Construction Zone Generation Mechanisms and Applications

Ragip Akbas

Stanford University
Center for Integrated Facility Engineering
M:4020, Stanford, CA, USA 94305-4020

rakbas@stanford.edu

Martin Fischer

Stanford University
Construction Engineering and Management Program,
Terman Engineering Center, Room 292
Stanford, CA, USA 94305-4020
fischer@stanford.edu

Abstract. Construction zones are geometric units of work in the construction planning process. Using construction parameters and geometric properties, construction zone generation mechanisms change the level of detail of the 3D geometry and corresponding activities of the planned project. We have defined and implemented a set of mechanisms for construction zone generation. *Decomposition* provides various ways to generate detailed zones, while *aggregation* generates combined zones. To support these mechanisms, we have formalized a geometric representation based on triangular meshes that can represent the activity workflow, project spatial hierarchy, local variations in productivity, and activity state information on a component at any given time. This paper summarizes the mechanisms and describes various planning/scheduling applications of these mechanisms, such as rapid plan evaluation, resource and workflow balancing in 3D, and association of schedules at multiple levels of detail. Our contributions are the formalization of these zoning mechanisms, representation and algorithms to support these mechanisms, and planning/balancing methods utilizing the mechanisms.

Keywords: construction planning; geometric algorithms; construction zones; line-of-balance techniques

1. INTRODUCTION

Today, it is difficult for construction superintendents and schedulers to consider and communicate alternative plans even with the existence of 3D models. There is also a gap between the techniques for high-level planning/scheduling and detailed operations level planning from research and practical perspectives. We designed construction zone generation mechanisms to address these problems.

Where and when an activity is ongoing is crucial knowledge that relates different sources of information. Scheduling at a more detailed level can decrease activity duration and increase the reliability in the construction schedules. Relating high-level and detailed schedules can prevent common inconsistencies between schedules at different granularity.

One of our goals is to make effective use of geometric models, extending existing techniques in computer graphics to build a general computational structure for construction planning. We want to make the geometry details transparent to the planner, capturing the reasons for planning as parameters in zoning mechanisms. Then, changing these parameters, planners can easily try different scenarios with different crews and methods and different workflow parameters.

1.1 Previous Work

We define construction zones as the geometric units of work in the construction planning process. A construction zone can be a part of a component, a component, a group of components or a volume.

We refer to the process of changing the level of detail of the building geometry and activities as *zone generation*. Previous zone generation approaches are mostly for automated planning and use user-defined volumetric elements. Their purpose is to simplify the automated planning process by creating sub-networks of activities and constraints within zones and combining them to generate the final plan.

Thabet and Beliveau [5] define zones for aggregating components using a hierarchy of boxes. The ZonePlanner research [6] uses volumetric grids to search for an optimal zoning plan. These systems do not address crucial knowledge of planners related to workflow, arrangement of work areas, and changes in production rates due to various factors. None of these systems analyze and make use of complex geometry. They suffer from the necessity to define building blocks for each case a priori and from the combinatorial complexity to try all possible merging alternatives.

Riley and Sanvido [3], in their research for space planning in multistory buildings, define work-area patterns to describe the directions for and locations of units of work completed for different activities and materials. However, they don't represent the patterns computationally. We provide a computational method for supporting similar patterns of construction on 3D models.

4D CAD, in its simplest form, is the ability to visualize the state of construction at any given project time. It uses the association of the building elements with the activities required to construct them. Therefore, every component has time attributes. We extend that to every polygon on the geometry. Therefore, every location on the geometry can support different time parameters.

1.2 Terminology

We represent a building element by association of semantic product information with geometry, similar to many product modeling approaches [1]. We use the term building element synonymously with an assembly, e.g., a stud wall, concrete slab, roof. Building elements or assemblies are organized in a spatial or organizational hierarchy, a tree, where only the leaf nodes can contain the actual components (Figure 1). An assembly is composed of one or more of its components or constituents. For example, components for a stud wall assembly are studs, dry wall, insulation, etc. Each component can have different partitions representing zones, the smallest possible partition being the unit size of that component. Each assembly has a type, e.g., floor slab. By the same token, each activity has an *activity type*, e.g., pour concrete.

There is no single correct spatial level of detail for building elements. For example, either the whole floor slab or part of this slab enclosed by a room can be a building element with different granularity. Similarly, the activity required to construct it can be for the whole slab or the slab bounded by the room.

We distinguish between three main component categories for zone generation purposes: *prefabricated*, *fixed size site-assembly* and *continuous site-assembly*. Prefabricated components are installed as a single step. Therefore, the component and the assembly are the same, and they cannot be partitioned. Fixed-size site-assembly components require on-site installation, are formed of well-defined units of work, and one or more different activities are necessary for their construction, e.g. stud walls or roofing. Continuous site-assembly components are either not formed of unit size elements or the elements are negligible in size, such as reinforced-concrete components.

In the rest of the paper we will first explain the zoning mechanisms and the required geometric representation for them. Then, we will focus on different applications using these mechanisms in areas where traditional planning and scheduling techniques are limited.

2. ZONE GENERATION APPROACH

2.1 Mechanisms

We have abstracted the common level of detail changes during the construction planning/scheduling process using a well-defined set of zoning mechanisms, which act on components and activities. Zoning mechanisms essentially describe planning and geometric parameters for how activities are installed. They are powerful in replicating the construction planner/scheduler's intent and support what-if analyses and visualization. They are also designed to be general and extensible. We have two main categories of mechanisms for zone generation: *decomposition* and *aggregation*. An intuitive way to think about these mechanisms is: *decomposition* creates and maintains zones formed by a part of a component, *aggregation* creates and maintains zones consisting of a group of components.

2.1.1 Decomposition

This mechanism generates detailed construction zones on components and detailed activities for the zones, thereby increasing the level of detail of the existing models. The operation for decomposition is Decompose(*c, a, t, D, F*), where
c: component to be decomposed,
a: activity to be decomposed,
t: type of decomposition, one of (grid, production),
D: direction set, which contains, in the case of grid decomposition, a direction vector and amount of decomposition in each direction. For production-based decomposition, the direction set is the starting location and a set of unit direction vectors at specified locations,
F: factor set, which contains the shape factor functions and parameters and time-space functions as explained in section 2.3.

After the decomposition, the application generates a set of *subcomponents* and *subactivities* and links them to each other. The geometric representation for each subcomponent is a partition of the geometry for the original component. Each subactivity is a result of splitting the original activity.

2.1.2 Aggregation

This mechanism groups components and sets the start and finish dates by associating detailed activities with the new group of components. Its types range from organizing the components within a region to ordering the activities necessary for their construction. It supports, for example, ordering of exterior walls given the starting location and direction.

The operation is *Aggregate*(*node*, *a, t, D, F*), where *node* is a node in the product hierarchy, *t* is the type of aggregation, *D* is the direction set, and *F* is the factor set. Note that *Aggregate* is different from the *Union* operator from solid modeling, in that its purpose is not to combine the geometry of individual components. Furthermore, *Aggregate* is not the inverse operator of *Decompose*.

2.2 Geometric representation for the mechanisms

Since the zoning mechanisms act on the geometry, geometric representation and algorithms are an important part of this research. We use triangle meshes for the geometric representation. Geometrically, a triangle mesh is a surface consisting of triangular faces pasted together along their edges. The mesh geometry can be denoted by a tuple $M = (K, V)$, where K specifies the connectivity of the mesh (the adjacency of the vertices, edges, and faces) and V is the set of vertex locations defining the shape of the mesh in $\mathbf{R}^3$. The vertex locations and the connectivity for the mesh can be acquired from any 3D CAD or modeling application. We can also build the connectivity for the faces using common computer graphics techniques.

Triangle mesh is a common and well-researched representation. However, as it is, it lacks the construction domain information needed to support the zoning mechanisms. We have extended the mesh representation to a tuple $M = (K, V, D, S)$, where D is local direction of workflow and production modifier at any location on a component, and S is the state of the component location as a function of time. D is associated with a (vertex, activity) pair, while the state S is associated with the faces (Figure 2c). These main extensions are as follows:

- Every vertex on the mesh contains a unit direction vector for each related activity to describe the local direction of workflow at that location.
- Every vertex on the mesh contains a productivity modifier for all related activities, $Pk \in [0,1]$. Originally, modifiers on all vertices are 1, meaning the activity production rate at the given location is equal to the production capacity.
- Every triangle contains a *local activity state* to represent the installation status at any given time. We created an *activity state machine* to unambiguously calculate the local activity states as a result of the zone generation mechanisms.

2.3 Production Rate Modifiers

The production rate is not always constant for activities anywhere on a component. We support variable local production rates using the productivity modifiers using the parameters stored with the geometry. The modifier can be a combination of the effects of the component's shape on the production rate (*shape factors*), effects of other ongoing activities (*time-space factors*), or other effects that we did not consider so far in our research (learning curve, availability of resources, etc.).

The application calculates the production rate value using the user-supplied functions and the shape representation. Shape functions are dependent on geometric parameters including height, distance to edge, and slope. Schedulers declare how production rates are affected by these parameters so that the software can calculate the local production rates. Time-space factors can decrease the production rate if two close activities are installed at the same time in the schedule. They can also stop an activity if a prerequisite component is not ready.

2.4 Spatial Structures

We use various spatial structures to aid the mechanisms. A *region* is a bounded volume, which encloses all the components that fall into it. Examples of regions for construction projects are a floor, a room, or a quadrant. We support a hierarchical region structure that is user defined and unique for a project. Other spatial structures we use are *octrees* for spatial search and *grids* for arrangements of some mechanisms.

3. APPLICATIONS

We now will use the defined mechanisms together with the geometric representation to show approaches to common construction planning-scheduling problems. Starting with plan evaluation on the simplest, single building element case, we will extend the application to multiple building elements at variable levels of detail. Building on the common line-of-balance convention of continuous utilization of resources, we will describe a hierarchical 3D scheduling approach using the zoning mechanisms. Then, we will consider extensions to the architecture, such as keeping an operational schedule synchronized with the master schedule, and defining construction method-specific zones using the mechanisms.

3.1 Rapid plan evaluation

This application is a direct consequence of the described mechanisms and the representation. Users can consider planning alternatives quickly by changing the direction of workflow, starting location, production rate and the sequence for the activities. The application uses the parameters in the mechanisms to traverse the geometry and generate the states for any location at any time.
We have a prototype application that allows the user to enter the parameters for the mechanisms, which become input for the actual mechanisms. Let's say the user wants to plan the pour for the slab in Figure 1. The user interactively defines how it will be constructed in the application, in this case specifying the direction of flow vector and amount (Figure 2a), which in turn generates the following mechanism:

Decompose(Floor2slab,Pour_Floor2slab, *grid*, *D, F*)

Note that if the user changes the direction of the activity or the amount of decomposition, only the directional information, *D,* changes from the initial parameters. The user can immediately visualize the effect of this change. If the user wanted to perform this plan evaluation manually, he would need to (1) modify the 3D model to split the geometry, (2) split the construction activities into subactivities, and (3) re-link them. Any other change requires the repetition of this tedious process.

For the columns over the slabs, the zones are groups of columns (Figure 2b). The mechanism is:

Aggregate(F2Columns, Rebar_F2Columns, *production*, (A-1, (-1,0,0)), *F*)

Obviously, construction projects are composed of many activities and components, and considering activities in isolation is not very valuable. The user can also decompose the same slab for the other associated activities, such as *rebar* and *form* by applying *decompose* mechanisms for each. In that case, different activities on the same component should follow each other without interference. The *activity state machine* guarantees that the activities do not violate their predefined ordering, i.e., the prerequisite activity is always installed before, and there are no two activities installed at the same location at the same time. The mechanisms are also applicable to more complex structures. Figure 3 shows production-based decomposition applied to an exterior enclosure element. The user specifies the starting location, direction of workflow, and effect of shape on production rate for different activity types required to construct the enclosure.

The following application focuses on the interactions of multiple activities on multiple building elements considering resources, providing an alternative scheduling approach when 3D models are available.

3.2 3D schedule balancing

Traditionally, line-of-balance (LOB) or linear scheduling techniques are used for activities following each other for linear or repetitive work, allowing planners to balance workflow by adjusting the production rates and start dates for activities [4]. They help plan for continuous utilization of resources. However, these methods are limited to fixed single level work areas corresponding to the flow. Activities in linear scheduling techniques should either be repetitive units or proceed linearly in a single axis-aligned direction - horizontal or vertical. To plan using LOB, the project is divided into planning units usually as large work areas. Although multiple activities can concurrently act on the same work area, LOB conservatively limits a single activity in the same work area at a time.

Our application extends these valuable techniques to many types of construction activities using the spatial and planning information and benefiting from the same zoning mechanisms explained in the previous application. It adds the ability to schedule at multiple levels and supports multiple activities on a component and complex shapes. The balancing process can range from single element/single region to hierarchical/multiple regions.

3.3 Maintaining master and operational schedules synchronously

The previous application examples ignored maintaining the level-of-detail relationships with the activities in the existing master schedule. Master schedules are the means for tracking resources and activities for construction projects. Detailed planning with zoning mechanisms and actual detailed site data adds valuable input to the schedule. Project scheduling software [2] provide hammock or summary activity types to summarize a group of activities. However, this relationship is error-prone and limited.

We associate the original activities with the activities resulting from the zoning mechanisms through the associated project locations. Whenever the activity date and duration are modified at a more detailed level, the activities in the master schedule can be updated. Similarly, users can propagate the effect of changes in the master schedule to the detailed schedules. This also allows the construction process visualization at two levels of detail.

3.4 Method changes on component

In all of the previous examples, the generated zones correspond to some planning units of work. Zones can also support changes in construction methods on a single component. Construction methods can vary because of geometric factors such as height and slope or existence of nearby components, such as scaffolding. We support ways to generate separate zones on building elements in such cases.

4. CONCLUSIONS

In this paper, we have briefly described construction zoning mechanisms and explained interactive planning/scheduling applications. We are validating the research by replicating the manual operation plans on several projects via the zoning mechanisms. The tested activity types include concrete pour schedules, exterior enclosure and roofing.

Zoning mechanisms assume an activity proceeds in a connected way and make a surface approximation for component geometry. Additionally, we use linear shape and time-space functions.

These mechanisms provide the possibility of many other types of visualization, analysis and optimization. Another important use is data collection by reverse-engineering the actual construction dates to infer the planning parameters. A space planning methodology can be adopted for the 3D balancing application to check for other types of spatial conflicts. Furthermore, functions to generate the temporary structures such as laydown, scaffolding and formwork on the fly can increase the value of this research.

5. REFERENCES:

1. IAI. 2001. *Industry Foundation Classes - Release 2.x Specifications.*

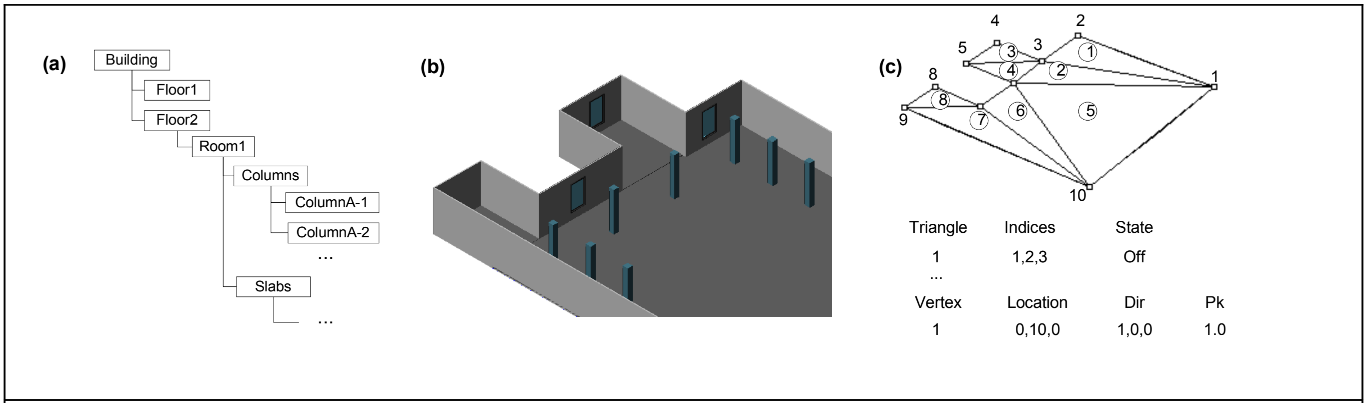
2. Primavera. 2000. *Primavera Project Planner*. Primavera Systems, Inc., Bala Cynwyd, PA.

3. Riley, D. R. and Sanvido, V. E. 1995. 'Patterns of Construction-Space in Multistory Buildings', *Journal of Construction Engineering and Management,* Vol. 121, no. 4, pp. 464-473.

4. Stradal, O. and Cacha, J. 1982. 'Time Space Scheduling Method.' *Journal of Construction Division*, Vol. 108, no. 3, pp. 445-457.

5. Thabet, W. Y. and Beliveau, Y. J. 1997. 'SCaRC: Space-Constrained Resource-Constrained scheduling system', *Journal of Computing in Civil Engineering*, Vol. 2, no. 1, pp. 48-59.

6. Winstanley, G. and Hoshi, K. 1993. 'Activity Aggregation in Model-Based AI Planning Systems', *I. Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacture (AI EDAM)*, Vol. 7, No.3, pp 209-228.



Triangle	Indices	State	
1	1,2,3	Off	
...			
Vertex	Location	Dir	Pk
1	0,10,0	1,0,0	1.0

Figure 1. Representation basics for the sample model. (a) Partial product composition hierarchy. (b) Partial 3D isometric view. (c) Triangle mesh representation for the slab including the extensions.

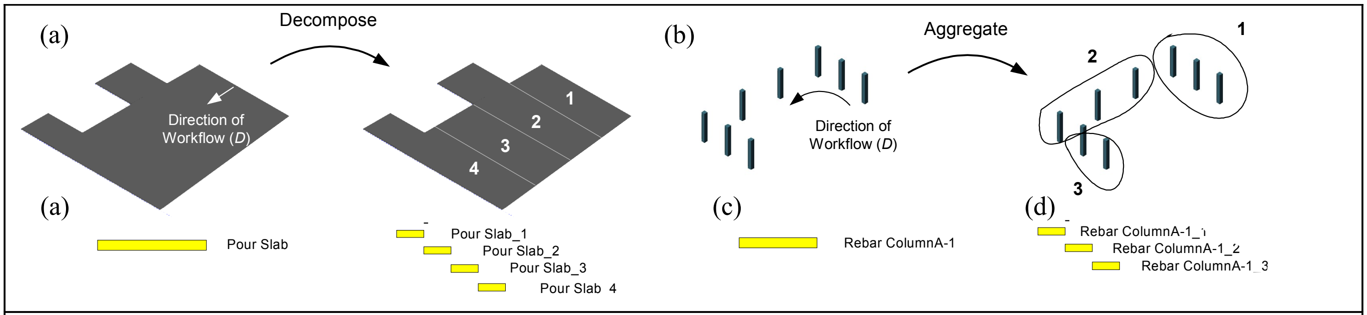


Figure 2. Examples of mechanisms. Decomposition splits the geometric representation to correspond to the units of work for *pour_slab*. Aggregation groups the concrete columns for *install_rebar* in this example.

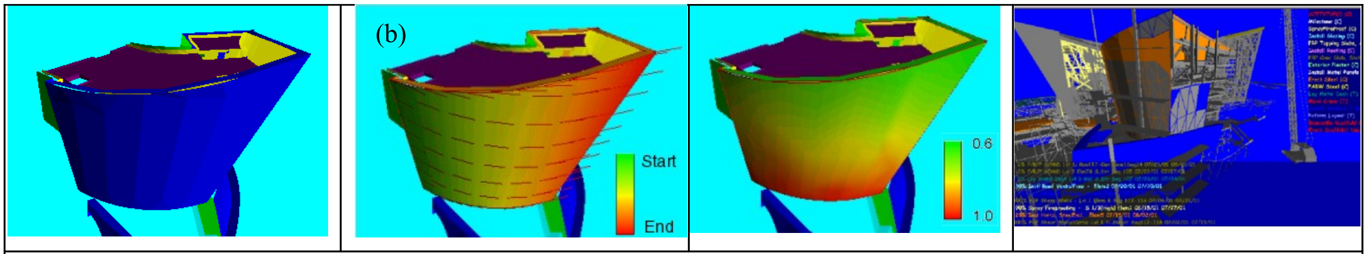


Figure 3. Zoning mechanisms applied to a complex exterior enclosure surface. (a) Original model. (b) Direction of workflow interpolated over the surface visualized by arrows and colors. (c) Height factor calculated on the surface. (d) Snapshot of 4D model after the mechanism is applied.