A Parametric 3d City Model: Basis for Decision Support in Inner-City Development

Nils Seifert¹, Michael Mühlhaus², and Frank Petzold³

1) Dipl.-Ing., Chair for Architectural Informatics, Department of Architecture, Technische Universität München, Munich, Germany. Email: nils.seifert@tum.de
2) Dipl.-Ing., Chair for Architectural Informatics, Department of Architecture, Technische Universität München, Munich, Germany
3) Dr.-Ing., Prof., Chair for Architectural Informatics, Department of Architecture, Technische Universität München, Munich, Germany

Abstract:
Semantic 3D city models provide an essential basis for the visualization and communication of planning measures, e.g. of densification strategies in inner-city areas. However, it appears that the simulation of amendments in the building codes and their impact on the built environment cannot be modeled efficiently in common city model schemata. In the paper, a parametric 3d city model is presented, in which the building representations do not consist of static geometric objects, but also depict the generic geometric relationships, in terms of the semantic and geometric constraints of the modeling of different building typologies and their topological interdependencies. Based on the concept of Petri nets, the parametric 3d city model has been prototypically implemented and evaluated on the basis of inner-city planning tools.

Keywords: urban planning, decision support, 3d city model, parameterization, Petri net

1. INTRODUCTION

Major cities and congested urban areas worldwide are recording large population growth (United Nations 2014). Continued high demand in the residential market coupled with rising per capita consumption of living space pose great challenges for cities and municipalities. Particularly large cities show a blatant housing shortage (Henger et al. 2015). Many cities are already exploiting their current building laws to the full and are now under pressure to amend building legislation to meet the increasing demand for affordable living space. To counter increasing demand for inner-city living space and additional social polarization due to rising living costs, they need to recognize and use the current potential for densification of inner-city residential and mixed living and commercial areas. In addition, long-term, new building legislation is necessary. It requires development of ideas and strategies across the entire city with the support of a broad segment of the population to avoid uncontrolled growth while simultaneously creating and preserving the identities of urban areas (Fink et al. 2011).

The social acceptability of potential planning measures during densification of inner-city neighborhoods is highly dependent on discussion of different planning alternatives and communication of their effects. Answering the question of what the consequences of densification measures would be on residents, particularly in terms of the quality of life, requires in-depth knowledge of planning strategies and the effect thereof, as well as in terms of citywide and contextual district factors. It is therefore essential to support political and planning discussions with objective criteria and reliable forecasts that precede any of the planning measures initiated by construction managers. Semantic 3D city models offer a valuable basis for this. A thorough evaluation of potential planning is made possible by the representation of spatial relationships and interactions, and by accessing analysis and simulation methods.

For some time, surveying authorities have been increasingly concentrating on developing a comprehensive 3D city model. Many cities and municipalities have already been converted to a digital data model, which combines information from cadastral maps and real estate books, and provides a uniform, basic dataset as well as uniform interfaces for data exchange. In recent years, CityGML has developed into the standard for storage and exchange of 3D city simulations, which in addition to geometry of urban and rural objects at different levels of detail (LOD), also accounts for relevant semantic data for urban planning, such as building usage type or their construction year (Kolbe 2009). These days, city models play a central role in qualifying urban planning primarily through their use as mediums for information and communication. In addition, research has found that approaches such as city modelling could be increasingly used to simulate, for example, social, infrastructural and climatic aspects (Biljecki et al. 2015). The purpose of this simulations is to include results as an evaluation of possible measures directly in planning processes and thus support political and participatory decision-making processes. Furthermore 3d city models enable the development of new forms of interaction e.g. for design support in urban architectural design (Schubert 2014).
Although information storage in urban planning is increasingly becoming part of semantic 3D city models, typical urban planning schemes cannot efficiently simulate building rules and their impact on the built environment. As an example, the following scenario will be used: The building authority would like to answer the question of how much floor space can be generated in a particular city district for a range of planning variations, where the regulated development limits for example for setbacks, eave heights and guideline building depth values are changed. Simulating such possible measures and their impacts on the urban structure requires a model in which the individual building representations not only comprise of static geometric elements (points, surfaces, solids), but also depict geometric generation correlations. In other words, they must include the semantic and geometric simulation dependencies of individual building types and their topological interdependencies.

2. CONCEPT OF THE PARAMETRIC 3D CITY MODEL

2.1 Typologies in architecture and urban planning

In architecture and urban planning abstracts/typologies refer to spatial structures that exist independently of actual shape decisions. They represent spatially and temporally typical solutions that are not attributable to the individual design concept, but rather on building culture conventions and traditions (Raith 2000). An example of neighborhoods strongly characterized by typologies are the block suburban areas that evolved in European cities in the second half of the 19th century. The basic element of the block structure is individual parcels built in a closed corner block building manner. By consistent perimeter development, mostly orthogonally arranged streets arise with different building densities, in clearly defined urban spaces. Apart from external structural relations, building typologies often imply the use of certain types of floor plans (for example lengthwise zoning of floor plans, succession of living spaces, adjacent rooms, developments and extensions). When looking at the neighborhoods today, in addition to perimeter block buildings with shapes from different eras, one can see that courtyard buildings in individual parcels have been pulled into the construction typology. The shape and design of the perimeter block areas were and largely are determined by two contradictory but mutually interacting forces: on the one hand, the demand for new work and living space as generating factors for development and densification of parcels; on the other hand, building legislation, which is heavily regulated and stipulates limits for the degree of development within the parcels. For historical reasons, therefore, this type of area differs from town to town, and from neighborhood to neighborhood, in terms of density levels (see. Figure 1).

2.2 Derivation of modeling correlations

Certain regional and building types show similarities, regardless of their use, not only in terms of structural and design features, but can also be attributed to common modelling correlations. For example, the simulation of a perimeter block, depending on the existing structure (parcelling of the block, thoroughfares) mainly follows fixed rules. These rules include consistent perimeter development, closed construction and phasing of building volumes and extensions of the borders to the inside of the block.

![Figure 1. Examples of European perimeter blocks with different grades of density](image-url)
A building simulation can be described by a model that in addition to these geometric generation rules, also includes on one hand geometric information and on the other dynamically changeable variables. For example, the first part contains the geo-referenced location of the corner of a building. The second part includes those values which one may like to change according to the runtime of the simulation. Should for example, one simulate the maximum height increase of a building in a city district by applying a specific configuration of building codes, it would be helpful if the buildings had a ‘height’ parameter, which the simulation could directly access and that these changes are reflected immediately in the model.

After fixed rules are identified for modeling perimeter block areas, the sequence of geometry creation can be derived (see Figure 3): Starting from a baseline (1), which represents the construction line, and in most cases the roadside boundary of a plot, an offset line (2) and the base (3) of the structure are generated by means of two extrusion directions and the ‘depth’ parameter. In a further step, the volume of building is generated by extruding this area by the ‘height’ parameter (4). In subsequent steps, the roof geometry is generated from other parameters (5; roof shape, - orientation, and height). This step-by-step running modeling process for the building can be presented as a dependency graph. It represents two essential ingredients of building modeling:

- The modeling structure: different geometric objects represent instances of each object type (e.g. line, surface, body). Object types imply the necessary functions for geometry creation and for calculation of key values.
- The process of modeling: references are used to set dependencies between geometrical objects and parameters (e.g. production of building volumes from plot areas and height).
In addition, geometric dependencies are conceivable between varieties of city model objects. So it may happen that a change in parameters of a building not only results in a change of state of the building object itself but also has a direct impact on the state of other objects, depending on the objects located at the structure. For example, a change in the 'depth' parameter of a corner block building has a direct impact on structurally connected extensions. When changing structure parameters here, ensure that any overlaps with other objects are avoided. Interdependencies between different building structures can be mapped by references between the inherent geometric objects. For example, the geometry of extensions can be set depending on geometric objects of the perimeter block building. The building itself can conversely have dependencies on other city model objects - for example, to construction lines or to the corners of a plot (see Figure 4).

![Diagram of parametric 3D city model and CityGML](image)

**Figure 5.** Concept of the parametric 3D city model
As shown in the example, modeling correlations can be derived for different building typologies and represented as a dependency graph. The idea behind the parametric 3D city model is to directly implement those dependencies in an object-orientated structure, so that the modeling immediately depicts possible changes of state of its inherent geometric objects and parameters (see Figure 5).

3. IMPLEMENTATION OF THE PARAMETRIC 3D CITY MODEL

3.1 Geometric relationships within the modeling structure of city model objects

Before presenting the concept for the implementation of the parametric city model, it is important to take a closer look on the relationships that exist among geometric objects. There can be two kinds of geometric relationships in the modeling structure of the city model objects:

- Structural relationship: the hierarchical “consists-of” relationship between geometric objects; example: two vectors are components of a line, i.e. a change of state of one vector inevitably implies a change of state of the line.
- Numerical relationship: the numerical “is-calculated-from” relationship among the dimensions; example: a vector $a$ is the result of the subtraction of the two vectors $b$ and $c$. A change of state of vector $b$ has to trigger the recalculation of vector $a$.

Structural relationships cause unilateral dependency, i.e. an object is dependent on one or multiple other objects involved in the structural relationship. Numerical relationships can imply bilateral dependency among the objects (when $a = b - c$ then $b = a + c$ and $c = b - a$).

3.2 The dependency graph as Petri net

Whenever a parameter or a geometric object within the modeling structure of city model objects is changed, the change of state has to be transmitted to all geometrically related objects. As an example one can imagine a vector that is component of a line. As soon as the vector changes its position, the information about the translation has to be transmitted to the line, which in turn has to recalculate its length and update its graphical representation. The line itself is then again component of a surface and so on (see Figure 7). This process of transitions among geometrically related objects can be imagined as a chain of events that runs through all involved geometric objects. Such a system of events can be modeled as a Petri net.

![Figure 6. A simple Petri net; initial state (1) and final state (2)](image)

Petri nets (also known as place/transformation nets) are abstract, formal models of information flow (Petri 1962). They find application in the design and analysis of organizational systems in which regulated flows of objects and information are significant (Reising 1992). A Petri net is a directed graph. The set of nodes is divided up into two disjoint sets: ‘places’ are passive elements (i.e. conditions) and ‘transitions’ are active elements (i.e. events that may occur). Petri nets can therefore be mathematically described as a triple $N = (P, T, F)$, where $P$ is a finite set of places, $T$ is a finite set of transitions and $F$ is a finite set of arcs. Arcs run from a place to a transition (input arc) or vice versa (output arc). In addition to the static elements represented in the graph, a Petri net can also have dynamic properties, resulting from its execution. These can e.g. be markers (called tokens) that represent data values transmitted by transitions (see Figure 6).

3.3 Implementation of the Petri net

The parametrical 3D city model has been implemented by means of an object orientated structure. A Petri net kernel provides the basic infrastructure to depict changes of state within the modeling structure of the city model objects. By supporting the concept of objects, each component of the net has its own attributes and behavior. The entity ‘place’ of the Petri net coincides with the geometric objects of different kinds (such as vector, line, surface, etc.). The entity ‘transition’ in turn represents an event that occurs between structurally or numerically related geometric objects when one of the objects changes its state.
The Petri net kernel was implemented in Java. It provides two basic components (see Figure 7):

- The interface Place; by implementing Place, the different geometric class types (e.g. line, surface, body) become part of the Petri net.
- The class Transition; transitions represent events that may occur between structurally or numerically related objects. They contain an input place and an output place (in, out) and a collection of methods that have to be executed when the transition fires.

The depiction of changes of state within the modeling structure is being performed in two steps:

1. ‘schedule’: find all objects involved in the transition process and put them in an topological order. Collect all methods that have to be executed when performing the transition and store them in the particular Transition object.
2. ‘execute’: fire the transitions, i.e. execute the changes of state of the objects and trigger all necessary calculations.

Figure 8. Example of numerical relationships (bilateral dependency) among objects. Two lines (line1 and line2) are parallel and therefore related my means of the object ‘parallelism’. Depending on which object triggers the transition, different execution paths are possible.
4. EVALUATION

4.1 Application of the parametric 3d city model

The parametric 3D city model was implemented as a prototype based on the presented concept (see Figure 9). It serves as the underlying data model for decision-support systems in urban planning. As part of ongoing research digital methods have been developed, that can serve as an informed basis for debate and argumentation in political decision-making and planning processes and in turn support the development of densification strategies that are well-suited to their urban context (Seifert et al. 2014). By monitoring the key building codes, and visualizing their impact on the urban structure they enable planners to check various strategies and their execution variants and to compare several alternative approaches. The implemented software prototypes were evaluated by means of exemplary inner-city planning scenarios.

Figure 9. Prototypical implementation of the parametric 3d city model: Simulation of building codes (left), visualization of modified building volumes (right)

The model or individual parts, i.e. the geometric components of city model objects and their dependencies can be stored as an XML document. In this, city model objects are formed from three main parts: geographical references, parameters and the modeling correlations of and among the city model objects.

4.2 Conversion of city model data

Official plan foundations have a positional accuracy in the centimeter range. For the implemented model a degree of abstraction was chosen which allows the building representations to fulfill the conditions to apply the rules of the building regulations. Bay windows, roof dormers, chimneys and similar subordinate components must be neglected to a certain extent. In contrast, for example, shape, height and orientation of the roofs are an important factor when calculating setbacks. In the CityGML exchange and storage format scheme, geometric and semantic data are provided that are needed to derive a parametric 3D city model. Methods have been developed that make it possible to convert a large part of the building automatically into the parametric 3D city model. The following steps are carried out to do this:

1. Abstraction of building representations: the building footprints are brought to the desired degree of abstraction by using edge smoothing algorithms. This can be specified in accordance with the limit values given in regionally regulated tolerances.

2. Comparing the city model objects with dependency models: By detecting e.g. parallel lines and congruent dots of different objects, interior and exterior dependencies for the city model are automatically detected and the appropriate object structure is applied.

3. Manual editing: An editor allows manual input of additional data and the correction of, for example erroneous interpretations of the parameterization algorithm (see Figure 10).

5. CONCLUSION

By using the methods developed and prototypical implementations it could be shown that a parametric 3D city model is absolutely useful in supporting decisions in urban planning. The possibility of simulating building regulations means systems can be created that enable an informed evaluation of possible planning measures to support the planning and political decision-making process with reliable forecasts and objective data. Petri nets provide an efficient way to depict model correlations within the parametric 3d city model. With the perimeter
block the concept has been evaluated by means of a significant building typology for a major part of European cities. In addition to the perimeter block areas it is suitable for further building and local types. Many European cities show substantial building areas of uniformly structured housing estates from the twenties and the post-war years and the large estates of the sixties and seventies with extensively homogeneous structuring and common basic spatial structures. Future research has to be focused on the conception of bidirectional interfaces for the conversion of CityGML data into the parametrical 3d city model and vice versa. Open questions in this context are e.g. the handling with ambiguous interpretations and the simultaneity of different dependency models.

Apart from that prototypical implementations have shown that a large part of the building objects can be automatically transferred from the geometric part of CityGML or the underlying official plan data into the parametric 3D city model. For the semantic part of the city model, the pre-requisite applies that semantics provided in the CityGML schemes that are essential for the simulation of building laws (e.g. the type of usage of buildings, floor levels, etc.) must also be available and of sufficient quality. It has been shown that a manual interface both to correct erroneous interpretations and also for the user to enter missing data is a necessary supplement.

Figure 10. Prototype for automatically detecting modelling correlations and dependencies in a 2D building plan.

Manual user interface to correct erroneous interpretations and to enter missing data.

REFERENCES


