CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

A Feature Ontology to Support Construction Cost Estimating

By

Sheryl Staub-French, Martin Fischer, John Kunz,
Boyd Paulson, and Kos Ishii

CIFE Working Paper #69
July 2002

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

WORKING PAPER #69

A FEATURE ONTOLOGY TO SUPPORT CONSTRUCTION COST ESTIMATING

Sheryl Staub-French[1], Martin Fischer[2], John Kunz[3], Boyd Paulson[4], Kos Ishii[5]

Abstract

Current building product models explicitly represent components, attributes of components, and relationships between components. These designer-focused product models do not represent the features of building components that are important for calculating construction costs, such as penetrations and component similarity. To provide product models that are useful to cost estimators, we need to transform current product models into estimator-focused product models that represent the features of building components that affect construction costs. Previous research efforts identify many of the different features that affect construction costs but they do not provide a formal and general way for practitioners to represent the features they care about according to their preferences. This paper presents the vocabulary we formalized to represent the different types of features of building product models that are important to cost estimators of building construction. The vocabulary allows estimators to represent their varied preferences for naming features, specifying features that result from component intersections and the similarity of components, and grouping features that affect a specific construction domain. The feature ontology provides the structure for transforming designer-focused product models into feature-based product models that support cost estimating. We also describe the framework we developed that uses the ontology to represent features in a project-independent way so that they can be reused from project to project to create estimator-focused feature-based product models from a given product model. Tests provide evidence for the power and generality of the feature ontology. The main contributions of the paper are the feature ontology and the framework developed to capture this knowledge from estimators.

[1] Assistant Professor, Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4, ssf@interchange.ubc.ca
[2] Associate Professor, Department of Civil and Environmental Engineering and (by Courtesy) Computer Science, Director, Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, fischer@stanford.edu
[3] Executive Director, Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, kunz@stanford.edu
[4] Professor, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305, paulson@stanford.edu
[5] Associate Professor, Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, ishii@cdr.stanford.edu

1. Introduction and Overview

3D modeling applications now support the design of complex products in many industries, including the building industry. Many architectural 3D modeling applications can export semantically rich product models using the industry standard Industry Foundation Classes (IFC's)(IAI 2001), enabling the sharing of product models with other software applications. IFC-based product models are object-oriented data models that explicitly represent components (e.g., 'IfcWall' and 'IfcBeam'), attributes of components (e.g., 'length' and 'fire-rated') and relationships between components (e.g., 'IfcRelConnectsElements'). Cost estimating applications leverage IFC-based models by extracting dimensional information from building components for quantity takeoff calculations (Timberline 2001). However, other types of design conditions impact the cost of constructing building components, such as openings, penetrations, and component similarity. Estimators have different preferences for describing these different design conditions and the impact they have on a specific component's construction cost. To provide product models that are useful to cost estimators, estimators need a vocabulary for describing the different types of design conditions that affect construction costs and a framework for representing the different design conditions generically in the computer so that this knowledge can be reused to support feature-based cost analysis.

Previous research efforts identify the different design conditions that affect construction costs (Hanna and Sanvido 1990; Fischer 1991; Thomas and Zavrski 2000; Thomas and Sackrakan 1994; de Sousa and Thomas 1996; Smith and Hanna 1993; Sanders and Thomas 1991). However, these researchers do not provide a formal way for practitioners to represent the design conditions they care about according to their preferences. In our research, we use features to describe the different design conditions that impact construction cost. Features are used extensively in the manufacturing industry to describe the parts of a product design that affect manufacturability, inspectability, serviceability, etc. (Cunningham and Dixon 1988; Shah 1991). However, the feature representations developed in the manufacturing industry do not fully support the representation of building product models. Specifically, building product models contain different features and different types of products, and the fragmentation of the building construction industry heightens the need for user customizability. Our research applies the manufacturing concept of features to building construction and extends it to represent the features that are useful to cost estimators.

We formalized a vocabulary (i.e., an ontology) using features to represent the different design conditions that affect construction costs. Figure 1a shows the building elements that affect construction costs that are explicitly represented in the IFC's (the specific example will be explained in the next section). The IFC's provide a designer-focused product model that explicitly represents 'components' and 'openings' as an attribute of components. This representation is incomplete because it does not represent most of the design conditions that are important to cost estimators. Our feature ontology enriches the current standard building product model representation by formalizing the representation of the variety of features of building components that affect construction costs (Figure 1b). The feature ontology enables the transformation of designer-focused product models into feature-based product models that support cost estimating.

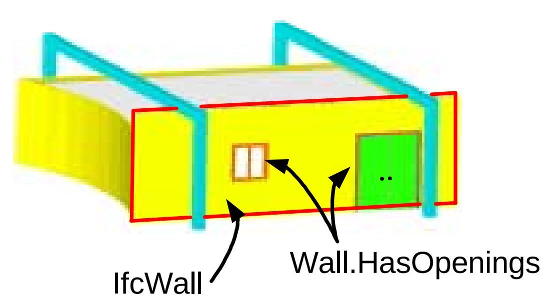


Figure 1a: The IFC's explicitly represent building components and their properties. The IFC's do not explicitly represent the features that affect construction costs.

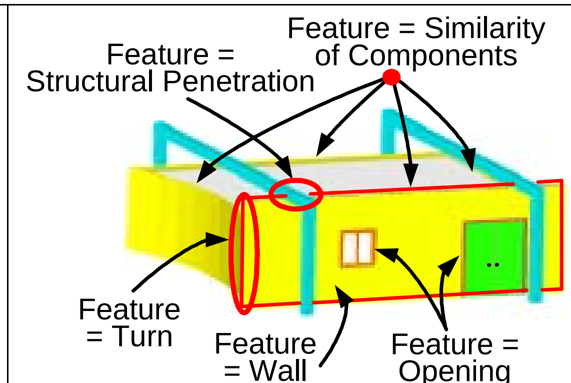


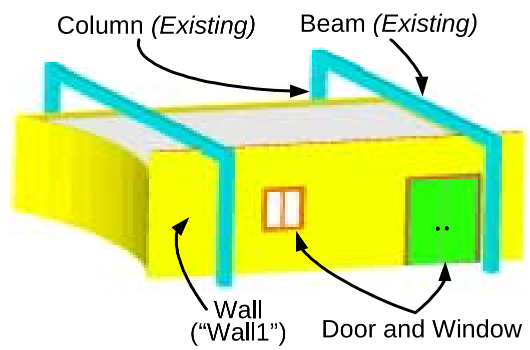
Figure 1b: Examples of features represented in the feature ontology we formalized to provide product models that are useful to cost estimators.

Figure 1: Comparison of product model representations using the IFC's and the feature-based representation formalized in our research. The feature ontology extends current product model representations by representing the features of building components that affect construction costs.

1.1 Case Example

This section describes use cases that illustrate the design conditions that affect drywall construction and concrete column construction. For each use case, we describe the different design conditions that estimators consider when creating a cost estimate.

For drywall construction, we describe two drywall estimators' perspectives to illustrate the different vocabularies estimators use to describe the design conditions they care about.



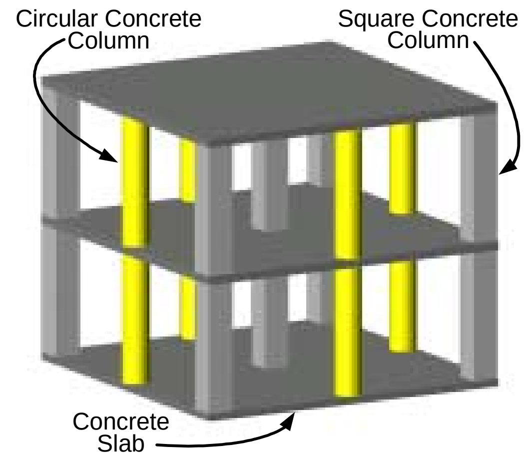


Figure 2a: Building components in the office project case study. The drywall estimator estimates the costs for constructing the four walls shown.	**Figure 2b:** Building components in the retail store case study. The concrete estimator estimates the costs for constructing the columns shown.

Figure 2: Building components in drywall and concrete column case studies. The case studies illustrate the different design conditions that affect construction costs.

Estimators must identify the design conditions that affect the project's activities, resources, and resource productivity rates that form the basis of a cost estimate for a particular design. Estimators from the same domain have different preferences for what design conditions they consider, and estimators from different domains consider different design conditions. Figure 3 shows the design conditions that two drywall estimators consider and the design conditions that the concrete column estimator considers.

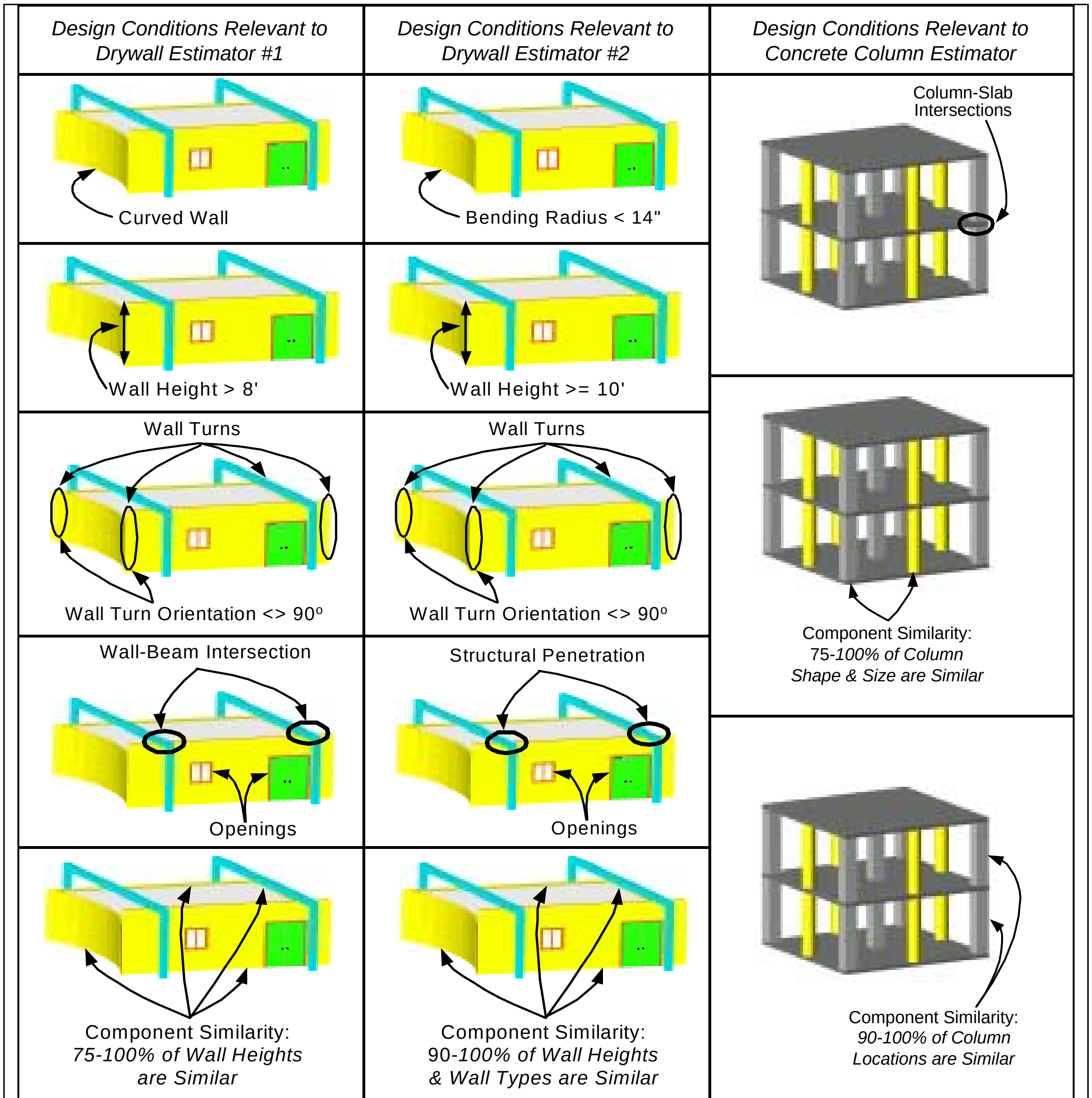


Figure 3: Design conditions that are important to drywall and concrete estimators for the two use cases. The two drywall estimators consider the same design conditions (e.g., 'wall turns' and the 'orientation of wall turns'), use different terms to describe the same design condition (e.g., 'wall-beam intersection' and 'structural penetration'), and have different preferences for describing the concept of component similarity (e.g., '75-100% of wall heights' and '90-100% of wall heights and types'). Drywall and concrete estimators consider similar design conditions (e.g., 'component similarity'), but some design conditions are unique to a specific domain (e.g., the 'column-slab intersections').

Today's cost estimating software (Timberline 2001) and concepts found in the literature (Laitinen 1998; Aouad et al. 1994; Froese 1996: Aouad et al. 1997; Stumpf et al. 1996; Slaughter 2000) allow estimators to represent their preferences for adding cost items to construct a component based on design conditions that are specific to the properties of the component, such as the curvature of a component. However, as Figure 3 shows, there are many other design conditions that affect construction costs in addition to component properties. Design conditions can be based on:

- properties of components (e.g., the 'curvature' and 'height' of the wall),
- groupings of components (e.g., the 'grouping of walls' based on component similarity),
- intersections of components (e.g., the 'structural penetration' resulting from the intersection of the wall and beam), and
- properties of component intersections (e.g., the 'orientation' of wall turns).

It is too time-consuming for estimators to manually identify all the project-specific design conditions and adjust the project's activities and resources accordingly for each project they estimate. Lacking automated support to identify and explicitly represent the important cost-incurring design conditions for a given product model, estimators require considerable time to prepare estimates and often employ ad hoc and error-prone methods, resulting in inefficiencies and inconsistencies in the cost estimating process and the resulting cost estimate. For automation, estimators need a vocabulary to describe design conditions that affect construction cost and a framework for representing them generically in the computer to enable the automatic generation of feature-based product models that support cost estimating.

We use the concept of *features* to represent the different design conditions that are important to cost estimators of building construction. We refer to components in a building product model, such as walls and columns, as "component features." Throughout the remainder of this paper, the terms "component feature" and "component" will be used interchangeably. We refer to features that result from the intersection of two components, such as openings and turns, as "intersection features." We refer to features that result from groupings of components, such as grouping walls based on component similarity, as "macro features."

To represent the features that affect construction costs, estimators need a vocabulary that allows them to:

- Represent the different types of features that affect construction costs,

- Customize how features are named, what component intersections are instantiated, and how component similarity is represented, and
- Specify the features and properties that affect a specific component's construction costs.

Our research formalizes an ontology of cost-driving features that enables estimators to transform designer-focused product models into feature-based product models that support cost estimating.

1.2 Research Goals

The goals of this research were to formalize an ontology to describe the different types of features that affect construction costs and to provide a formal and computer-interpretable way for estimators to specify the features that affect a specific component's construction costs. The use case illustrates that the ontology needs to be formal, general, and flexible to represent the different features that affect construction costs:

(1) Formal: Estimators need a structured way to represent the features of building product models that affect construction costs. The formal representation should include all the attributes necessary for estimators to describe the different design conditions that affect construction costs (Figure 3).

(2) General: Estimators need to represent features independent of a specific project or product model. A generic and computer-interpretable representation of features enables estimators to reuse this knowledge from project to project. The generic representation of features can be leveraged to automatically create a project-specific feature-based product model that supports cost estimating. Finally, the representation of features also has to be general enough to support cost estimating of different construction domains.

(3) Flexible: The use case demonstrates that estimators have different preferences for describing the different design conditions that affect construction costs (Figure 3). The ontology must be flexible enough to represent estimators' varied preferences for naming features, specifying relevant component intersections, defining component similarity, and specifying the features that affect a specific component's construction costs.

The ontology formalized in this research to represent the different features that affect construction costs meets these criteria. The ontology provides a vocabulary that abstracts the different design conditions estimators consider when estimating the cost of building construction. The next section describes the related research background.

2. Related Research Background

To represent the features and properties that are important to cost estimators, this research combines and extends previous research in construction cost estimating and product modeling.

2.1 Prior Research on Construction Cost Estimating

Many researchers identify the design conditions that affect the cost of building construction (Hanna and Sanvido 1990; Fischer 1991; Thomas and Zavrski 2000; Thomas and Sackrakan 1994; de Sousa and Thomas 1996; Smith and Hanna 1993; Sanders and Thomas 1991). For example, Hanna and Sanvido (1990) recognize that component similarity limits the applicability of different formwork systems, and Thomas and Zavrski (2000) recognize that penetrations and turns affect resource productivity. However, these researchers do not provide a vocabulary or a framework for practitioners to specify the design conditions that matter to them. They do not provide a formal way for practitioners to specify new design conditions (e.g., the wall-beam intersection) or customize existing design conditions (e.g., how component similarity is represented) based on their preferences.

2.2 Prior Research on Product Modeling

Several researchers represent components, attributes of components, and relationships between components in building product models explicitly and generally. They represent many of the components and component properties that affect the cost of building construction (Bjork 1987; Gielingh 1988; IAI 2001). However, they do not explicitly represent many of the design conditions that affect construction costs, such as penetrations and component similarity. We use the building components and component properties represented in the IFC's but extend building product models to represent the design conditions that affect construction costs. We use the concept of *features* to represent the design conditions that are important to cost estimators of building construction. Product features are used extensively in manufacturing to describe the geometric forms or entities in a product model that are important in some aspect of the manufacturing process (Cunningham and Dixon 1988; Shah 1991). This research focuses on the product features that are important for estimating the cost of building construction.

Several researchers use features to represent the building components that are important in the design process (van Leeuwen 1999; Clayton et al. 1996). However, they do not represent features that result from intersections of components and the similarity of components that are important for building construction. Dixon and Cunningham (1988) formalize 'intersection features' to represent features that emerge from intersections of primitive and add-on features (e.g., corners). They formalize 'macro features' as pre-specified combinations of primitives (e.g., boxes). We extend the definition of "intersection features" to represent building designs by defining intersection features as the intersection of component features. We extend the definition of macro features to represent the concept of component similarity. We define macro features as pre-specified combinations of other features. We represent component similarity as "groupings" of components based on the feature property *similarity*.

Dixon and Cunningham (1988) formalize the classification of 'intersection features' and 'macro features' but they do not formalize attributes of each feature type to enable practitioners to create or customize instances of intersection and macro features. We formalize the different attributes of intersection and macro features to provide a formal way for estimators to represent the component intersections that affect construction costs and to define component similarity according to their preferences.

The IFC's explicitly represent the connectivity between components using the 'RelatingElement' and 'RelatedElements' attributes, which we use to represent intersection features. However, the IFC's do not provide a way to filter the component feature's connections that are important to an estimator. For example, the wall's connection with the ceiling and floor were not important to the drywall estimator while the wall's connections to other walls ('wall turns') and to the beam ('structural penetration') were important because these connections impact construction costs. Moreover, some component connections are not explicitly represented in IFC-based product models because the designer does not intend for the components to be connected. For example, the connection between the wall and the beam emerges based on the architectural and structural designs. This research provides a formal way to represent the component connections that affect construction costs by instantiating these connections as intersection features.

Estimators need to be able to specify the features that affect a specific component's construction costs. Many researchers provide a formal way to group features based on how they influence manufacturing processes (Shah 1991; Hyer and Wemmerlov 1984; Cunningham and Dixon 1988). For example, Cunningham and Dixon (1988) determined that 'feature sets' could be deduced by a process-activity pair, such as a feature set for the activity of manufacturability evaluation in the process of injection molding. A corresponding example for this research would be a feature set for the activity of cost estimating in the process of drywall construction. However, they do not represent the feature sets that are important to estimators of building construction, and they do not provide a flexible representation that allows practitioners to specify the features that should be assigned to a feature set. For example, the feature set for wall construction includes 'openings,' 'turns,' and 'structural penetrations.' The IFC's represent 'property sets' for different component types to represent the properties that are important to designers. However, the IFC's do not use property sets to represent the properties that are important to cost estimators, and they do not provide a formal way for estimators to specify the properties of components (e.g., 'curvature' of walls) and intersection features (e.g., 'orientation' of turns) that affect construction costs. Our research extends the application of *feature sets* and *property sets* to cost estimating of building construction.

In summary, previous research efforts in cost estimating identify many of the design conditions that affect construction costs, but they do not provide a formal way for practitioners to represent the design conditions they care about. Prior research efforts in feature-based product modeling represent components, the connectivity between components, and feature sets and property sets of components. However, they do not provide a flexible representation that allows estimators to represent the component connections that affect construction costs, and they do not provide a formal way for estimators to specify their preferences for representing component similarity. Moreover, they do not provide a way for estimators to specify the feature sets that are important for a specific construction domain and the property sets of features that affect construction cost.

3. Representing Features that Affect Construction Costs

So far this paper has demonstrated the need to provide a vocabulary to represent the features and properties that are important to cost estimators of building construction. Our feature ontology classifies the features that affect cost, formalizes attributes to describe each feature type, and represents the sets of features and properties that affect costs for a specific construction domain.

To create the ontology, we identified the different design conditions that affect construction costs by reviewing previous research in this area and interviewing cost estimators. We interviewed 14 different cost estimators from five different construction domains. We interviewed two general contractors and twelve subcontractors that self-perform construction work on drywall, structural concrete, ductwork, process piping, and electrical systems. We implemented three case studies on two drywall construction projects and one case study on a concrete column construction project. We abstracted the different vocabularies used by estimators to describe the design conditions that affect construction costs for the types of conditions that we studied.

We implemented the feature ontology in a software prototype called Feature Generator (FeaGen) (Figure 4). FeaGen provides the framework that uses the ontology to represent estimators' preferences for the features and properties that affect construction costs. Estimators represent the relevant intersection features and customize component similarity in FeaGen in Feature Specification templates that use the attributes of the different feature types formalized in the feature ontology. Estimators also specify the sets of features and properties that affect a specific component's construction costs. FeaGen represents the cost-driving features and properties specified by the estimator generically so that they can be reused from project to project. Each time an estimator wishes to create an estimate for a specific project, FeaGen uses the generic cost-driving features to create a project-specific feature-based product model that represents the features and properties that are important to the cost estimator. We created another software prototype called Activity-based Cost Estimating (ACE) that uses the estimator-focused feature-based product model to generate and maintain construction cost estimates (Staub-French et al. 2002b).

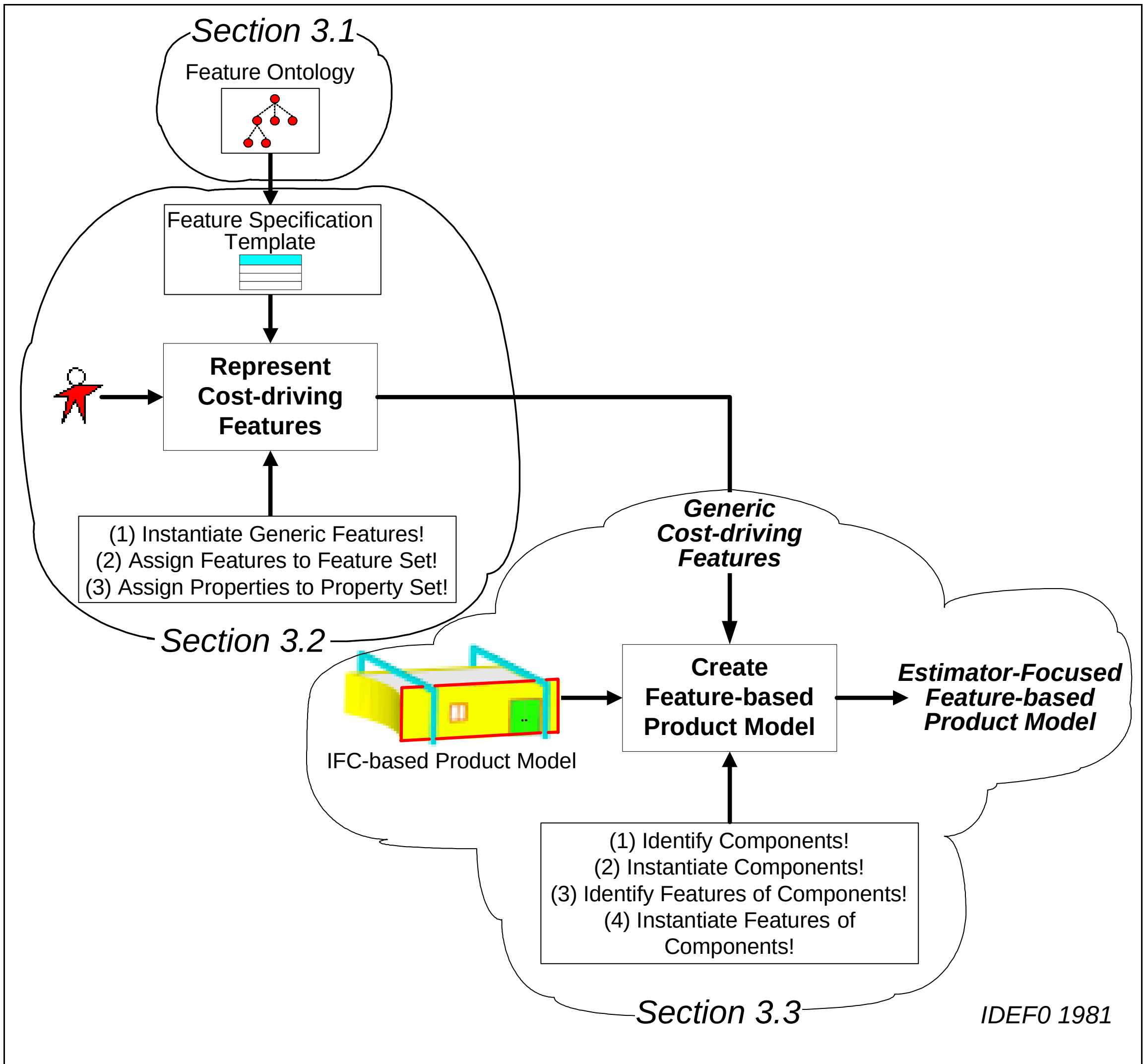


Figure 4: FeaGen uses the attributes of the feature ontology to capture estimators' preferences for the features and properties that affect construction costs in Feature Specification templates. FeaGen represents the cost-driving features and properties specified by the estimator generically so that they can be reused from project to project. Each time an estimator wishes to create an estimate for a specific project, FeaGen uses the generic cost-driving features to create estimator-focused feature-based product models given an IFC-based product model.

The next sections describe how the ontology represents the features that affect construction costs. We describe the feature classification and the different attributes of each feature type (Section 3.1). We then describe how FeaGen uses the attributes of the ontology to provide a framework for estimators to represent their preferences for cost-driving features generically (Section 3.2). Then we describe how we leverage the generic cost-driving features specified by the estimator to generate project-specific feature-based product models that support cost estimating (Section 3.3).

3.1 A Feature Ontology

The research challenge with respect to formalizing a vocabulary for estimators is that different design conditions exist in a given product model, that different types of design conditions affect construction costs, that estimators have different preferences for representing design conditions, and that different design conditions affect different construction domains. The ontology we formalized addresses these challenges by providing a general way to describe design conditions using features. The ontology represents features independent of a particular project or product model, allows estimators to customize features according to their preferences, and allows estimators to specify the specific features and properties that affect a component's construction cost. Appendix A shows the entire feature ontology.

3.1.1 Classifying Features

We classify features to enable estimators to represent instances of each feature type according to their preferences. The case demonstrated that features can be components, features can emerge from intersections of components, and features can emerge from groupings of components based on their similarity. Consequently, we classified features based on these different views of building components in a product model. Each feature type has different attributes that help estimators represent feature instances, which we discuss in the next section. We classify features into three different types:

1. Component Features: Features that result from components in an IFC-based building product model, such as walls and columns.
2. Intersection Features: Features that result from intersections of components, such as penetrations and turns.

3. Macro Features: Features that result from pre-specified combinations of other features. We focused on macro features that emerge based on the similarity of components of the same type.

Figure 5 shows the three feature types formalized in this research and examples of each feature type from the motivating case.

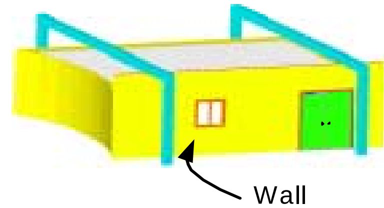
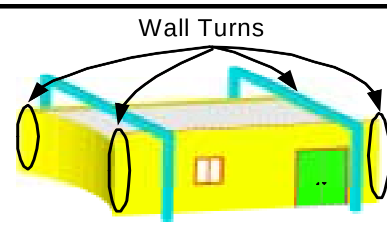
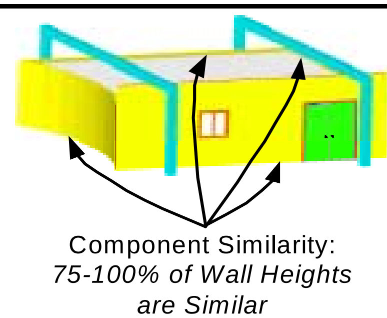
Feature Type	Feature Type Definition	Example from Case
Component Features	Features that result from components in an IFC-based building product model.	
Intersection Features	Features that result from intersections of components.	
Macro Features	Features that result from pre-specified combinations of other features. We focused on macro features that emerge based on the similarity of components.	

Figure 5: The ontology classifies features into three types: (1) Component Features, (2) Intersection Features, and (3) Macro Features. The feature classification enables estimators to represent instances of each feature type according to their preferences.

3.1.2 Common Attributes of Features

The common attributes of the three feature types formalized in our research enable estimators to represent their varied preferences for naming features, specifying the component intersections that are important to them, defining component similarity, and specifying the features and properties that affect a specific component's construction costs. Figure 6 shows the common attributes of each feature type. Some attributes are common to all features, and some attributes are specific to a feature type. The common attributes of all features are:

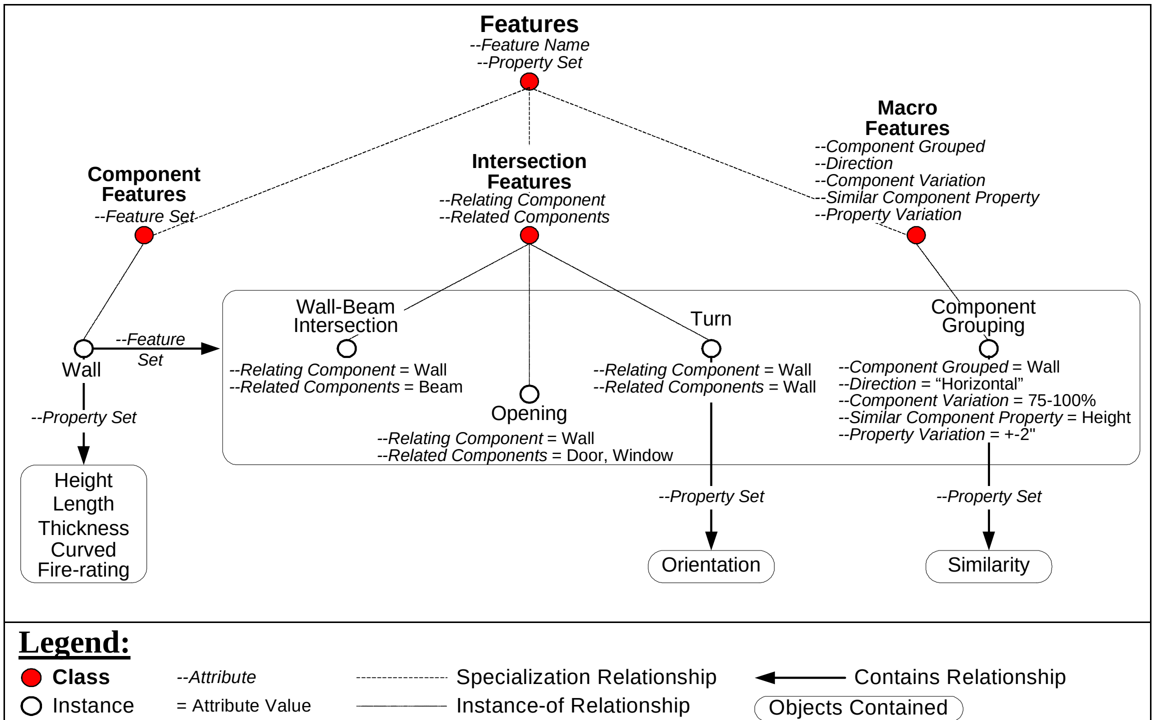


Figure 6: Feature ontology that represents the common attributes of the three feature types and example feature instances from the motivating case. Each component feature knows the estimator's preference for what features and properties influence the cost of its construction using the 'feature set' and 'property set' attributes. Each intersection feature knows the estimator's preference for what properties of the feature affect a component feature's construction cost using the 'property set' attribute and what component intersections are important using the 'related component' and 'relating components' attributes. Each macro feature represents the estimator's preference for defining component similarity in terms of the component properties that need to be similar and the amount of variation that is allowed to exist.

(1) Feature Name: The estimator's preference for naming the feature. The feature name must be unique. This attribute is necessary because estimators use different terms for describing the features that are important to them. For example, in the motivating case, the two drywall estimators called the intersection of the wall and the beam a 'wall-beam intersection' and a structural penetration' (Figure 3).

(2) Property Set: The estimator's preference for the properties of a feature that affect a component's construction costs. This attribute allows estimators to filter the properties of a feature that are important for cost estimating. For example, the estimator in the motivating case represents 'orientation' in the property set of the feature *turn*. I use "properties" to describe the attributes of features to be consistent with the terminology used in the IFC's.

The next sections describe the common attributes specific to each feature type.

3.1.2.1 Common Attributes of Component Features

The attributes of component features enable estimators to specify the features and properties that affect a specific component's construction costs. The attribute specific to component features is:

(1) Feature Set: The estimator's preference for the features that affect a specific component's construction costs. For example, the estimator in the motivating case represents 'wall-beam intersections,' 'turns,' and 'openings' in the feature set for walls.

3.1.2.2 Common Attributes of Intersection Features

The two attributes of intersection features are based on the attributes used by the IFC's to represent the relationships between components (IAI 2001). Estimators have different preferences for what component intersections are important for estimating a specific component's construction cost and how to name the component intersections. The attributes of intersection features allow estimators to specify the cost-driving features that result from specific intersections of components. The two attributes of intersection features are:

(1) Relating Component: The component class that the estimator is estimating. From the motivating case, the estimators consider 'wall' and 'column' components.

(2) Related Component(s): The component classes of the intersecting components for the 'related component.' For example, to represent the 'wall-beam intersection' feature, the estimator specifies 'wall' for the 'relating component' and 'beam' for the 'related components.' To represent the 'opening' feature, the estimator specifies 'wall' for the 'relating component' and 'door' and 'window' for the 'related components.'

3.1.2.3 Common Attributes of Macro Features

The attributes of macro features provide a formal way for estimators to represent the concept of component similarity according to their preferences. Estimators have different preferences for what component properties need to be similar and how much variation is acceptable for component similarity to exist. The attributes formalized to represent component similarity are:

(1) Component Grouped: The component that is being evaluated for similarity. In the motivating case, the estimator specifies the 'wall' component class to evaluate for component similarity. The component grouped can be different from the component being estimated. For example, the similarity of 'columns' can affect the formwork operations and cost for constructing the 'slab' component.

(2) Direction: The ontology represents the direction for which component similarity will be assessed as either 'horizontal' or 'vertical.' The horizontal direction represents similarity across a single floor. The vertical direction represents similarity across floors. In the motivating case, the drywall estimator evaluates the similarity of the wall components in the horizontal direction, and the concrete estimator evaluates the similarity of the column components in the vertical direction.

(3) Component Variation**:** The overall variation of the components (see (1)) allowed to achieve component similarity. This attribute is needed because estimators have different preferences for the degree of similarity that must be achieved for component similarity to exist. In the motivating case, the estimator prefers that 75-100% of the walls have similar heights (see (4)) when component similarity exists.

(4) Similar Component Properties: The component properties (or property) of the component grouped that will be compared to determine whether the components are similar. In the motivating case, one drywall estimator analyzes the property 'height' and the other drywall estimator analyzes the properties 'height' and 'type' to assess the similarity of wall components.

(5) Property Variation: The variation in the value for the similar component property allowed to achieve similarity. For example, if an estimator specifies 2" for the property variation, then the estimator views wall #1 as similar to wall #2 if its height is at most 2" shorter or taller than wall #2.

FeaGen uses the ontology in its framework to enable estimators to represent the cost-driving features that affect a specific component's construction costs.

3.2 A Framework for Estimators to Represent Cost-driving Features

We implemented the ontology in FeaGen to provide a formal way for estimators to specify the features and properties that affect a component feature's construction costs, to represent the cost-driving component intersections as intersection features, and to define component similarity according to their preferences.

3.2.1 Represent the Generic Features and Properties that Affect a Component's Construction Costs

FeaGen uses the 'feature set' and 'property set' attributes of the ontology in its framework to allow estimators to specify the features and properties that affect a specific component's construction cost. We represent the same component classes that are represented in the IFC's. Figure 7 shows the User Interface from FeaGen and an example 'feature set' and 'property set' specified by the estimator from the motivating case to estimate wall components. If estimators want to add a new feature or property to the 'feature set' or 'property set,' they simply select the generic feature or property from the available features and properties and add it to the 'feature set' or 'property set.' If the desired intersection or macro feature is not available, then estimators need to first create the feature before adding it to the 'feature set,' which is described in the next section. FeaGen represents this knowledge in a project-independent way so that it can be reused from project to project to identify the relevant features and properties when creating a feature-based product model that supports cost estimating.

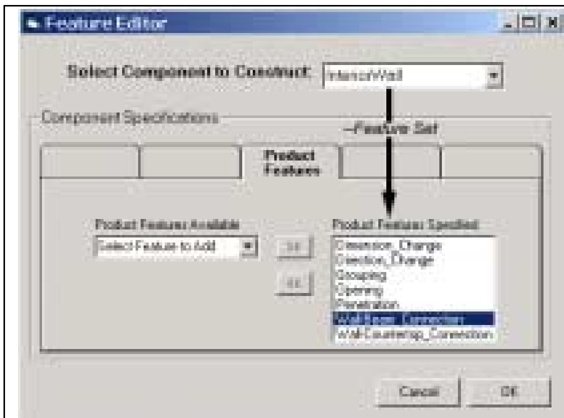


Figure 7a: Example 'feature set' for walls specified by the estimator from the motivating case.

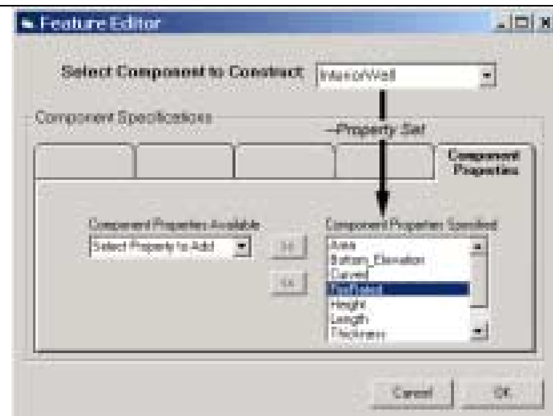


Figure 7b: Example 'property set' for walls specified by the estimator from the motivating case.

Figure 7: FeaGen uses the 'feature set' and 'property set' attributes of the feature ontology in its framework to capture estimators' preferences for the features and properties that affect a specific component's construction costs.

3.2.2 Represent Generic Instances of Intersection Features

FeaGen uses the ‘relating component’ and ‘related components’ attributes of the ontology in its framework to allow estimators to specify the component intersections that affect construction costs using the terminology that the estimator prefers. The ‘property set’ attribute allows estimators to specify the properties of the intersection feature that affect the construction cost of the ‘related component.’ Figure 6 shows instances of intersection features from the motivating case and the attribute values for each feature. If estimators want to create a new intersection feature, they simply have to specify the attribute values of intersection features in the Feature Specification template. For example, if estimators want to represent the intersection of the wall and the ceiling as a feature, they simply have to specify the ‘feature name’ (e.g., “wall-ceiling intersection”), specify “wall” in the ‘relating component’ attribute, specify “ceiling” in the ‘related components’ attribute, and if applicable, specify the relevant properties using the ‘property set’ attribute. Based on the estimator’s selections in the ‘related components’ and ‘property set’ attributes, FeaGen knows what component intersections and feature properties to identify in a given product model for the component specified in the ‘related component’ attribute to create an estimator-focused feature-based product model.

3.2.3 Represent Generic Instances of Macro Features (Component Similarity)

FeaGen leverages the attributes of the ontology to provide a framework that allows estimators to represent their preferences for defining component similarity. Figure 8 shows the Feature Specification template created to capture estimators’ preferences for defining component similarity. Estimators specify the properties of the component that need to be evaluated for similarity using the ‘similar component property’ attribute and the degree of similarity that needs to exist using the ‘component variation’ and ‘property variation’ attributes. Estimators can use these attributes to represent a variety of definitions for component similarity. Figure 8 shows an estimator’s preference that 90-100% of the wall heights be similar for component similarity to exist.

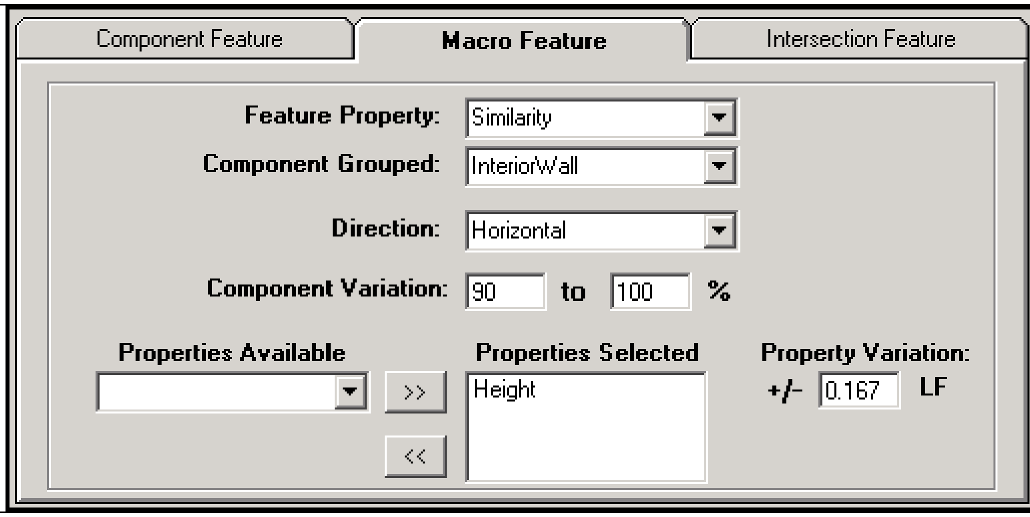


Figure 8: FeaGen leverages the feature ontology to provide a framework for estimators to define component similarity. The example from the motivating case shows an estimator's preference that 90-100% of the walls have similar wall heights for component similarity to exist.

The next section describes how FeaGen creates a project-specific feature-based product model using the generic cost-driving features specified by the estimator.

3.3 Create Feature-based Product Models to Support Construction Cost Estimating

FeaGen transforms a designer-focused product model into a feature-based product model that supports cost estimating uses the generic cost-driving features specified by the estimator (Figure 9). In FeaGen, the input 3D model is created using Bricsnet Architecturals (Bricsnet Architecturals 2001). The representation of product models in Bricsnet is very similar to product models represented using the IFC's. FeaGen analyzes the geometry and topological relationships between the components in the input IFC-based product model to identify the generic cost-driving features and properties specified by the estimator. The output is a project-specific feature-based product model that explicitly represents the features and properties that are important to the cost estimator.

To create a project-specific feature-based product model, FeaGen executes the four steps shown in Figure 9:

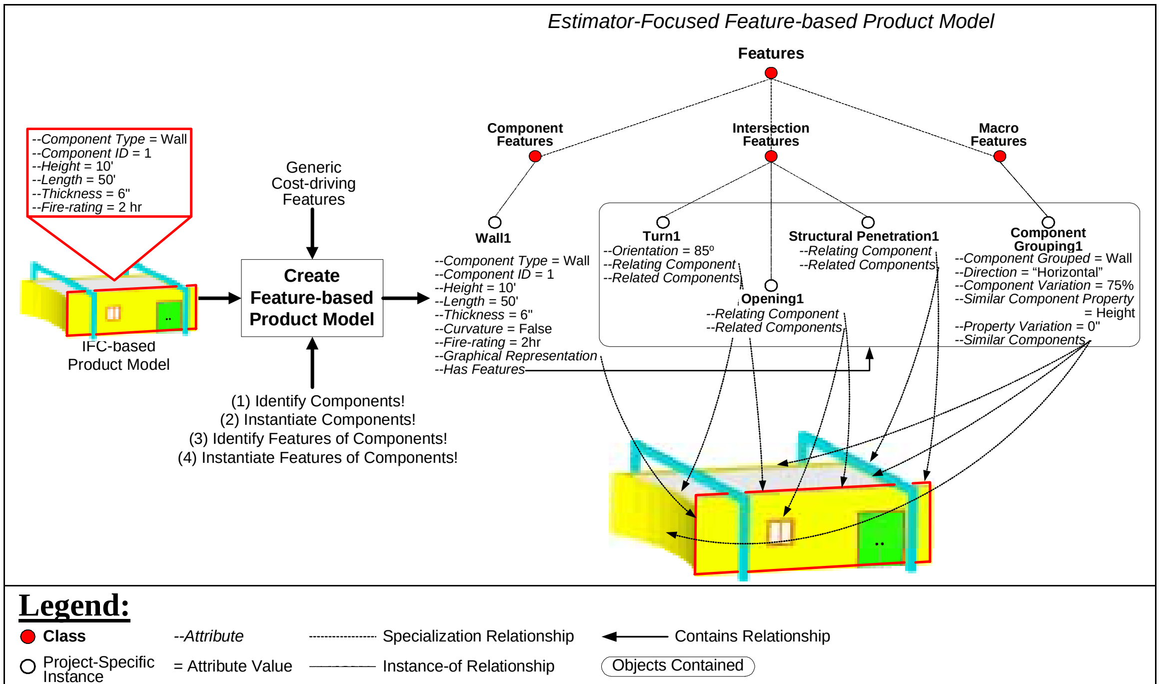


Figure 9: FeaGen transforms an IFC-based product model into a project-specific feature-based product model that explicitly represents the cost-driving features specified by the estimator.

(1) Identify Components: The estimator selects the component class to estimate. In the motivating case, the estimator selects the component class "wall." FeaGen identifies instances of the component in the input IFC-based product model. In IFC-based product models, components are explicitly represented as features so FeaGen simply has to query the input product model to identify them.

(2) Instantiate Components: FeaGen creates instances for each component identified in the input IFC-based product model. For example, FeaGen creates an instance of the component "Wall1." FeaGen relates the new component instance to its corresponding geometry using the 'graphical representation' attribute. Then, based on the ontology described in the previous section, FeaGen uses the 'property set' attribute for the component being estimated to determine what component properties are important to the cost estimator. For each component property in the 'property set,' FeaGen copies the appropriate attribute values from the corresponding component in the input IFC-based product model to the output model.

(3) Identify Features of Components: FeaGen uses the 'feature set' attribute for the component being estimated to determine what features are important to the cost estimator. Then, FeaGen analyzes the input IFC-based product model to determine whether those features exist.

(a) Identify Intersection Features: To identify intersection features, FeaGen reasons about the topological relationships between components, which are represented in different ways depending on the intersecting components. Figure 10 shows different representations of the connectivity between components using a Bricsnet 3D model and an IFC-based product model. Some relationships between components are represented explicitly in IFC-based product models. For example, the connections between the wall and the door and between the two walls are explicit in an IFC-based product model because the architect intends for these components to be connected. Consequently, to determine whether the components are intersecting, FeaGen analyzes the relationships between these components using the objects and attributes shown in Figure 10. In contrast, some relationships between components are implicit in IFC-based product models. For example, the connection between the wall and the beam is implicit because it emerges based on the architectural and structural designs. Consequently, conflict detection mechanisms are needed to determine if these components are intersecting. In FeaGen, users need to identify these types of component intersections manually.

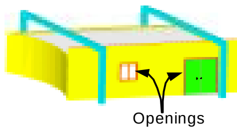
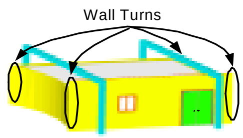
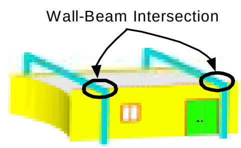
Example Intersection Features from Motivating Case	*Bricsnet Objects and Attributes Analyzed*	*IFC Objects and Attributes Analyzed*	*Description of Reasoning Process*
	<Form.HasOpenings> <Form> <Form.TypeOf>	<IfcBuidlingElement.HasOpenings> <IfcBuildingElement> <IfcBuildingElement.UserDefinedType>	FeaGen queries the objects in the 'HasOpenings' attribute of the component to identify instances of the 'opening' feature.
	<Form.Links.Connections> <Form> <Form.TypeOf>	<IfcBuildingElement.ConnectedTo> <IfcRelConnectsElements.RelatedElement> <IfcBuildingElement> <IfcBuildingElement.UserDefinedType>	FeaGen queries the objects connected to the component in the 'ConnectedTo' attribute to find the intersecting components, and then analyzes the 'TypeOf' attribute of each intersecting component to determine if it is appropriate.
	<Form> <Form.Typeof = Wall> <Form> <Form.Typeof = Beam>	<IfcWall> <IfcBuildingElement.UserDefinedType> <IfcBeam> <IfcBuildingElement.UserDefinedType>	The topological relationships between these components are not explicit. Users identify these intersection features manually in FeaGen.

Figure 10: Objects and attributes of Bricsnet and IFC models analyzed to identify intersection features. Some component intersections are explicitly represented in IFC-based product models, such as the wall and door connection. However, some component intersections are implicit in IFC-based product models, such as the wall and beam connection, and require design conflict detection mechanisms or manual interpretation to identify.

(b) Identify Macro Features (Component Similarity): To identify component similarity, FeaGen reasons about the properties of building components of the same type to determine if the property values are similar. FeaGen identifies the relevant instances of building components in the IFC-based product model based on the component class specified by the estimator in the 'component grouped' attribute. If the estimator specified "horizontal" in the 'direction' attribute, then FeaGen evaluates the building components on a single floor. If the estimator specified "vertical" in the 'direction' attribute, then FeaGen evaluates the building components on all the floors. Then, FeaGen analyzes each property of the building component specified in the 'similar component properties' attribute. FeaGen cycles through each building component instance and compares it to the previous one to determine if the components are similar. FeaGen compares the property values to determine if the variation is acceptable based on the estimator's preferences in the 'property variation' attribute. If the value of the component property is within an acceptable range, then FeaGen considers that component to be similar and adds it to a collection containing the similar components. After

FeaGen has evaluated all the properties of all the components, it calculates the percentage of similar components by dividing the number of similar components collected by the number of components evaluated. If the percentage calculated is within the range specified by the estimator in the 'component variation' attribute, then FeaGen considers the components to be similar.

(4) Instantiate Features of Components: FeaGen creates feature instances for each intersection feature identified in the input product model. For example, for "Wall1", FeaGen creates instances of intersection features for the 'wall-beam intersection', the 'openings', and the wall 'turns.' FeaGen relates the intersection features instantiated to the component using the 'has features' attribute of the component in the project-specific feature-based product model. Then, FeaGen uses the 'property set' attribute of the intersection feature to determine what properties of the feature are important to the cost estimator. For each property in the 'property set,' FeaGen analyzes the intersecting components in the input product model to determine the value of the property. For example, FeaGen analyzes the two walls for the feature 'turn' to determine the orientation of the wall turn and assigns the corresponding attribute values to the feature property in the project-specific feature-based product model. For macro features, FeaGen adds the instances of similar components to the 'similar components' attribute in the project-specific feature-based product model (Figure 9).

FeaGen generates a project-specific feature-based product model that explicitly represents the features and properties that are important to estimators. The feature ontology provides the blue-print for the additions and changes needed to transform an IFC-based product model into a product model that is useful to cost estimators of building construction. In other words, the feature ontology provides the map to relate an IFC-based product model to an estimator-focused product model. Our tests show that the estimator-focused feature-based product model enables estimators to generate and maintain cost estimates more quickly, consistently, and accurately than cost estimating applications that leverage IFC-based product models.

4. Validation

The goals of this research were to provide a formal, general, and flexible way for estimators to represent the different design conditions that affect construction costs. Our tests provide evidence that the feature ontology meets these criteria. We performed a charrette test (Clayton et al. 1998) and three retrospective tests to demonstrate the power and generality of the feature ontology.

To demonstrate power and generality, we provide evidence that shows that the feature ontology enabled cost estimators to:

- Represent the design conditions that affect construction costs more explicitly than IFC-based product models,
- Represent the design conditions that affect construction costs for two different component types,
- Specify their preferences for representing the design conditions that affect construction costs, and
- Generate and maintain cost estimates more accurately, consistently, and quickly with feature-based product models than with IFC-based product models.

The next sections describe the evidence for power and generality in more detail.

4.1 Evidence that cost estimators can represent the design conditions that affect construction costs more explicitly than IFC-based product models

To evaluate the extent to which the feature ontology can represent features explicitly, we wanted to show that the formal structure of the feature ontology enabled estimators to represent the variety of features that affect construction costs explicitly. We created a theoretical ideal to represent the component, intersection, and macro features that are important for interior wall and concrete column construction. We crafted this ideal based on interviews with five estimating experts of interior wall and concrete column construction. Table 1 shows the different features represented in the theoretical ideal for the four test cases. The macro features represented in the theoretical ideal focus on representing different definitions of component similarity and exclude other types of macro features. We compared the different features represented in the feature ontology and in the IFC's with the theoretical ideal to assess the ability of the feature ontology to support the explicit representation of features relevant for cost estimators. We assigned one point for each feature represented explicitly using the feature ontology and the IFC's. Table1 shows the features represented explicitly using the feature ontology and the IFC's. These results suggest that the feature ontology explicitly represents more cost-driving features than the IFC's. These results also show that the features formalized in the feature ontology approach the theoretical ideal for the three feature types shown.

Features	Theoretical Ideal		Feature Ontology	IFC's
Component Features	Walls	1	1	1
	Columns	1	1	1
Intersection Features	Turns	1	1	0
	Openings	1	1	1
	Penetrations	1	1	0
	Wall-Beam Intersections	1	1	0
	Wall-Countertop Intersections	1	1	0
	Column-Slab Intersections	1	1	0
Macro Features (Component Similarity)	Similarity of Height	1	1	0
	Similarity of Width	1	1	0
	Similarity of Type	1	1	0
	Similarity of Shape	1	1	0
	Similarity of Location	1	0	0
Features Represented Explicitly		**13**	**12**	**3**

Table 1: Comparison of the different features represented in the feature ontology and in the IFC's with the theoretical ideal to assess the ability of the feature ontology to support the explicit representation of features. The theoretical ideal represents the component, intersection, and macro features that estimating experts confirmed are important for interior wall and concrete column construction. These results suggest that the feature ontology represents cost-driving features more explicitly than the IFC's and that the features represented in the feature ontology approach the theoretical ideal. Hence, the validation tests demonstrate the power of the feature ontology to represent the features that affect construction cost explicitly.

4.2 Evidence that cost estimators can represent the design conditions that affect construction costs for two different component types

To demonstrate the generality of the feature ontology, we modeled costs for two different component types in three retrospective test cases. Different features and feature properties impact costs for these two component types. Table 2 shows the different features represented in the feature ontology for each component type. The ability of practitioners to represent different features for different component types demonstrates the generality of the feature ontology.

Component Features	Intersection Features	Macro Features *(Component Similarity)*
Walls	Turns	Similarity of Height
	Openings	Similarity of Width
	Penetrations	Similarity of Type
	Wall-Beam Intersections	
	Wall-Countertop Intersections	
Columns	Column-Slab Intersection	Similarity of Height
		Similarity of Width
		Similarity of Length
		Similarity of Shape

Table 2: Features represented in the feature ontology for two test cases on walls and columns. The ability of practitioners to represent different features for different component types provides evidence for the generality of the feature ontology.

4.3 Evidence that cost estimators can specify their preferences for representing the design conditions that affect construction costs

As evidence that the feature ontology is general, we demonstrated that 13 different estimators could specify their preferences for representing the features that affect construction costs in the different test cases. We tested whether four different estimators of drywall construction can specify their preferences for representing features in FeaGen. We also tested whether estimators from two different construction domains can represent their preferences in the three retrospective test cases. Finally, we tested whether the eight practitioners in the charrette test can specify their preferences for representing features in FeaGen. These tests demonstrate generality across user types and suggest that the feature ontology is sufficiently flexible to represent different estimators' preferences.

4.4 Evidence that cost estimators can generate and maintain cost estimates more accurately, consistently, and quickly with feature-based product models than with IFC-based product models

We developed a prototype cost estimating application called ACE to test whether the feature ontology helped estimators to generate and maintain cost estimates more accurately, consistently, and quickly than current methods. ACE automatically customizes activities and resources based on the features in the estimator-focused feature-based product model created in FeaGen (Figure 11). In the remainder of this section, we first describe how ACE generates and maintains cost estimates with estimator-focused feature-based product models, and then describe how we used ACE to validate the feature ontology.

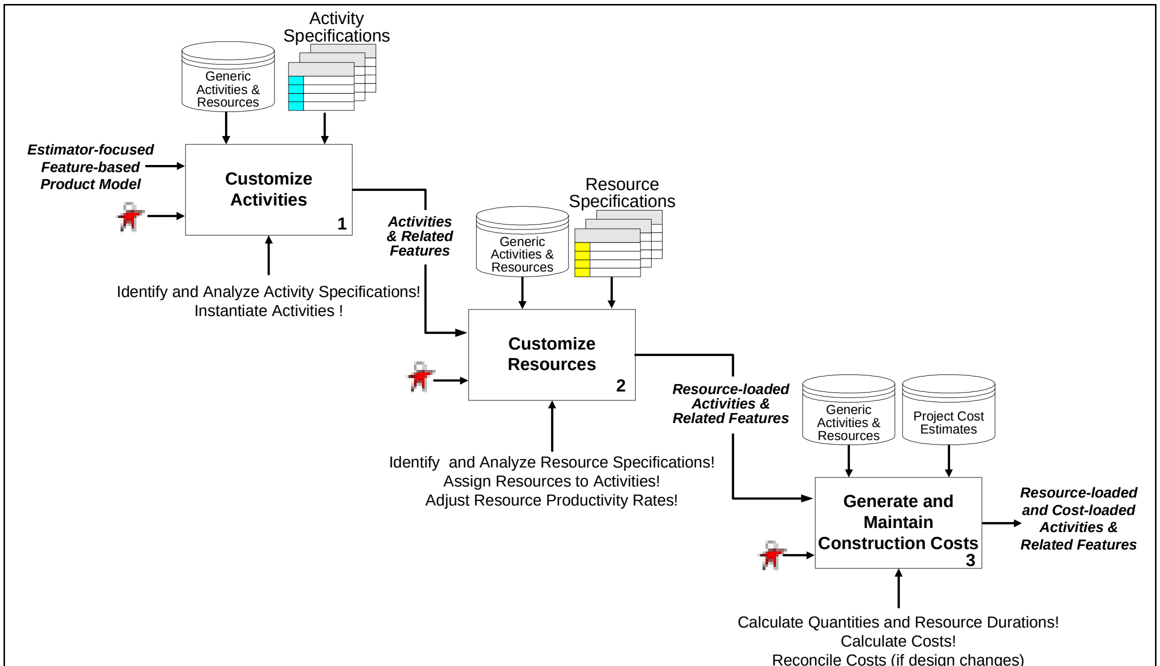


Figure 11: The different steps of the activity and resource customization process we formalized and implemented in ACE to support the generation and maintenance of construction cost estimates with feature-based product models. We used ACE to validate the feature ontology by showing that estimator-focused feature-based product models enable estimators to generate and maintain cost estimates more accurately, consistently, and quickly than IFC-based product models.

Estimators using ACE first represent their preferences for how the cost estimate should be adjusted for each of the generic cost-driving features specified by the estimator in FeaGen. Estimators adjust the project's activities, resources, and resource productivity rates to account for the cost impact of different features. We created different templates to provide a formal way for estimators to specify the features that affect activities (Activity Specification templates) and the features that affect resources (Resource Specification templates) (Staub-French et al. 2002a). Figure 12 shows the attributes of Activity and Resource Specifications and examples of estimators' rationale for adjusting activities and resources for the features from the motivating case. Activity Specification templates capture estimators' rationale about how and when activities are required for different features. Estimators fill in Activity Specification templates by specifying the *feature* that requires the activity, the *design condition* that dictates when the feature requires the activity, the *activity* (represented as an action-object pair) to instantiate if the feature exists and the design condition is satisfied, and the *cost implication* of the activity. Resource Specification templates capture estimators' rationale about when resources are required for a given activity and when and how to adjust resource productivity rates for different features. Estimators fill in Resource Specification templates by specifying the *activity* (represented as an action-object pair), the *resource*, and the *design condition* that dictates when the *feature* affects the resource, and if applicable, the *adjustment* to make to the resource's productivity. ACE leverages the estimator's rationale captured by the templates to generate and maintain construction cost estimates.

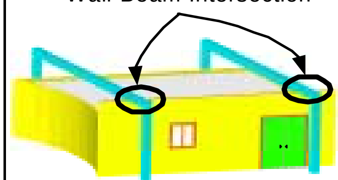
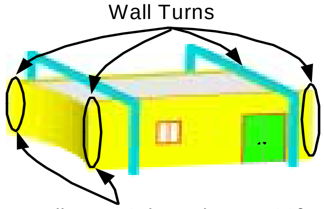
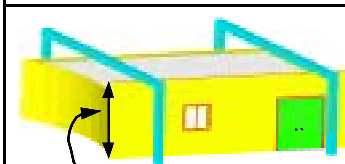
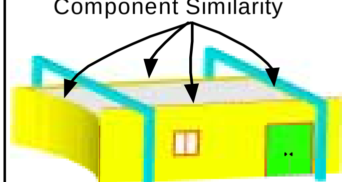
Relevant Design Conditions	*Estimator's Rationale about Activities and Resources Required for Design Conditions*	*Estimators Rationale Input in Activity and Resource Specification Templates*
Wall-Beam Intersection 	Add Activity "Apply Caulk" if the intersecting wall is fire-rated.	*Activity Specification #1* **Feature**: Wall **Action**: Layout **Object**: Wall **Cost Implication**: Resource Design Condition – **Feature**: Wall Design Condition – **Constraint**: Curved = True
Wall Turns  Wall Turn Orientation <> 90°	Add Activity "Layout Wall" if the orientation of the wall turn is not 90°.	*Activity Specification #2* **Feature**: Turn **Action**: Layout **Object**: Wall **Cost Implication**: Resource Design Condition – **Feature**: Turn Design Condition – **Constraint**: Orientation <> 90
 Wall Height	Use Rolling Scaffolding if the wall height is between 9' - 13'.	*Resource Specification #1* **Action**: Install **Object**: Metal Stud **Resource**: Rolling Scaffolding Design Condition – **Feature**: Wall Design Condition – **Constraint**: 9'<= Height <=13'
Component Similarity 	Increase the base crew productivity rate 10% if 75-100% of the walls have the same height.	*Resource Specification #2* **Action**: Install **Object**: Metal Stud **Resource**: Crew P.R. **Adjustment**: Increase 10% Design Condition – **Feature**: Grouping Design Condition – **Constraint**: 75-100% Walls have Similar Heights

Figure 12: Estimators use Activity and Resource Specification templates in ACE to represent their preferences for how activities and resources should be adjusted for each of the generic cost-driving features specified by the estimator in FeaGen.

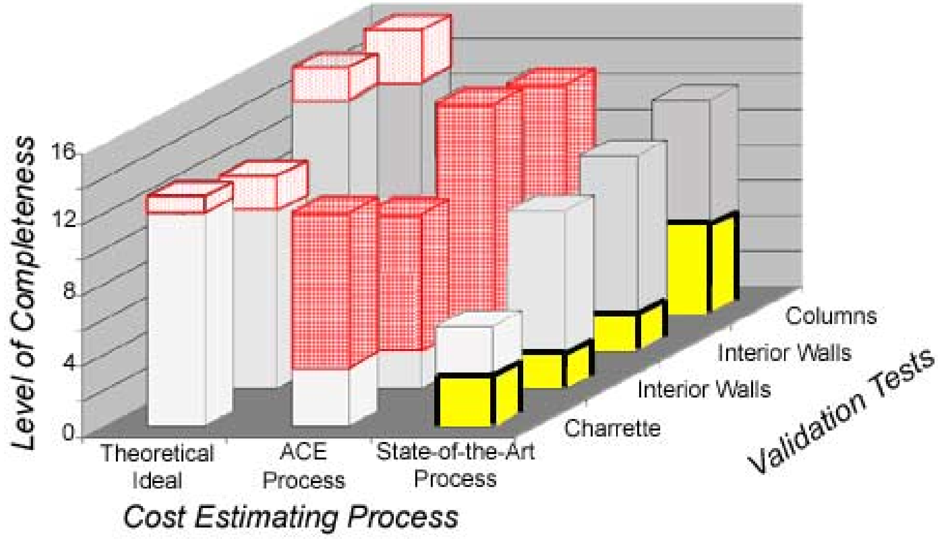
In ACE, we implemented a formal process that automatically customizes activities and resources when generating and maintaining cost estimates for estimator-focused feature-based product models (Staub-French et al. 2002b). Figure 11 shows that ACE executes three steps to generate and maintain construction cost estimates for feature-based product models:

(1) ***Customize Activities*:** ACE customizes the activities for each component feature being estimated based on the estimator's rationale in Activity Specifications and the features in the estimator-focused feature-based product model.

(2) ***Customize Resources:*** ACE customizes each activity's resources and resource productivity rates based on the estimator's rationale in Resource Specifications and the particular features in the estimator-focused feature-based product model.

(3) ***Generate and Maintain Construction Costs:*** ACE calculates each activity's quantities and duration to determine the activity's labor and material cost. If the estimate is based on a revised design, ACE identifies the cost information affected and reconciles the activities and resources so that the design and estimate remain in balance. ACE creates cost estimates consisting of resource-loaded and cost-loaded activities that are explicitly related to the features in the estimator-focused feature-based product model.

We used ACE to demonstrate that the estimator-focused feature-based product model helps estimators to generate and maintain cost estimates more accurately, consistently, and quickly than IFC-based product models (Staub-French 2002). We evaluated the *level of completeness* of estimates generated by 13 estimators using ACE and compared them to estimates generated by the same estimators using Timberline's state-of-the-art Precision Estimating (PE) software (Timberline 2001). We used level of completeness to measure the extent to which estimators accounted for the cost impacts of features explicitly. If estimators used ad hoc methods or overlooked the cost impact of features, they received a lower score for completeness. We defined a theoretical ideal to represent the "most complete" estimate for each test case. We crafted the theoretical ideal based on interviews with estimating experts of interior wall and concrete column construction. The theoretical ideal represents the cost impacts of features explicitly and excludes ad hoc methods used by estimators. The results of the validation tests demonstrate that the estimator-focused feature-based product model enabled estimators using ACE to generate and maintain more complete cost estimates than the same estimators using state-of-the-art software that uses IFC-based product models. Estimators can generate and maintain cost estimates that are less ad hoc and contain fewer omissions with feature-based product models than with IFC-based product models. The charrette test also demonstrated that practitioners using ACE were able to more consistently identify the correct cost impact and identify the cost impacts 17% faster using ACE when compared with state-of-the-art tools using IFC-based product models. Figure 13 summarizes the level of completeness results for the four validation tests. Therefore, the four validation tests demonstrate the power of the feature ontology by showing that the estimator-focused feature-based product model helped estimators to account for the cost impact of features more accurately (completely), consistently, and quickly using ACE than with state-of-the-art software that uses IFC-based product models.



Legend:

Improvement in *level of completeness* of estimates generated by practitioners using ACE when compared with state-of-the-art tools.

Disparity in *level of completeness* of estimates generated by practitioners using ACE when compared with the theoretical ideal.

Extent to which state-of-the-art tools helped the estimator to account for the cost impact of features explicitly.

Figure 13: Summary of results for the four validation tests. We tested the level of completeness of estimates generated by practitioners using ACE compared with state-of-the-art tools. We used level of completeness to measure the extent to which the estimator-focused feature-based product model helped estimators to account for the cost impacts of features explicitly when generating and maintaining cost estimates. The theoretical ideal represents the "most complete" estimate for each test case. The height of each bar indicates the level of completeness achieved by estimators using ACE and state-of-the-art tools relative to the theoretical ideal for each validation test. Results of the four tests show that estimators using the feature-based product model can generate substantially more complete cost estimates than with IFC-based product models and approaches the theoretical ideal.

5. Conclusions

Current industry standard representations of building product models, such as IFC-based product models, do not represent many of the different design conditions that affect construction costs. Our research formalizes a vocabulary to describe the different types of design conditions that affect construction costs. The feature ontology enables estimators to represent their preferences for the different features that affect a specific component's construction costs. We also formalized a framework to capture this knowledge from estimators and represent features in a project-independent way so that this knowledge can be reused to create feature-based product models that support construction cost estimating.

The feature ontology presented in this paper is limited in several ways. The feature ontology does not represent all the types of features that affect construction costs. For example, material features, such as the workability of concrete, can affect construction costs. Similarly, the proximity of a duct run to a pipe run can lead to increased congestion and affect the cost of installing pipe. The feature ontology is also unable to represent similarity of location, as shown in the theoretical ideal in Table 1. Although we implemented mechanisms to identify most intersection features and component similarity, we do not claim that these mechanisms are general to identify these features for different types of product model representations. We also recommend additional testing to validate the generality of the feature ontology.

Automating the generation of feature-based product models that support construction cost estimating has the potential to significantly reduce the time it takes to generate and maintain construction cost estimates. Today, estimators spend significant amounts of time analyzing building designs to identify all the project-specific instances of cost-incurring design conditions in a given product model. If these design conditions could be identified automatically, estimators could provide cost feedback in significantly less time. As a result, project teams could perform what-if analyses on different designs and explore a larger variety of design alternatives to identify the lowest cost design. Moreover, estimators could provide feedback to designers on the specific features that are impacting construction costs. Hence, project teams can leverage feature-based product models to develop more cost-effective and constructable designs in less time.

6. Acknowledgements

We gratefully acknowledge the support of the Center for Integrated Facility Engineering at Stanford University. We also thank the National Science Foundation for the partial support of this work under grant 9625228. We also acknowledge the financial support of the Future Professors of Manufacturing Program, Hathaway-Dinwiddie Construction Company, Flad & Associates, Mazzetti & Associates, Rosendin Electric, Paragon Mechanical, and Rountree Plumbing and Heating. We also thank the following people and their respective companies for their participation in this research:

- Melody Spradlin, Gregg Thoman, and Jeff Lindell from Hathaway Dinwiddie Construction Company,
- Chris Crouse from Rountree Plumbing and Heating,
- Chris Sorauf from Rosendin Electric,
- Jim Brady from Paragon Mechanical,
- Craig Vargas from California Drywall,
- Christopher Stewart from Flad and Associates,
- Dean Read, Jim Ray, Mark Qualcomm, and Jay Wilson from DPR Construction,
- John Sempek from Newcon Construction, Inc.,
- Bill Russell from Vance Brown, Inc., and
- Jim Dick from Pankow.

7. References

Aouad, G., Betts, M., Brandon, P., Brown, F., Child, T., Cooper, G., Ford, S., Kirkham, J., Oxma, R., Sarshar, M., and Young, B. (1994). "*ICON: Integration of Construction Information*." Department of Surveying and Information Technology Institute, University of Salford, Salford.

Aouad, G., Child, T., Marir, F., and Brandon, P. (1997). "*Open Systems for Construction (OSCON), draft industry report*." Department of Surveying, University of Salford, Salford.

Bjork, B. C. (1991). "A Unified Approach for Modeling Construction Information." Building and Environment, 27(2), 173-194.

Bricsnet (2001). Architecturals, Users Documentation, Portsmouth, New Hampshire.

Clayton, Mark J., Kunz, John C., and Fischer, Martin A. (1996). "Rapid Conceptual Design Evaluation Using a Virtual Product Model." Engineering Applications of Artificial Intelligence, 9(4), 439-451.

Clayton, Mark. J; Kunz, John C.; and Fischer, Martin A. (1998). "The Charrette Test Method." *Technical Report 120*, Center for Integrated Facility Engineering, Stanford, CA.

Cunningham, J.J. and Dixon, J.R. (1988). "Designing with Features: The Origin of Features," *ASME Computers in Engineering Conference*, San Francisco, CA, Aug. 1988, 237-243.

de Sousa, U.E.L. and Thomas, H.R. (1996). "Development of an Explanatory Model for Concrete Formwork Productivity." *7th International Symposium on the Organization and Management of Construction*, CIB, W-65, Glasgow, Scotland, Aug. 28, 1996, (2)27-38.

Fischer, M. (1991). "Constructibility Input to Preliminary Design of Reinforced Concrete Structures." *Technical Report 64*, Center for Integrated Facility Engineering, Stanford.

Froese, T., Yu, K., and Shahid, S. (1996). "Project Modeling in Construction Applications," *Computing in Civil Engineering: Proceedings of the Third Congress*, ASCE, Anaheim, June, 1996, 572-578.

Gielingh, W.F. (1988). "General AEC Reference Model," ISO TC184/SC4/WG1 doc. 3.2.2.1. TNO report BI-88-150.

Hanna, A.S., and Sanvido, V.E. (1990). "Interactive Vertical Formwork Selection." *Concrete International: Design and Construction*, 12(4), 26-32.

Hyer, N.L., Wemmerlov, U. (1992). "Group Technology." *The Handbook of Industrial Engineering*, editor Gavriel Salvendy. New York: John Wiley and Sons, 464-489.

IDEF0. (1981). "Integrated Computer-Aided Manufacturing (ICAM), Architecture Part II, Vol. IV- Function Modeling (IDEF-0)." SoftTech, Inc., Waltham, MA.

International Alliance of Interoperability (IAI) (2001). "IFC 2x Extension Modeling Guide", Available from http://www.iai.org.uk.

Laitinen, J. (1998). "Model Based Construction Process Management," *PhD Thesis*, Royal Institute of Technology. Stockholm, Sweden.

Sanders, S.R. and Thomas, H.R. (1991). "Factors Affecting Masonry-Labor Productivity." *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 117(4), 626-644.

Shah, J.J. (1991). "Assessment of Features Technology," *Computer-Aided Design*, 23(5), 331-343.

Slaughter, S. (2000). “The Link Between Design and Process: Simulation Modeling of Construction Activities.” *Proceedings of the Construction Congress VI*, Orlando, Florida, 1051-1057.

Smith, G. R., and Hanna, A. S. (1993). “Factors Influencing Formwork Productivity.” *Canadian Journal of Civil Engineering*, 20(1), 144-153.

Staub-French, Sheryl (2002) “Feature-Driven Activity-based Cost Estimating.” *PhD Thesis*, Stanford University, Stanford.

Staub-French, S., Fischer, M., Kunz, J., Paulson, B. and Ishii, K. (2002a). “An Ontology for Relating Features of Building Product Models with Construction Activities to Support Cost Estimating.” *Working Paper #70*, Center for Integrated Facility Engineering, Stanford.

Staub-French, S., Fischer, M., Kunz, J., Paulson, B. and Ishii, K. (2002b). “A Formal Process to Create Resource-loaded and Cost-loaded Activities Related to Feature-based Product Models.” *Working Paper #71*, Center for Integrated Facility Engineering, Stanford.

Stumpf, A., Ganeshan, R., Chin, S. and Liu, L. (1996). “Object-Oriented Model for Integrating Construction Product and Process Information.” *Journal of Computing in Civil Engineering*, 10(3), 204-212.

Thomas, H.R. and Sackrakan, A. (1994). “Forecasting Labor Productivity Using the Factor Model.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 120(1), 229-239.

Thomas, H.R. and Zavrski, I. (1999). “Construction Baseline Productivity: Theory and Practice.” *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 125(5), 295-303.

Timberline Software Company (2001). Precision Estimating Extended and CAD Integrator, Users Documentation, Beaverton, Oregon.

van Leeuwen, J.P. (1999). “Modelling Architectural Design Information by Features,” Ph.D. Thesis. Eindhoven, The Netherlands: Eindhoven University of Technology.