A space model for personal lighting control

Georg Suter, PhD, Associate Professor
Filip Petrushevski, Dipl.-Ing., Researcher
Milos Sipetic, Dipl.-Ing., Researcher
Design Computing Group, Vienna University of Technology
georg.suter@tuwien.ac.at

ABSTRACT:

We describe a space model which supports spatial data needs of a personal lighting control (PLC) system. The potential benefits of such a system include improved individual control of the visual environment, user satisfaction, and energy effectiveness. In our system, lighting agents dynamically query the space model for luminaires and windows that surround a given user location. Agents use that data to adjust luminaire outputs and window shades with respect to user preferences. We introduce a user scenario and the architecture of a PLC system prototype which we have developed. Spatial data requirements of lighting agents are described next. These depend on the lighting agent's mode. Supported modes include artificial lighting, artificial and natural lighting, and artificial and natural lighting and shading. The space model consists of a lighting layout. Layout elements include whole spaces, partial spaces or subspaces, and space elements that are relevant for lighting, such as luminaires, windows, and furnishing elements. Selected spatial relations between layout elements are modeled explicitly as a geometric network. The network supports efficient graph queries, such as nearest neighbor or shortest path queries. In our system, good query performance is necessary to ensure short system response times. The network is derived automatically from space and space element data defined in a commercial BIM system for architectural design. Query requests of lighting agents are processed by a space model server. The server decomposes a query request into a subquery sequence. Each subquery involves tracing of the lighting layout's network, or a subnetwork extracted from it. Subqueries are illustrated with the example of a test space where we have deployed our system prototype. We describe the implementation of the lighting agent and the space model server and conclude with a discussion of open issues.

KEYWORDS: *Building information modeling, Space models, Building automation systems*

1. INTRODUCTION

Personal control of indoor lighting is increasingly feasible due to advances mobile computing and building automation systems. Studies suggest that personal control of luminaires improves user satisfaction and, possibly, productivity (see, for example, Newsham 2004). It may further mitigate the over-illumination problem in facilities, which adversely affects human health and energy use (see, for example, Galasiu 2009). Existing personal lighting control (PLC) systems are typically deployed in offices. Users may adjust dimming levels of luminaires that are near their workplaces via remote control devices or GUIs on PCs or smartphones (see, for example, Shin 2009, Krioukov 2011). However, in order to minimize system complexity and maintenance, luminaires tend to be aggregated into large, static (that is, hard-wired and/or hard-coded) control groups. This may negatively affect the effectiveness of PLC systems. First, existing systems are not flexible enough to respond to changing user needs as control of lighting conditions is limited by static control groups. For example, when a user moves to a different location, existing systems must typically be reconfigured. Second, the need for shared control of luminaires by multiple users is more likely in large control groups than in small ones. Third, large control groups may result in over-illumination.

To address these issues, we have designed and implemented a prototype of a PLC system (Suter et al. 2011). The system's core modules are:

- a lighting control agent (LCA) which adjusts dynamically derived luminaire control groups based on lighting preferences, location and illuminance data retrieved from a user's smartphone,
- a space model which addresses LCAs' spatial data needs, and
- a space model server (SMS) which provides a query interface for LCAs and supports high concurrency and fast response times.

The focus of this paper is on the description of the space model structure and query processing of SMS for the envisioned PLC system. In section 2, we give an overview of a system prototype which we have designed and implemented. Spatial data needs of LCA are identified in section 3. The description of the space model structure in section 4 is divided into two parts. Space model concepts which we have developed in previous work are reviewed, and the space model for PLC is introduced. In section 5, we describe and illustrate the processing of LCA query requests by SMS. LCA and SMS implementations are discussed in Section 6. Open issues are identified in Section 7.

2. PERSONAL LIGHTING CONTROL SYSTEM

2.1 User scenario

The concept of personal lighting control, as we envision it, is illustrated with a user scenario (Suter et al. 2011).

Start situation: User N. enters his office at 7 a.m. on a winter day. Ceiling luminaires in the office are turned on to provide basic visual comfort. No other users are present yet in the office, where three workplaces are shared on a first-come-first-serve basis. A PLC application is installed on N.'s smartphone.

Flow of events: N. approaches an available workplace and puts his smartphone face up on the work desk. The smartphone detects N.'s location. The PLC application is triggered. The application uses N.'s location, his lighting preferences, and data from an illuminance sensor on the smartphone to adjust output levels of luminaires that surround N.'s desk. While the desk luminaire and ceiling luminaires that are near the desk are adjusted to high output, distant ceiling luminaires are adjusted to low output as they have less influence on illumination of the work area. Luminaire outputs are set according to N.'s preferences for computer work, which is the assumed default activity. Resulting desk illuminance is 495 lx, which is within N.'s preferred illuminance interval of 480 – 520 lx for computer work.

End situation: N. sets up his laptop and starts working. Lighting conditions correspond to his preferences for computer work.

2.2 System architecture

We have developed requirements for a PLC system prototype from the scenario. The system adopts a client/server architecture (Figure 1, Suter et al. 2011). Compared with a service-oriented architecture, the benefit of a client/server architecture is its relative ease of implementation. On the client side (that is, on the user's smartphone), the main module is the Personal Lighting Manager (PLM), which reads illuminance and symbolic location data from, respectively, Illuminance Sensor (IS) and Near Field Communication (NFC) modules. NFC is chosen for symbolic location sensing because the technology is increasingly supported on smartphones (Juniper 2011). NFC tags are mounted on users' desks.

Both IS and NFC process sensor data streams and generate events due to changes in illuminance levels or location. PLM receives these events, interprets them and, if necessary, sends sensor data together with the user's lighting preferences to the Lighting Control Agent (LCA) module. Presently implemented preferences include illuminance levels and size of the region managed by LCA. In a large, open plan office spaces, for example, regions would typically be smaller than these spaces. IS sends updates only when there are significant changes in lighting conditions. LCA is responsible for translating lighting requests into luminaire and window shade control actions. It interfaces with the lighting automation system (LAS) module to execute these actions. LCA queries the space model server (SMS) module to obtain luminaires and windows that are near the user location. LCA uses a closed-loop control algorithm to iteratively adjust luminaire outputs until a targeted illuminance at the user's location is reached. Moreover, it deploys window shades if necessary. At present, LCA deals only with single user situations.

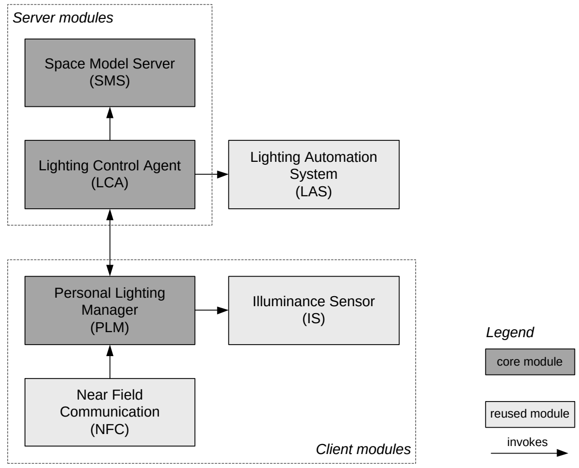

Fig. 1: Architecture of the personal lighting control system.

3. SPATIAL DATA REQUIREMENTS

LCA operates in three modes which determine what luminaire and window data it needs from the space model. The first mode is artificial lighting. In this mode, LCA needs luminaires that are near a user location. Luminaires and the user location must be contained in the same space. Each luminaire must include its distance (according to a distance measure) relative to the luminaire that is nearest to the user's location. LCA computes luminaire output based on that distance. Luminaires that are near the user location have higher output than distant ones. The maximum distance (that is, the size of the region managed by LCA) is defined in user preferences.

The second LCA mode is artificial and natural lighting under diffuse sky conditions. In this mode, natural light is used to illuminate a space. Natural light is complemented by artificial light if necessary to achieve minimum illuminances. Similar to artificial lighting mode, LCA needs luminaires that are near a user location. In addition to the distances of luminaires to the luminaire that is nearest to the user location, distances of luminaires to the nearest window must be returned as well. LCA considers both distances when it computes a luminaire's output. Assuming that they have the same distance to the user location, the output of a luminaire that is near windows is lower than the output of a distant luminaire.

The third LCA mode is artificial and natural lighting and shading. This mode is similar to artificial and natural lighting insofar as natural lighting is complemented by artificial lighting, if necessary. In addition to luminaires that are near a user location, LCA needs to know about nearby windows with a line-of-sight to the sun. Buildings, trees, or clouds may obstruct line-of-sight. LCA shades these windows. A shade may be either open or closed. Distances of luminaires with respect to nearest non-shaded windows must be returned together with distances to the luminaire that is nearest to a user location.

4. SPACE MODEL STRUCTURE

4.1 Overview

The space model for PLC consists of a network-based space layout. The layout supports high spatial granularity, rich spatial relations, and efficient local queries. Concepts of network-based space layouts are reviewed first. A layout that specifically meets spatial data needs for PLC is described subsequently.

4.2 Space layout concepts

Network-based space layouts incorporate aspects of existing architectural space models (Bjoerk, 1992; Eastman and Siabiris, 1995; Ekholm, 2000; BuildingSmart, 2010). For a detailed description, the reader is referred to Suter (2010a and 2010b). A layout includes spaces and space elements that model a particular spatial view of a building. For example, a set of layouts may model a building's circulation system, including wings, floors, circulation spaces, doors, and furnishing elements. Spaces in other layouts may model heating, ventilation, and air conditioning zones. Space elements in these layouts would include air inlets and outlets as well as temperature sensors. Selected spatial relations between layout elements are modeled explicitly as a geometric network for efficient local querying with graph algorithms.

Figure 2 shows four partial views of an example network-based space layout. Each view features different spatial relations between layout elements. There are four types of layout elements: whole spaces, subspaces, space boundary elements, and space elements. A whole space is a space which is bounded on all sides by space boundary elements. A space boundary element is part of an immaterial layer with zero thickness that bounds a whole space. A partial space or subspace is a space which is contained in a whole space. It may or may not be bounded on all sides by space boundary elements and may surround space elements. Different types of subspace volumes are supported. In the example layout in Figure 2, subspace volumes correspond to geodesic Voronoi cells (Aurenhammer and Klein, 2001) that are derived from whole space boundaries (used as obstacles) and subspace positions (used as sites). Other types of subspaces such as spherical subspaces are supported as well. Space elements are (physical) objects, including windows, desks, or luminaires that are either contained in or enclose a whole space. Space elements have attributes that indicate if they are contained in or enclose a whole space. A desk is a space element that is contained in a whole space, and a door one that encloses a whole space. This distinction of space elements matters because they participate in different spatial relations. A layout has a layout element network, which is a directed, weighted graph with layout elements as nodes and spatial relation elements as edges. Layout and spatial relation elements have weights, which facilitates layout queries with graph algorithms.

Support for the generation of space layouts is desirable to minimize manual modeling effort. In particular, the generation of a layout element network and subspaces are time consuming and error-prone tasks. A conversion routine is described in Suter (2010a) which automatically derives the layout element network and subspaces. The routine evaluates spatial constraints on layout elements and spatial relations to identify and resolve spatial inconsistencies. Figure 3 illustrates the conversion routine with the floor layout of an existing office building in Vienna where our test space for validating the PLC system is located. For improved visualization, the figure only includes a subset of spatial relations in the layout element network. Rooms, luminaires, and desks are modeled as, respectively, whole spaces and space elements in a BIM system for architectural design (Autodesk 2011). The geometric network and subspaces are derived using a solid modeling engine (Spatial 2011). Subspaces are derived from luminaire and desk templates in which default subspaces are defined. Subspace volumes are derived from two-dimensional Voronoi diagrams (Aurenhammer 2000). These volumes are not shown in Figure 3.

4.3 Lighting layout

The space layout for PLC models elements and spatial relations that are relevant for artificial and natural lighting. On the one hand, it models physical elements, that is, luminaires, windows, and desks. On the other hand, it includes related whole spaces and subspaces. The adjacency relation between subspaces (A_{SS}) and the surround relation between subspaces and space elements ($S_{SS,SE}$) constitute a layout element network together with subspaces, luminaires, windows, and desks. The layout in Figure 3 shows these relations. The corner office, which is enlarged in Figure 3, is our test space. The office includes five desks and six luminaires that are arranged in an irregular 2x3 grid. For simplicity, task lights, chairs, cabinets, and doors are not included in the layout. Five windows admit natural light from two sides into the office. Each window may be shaded individually. The lighting layout also includes an external whole space, which models a building’s site. This whole space is useful to determine which windows are exposed to direct sunlight (Section 5.6). For clarity, it is not shown in Figure 3.

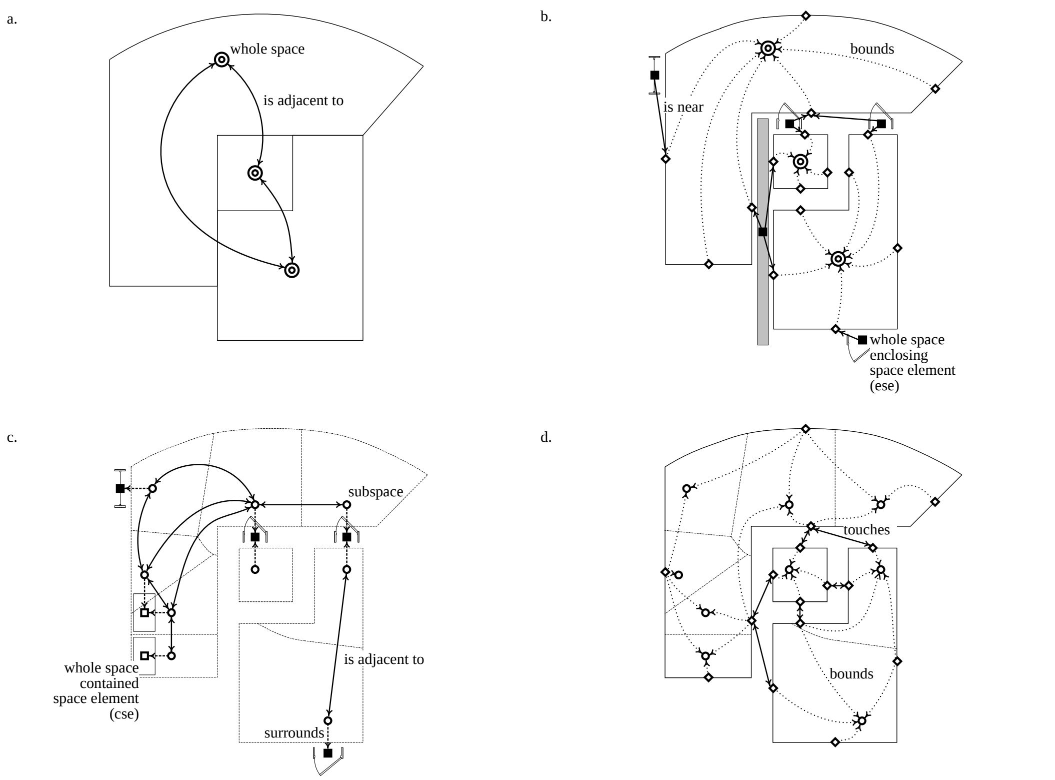

Fig. 2: Layout elements and spatial relations in a network-based space layout. a. Adjacency relation between whole spaces. b. Boundary relation between space boundary elements and whole spaces, proximity relation between space elements and space boundary elements. c. Adjacency relation between subspaces, surround relation between subspaces and space elements. d. Boundary relation between space boundary elements and subspaces, touch relation between space boundary elements.

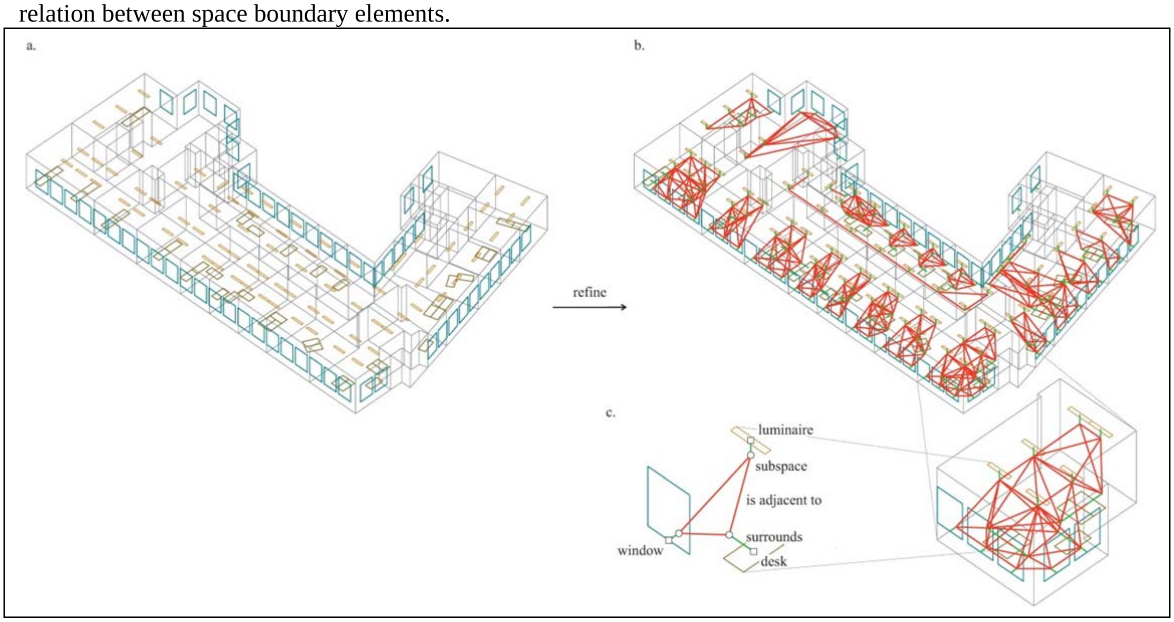

Fig. 3: Derivation of a layout element network and subspaces. a. Whole spaces and space elements created in a BIM system for architectural design. b. Converted layout with automatically derived layout element network and subspaces.

5. QUERY PROCESSING

5.1 Overview

LCA queries the lighting layout for the set of luminaires and windows that are near a given user location. LCA sends high-level query requests to SMS to minimize data communication between these modules. A query request is divided by SMS into a sequence of sub-queries. An LCA query request has two parameters. The first parameter is the lighting mode, which is either artificial lighting (AL), artificial and natural lighting (AL-NL), or artificial and natural lighting and shading (AL-NL-S). The second parameter is the user location, which corresponds to the unique identifier of the desk where a user has checked in (Section 2.1). Figure 4 shows how SMS processes LCA query requests. Most subqueries use graph algorithms to trace the lighting layout's layout element network or a subnetwork retrieved from it. Graph algorithms include connected component, nearest neighbor, and shortest path queries (Bondy and Murty 2010). In the following, each subquery is described in more detail.

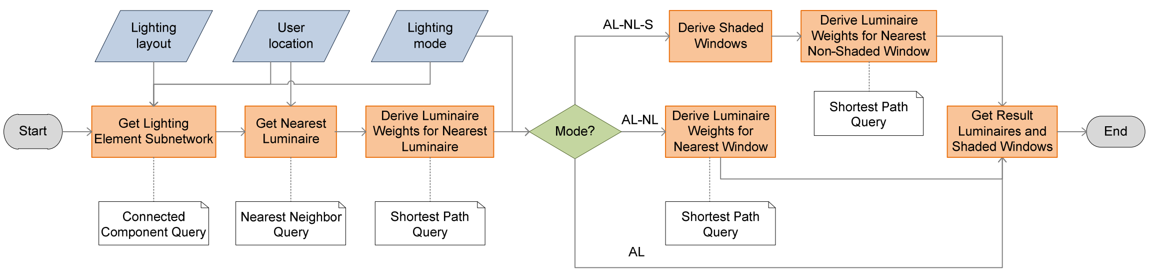

Fig. 4: Processing of an LCA query request by SMS for artificial lighting (AL), artificial and natural lighting (AL-NL), and artificial and natural lighting and shading (AL-NL-S) modes.

5.2 Get lighting element subnetwork

The first subquery in the processing of a LCA query request retrieves a connected component of the lighting layout's layout element network. In AL mode, a connected component consists of desks, luminaires and A_{SS} and $S_{SS,SE}$ relations in the whole space that contains the user location (that is, the user's desk). The latter is the start node in the connected component query. In order to limit tracing to the whole space that contains the user location, a filter predicate is used that designates windows as barriers (note that walls and doors are not modeled in the lighting layout). Without windows acting as barriers, the connected component query would proceed to external subspaces (not shown in Figure 3), which are modeled explicitly in the lighting layout to support the derivation of shaded windows (Section 5.6). Barrier elements may or may not be included in the result network of a connected component query. In AL-NL and AL-NL-S modes, windows are included, whereas they are excluded in AL mode. The network of the corner office in Figure 3 is an example of the subnetworks that are retrieved from a lighting layout in AL-NL and AL-NL-S modes.

5.3 Get nearest luminaire

The luminaire that is nearest to a given user location is determined by a nearest neighbor query. With the user location as a start node, the lighting element subnetwork obtained in the previous subquery is traced. Path weight is used as a distance measure in this query. Path weight refers to the sum of edge weights in a path, where a weight corresponds to the Euclidean distance between incident nodes. Nearest neighbor candidates must meet a filter predicate. Multiple nearest luminaires are feasible. Figure 5 illustrates the retrieval of nearest luminaires for three user locations.

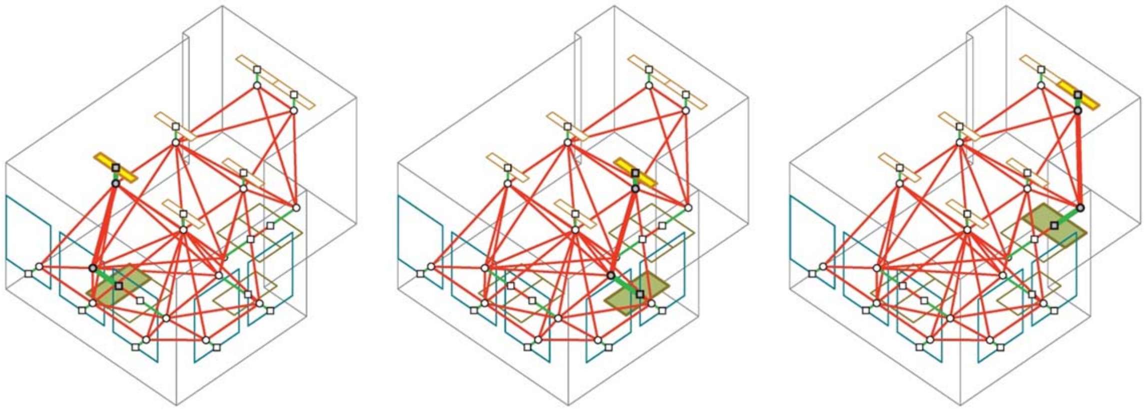

Fig. 5: Illustration of 'Get nearest luminaire' subquery.

5.4 Derive luminaire weights for nearest luminaire

According to the user scenario (Section 2.1), luminaire outputs are set relative to their distance to the user location. In this subquery, the distance of each (source) luminaire in the lighting element subnetwork with respect to the (target) luminaire that is nearest to the user location is derived. This is done with a shortest path query. Figure 6 illustrates weight derivation of luminaires for three nearest luminaires. Path length (that is, the number of edges in the path) is used as distance measure to derive weights. Two filter predicates are used. The first filter selects source luminaires for which weights are derived. In each lighting mode, all luminaires in the lighting element subnetwork are selected as source luminaires. The second filter selects target luminaires, which corresponds to the luminaire that is nearest to the user location.

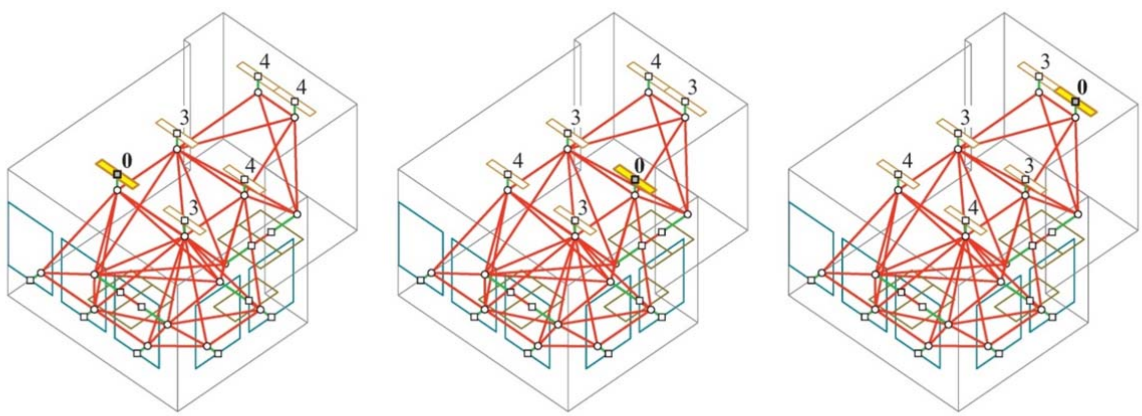

Fig. 6: Illustration of 'Derive luminaire weights for nearest luminaire' subquery.

5.5 Derive luminaire weights for nearest window

In AL-NL mode, output levels of luminaires that are near windows are reduced more than output levels of distant luminaires. Respective luminaire weights relative to nearest windows are derived in this subquery. Figure 7a illustrates the subquery. Again, path length is used as a distance measure. All luminaires in the lighting element subnetwork are selected as sources for which weights are derived, and all windows are selected as targets.

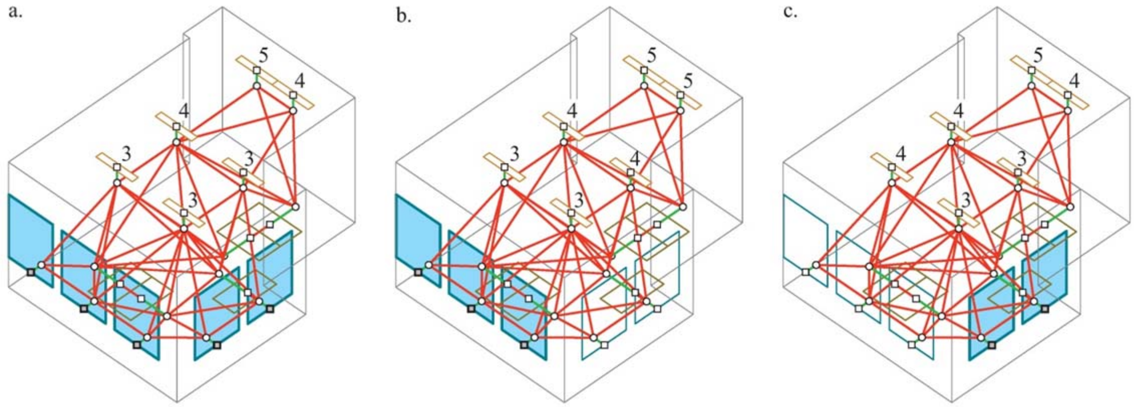

Fig. 7: Illustration of 'Derive luminaire weights for nearest (non-shaded) window' subquery. a. All windows admit natural light. b. Two shaded windows. c. Three shaded windows.

5.6 Derive shaded windows

Shaded windows need to be derived in AL-NL-S mode. For that purpose, the lighting layout includes an external whole space which models a building's site. The volume of the external whole space is derived from the volumes of the building in question as well as nearby buildings and other obstructing objects, such as trees. Each window that encloses an internal and the external whole space (and is thus part of the building facade) is surrounded by two subspaces that are contained in, respectively, an internal and the external whole space. The sun's position at a given time is modeled as a solar subspace in the external whole space. A window is exposed to direct sunlight:

1. if it is surrounded by an external subspace that is adjacent to the solar subspace (that is, an A_{SS} edge connects the two subspaces in the layout element network of the lighting layout), and
2. if there is a line-of-sight between the two subspaces.

An A_{SS} edge has a Boolean attribute that indicates if line-of-sight exists between its subspaces. Attribute values are derived during the layout conversion routine (Section 4.2). It is thus feasible for SMS to derive the set of shaded windows by tracing the lighting layout's layout element network. Figure 8 illustrates the derivation of shaded windows for a floor in the example building. A winter day with low solar elevation angles is assumed. In the morning, windows on the facade facing Southeast receive direct sunlight. At noon, this is true for windows on Southeast and Southwest facades, and in the afternoon for windows on the Southwest facade. Windows facing the courtyard do not receive direct sunlight on that day.

A benefit of the described method to derive shaded windows is that SMS does not need to perform computation intensive spatial reasoning during query processing. However, we have not yet resolved the issue of how to best model dynamically changing solar subspaces and incident A_{SS} edges. One approach is to add timestamps to solar subspaces. Queries could use a timestamp filter to identify the solar subspace and incident A_{SS} edges that match a given timestamp. Another unresolved issue is how to consider indirect sunlight reflecting e.g. from nearby buildings.

5.7 Derive luminaire weights for nearest non-shaded window

When shaded and non-shaded windows are known in AL-NL-S mode, the next subquery derives luminaire weights with respect to non-shaded windows. This subquery is similar to the derivation of luminaire weights with respect to all windows in AL-NL mode (Section 5.5). The only difference concerns the selection of non-shaded windows as destination nodes. Figures 7b and 7c illustrate the effects of shaded windows on luminaire weights. When all windows are shaded, then the space does not receive natural light and this subquery is skipped.

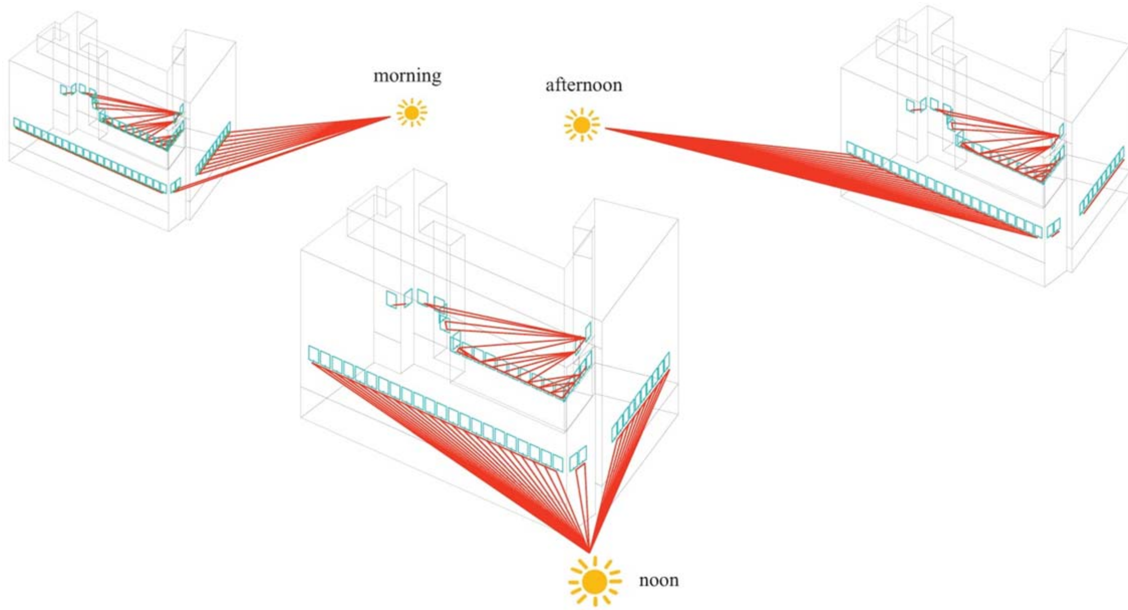

Fig. 8: Illustration of the 'Derive shaded windows' subquery.

5.8 Get result luminaires and shaded windows

The last subquery in the processing of an LCA query request involves the retrieval of luminaire and shaded window result sets from the lighting element subnetwork. Each luminaire in the result set includes a weight for the distance to the luminaire that is nearest to the user location, and a weight for the distance to the nearest non-shaded window. In AL and AL-NL modes, the shaded window result set is always empty.

6. IMPLEMENTATION

We have implemented a prototype of the described PLC system. The prototype has been deployed in our test space and supports the user scenario (Section 2.1). SMS processes LCA query requests by accessing a lighting layout of the office building in which the test space is located (Figure 3). At present, the system prototype supports AL mode. The lighting layout in SMS is implemented using Boost Graph Library (BGL, Boost 2011). BGL is a compact, efficient, and flexible graph framework that lets clients customize graph representation methods and algorithms. Communication between LCA and SMS is implemented using the Boost.Asio, which is a C++ library for asynchronous networking and low level input/output programming (Kohlhoff 2008). Asynchronous network programming supports a large number of concurrent network connections.

7. CONCLUSION

We are currently testing and evaluating the prototype system. Lighting quality and response times for the whole system and SMS will be measured in the test space. Load and stress tests will be performed in virtual environments. A direction for future work concerns the development and evaluation of alternative luminaire weight derivation methods. For example, the weight of a luminaire with respect to windows may be derived by addition of weights or lengths of shortest paths from the luminaire to each window. Resulting weights would better approximate natural lighting distribution in spaces than luminaire weights in Figure 7.

ACKNOWLEDGEMENT

The work presented in this paper is supported by grant Austrian Science Fund (FWF): P22208.

REFERENCES

Aurenhammer, F. & Klein, R. (2000), Voronoi diagrams, *in* J. Sack & G. Urrutia, ed., 'Handbook of Computational Geometry, Chapter V', Elsevier Science Publishing, , pp. 201--290.

Autodesk (2011), 'AutoCAD Architecture 2012 user's guide'.

Bjoerk, B.-C. (1992), 'A conceptual model of spaces, space boundaries and enclosing structures', *Automation in Construction* **1**(1), 193-214.

Bondy, A. & Murty, U. (2010), *Graph Theory*, Springer.

Boost (2011), 'Boost Graph Library', Boost, Technical report, Boost.

BuildingSmart (2010), 'Industry Foundation Classes IFC2x4', BuildingSmart, Technical report, BuildingSmart.

Eastman, C. & Siabiris, A. (1995), 'A generic building product model incorporating building type information', *Automation in Construction* **3 (1)**, 283-304.

Ekholm, A. & Fridqvist, S. (2000), 'A concept of space for building classification, product modelling and design', *Automation in Construction* **9 (3)**, 315-328.

Galasiu, A. D. & Newsham, G. R. (2009), Energy savings due to occupancy sensors and personal controls: a pilot field study, *in* 'Proceedings of Lux Europa'.

Hillier, B. & Hanson, J. (1989), *The Social Logic of Space*, Cambridge University Press.

ISO (2004), 'Standard for the exchange of product model data (ISO 10303)'(ISO 10303), TC184/SC4, Technical report, ISO.

ISO/IEC (2008), 'SQL:2008 (ISO/IEC 9075)'(ISO/IEC 9075), Technical report, ISO/IEC.

Kohlhoff, C. M. (2008), 'Boost.Asio Library', Boost, Technical report, Boost.

Krioukov, A.; Dawson-Haggerty, S.; Lee, L.; Rehmane, O. & Culler, D. (2011), 'A Living Laboratory Study in Personalized Automated Lighting Controls''BuildSys: 3rd ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings, Seattle WA'.

Liggett, R. S. (2000), 'Automated facilities layout: past, present and future', *Automation in Construction* **9**(2), 197 - 215.

Newsham, G.; Veitch, J.; Arsenault, C. & Duval, C. (2004), Effect of dimming control on office worker satisfaction and performance, *in* 'IESNA Annual Conference Proceedings, Tampa, Florida', pp. 19-41.

Shin, C. & Woo, W. (2009), 'Service Conflict Management Framework for Multi-user Inhabited Smart Home', *Journal of Universal Computer Science* **15**(12), 2330-2352.

Spatial (2011), '3D ACIS Modeler'.

Suter, G. (2010a), Outline of a schema for network-based space layouts, *in* K. Menzel & R. Scherer, ed., '8th European Conference on Product and Process Modelling, European Group for Intelligent Computing in Engineering (EG-ICE)', Taylor & Francis, Cork, Ireland.

Suter, G. (2010b), Topological constraints for consistency checking of network-based space layouts, *in* W. Thabet, ed., '27th International Conference on Applications of IT in the AEC Industry'.

Suter, G.; Petrushevski, F. & Sipetic, M. (2011), Design of a personalized lighting control system enabled by a space model, *in* Jeff Haberl, ed., '11th International Conference for Enhanced Building Operations'.