

Article

Improving Control Efficiency of Dynamic Street Lighting by Utilizing the Dual Graph Grammar Concept

Igor Wojnicki *  and Leszek Kotulski

Department of Applied Computer Science, AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland; kotulski@agh.edu.pl

* Correspondence: wojnicki@agh.edu.pl; Tel.: +48-508-444-450

Received: 5 December 2017; Accepted: 6 February 2018; Published: 9 February 2018

Abstract: The paper introduces a definition of *dual graph grammar*. It enables two graphs to share information in a synchronized way. A smart city example application, which is an outdoor lighting control system utilizing the *dual graph grammar*, is also demonstrated. The system controls dimming of street lights which is based on traffic intensity. Each luminaire's light level is adjusted individually to comply with the lighting norms to ensure safety. Benefits of applying the *dual graph grammar* are twofold. First, it increases expressive power of the mathematical model that the system uses. It becomes possible to take into account complex geographical distribution of sensors and logical dependencies among them. Second, it increases the system's efficiency by reducing the problem size during run-time. Experimental results show a reduction of the computation time by a factor of 2.8. The approach has been verified in practice.

Keywords: graph transformations; *dual graph grammar*; road lighting; street lighting; efficiency; problem size reduction

1. Introduction

Applying the graph theory to create formal models of intelligent spaces has already been proven useful, as they enable interactions between smart city components [1,2]. Synergy of knowledge regarding different aspects of the space under consideration can result in added value [3]. An illustrative example is street and road lighting [4]. A formal architectural model can represent features such as streets, pavements, sidewalks, medians, buildings, trees etc. Having such a model enables outdoor lighting design optimization and its dynamic control [1].

Lighting design is about defining the placement of poles and luminaires as well as their parameters, such as luminaire type, its power, overhang, tilt, rotation etc. Design optimization is aimed at providing particular values of these parameters to maximize safety and minimize energy usage.

Dynamic control is about adjusting lighting levels on demand, based on the current needs, taking into account such factors as traffic intensity or weather conditions. Having a graph-based formal model enables easy deployment of control schemas, even exceedingly sophisticated ones. It follows the idea of data-driven design which separates data, the formal model, from its processing, the control rules. From a practical point of view, it also facilitates deployments in new locations. The control rules do not have to be updated or reprogrammed—it is sufficient to provide a new model of the environment. Such graph-based dynamic control results in increased efficiency, both in terms of lighting energy consumption reduction and decreased deployment effort.

In order to make the environment safe, its illumination has to comply with lighting norms, such as CEN/TR 13201-1: 2004 or 2014 [5,6]. While designing a lighting system, proper selection of the lighting class, applied to a given area being illuminated, is most important. A lighting class is a set of lighting

requirements such as illumination, uniformity etc. Particular classes, such as *me2*, *me3* etc. deliver well-defined lighting levels.

The main driver for this paper is based on experience from a research project which resulted in deployment of 3768 light points in Kraków, Poland. It has been observed that the sensor infrastructure becomes more and more complex, delivering more and more information. It is often redundant to provide comparative results or robustness to failures. In Kraków, the number of ambient light sensors for outdoor lighting has increased 15 times over just one year (2016). Instead of a single sensor in an electric cabinet which supplies three circuits of 50 luminaires each, there is one for every 10 luminaires, mounted on the lamp pole. Similarly, the number of traffic intensity sensors has increased as well. For a typical intersection, the number of traffic intensity detectors (induction loops) has increased by a factor of 2. For example, in the case of a 4-way intersection with 10 inbound and 6 outbound lanes, it has increased from 10 to 20.

This paper focuses on providing a formal basis to incorporate knowledge regarding multiple sensors into the lighting control model by introducing the *dual graph grammar* concept. As a result, the model remains easy to comprehend, while the computing power needed to process and interpret it is decreased.

2. Motivation and Related Research

The concepts of smart cities and smart environments, as well as technologies related to them, are gaining traction. The main technological enablers are Internet of Things (IoT) and sensor nets resulting in broad availability of data [4,7,8]. The main drivers are safety, convenience and, last but not least, economic factors such as energy savings [9,10]. Often, combination of smart technologies with renewable energy sources offers added economic value [11]. Such solutions can be applied at the scale of an entire city [12], a single block, or indoor [13–15]. They are often based on rules and configured for particular deployments [16–19]. However, a lack of underlying formal models often results in harder and more costly adjustment to evolving requirements or deployment at new locations.

Graphs seem to be an appropriate tool to develop such models [1,20]. However, one of the challenges is the efficiency of graph transformations. The main problems are graph parsing and determination of membership. With strong limitations on the graph structure, the membership problem is of $O(n^2)$ computational complexity [21]. However, in general it remains a higher-order polynomial. The efficiency depends on the size of the entire graph. One of the methods of its reduction is the concept of hierarchical graphs [22,23], where the nodes can also be graphs introducing more detail at the graph's lower levels. While the concept provides a convenient structure, it also increases the graph size and processing complexity. Real-world cases indicate a need for a representation which would not require increased processing power. An example of such a case is generation of dynamic control rules for an outdoor lighting system. Initially, a flat graph structure, the Control Availability Graph (CAG), was proposed for this purpose [1,20]. It is an extract of architectural features that are necessary to provide dynamic control, hence the name. Dynamic control is provided by software performing the following tasks:

1. obtaining data from sensors and updating the CAG accordingly (traffic intensity and ambient light intensity sensors are considered),
2. firing the control rules in the form of graph transformations, which results in conclusions being stored in the CAG,
3. delivering the conclusions to the actuators, which set power levels of the appropriate luminaires.

The more luminaires are under control, the larger the CAG becomes. Since the control software is meant to be used as an embedded system, the performance requirements play a major role.

To present the CAG concept let us introduce an example intersection which is given in Figure 1. To control lighting intensity, the following formal structure has been defined. There is a single street segment *s1*, which is a length of a street with uniform lighting conditions. It covers four

luminaires $l1, \dots, l4$. There is also an ambient light detector $da1$ and a traffic intensity detector $dt1$. A corresponding CAG is given in Figure 2. Some of the graph indices, labels and attributes are shown. The following graphical convention is used. Vertex labels are indicated within vertices, indices, if present, are prefixed with a “/” character. For example, a vertex indicated as $dt/dt1$ has the label dt and an index of $dt1$. In this case, the vertex represents a traffic intensity detector, hence the label dt . Indices are arbitrary symbols to increase readability and correspondence with real objects. Edge labels can also be present. For example, between vertices $s1$ and $c1$, there is a label indicating the lighting class $me3$. Attribute names and values are indicated as well, for example $p(0.8)$ is an attribute p holding the value of 0.8 which indicates luminaire’s power level.

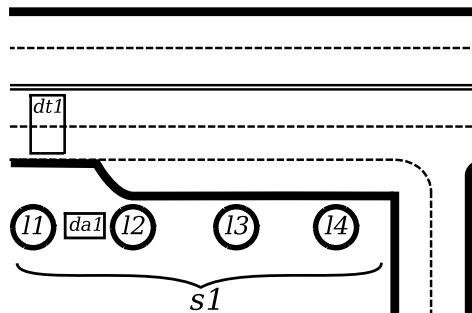


Figure 1. A street segment $s1$, with luminaires $l1, \dots, l4$, a traffic intensity detector $dt1$ and an ambient light detector $da1$.

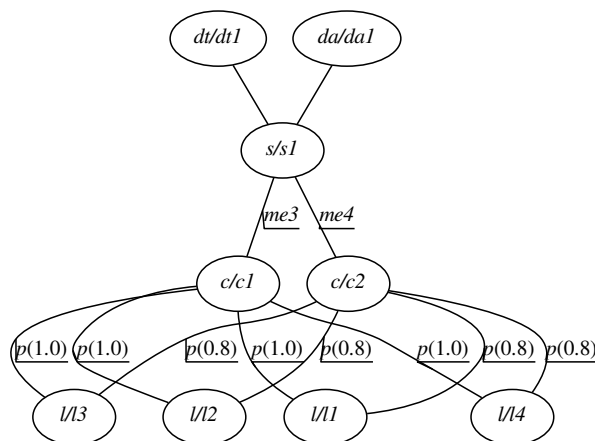


Figure 2. A street segment, Control Availability Graph logical model, representing all features from Figure 1 and luminaire configurations: a traffic intensity detector $dt/dt1$, an ambient light intensity detector $da/da1$, a segment $s/s1$, luminaire configurations $c/c1, c/c2$, luminaires $l/l1, l/l2, l/l3, l/l4$.

The graph consists of a vertex labeled as dt which represents a traffic intensity detector, da which represents an ambient light detector, s which represents a segment, c which represents a luminaire configuration, and finally l representing a luminaire. Edge labels between vertices c and s represent lighting classes which are provided by a given configuration on a given segment. With each lighting class, the power level of the luminaries belonging to the segment is defined. It is represented as the p attribute of an edge between vertices labeled c and l . Its value is a coefficient by which a luminaire’s power setting has to be adjusted, ranging from 0 (0%) to 1 (100%). It indicates how much the luminaire should be dimmed. Assuming that the power setting is expressed as $p(x)$, its relationship with the dimming is expressed as: $x = 1 - d$. The settings for a given configuration which supports an indicated norm are computed based on photometric calculations according to the lighting norms while building the graph. During run-time, when control is carried out, the values of the sensors allow to determine

the proper lighting class, represented as an edge between s and c . The power settings for each of the luminaires are selected accordingly based on the class.

In case of cities with thousands of lamps, such a graph consists of hundreds of thousands of nodes and edges, thus both the readability of the model and efficiency of graph grammars are major challenges.

The CAG is built by performing photometric calculations based on the lighting norms. The examples in this paper, as well as the deployment in Kraków, utilize CEN/TR 13201-1:2004 norm [5] even though the norm was updated in 2014 [6]. It is because the deployment was designed in 2015 and 2016 at which time the 2014 update was not acknowledged in Poland yet.

In this paper, we introduce the *dual graph grammar* concept. It results in a graph-based model with low computational complexity and flexibility where needed. An example application is a dynamic street lighting control system. It enables efficient adjustment of lighting to particular needs and capabilities by providing high-performance processing. The adjustment can be based on far more complex logical relationships among sensors than the ones presented in Figure 2.

There is also additional motivation to use as many sensors as available. The more sensor information is available, the more precise the control process is, which can lead to increased energy savings. For example, if there are redundant traffic intensity sensors taken into consideration, not delivering the measurements by one of them does not influence the control process. If there is no redundancy, a malfunctioning traffic detector forces the control system to assume the maximum value of allowed traffic intensity. This results in delivering more light than necessary and consuming more energy.

It needs to be pointed out that even insignificant energy savings at a single luminaire lead to substantial savings for a city, due to the effect of scale. A city with a population of 1 million has approximately 80,000 light points (it has been experimentally established that there is a relationship between population density and the number of light points in urbanized areas, there are 8 light points per 100 citizens for Poland), which consume from 20 to 50 GWh per year. The estimation is based on the assumption that the wattage of a single luminaire varies from 60 to 150 (both LED and HPS light sources) on average.

3. Problem Statement

A synthesis of the CAG model, which is based on photometric calculations and a rule-based control approach were proposed in [1]. The control process works efficiently as long as the sensor infrastructure is built from the ground up, having only one detector of a given kind at any street segment. However, in the real world, there are usually multiple detectors of a kind at one segment which are part of an Intelligent Transportation System (ITS) or a sensor network which is already deployed. If so, input to the lighting control system should be based on a result of calculations on data from these multiple detectors. Extending the model, thus the graph grammar, to cover these additional computations, while perfectly justified, leads to some undesired side effects. First of all, the model becomes less clear, focusing on the sensors and relationships between them, instead of binding sensors with segments, configurations and luminaires. Then, the control rules have to take care of choosing the most appropriate sensor from a group and applying the appropriate calculations. It makes them more complex, challenging to verify, and can also lead to decreased processing performance of such a model during run-time.

To handle these issues, the concept of a *dual graph grammar* is introduced. Such a grammar provides double expressiveness of the model: efficient calculations on sensor data and logical organization of the infrastructure. It is created by combining two separate graph grammars and explicitly stating the synchronization conditions between the graphs they generate. As a result, there will be two synchronized graphs representing these two aspects. Such an approach will also improve run-time performance, since the two graphs can be processed independently.

Before introducing formal definitions, let us focus on an example which explains the problem. As mentioned earlier, in highly urbanized areas, the example shown in Figure 1 becomes unrealistic. The actual detector distribution would be close to that presented in Figure 3 instead. The difference is that for multi-lane streets, there is usually at least one detector per lane. Sometimes, the traffic intensity detectors are doubled or tripled and their placement could be before or after an intersection. Considering the example in Figure 3, in order to calculate the traffic intensity dt in segment $s1$, one needs to perform:

$$dt = \max(dt1, dt3) + \max(dt2, (dt4 + dt5))$$

where $dt1, \dots, dt5$ express traffic intensity measured by respective traffic intensity detectors. In this case, $dt1$ and $dt3$ are redundant, measuring traffic intensity in the same lane. Similarly, $dt2$ and a sum of $dt4$ and $dt5$ are redundant too.

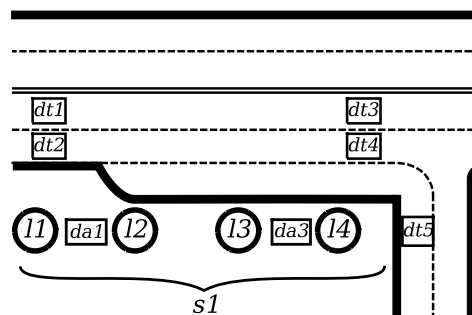


Figure 3. A real world example of a street segment $s1$, with luminaires $l1, \dots, l4$, and multiple traffic intensity detectors $dt1, \dots, dt5$ and ambient light detectors $da1, da2$.

There could also be more than one ambient light sensor for a single segment. These sensors are often located on the poles, together with luminaires. The actual ambient light intensity, taking into account multiple ambient light detectors, should be calculated as:

$$da = \min(da1, da2)$$

where $da1$ and $da2$ represent ambient light intensity measured by respective detectors. The $\min()$ function is used since the lighting level at the segment has to be adjusted to the lowest value of ambient light detected, to avoid the segment being underlit.

A model with all the sensors taken into consideration could look like the one given in Figure 4. However, as mentioned earlier, it is not practically feasible, because it lacks the knowledge about what particular calculations should be performed; spatial and logical relationships among the detectors are not there. Thus, in order to make it usable, the model gets simplified, taking into consideration only a subset of the detectors, as shown in Figure 5. Then, by a rule of thumb, traffic intensity for all traffic detectors should be added up, while ambient light intensity is calculated as a minimum over all detectors for a given segment.

However, there are still some issues:

1. no detector redundancy is taken into consideration, which increases risk of traffic intensity misjudgment due to sensor failure or noise,
2. sensors to be included in the CAG have to be hand-picked,
3. data aggregation from multiple detectors is driven by control rules; it is performed on an arbitrary number of detectors, which might result in run-time performance degradation, since the number of detectors can vary from segment to segment.

Let us show how the proposed *dual graph grammar* can solve these issues.

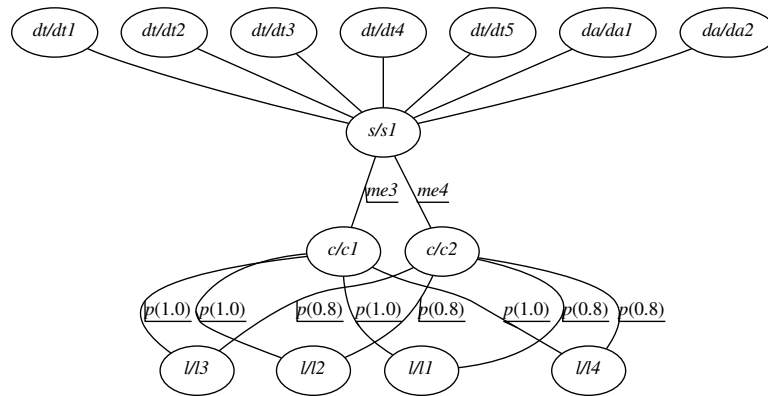


Figure 4. A street segment, logical model: a Control Availability Graph; no logical relationships among detectors.

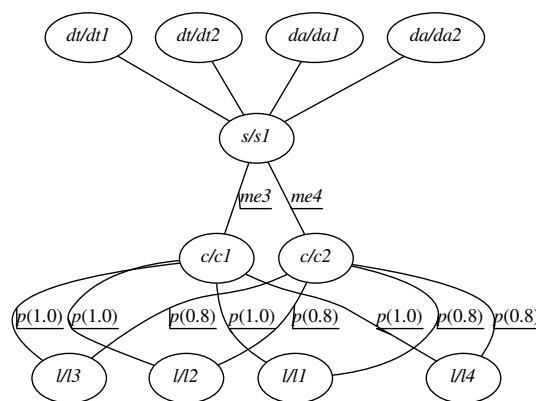


Figure 5. A street segment, logical model: a Control Availability Graph; a simplified approach which takes into account a subset of detectors.

4. Dual Graph Grammar Representation

As mentioned in the previous section, the CAG structure does not directly represent the complexity of the influence of multiple sensors on lighting segments and subsequently on the luminaires to be controlled. Hard-coding it, which is using general-purpose programming languages to represent this influence, would make it hard to analyze or modify, since it would not be part of the graph. On the other hand, extending the CAG to perform additional processing to cover the influence would increase its size and decrease the overall processing efficiency, since the complexity of graph parsing algorithms depends on the graph size. To solve these problems, we introduce the *dual graph grammar* representation, in which two graph grammars are used to generate a consistent model in such a way that they are separate graphs with common elements. The synchronization of both graphs is provided by the validation properties associated with each of the grammars. As a result, the two graphs have separate purposes: one is used for control, and the other—to express relationships among sensors. Let us define this structure formally.

Definition 1. A dual graph grammar $[\Psi, \Theta]$ is a pair of graph grammars:

$$\Psi = (\Sigma_{\Psi}, \Gamma_{\Psi}, \Delta_{\Psi}, \Phi_{\Psi}, S_{\Psi}, \Pi_{\Psi})$$

$$\Theta = (\Sigma_{\Theta}, \Gamma_{\Theta}, \Delta_{\Theta}, \Phi_{\Theta}, S_{\Theta}, \Pi_{\Theta})$$

where:

- Σ_{Ψ} and Σ_{Θ} are the sets of node labels,

- $\Delta_{\Psi} \subset \Sigma_{\Psi}$, $\Delta_{\Theta} \subset \Sigma_{\Theta}$ are the sets of terminal node labels,
- Γ_{Ψ} and Γ_{Θ} are the sets of edge labels,
- Φ_{Ψ} and Φ_{Θ} are the sets of transformation rules,
- S_{Ψ} and S_{Θ} are the starting graphs,
- Π_{Ψ} and Π_{Θ} are the validation graph grammar conditions,

and the following condition is fulfilled: $\Sigma_{\Psi} \cap \Sigma_{\Theta} \neq \emptyset$.

Applications of transformation rules on a graph are atomic. This means that two productions on a single subgraph are mutually exclusive.

Definition 2. A dual graph grammar $[\Psi, \Theta]$ is synchronized if both validation conditions Π_{Ψ} and Π_{Θ} are met.

Let us explain how the proposed *dual graph grammar* notation supports the problem of maintaining the influence of multiple sensors on a given segment. The basis for this concept is the notion of a *virtual detector*. It represents a data source with values calculated based on information from actual detectors. There is only one virtual detector of a given type in relation with a segment. In this case, a node labeled by *dva* will represent such a detector, providing ambient light intensity. Similarly, a node labeled by *dvt* (virtual traffic detector) will represent traffic intensity. We assume that *dva* and *dvt* are the only node labels that belong to $\Sigma_{\Psi} \cap \Sigma_{\Theta}$.

The Ψ graph grammar will provide the logical relationships among entities; it enables control by defining a new Control Availability Graph: CAG_{Ψ} . Its role is to:

- maintain information about the lighting infrastructure to be controlled,
- optimize the internal structure of the lighting system (for more details see [24]),
- evaluate the current CAG_{Ψ} parameters to provide actual control.

The Θ grammar will define the evaluation rules for virtual detectors—a Detector Structure Graph (DSG_{Θ}). Its role is to:

- maintain information about the detectors' logical structure and evaluation conditions of sensory data resulting in virtual detectors' values,
- evaluate the virtual detectors' values.

Let us define CAG_{Ψ} and DSG_{Θ} formally.

Definition 3. CAG_{Ψ} is an attributed graph over the set of node labels Σ_{Ψ} and the set of edge labels Γ_{Ψ} which is generated by the Ψ grammar:

$$CAG_{\Psi} = (V_{\Psi}, E_{\Psi}, lab_{\Psi}^V, lab_{\Psi}^E, \Sigma_{\Psi}, \Gamma_{\Psi}, att_{\Psi}^V, att_{\Psi}^E, A_{\Psi}^V, A_{\Psi}^E)$$

where:

- V_{Ψ} is a finite, nonempty set of graph nodes identified unambiguously by some injective indexing function $Index : V_{\Psi} \rightarrow IS$, where IS is a set of indexing symbols (usually \mathbb{N} is used but in some cases it is easier to provide arbitrary symbols to increase readability),
- $E_{\Psi} \subseteq V_{\Psi} \times V_{\Psi}$ is a set of edges,
- $\Psi_lab_{\Psi}^V : V_{\Psi} \rightarrow \Sigma_{\Psi}$ is a node labelling function,
- $\Psi_lab_{\Psi}^E : E_{\Psi} \rightarrow \Gamma_{\Psi}$ is an edge labelling function,
- $\Sigma_{\Psi} = \{s, l, dvt, dva, c\}$ is a set of node labels, where:
 - s represents segments,
 - l represents luminaires,

- *dot* and *dva* represent virtual traffic and ambient light intensity detectors, respectively,
- *c* represents a configuration, which is a set of adjustments, associated with a relevant set of luminaires at a given segment,
- $\Gamma_{\Psi} = \{me2, me3, me4, me5\}$ is a set of edge labels, where: *me2*, *me3*, *me4*, *me5* are lighting classes,
- $att_{\Psi}^V : V_{\Psi} \times \Sigma_{\Psi} \rightarrow 2^{A_{\Psi}^V}$ is a node attributing function, such that for $x \in V_{\Psi}, l \in \Sigma_{\Psi}, a \in A_{\Psi}^V$ $att_{\Psi}^V(x, l)(a)$ is a value of the attribute *a*,
- $att_{\Psi}^E : E_{\Psi} \times \Gamma_{\Psi} \rightarrow 2^{A_{\Psi}^E}$ is an edge attributing function, such that for $x \in E_{\Psi}, l \in \Gamma_{\Psi}, a \in A_{\Psi}^E$ $att_{\Psi}^E(x, l)(a)$ is a value of the attribute *a*,

Intuitively, we feel that each segment *s* is connected by an edge with exactly one *dot* and one *dva*. Formally, the validation condition Π_{Ψ} of the grammar is as follows:

$$\begin{aligned}
 \forall v_1, \in V_{\Psi} : lab_{\Psi}^V(v_1) &= l, \\
 \exists! v_2, \in V_{\Psi} : lab_{\Psi}^V(v_2) &= dot, \\
 \exists! v_3, \in V_{\Psi} : lab_{\Psi}^V(v_3) &= dva, \\
 \exists v_4 \in V_{\Psi} : lab_{\Psi}^V(v_4) &= s, \\
 \exists v_5 \in V_{\Psi} : lab_{\Psi}^V(v_5) &= c, \\
 \{(v_3, v_4), (v_2, v_4), (v_4, v_5)(v_5, v_1)\} &\subset E_{\Psi}
 \end{aligned} \tag{1}$$

Definition 4. DSG_{Θ} is an attributed directed graph over the set of node labels Σ_{Θ} and the empty set of edge labels Γ_{Θ} which is generated by the Θ grammar:

$$DSG_{\Theta} = (V_{\Theta}, E_{\Theta}, lab_{\Theta}^V, \Sigma_{\Theta}, att_{\Theta}^V, A_{\Theta}^V)$$

where:

- $V_{\Theta}, E_{\Theta}, lab_{\Theta}^V, att_{\Theta}^V, A_{\Theta}^V$ are defined similarly as in Definition 3,
- $\Sigma_{\Theta} = \{dot, dva, dt, da, +, -, \times, /, min, max, val\}$ is a set of node labels, where:
 - *dot* and *dva* represent virtual traffic and ambient light intensity sensors, respectively,
 - *dt* and *da* represent actual traffic and ambient light intensity sensors, respectively,
 - *+*, *−*, *×*, */*, *min*, *max* represent arithmetic operators,
 - *val* represents constant arbitrary values.

Formally, the validation condition Π_{Θ} makes sure that there are no non-terminals in the graph:

$$\forall v \in V_{\Theta} : lab_{\Theta}^V(v) \in \Delta_{\Theta} \tag{2}$$

While DSG_{Θ} is being built there may be vertices with labels being non-terminals. In such a case, the graphs are not in a synchronized state since according to Definition 2, the validation conditions are not met.

5. Solving the Problem

To summarize, in order to handle the issues presented in Section 3, the following assumptions are made. There is a dual graph grammar $[\Psi, \Theta]$ defined. The Ψ graph grammar is used to represent the control structure of the lighting system stored in a CAG_{Ψ} . The Θ graph grammar represents the evaluation of virtual sensors, defined by a DSG_{Θ} .

As a result, the following proposals can be made:

- p1 the model of the dynamic lighting control is a graph, which consists of: CAG_{Ψ} and DSG_{Θ} ,
- p2 there are virtual detectors introduced, represented by the graphs' synchronization vertices,

- p3 there is only a single virtual detector of a given type (traffic, ambient light, etc.) in relation to an individual segment,
- p4 logical relationships between detectors and a virtual detector are expressed as a graph.

Application of the proposed *dual graph grammar* has the following properties:

1. it separates the logical relationships between segments, detectors and luminaires from calculations on sensor data; thus, control is separated from evaluation and both can be processed separately during run-time,
2. it simplifies the control process graph transformations, since there is only one edge between a segment and a detector of a given kind,
3. a detector can be used in calculations for multiple virtual detectors,
4. there is a number of vertices to be processed upon executing control being reduced, since only the Ψ graph is considered; this enhances run-time performance of the control system,
5. full knowledge necessary for control is expressed as a graph, subject to analysis and validation,
6. formal graph folding enables extracting and viewing a CAG_{Ψ} or a DSG_{Θ} , allowing for a clear representation of control availability or logical relationships among detectors, respectively,
7. high-performance processing of DSG_{Θ} is possible since it represents a parse tree,
8. calculations (DSG_{Θ}) can be easily updated without influencing CAG_{Ψ} ,
9. control availability (CAG_{Ψ}) can be easily updated without influencing DSG_{Θ} .

Applying the proposed *dual graph grammar* approach to the example in Figure 3 results in a model presented in Figure 6 (CAG_{Ψ}) and Figure 7 (DSG_{Θ}).

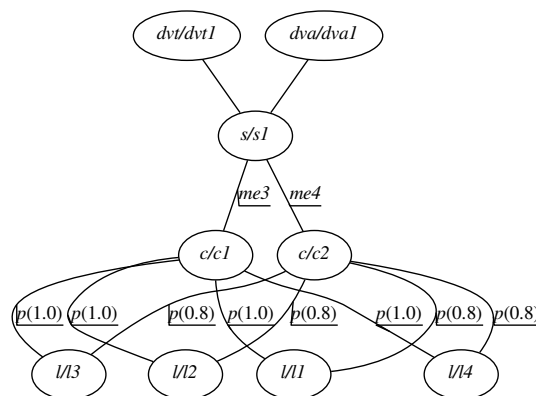


Figure 6. A street segment, logical model: CAG_{Ψ} , based on the Ψ (structure) grammar.

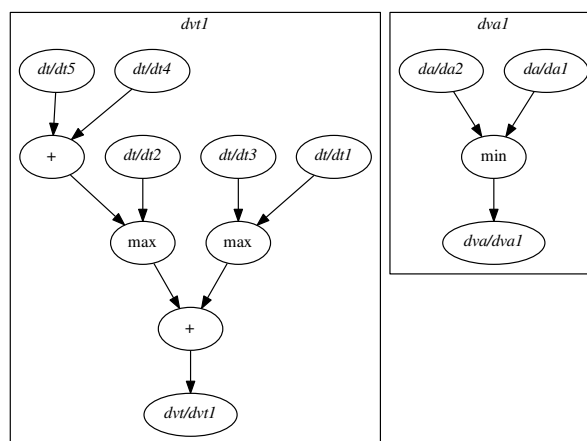


Figure 7. A street segment, evaluation model; DSG_{Θ} complying with the Θ (evaluation) grammar.

6. A Real World Example

Having defined the approach, let us present an intuitive real world example. It is an excerpt from a large-scale deployment of dynamic lighting control system in Krakow, Poland. The excerpt describes a single intersection presented in Figure 8. There are five segments: s_1, s_2, s_3, s_4, s_5 . The intersection is equipped with inductive loop traffic intensity sensors: $D_{2_11}, D_{2_12}, D_{2_21}, D_{2_22}, D_{4L_2}, D_{4L}, D_{4_22}, D_{4_21}, D_{4_12}, D_{4_11}, D_{3_11}, D_{3_12}, D_{3_21}, D_{3_22}, D_{1_12}, D_{1_11}, D_{1_22}, D_{1_21}, D_{24_1}$. There are no other detectors considered.

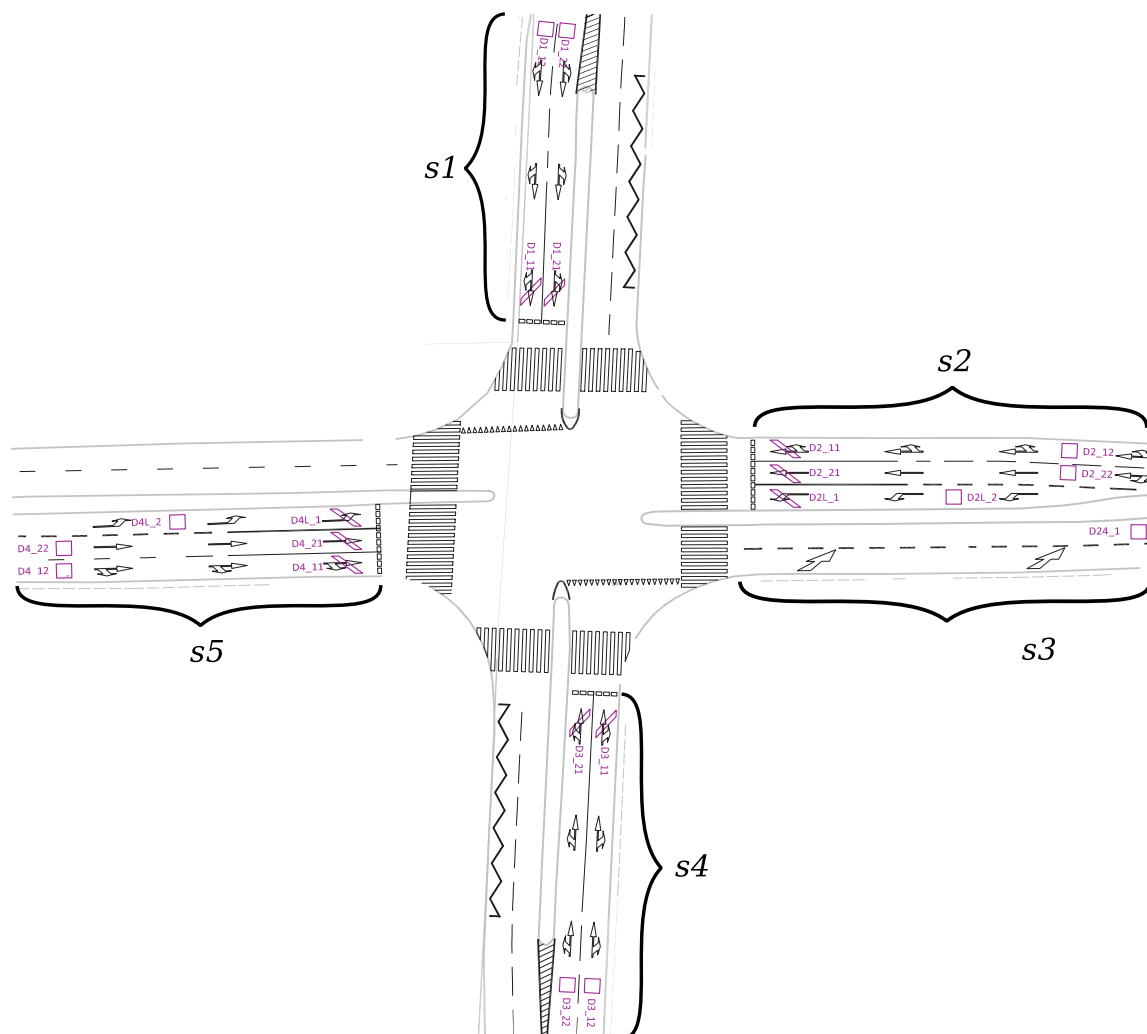


Figure 8. An example intersection, Krakow, Poland with inductive loops: $D_{2_11}, D_{2_12}, D_{2_21}, D_{2_22}, D_{4L_2}, D_{4L}, D_{4_22}, D_{4_21}, D_{4_12}, D_{4_11}, D_{3_11}, D_{3_12}, D_{3_21}, D_{3_22}, D_{1_12}, D_{1_11}, D_{1_22}, D_{1_21}, D_{24_1}$, and segments: s_1, s_2, s_3, s_4 and s_5 .

Influence of traffic data from the virtual detectors on particular segments, assignments of lamps (l_1, \dots, l_{14}) to the segments as well their configurations (c_1, \dots, c_6)—represented in CAG_{Ψ} —are shown in Figure 9. Since there are no luminaires under control in s_1 and s_5 , there is no edge connecting s_1 or s_5 with any vertex labeled with c . The ambient light detectors are omitted for simplicity, so the available CAG_{Ψ} vertex labels are defined as: $L_V = \{s, l, dot, c\}$. Furthermore, $\Gamma_{\Psi} = \{me_{3c}, me_{4b}, me_5\}$ since the set of lighting classes is adjusted to cover me_{3c}, me_{4b}, me_5 . There are classes me_{4b} and me_5 available on the segment s_4 , while me_{3c} and me_{4b} are available on s_2 and s_3 . These lighting classes were selected due to physical features of the segments according to the lighting norms. Correspondence between lighting

classes and the power settings of the luminaires have been verified by using photometric calculations. Similarly, the synchronization condition Π_{Ψ} is relaxed, due to a lack of ambient light detectors:

$$\begin{aligned} \forall v_1, v_2 \in L_V : lab_V(v_1) = l, lab_V(v_2) = dt \\ \exists v_4 \in L_V : lab_V(v_4) = s, \\ \exists v_5 \in L_V : lab_V(v_5) = c, \\ \{(v_2, v_4), (v_4, v_5)(v_5, v_1)\} \subset E \end{aligned} \quad (3)$$

For each of the configurations, for the given class, there are lamp settings defined as edges between c and l . They indicate power setting for a particular lamp, provided by a value of attribute p . For example, there is an edge between $s4$ and $c5$, which represents a configuration for establishing the $me4b$ lighting class on $s4$. It is carried out by setting the luminaires $l6$ – $l4$ to full power, which is indicated by the attribute p set to 1.0 ($p(1.0)$). For comparison, establishing the $me5$ class on the same segment would set the luminaires to 80% ($p(0.8)$). The actual power settings were computed by running photometric calculations while building CAG_{Ψ} .

The DSG_{Θ} part, which represents virtual sensor evaluations, is presented in Figure 10. The disconnected directed graph forms arithmetic expression parse trees. As a visual aid, the vertices representing the sensors (dv label) that are placed in the same segment as the virtual sensor are additionally enclosed in solid lines labeled with the virtual sensor index. For example, there is a virtual sensor $dv2$ which belongs to the same segment as the detectors $D2_{11}$, $D2_{12}$, $D2_{21}$, $D2_{22}$, thus it is enclosed by a solid line labeled $dv2$. Nodes representing arithmetic functions are labeled with their respective symbols e.g., \max , $+$, \times etc. The value of $dv2$ is evaluated as: $\max(D2_{11}, D2_{12}) + \max(D2_{21}, D2_{22})$. There are also vertices representing constants; their values are indicated using val attributes. For example, there are two nodes attributed $val(0.3)$ and $val(0.7)$ in the enclosure $dv3$. Their respective values are used for evaluation of $dv3$, which is a result of: $\max(D24_{11}, \max(D1_{12}, D1_{11}) \times 0.3 + 0.7 \times \max(D4_{12}, D4_{11}) + 0.3 \times \max(D3_{11}, D3_{12}) + \max(D4_{22}, D4_{21}))$. For this particular virtual detector, measurements provided by detectors in other segments are also taken into consideration, which increases the connectivity of the graph.

As a result, the size of CAG_{Ψ} , in comparison with CAG , is reduced. For the example intersection, considering all sensors, the CAG consists of 19 traffic intensity detectors (dt), 5 segments (s), 14 luminaires (l) and 6 configurations (c), which results in a total of 44 vertices. There are 21 edges between traffic detectors (dt) and segments (s), 6 edges between segments (s) and configurations (c), and 28 edges between configurations (c) and luminaires (l) which results in 55 edges total. For comparison, CAG_{Ψ} consists of the following vertices: 5 virtual traffic intensity detectors (dv), 5 segments (s), 14 luminaires (l) and 6 configurations (c), with a total of 30 vertices. There are the following edges: 5 between virtual traffic detectors (dv) and segments (s), 6 between segments (s) and configurations (c), and 28 between configurations (c) and luminaires (l) which results in 39 total. The size of a graph is expressed as the number of its edges. In this case, it was reduced by 29%, from 55 to 39. The computational complexity of the dynamic control algorithms is $O(n^3)$. This is due to the properties of the Ψ grammar which do not comply with the limitations proposed in [21]. Therefore, the aforementioned graph size reduction decreases the number of computations performed during run-time from 166,375 to 59,319, giving a reduction factor of 2.8. Such a reduction of computation time significantly increases the scalability of the control system.

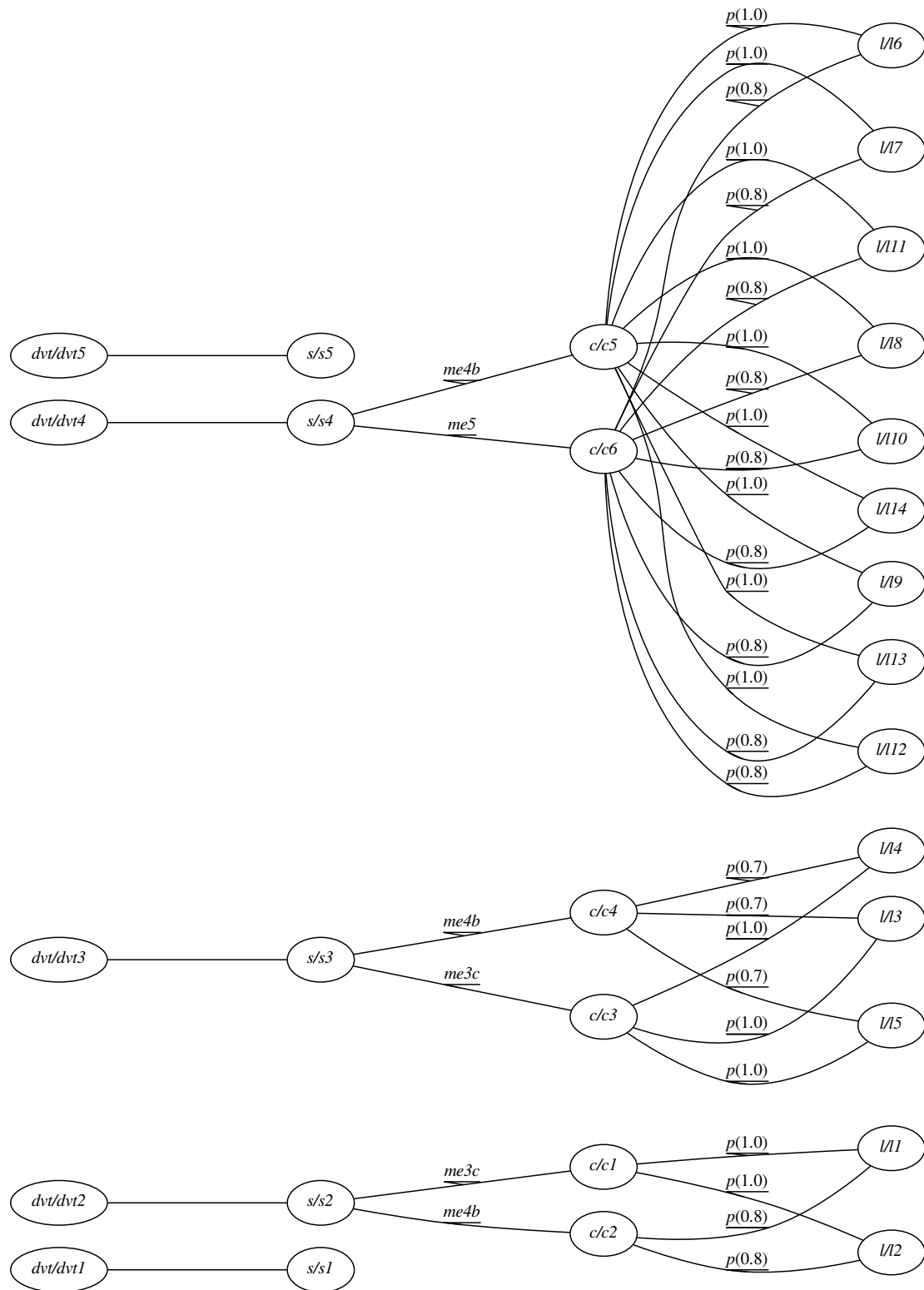


Figure 9. An example intersection, Krakow, Poland; CAG Ψ complying with the Ψ (structure) grammar.

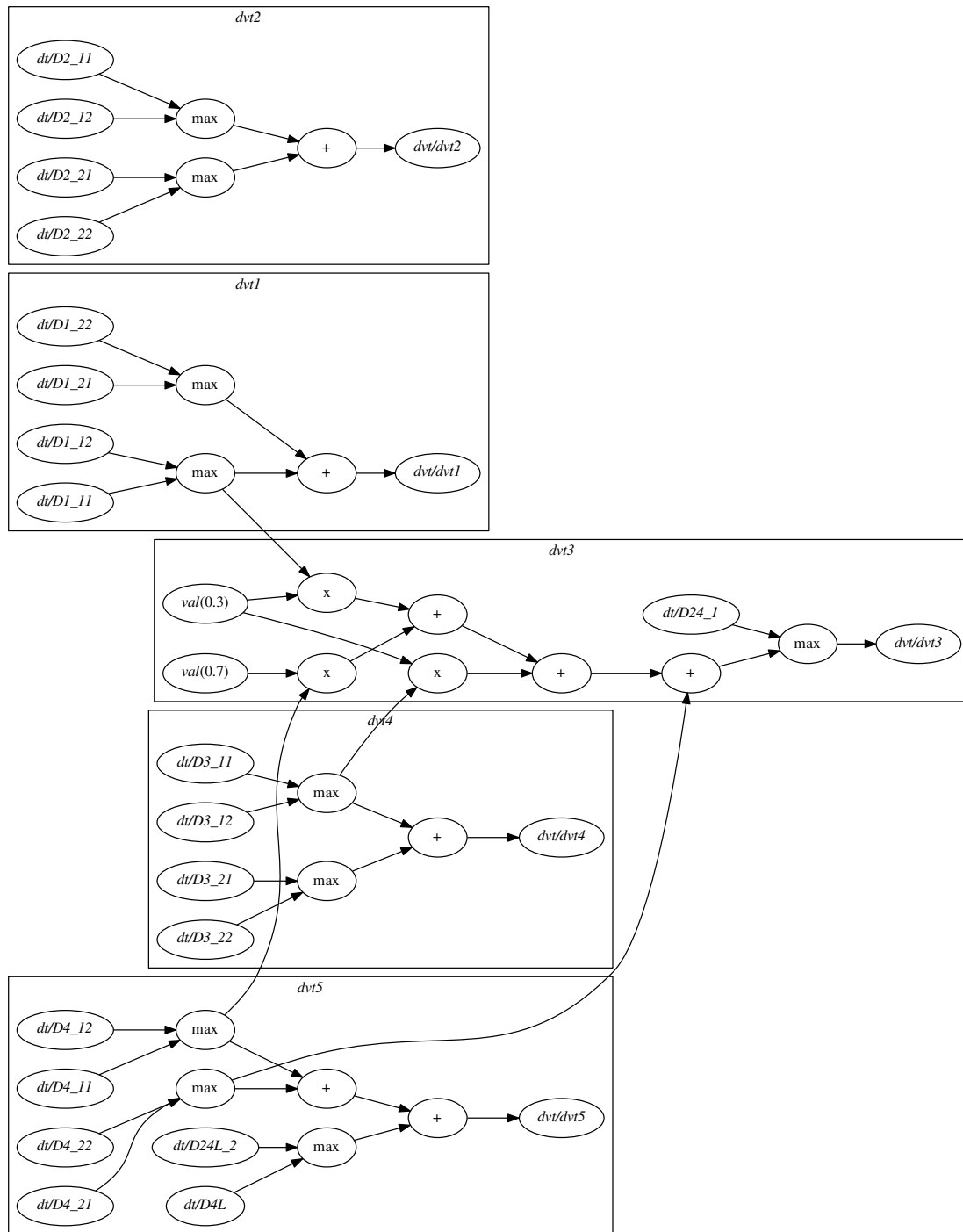


Figure 10. An example intersection, Krakow, Poland; DSG_{Θ} complying with the Θ (evaluation) grammar.

7. Conclusions

Utilizing graph-based models for control systems provides a data-driven approach, which can increase the reusability of software. A graph model describes the environment, while software provides means for its processing. Upon new deployments, only the model needs to be adjusted, while the software remains unchanged. The proposed *dual graph grammar* increases processing performance and readability of such models.

A dynamic outdoor lighting control system is a practical example which confirms these claims. It is built around the dual nature of the problem: assuring proper control processing (CAG_{Ψ}) and

utilizing relationships among sensors (DSG_{Θ}). These two aspects form a single graph defined by a *dual graph grammar* $[\Psi, \Theta]$. Depending on particular needs, the graph can be folded, presenting just one of the two components. Since a formal graph grammar synchronization can be defined, the interactions between the two components are smooth. The graphs can be processed independently as long as the synchronization conditions are met. If they are not met, the processing has to be ceased until the graphs become synchronized again. The aspects are clearly separated, which decreases the risk of update issues or further development of the model. The updates can regard changing physical conditions, such as deploying new sensors or retiring old ones, as well as updating the control logic requirements, which would cause the control system to behave differently.

As a result, the proposed *dual graph grammar* provides a model that on one hand is clean and readable, but on the other significantly reduces the processing time. Lowering the computing power requirements by a factor of 2.8 increases the scalability of the dynamic control system. In case of complex relationships among sensors, it ensures high processing capability, since they are expressed as arithmetic expression parse trees. This enables efficient computation of the values delivered by the virtual detectors.

Acknowledgments: This research was supported by the AGH University of Science and Technology grant number 11.11.120.859.

Author Contributions: I.W. formulated the problem, which was jointly researched by I.W. and L.K. L.K. supervised the theoretical background, I.W. and L.K. wrote the paper, I.W. designed and conceived the experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wojnicki, I.; Ernst, S.; Kotulski, L.; Sędziwy, A. Advanced street lighting control. *Expert Syst. Appl.* **2013**, *41*, 999–1005.
2. Wojnicki, I.; Kotulski, L. Controlling Complex Lighting Systems. In *Complex Systems and Dependability; Advances in Intelligent and Soft Computing*; Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 170, pp. 305–317.
3. Neirotti, P.; De Marco, A.; Cagliano, A.C.; Mangano, G.; Scorrano, F. Current trends in Smart City initiatives: Some stylised facts. *Cities* **2014**, *38*, 25–36.
4. Merlino, G.; Bruneo, D.; Distefano, S.; Longo, F.; Puliafito, A.; Al-Anbuky, A. A Smart City Lighting Case Study on an OpenStack-Powered Infrastructure. *Sensors* **2015**, *15*, 16314–16335.
5. European Committee for Standardization (CEN). *CEN/TR 13201-1: 2004, Road Lighting. Selection of Lighting Classes*; Technical Report; European Committee for Standardization: Brussels, Belgium, 2004.
6. European Committee for Standardization (CEN). *CEN/TR 13201-1: 2014, Road Lighting—Part 1: Guidelines on Selection of Lighting Classes*; Technical Report; European Committee for Standardization: Brussels, Belgium, 2014.
7. De Paz, J.F.; Bajo, J.; Rodríguez, S.; Villarrubia, G.; Corchado, J.M. Intelligent system for lighting control in smart cities. *Inf. Sci.* **2016**, *372*, 241–255.
8. Leduc, G. Road traffic data: Collection methods and applications. In *Working Papers on Energy, Transport and Climate Change*; JRC: Tokyo, Japan, 2008; Volume 1.
9. Wojnicki, I.; Ernst, S.; Kotulski, L. Economic Impact of Intelligent Dynamic Control in Urban Outdoor Lighting. *Energies* **2016**, *9*, 314, doi:10.3390/en9050314.
10. Kim, J.T.; Hwang, T. Feasibility Study on LED Street Lighting with Smart Dimming Systems in Wooi Stream, Seoul. *J. Asian Archit. Build. Eng.* **2017**, *16*, 425–430.
11. Kovács, A.; Bártai, R.; Csáji, B.C.; Dudás, P.; Háty, B.; Pedone, G.; Révész, T.; Váncza, J. Intelligent control for energy-positive street lighting. *Energy* **2016**, *114*, 40–51.
12. Wojnicki, I.; Szwed, P.; Chmiel, W.; Ernst, S. Ontology Oriented Storage, Retrieval and Interpretation for a Dynamic Map System. In *Multimedia Communications, Services and Security; Communications in Computer and Information Science*; Dziech, A., Czyżewski, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 287, pp. 380–391.

13. Mahdavi, A. Predictive simulation-based lighting and shading systems control in buildings. *Build. Simul.* **2008**, *1*, 25–35.
14. Mady, A.; Boubekeur, M.; Provan, G.; Ryan, C.; Brown, K. *Intelligent Hybrid Control Model for Lighting Systems Using Constraint-Based Optimisation*; Advances in Intelligent and Soft Computing; Springer: Berlin/Heidelberg, Germany, 2010; Volume 73, pp. 249–259.
15. Doulos, L.; Tsangrassoulis, A.; Topalis, F.V. Multi-criteria decision analysis to select the optimum position and proper field of view of a photosensor. *Energy Convers. Manag.* **2014**, *86*, 1069–1077.
16. Atis, S.; Ekren, N. Development of an outdoor lighting control system using expert system. *Energy Build.* **2016**, *130*, 773–786.
17. Mahoor, M.; Salmasi, F.R.; Najafabadi, T.A. A Hierarchical Smart Street Lighting System With Brute-Force Energy Optimization. *IEEE Sens. J.* **2017**, *17*, 2871–2879.
18. Lau, S.P.; Merrett, G.V.; Weddell, A.S.; White, N.M. A traffic-aware street lighting scheme for Smart Cities using autonomous networked sensors. *Comput. Electr. Eng.* **2015**, *45*, 192–207.
19. Marino, F.; Leccese, F.; Pizzuti, S. Adaptive Street Lighting Predictive Control. *Energy Procedia* **2017**, *111*, 790–799.
20. Wojnicki, I.; Kotulski, L.; Sędziwy, A.; Ernst, S. Application of Distributed Graph Transformations to Automated Generation of Control Patterns for Intelligent Lighting Systems. *J. Comput. Sci.* **2017**, *23*, 20–30.
21. Flasiński, M. On the parsing of deterministic graph languages for syntactic pattern recognition. *Pattern Recognit.* **1993**, *26*, 1–16.
22. Drewes, F.; Hoffmann, B.; Plump, D. Hierarchical Graph Transformation. *J. Comput. Syst. Sci.* **2002**, *64*, 249–283.
23. Taentzer, G. Hierarchically distributed graph transformation. In *Graph Grammars and Their Application to Computer Science*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1994; pp. 304–320.
24. Wojnicki, I.; Kotulski, L. Street Lighting Control, Energy Consumption Optimization. In Proceedings of the 16th International Conference Artificial Intelligence and Soft Computing, ICAISC 2017, Zakopane, Poland, 11–15 June 2017; Lecture Notes in Computer Science; Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M., Eds.; Springer: New York, NY, USA, 2017; Volume 10246, pp. 357–364.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).