSM node.

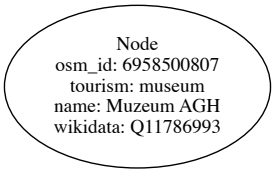


**Fig. 1.** Graph representation of an OpenStreetMap node.

In the graph, all nodes which are members of a way will have `MEMBER_OF` relationships towards the way, and the order is indicated by the `order` property.

Cypher code representing an example node is provided in lst. 2, and its visual representation is in fig. 2.

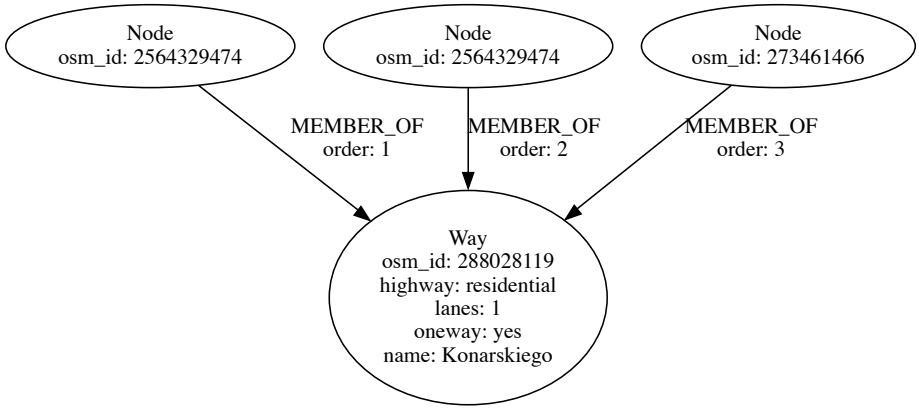


**Fig. 2.** Graph representation of an OpenStreetMap way; node attributes ommitted for clarity.

**Relations** OpenStreetMap *relations* model relationships among other OSM objects. They are ordered lists of nodes, ways and/or other relations. Relation members may have *roles*, such as:

- `outer`, signifying that a *way* is part of an outer boundary of an area,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

```
MERGE (w1:Way {
  osm_id: 288028119, highway: "residential", lanes: 1,
  oneway: "yes", name: "Konarskiego"})
MERGE (n1:Node {osm_id: 2564329474})
MERGE (n2:Node {osm_id: 2720618781})
MERGE (n3:Node {osm_id: 273461466})
MERGE (n1)-[:MEMBER_OF {order: 1}]->(w1)
MERGE (n2)-[:MEMBER_OF {order: 2}]->(w1)
MERGE (n3)-[:MEMBER_OF {order: 3}]->(w1);
```

Listing 2: Cypher representation of an OSM way.

- `subarea`, signifying that an area is part of another, larger area (e.g. a city district is part of the city itself),
- `admin_centre`, signifying that a given point (*node*) is designated as the administrative centre of a city.

Relations can be used to model any relationships, from administrative areas to bus lines. The relation's tags determine its semantics.

We model relations similarly to ways, i.e. the identifier is stored as the `osm_id` attribute, and all tags as other attributes.

Each relation member will have a `MEMBER_OF` relationship towards the relation, with its order and role indicated by the `order` and `role` properties, respectively.

Due to space limitations and the structural similarity of the graph representation of relations and ways, an example is not provided.

**Fig. 3.** Locations of 6 camera-based traffic sensors in Siechnice; 3 of these are located in one intersection in the city centre.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

### 4.2 Sensor modelling

To discuss the issue of sensor modelling, let us consider an example of Siechnice. It is is a town in in south-western Poland with a population of 9,302 (as of 2022). The local authorities are keen on innovation, and in the years 2019–2022, Siechnice introduced several intelligent solutions in cooperation with AGH University, within the Human Smart Cities project.

There are 6 cameras with vehicle-counting functionality deployed in the city. Their locations have been shown in Figure 3. Each camera is described by its geographic location, its viewing angle (degrees) and its azimuth (degrees).

The cameras are modelled as nodes with label *Sensor*, add attributes as shown in 4. Please note that at the moment, sensors

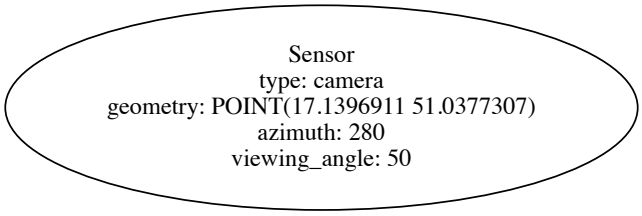


**Fig. 4.** Graph representation of a camera-based sensor.

### 4.3 Sensor assignment

In the presented example, each camera is configurable with regard to detection zones, which can be used to measure the traffic intensity in a particular lane, as long as that lane is within the field of view of the camera.

Please note that, at this stage, both the cameras and the streets are parts of the graph, as *Sensor* and *Way* objects, respectively. However, there are no relationships defined between them yet.

To assign the sensors to the street fragments they are able to monitor, we use the STGT methodology, formally presented in Section 3.1.

The proposed procedure consists of two stages:

1. estimating the field of view of each camera,
2. determining the streets within that field of view.

The first stage involves querying the graph for all cameras and, for each one, constructing shapes (geometries) which model the field of view. These are added to the graph as separate nodes. To estimate the area "seen" by each camera, we will first use a procedure to generate an isosceles triangle, where:

– the *apex* is at the location of the camera,

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

– the length of the *arms* represents a reasonable viewing range of the camera (e.g. 40 meters),
– the angle of the *axis* is the azimuth of the camera,
– the angle between the *arms* equals the viewing angle.

Therefore, at the first stage, the graph, containing the *Sensor* nodes, is supplemented with *FieldOfView* nodes, using the transformation presented in 5. The new node has a `geometry` attribute, with a polygon feature modelling the field of view. As indicated in sec. 3.1, this would be calculated using an external tool, such as GeoPandas, or a spatial database (e.g. PostGIS).

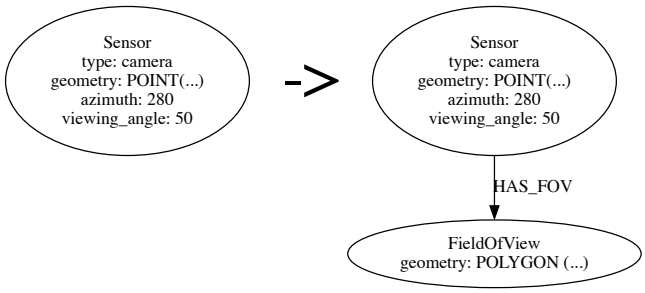


**Fig. 5.** Graph transformation adding the field of view to a camera-based sensor.

The external tool is used to find the pairs of *FieldOfView* and *Way* nodes with intersecting geometries. In this step, no new nodes are added, but a *COVERS* relationship is added between the two "intersecting" nodes, as shown in fig. 7. Therefore, the spatial relationships *detected* in the data are now *materialized*, making them suitable for further analysis.

In the last stage, a *SEES* relationship between the *Sensor* node and the *Way* is added, using only the relationships detected and materialised in the previous step, as shown in fig. 8. Therefore, the relationship between the sensor and the road is materialised, which means that the fact that a given camera is able to measure traffic on a given road.

It should also be noted that this approach makes the process well-documented, replicable and reversible; these characteristics alone constitute added value of the graph-based methodology compared to approaches based solely on GIS tools or spatial databases. However, further benefits are presented in sec. 4.4 below.

### 4.4 Traffic flow modelling

At this stage, we have assigned the camera sensors to the roads they are able to monitor in a direct manner. However, given the usual sparsity of the sensor system, the traffic intensity in most streets fragments is not monitored, and therefore we are unable include them e.g. in the dynamic streetlight dimming system.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

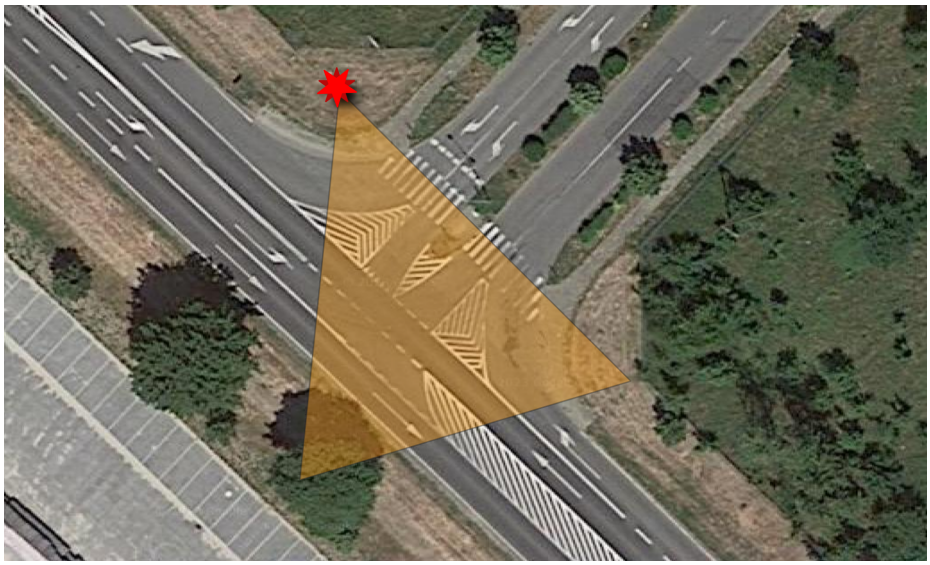

**Fig. 6.** Visualisation of the created field of view on a map.

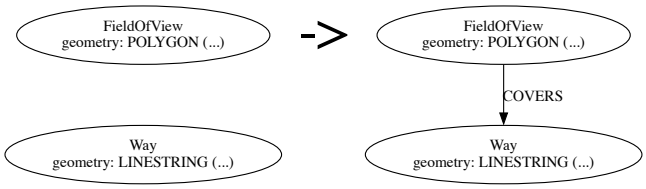


**Fig. 7.** Materialiasation of the *COVERS* relationship between the camera's field of view and the geometries.

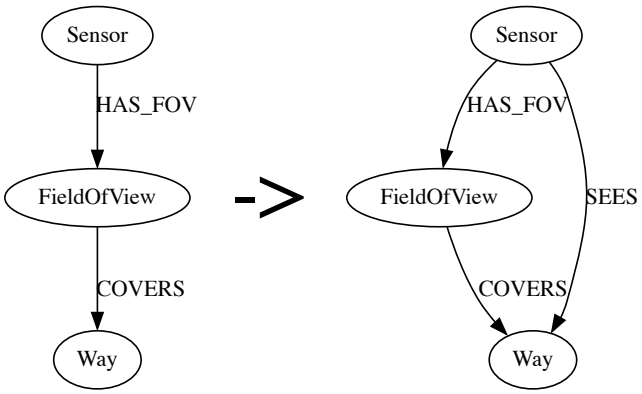


**Fig. 8.** Materialiasation of the *SEES* relationship between the camera-based sensor and the road.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

**Fig. 9.** Results of estimation certainty propagation in the city of Siechnice.

However, the graph-based structure makes it appropriate for network analysis, which may result both in no-cost expansion of the dimming system's coverage and in determination of the candidates for expansion.

Since the steps of this procedure use traditional graph transformation, and due to space limitations, only a brief, textual description is provided:

1. New *Flow* nodes, representing possible traffic flows, are introduced into the graph – two for each two-way street, and one for each one-way street; each *Flow* node has a *ON_STREET* edge towards "its" street (*Way* node), with the `direction` attribute indicating the flow direction (forward or reverse).
2. Possible street-to-street flows are modelled by introducing *CONTINUES_INTO* edges between *Flow* nodes; e.g., for an intersection where a vehicle can go straight on, turn left or turn right, there will be 3 edges towards the respective destination flows.
3. A `certainty` attribute with the value of 1 is added to *Flow* nodes belonging to *Way* nodes with *SEES* relationships from *Sensor* nodes.
4. A *certainty propagation* network algorithm is executed, where the values of the `certainty` attribute are propagated to continuation *Flow* nodes, where the value is divided by the number of possible continuations; e.g., for the example presented in point 2 above, if the source *Flow* node's `certainty` equals 1, the three destination *Flow* nodes will have a `certainty` of $\frac{1}{3}$. This procedure is repeated until there are no modifications of certainty of any *Flow*.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

As a result, it is possible to estimate the certainty with which it is possible to *estimate* the traffic intensity in each street. Example results for the city of Siechnice are presented in fig. 9, where the colour indicates the certainty value. This allows us to distinguish:

- streets with reasonably high traffic estimation certainty – the ones which can be equipped with *virtual sensors* and included in the dynamic dimming system,
- streets with low certainty – which should be the natural candidates for deployment of new sensors.

## 5 Conclusions and future work

The graph formalism is suitable for modelling smart city concepts by gathering knowledge in a robust way which results in a digital twin. Labelled and attributes nodes and edges are suitable for expressing semantic, qualitative and quantitative features. On top of that graph transformations enable simulation, inference and general interpretation of such knowledge providing actual incentives, such as energy savings, increase of comfort, smart control, or development planing.

However there is a procedural gap between actual reality to be modelled and its representation being the digital twin. It is data integration. Establishing a proper model is often based on semantic fuzziness of natural language such as: *near by*, *looks at*, *crosses over*, *leads to* etc., that binds geographical locations of the modelled physical features. It poses a challenge if such a model is to be established automatically from large heterogeneous datasets.

The gap can be sealed with the proposed STGT methodology. It enables graph transformations to be triggered by spatial locations. Thanks to that questions like: *what street does the camera overlook*, *can traffic flow from street A to street B at this intersection*, *is this sensor redundant* do not have to be answered by human experts but instead the answers can be inferred at large. It results in an ability to integrate heterogeneous data in a single, coherent, graph-based model.

The proposed approach has been proven within the framework of the pilot projects in Kraków and Siechnice, Poland. It is to be tested at scale as a main tool to assess modernization capacity of large urban areas in 2023.

## References

1. Deakin, M., Al Waer, H.: From intelligent to smart cities. Intelligent Buildings International **3**(3), 133–139 (Jul 2011). https://doi.org/10.1080/17508975.2011.586673
2. Ernst, S., Kotulski, L.: Estimation of Road Lighting Power Efficiency Using Graph-Controlled Spatial Data Interpretation. In: Paszynski, M., Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A. (eds.) Computational Science – ICCS 2021. pp. 585–598. Lecture Notes in Computer Science, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-77961-0_47

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31

3. Ernst, S., Starczewski, J.: How Spatial Data Analysis Can Make Smart Lighting Smarter. In: Nguyen, N.T., Chittayasothorn, S., Niyato, D., Trawiński, B. (eds.) Intelligent Information and Database Systems. pp. 272–285. Lecture Notes in Computer Science, Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-73280-6_22
4. European Committee for Standarization: CEN/TR 13201-1: Road lighting – Part 1: Guidelines on selection of lighting classes. Tech. rep., European Committee for Standarization (Dec 2014)
5. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology **29**, 36–52 (May 2020). https://doi.org/10.1016/j.cirpj.2020.02.002, https://www.sciencedirect.com/science/article/pii/S1755581720300110
6. Kotulski, L., Szpyrka, M.: Graph representation of hierarchical Alvis model structure. In: Proc. of the 2011 International Conference on Foundations of Computer Science FCS'11 (Part of Worldcomp 2011). pp. 95–101. Las Vegas, Nevada, USA (Jul 2011)
7. Mahoor, M., Salmasi, F.R., Najafabadi, T.A.: A Hierarchical Smart Street Lighting System With Brute-Force Energy Optimization. IEEE Sensors Journal **17**(9), 2871–2879 (May 2017). https://doi.org/10.1109/JSEN.2017.2684240
8. Paiho, S., Tuominen, P., Rökman, J., Ylikerälä, M., Pajula, J., Siikavirta, H.: Opportunities of collected city data for smart cities. IET Smart Cities **4**(4), 275–291 (2022). https://doi.org/10.1049/smc2.12044
9. Pardo-Bosch, F., Blanco, A., Sesé, E., Ezcurra, F., Pujadas, P.: Sustainable strategy for the implementation of energy efficient smart public lighting in urban areas: Case study in San Sebastian. Sustainable Cities and Society **76**, 103454 (Jan 2022). https://doi.org/10.1016/j.scs.2021.103454
10. Szpyrka, M., Matyasik, P., Biernacki, J., Biernacka, A., Wypych, M., Kotulski, L.: Hierarchical Communication Diagrams. COMPUTING AND INFORMATICS **35**(1), 55–83 (May 2016)
11. Wojnicki, I., Ernst, S., Kotulski, L.: Economic Impact of Intelligent Dynamic Control in Urban Outdoor Lighting. Energies **9**(5), 314 (May 2016). https://doi.org/10.3390/en9050314
12. Wojnicki, I., Ernst, S., Kotulski, L., Sędziwy, A.: Advanced street lighting control. Expert Systems with Applications **41**(4, Part 1), 999–1005 (Mar 2014). https://doi.org/10.1016/j.eswa.2013.07.044
13. Wojnicki, I., Kotulski, L.: Improving Control Efficiency of Dynamic Street Lighting by Utilizing the Dual Graph Grammar Concept. Energies **11**(2), 402 (Feb 2018). https://doi.org/10.3390/en11020402

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_31