Advanced Engineering Informatics 28 (2014) 395–411

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

# Operations on network-based space layouts for modeling multiple space views of buildings ☆

Georg Suter *, Filip Petrushevski, Miloš Šipetić

*Design Computing Group, Faculty of Architecture and Planning, Vienna University of Technology, Treitlstrasse 3, A-1040 Vienna, Austria*

ARTICLE INFO

*Article history:*
Received 20 October 2012
Received in revised form 14 February 2014
Accepted 10 June 2014
Available online 21 August 2014

*Keywords:*
Space modeling
Building information modeling
Spatial query languages

ABSTRACT

Space layouts are created by designers to model a building's spaces and related physical objects. Building services designers commonly reuse space layouts created by architectural designers to develop their designs. However, reuse tends to be limited due to differences in designers' space views. In order to address this issue of modeling multiple space views, we define a set of novel operations that can be used by designers to generate new space layouts from existing layouts. Fundamental operations include *select*, *aggregate*, and *decompose* operations. The *select* operation facilitates reuse of space layouts created in building information modeling (BIM) authoring systems. Signatures and processing of these operations are defined. We use an existing schema for network-based space layouts to represent space layouts. In a network-based space layout, specific spatial relations between layout elements are explicitly modeled as a directed, weighted graph or network. Processing of certain operations involves traversal of a spatial relation network with graph algorithms to determine layout modifications. Symmetric difference and *overlay* operations are defined as additional operations. They are composed of *union*, *intersect*, and *subtract* operations, which are fundamental operations. Fundamental and additional layout operations may be composed into expressions to model domain-specific space views. We have extended an existing layout modeling system with implementations of these layout operations. The system relies on geometric and solid modeling as well as graph libraries to represent layouts and process operations. The feasibility of modeling of multiple space views with layout operation expressions is shown with an example in which a security lighting layout of a floor of an existing office building is automatically generated from an architectural layout.



## 1. Introduction

Space layouts are created by designers to model a building's spaces and related physical objects, such as walls, windows, furnishing and technical equipment elements. Designers increasingly use building information modeling (BIM) authoring systems to develop space layouts [1]. Sharing of space layouts is facilitated by data exchange methods. The space schema in the Industry Foundation Classes (IFC), for example, defines rich data structures for spaces, space boundaries, and space enclosures [2].

It is common practice for building services designers to reuse space layouts created by architectural designers to develop their designs. However, reuse tends to be limited due to substantial differences in designers' space views. This issue of multiple, domain-specific views of objects is well-known in BIM [3,4]. For an architectural designer, a space typically includes space boundaries, windows, doors, as well as visually relevant furnishing and equipment elements. By contrast, in the space view adopted by an indoor climate control system designer, spaces are often zones that span multiple rooms with similar orientation, function, or thermal requirements. Space enclosure elements with high thermal flux, air inlets/outlets, or indoor air temperature sensors are of concern. A lighting designer in turn is usually interested in zones within rooms that are related to luminaires, shades, walls, windows, illuminance sensors, or occupancy sensors.

While BIM authoring systems and IFC provide general space data modeling and exchange capabilities, support for automated generation of space layouts that model multiple, domain-specific space views, as outlined above, is limited. Building services designers may thus only partially reuse space layouts created by architectural designers. As a workaround, they need to manually recreate space layouts according to their space views. This existing workflow is inefficient, error-prone, and a significant barrier for

☆ Handled by W.O. O'Brien.

* Corresponding author. Tel.: +43 1 58801 27220; fax: +43 1 58801 27299.
*E-mail addresses:* georg.suter@tuwien.ac.at (G. Suter), filip.petrushevski@tuwien.ac.at (F. Petrushevski), milos.sipetic@tuwien.ac.at (M. Šipetić).

http://dx.doi.org/10.1016/j.aei.2014.06.004
1474-0346/

**Nomenclature**

| | | | |
|---|---|---|---|
| $A$ | "*is adjacent to*" relation | $se_{c\_ws}$ | space element contained in a whole space |
| $le$ | layout element | $se_{pe\_ws}$ | space element partially enclosing a whole space |
| $N$ | "*is near*" relation | $sr$ | spatial relation element |
| $O$ | "*overlaps*" relation | $SRN$ | spatial relation network |
| $PB$ | "*partially bounds*" relation | $ss$ | subspace |
| $PE$ | "*partially encloses*" relation | $T$ | "*touches*" relation |
| $sbe$ | space boundary element | $ws$ | whole space |
| $se$ | space element | | |

seamless collaboration in multi-disciplinary design teams. According to [5], reduced errors, shorter cycle times of workflows, and reduced rework are viewed as highly important for improved quality and productivity of design firms in architecture, engineering, and construction.

In order to address the issue of modeling multiple space views, we describe a set of novel operations that can be used by designers to generate new space layouts from existing layouts. We build on and extend previous work, in which we have developed initial specifications for these operations [6,7]. Layout operations may be composed into expressions to model domain-specific space views. For example, a layout operation expression may be defined to generate an indoor climate control layout from an architectural layout. We use an existing schema for network-based space to represent space layouts [8]. In a network-based space layout, spatial relations between layout elements are explicitly modeled as a directed, weighted graph or network. Processing of certain operations involves tracing of this spatial relation network with graph algorithms to determine layout modifications.

The remainder of this paper is organized as follows. A survey of related work is included in Section 2. Network-based space layout concepts are reviewed in Section 3. Fundamental layout operations are defined in Section 4. Additional layout operations that are compositions of fundamental operations are introduced in Section 5. The implementation of fundamental and additional layout operations in an existing layout modeling system is described in Section 6. Section 7 presents an example of a layout operation expression. Open issues and future work are discussed in Section 8.

## 2. Related work

Related work is reviewed, including IFC model view definitions, modeling and query languages, and ontology-based reasoning. We highlight aspects that deal with spatial modeling.

### 2.1. IFC model view definitions

An IFC model view definition is a subset of the original IFC schema that is relevant for specific applications and life-cycle phases [9]. The IFC Coordination View defines a view for architectural, mechanical, and structural design coordination [10]. It includes *IfcSpace*, *IfcSpatialStructureElement*, and *IfcZone* classes. A building may be hierarchically divided into *IfcSpatialStructureElement* levels, such as sections, floors and rooms. *IfcSpace*s and building elements may be attached to these levels. Alternatively, *IfcSpace*s may be aggregated into *IfcZone*s. In contrast to *IfcSpatialStructureElement*s, the latter do not have a shape. Their structure does not need to be hierarchical, that is, an *IfcSpace* may belong to one or more *IfcZone*s.

Weise et al. [11] describe a schema for the definition of arbitrary IFC model views. A data subset is retrieved in two steps from an IFC model instance. In the first step, objects are selected dynamically from the original model by filtering based on criteria supplied by users at runtime. An example is the selection of objects on a particular floor. In the second step, attribute data or references to objects that are not needed are removed based on a pre-defined, static model view.

### 2.2. Modeling languages

GLIDE and EDM are early examples of modeling languages [12,13]. Eastman and Siabiris [14] use composition and specialization in EDM to model views of constructed spaces and use spaces. Constraints verify if required relations between model entities are satisfied. An example is a constraint that checks for correct alignment of faces of constructed space and use space polyhedrons.

Zamanian et al. [15] describe a framework for modeling spatial abstractions of buildings. Decomposition, selection, and composition operations are applied to spatial configurations that are based on a spatial representation scheme. In this scheme, disjoint geometric elements (vertices, faces, solids) are generated by user-defined carrier elements (points, lines, planes). Domain-specific spatial views are defined by selection and composition operations and generated in three steps. First, Euclidean space is partitioned recursively until the desired level of granularity in a spatial configuration is reached. Second, atomic elements in the configuration are selected based on user-defined criteria. Third, selected atomic elements are composed to create new elements. The composition operation ensures that new elements are disjoint.

Stouffs et al. [16,17] define the SORTS language for manipulating nested property structures. Although the language is not domain-specific, its development is motivated by the need for representational flexibility in building design. Property structures are composed with addition and sum operations. Furthermore, they may be combined using addition, subtract, and product operations. Primitive data constructs have specific, built-in behaviors under these operations. For example, the addition of two properties that model boundaries of solids corresponds to a merge (union) of these solids.

Grabska et al. [18–20] describe a visual design language that uses attributed, hierarchical hypergraphs to represent designs. Designers create hypergraphs representations of space layouts, for example, with operations that include hypergraph development and suppression. These operations add or remove levels of detail in a spatial hierarchy. Szuba [21] combines the hypergraph representation with a graph rewriting system to reason about spatial requirements. Activity requirements that are modeled as paths are matched against access graphs that are extracted from a space layout created in a BIM authoring system. This matching is nontrivial as it considers indirect access relations between rooms via circulation spaces.

The BERA language developed by Lee [22] provides high-level constructs to check building models against pedestrian circulation or space requirements. Users define simplified but extensible