Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2003), *17*, 133–154. Printed in the USA.
Copyright © 2003 Cambridge University Press 0890-0604/03 $16.00
DOI: 10.1017/S0890060403172034

A feature ontology to support construction cost estimating

SHERYL STAUB–FRENCH,[1] MARTIN FISCHER,[2,3] JOHN KUNZ,[3] KOS ISHII,[4] AND BOYD PAULSON[2]

[1]Department of Civil Engineering, University of British Columbia, 2324 Main Mall, Vancouver, British Columbia V6T 1Z4 Canada
[2]Department of Civil and Environmental Engineering, Stanford University, Stanford, California 94305, USA
[3]Center for Integrated Facility Engineering, Stanford University, Stanford, California 94305, USA
[4]Department of Mechanical Engineering, Stanford University, Stanford, California 94305, USA

(RECEIVED November 21, 2002; ACCEPTED June 30, 2003)

Abstract

Construction cost estimators are confronted with the challenging task of estimating the cost of constructing one of a kind facilities. They must first recognize the design conditions of the facility design that are important (i.e., incur a cost) and then determine how the design conditions affect the cost of construction. Current product models of facility designs explicitly represent components, attributes of components, and relationships between components. These designer-focused product models do not represent many of the cost-driving features of building product models, such as penetrations and component similarity. 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 formal ontology we developed to represent construction knowledge about the cost-driving features of building product models. The ontology formalizes three classes of features, defines the attributes and functions of each feature type, and represents the relationships between the features explicitly. The descriptive semantics of the model allow 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. A software prototype that implements the ontology enables estimators to transform designer-focused product models into estimator-focused, feature-based product models. Our tests show that estimators are able to generate and maintain cost estimates more accurately, consistently, and expeditiously with feature-based product models than with industry standard product models.

Keywords: Activity-Based Cost Estimating; Building Construction; Computer-Integrated Design and Construction; Cost-Driving Product Features; Formal Ontologies; Product and Process Modeling

1. INTRODUCTION

Understanding how the building design influences construction costs is a challenging task for estimators. Estimators must identify the design conditions that are important (i.e., incur a cost) and adjust the activities, resources, and resource productivity rates that form the basis of a cost estimate for a particular design. Current architectural 3-dimensional (3-D) modeling applications can now export semantically rich product models using Industry Foundation Classes [IFCs, International Alliance of Interoperability (IAI), 2001], the primary product model exchange standard for the architecture, engineering, and construction industry. Industry standard IFC-based product models enable the sharing of product models with other software applications. Cost estimating applications leverage IFC-based models by extracting dimensional information for quantity takeoff calculations (Timberline Software Company, 2001). However, in addition to dimensional information, 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 for cost estimating, estimators need a vocabu-

Reprint requests to: Sheryl Staub–French, Department of Civil Engineering, University of British Columbia, 2324 Main Mall, Vancouver, BC V6T 1Z4, Canada. E-mail: sherylsf@civil.ubc.ca

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

lary for describing the different types of design conditions that affect construction costs and a framework to capture this knowledge from estimators so that it can be reused from project to project in feature-based cost analysis.

Previous research efforts identify the different design conditions that affect construction costs (Hanna & Sanvido, 1990; Fischer, 1991; Sanders & Thomas, 1991; Smith & Hanna, 1993; Thomas & Zavrski, 1999). 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, and so forth (Cunningham & Dixon, 1988; Shah, 1991). However, the feature representations developed in the manufacturing industry do not fully support the representation of product models for buildings. Our research applies the manufacturing concept of features to building construction and extends it to represent the features that are useful to construction cost estimators.

This paper describes the ontology we formalized to represent cost-driving features of building product models. The feature ontology classifies the features that affect cost, defines attributes to describe each feature type, and represents the relationships between features explicitly. The feature ontology represents three types of features: *component features* result from components in standard building product models, *intersection features* result from intersections of components, and *macrofeatures* result from the similarity of components. The vocabulary defined in the ontology enables 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. We implemented the ontology in a knowledge-based software prototype that analyzes the geometry of components and the topological relationships between components in standard building product models to identify the cost-driving features that are important to a particular cost estimator. The feature ontology enriches the current representation of the standard building product model by transforming designer-focused product models into estimator-focused feature-based product models that support feature-based cost estimating. The main contributions of the paper are the feature ontology and the computer-interpretable templates that use the attributes of the ontology to capture this knowledge from estimators.

1.1. Case examples

This section describes two case examples that illustrate the practical motivation for this research. The case examples shown in Figure 1 highlight several design conditions that affect drywall and concrete column construction. For each case example, 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.

Estimators must identify the design conditions that affect the project's activities, resources, and resource productivity rates when configuring 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 2 shows the design conditions that two

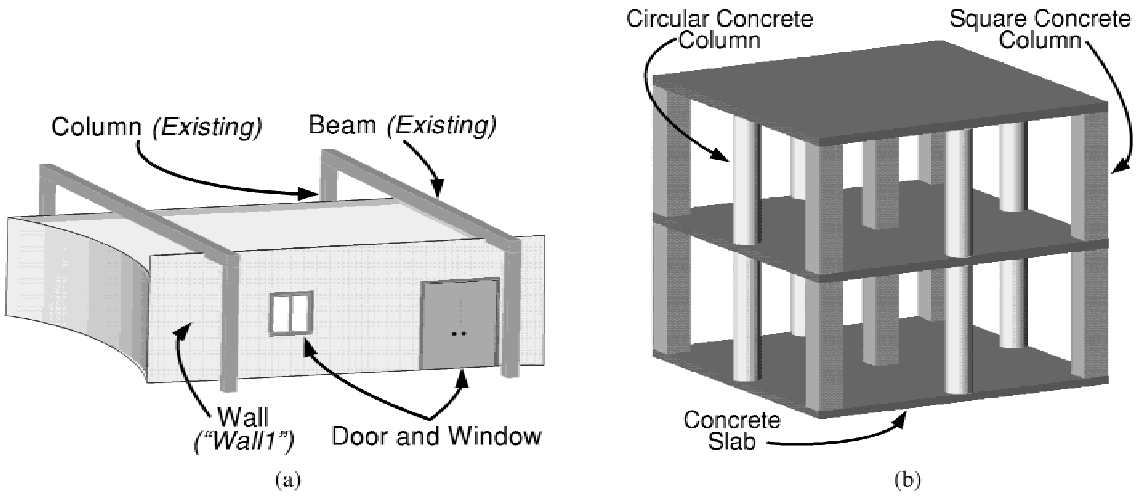


Fig. 1. The building components in drywall and concrete column case examples illustrating the different design conditions that affect construction costs. (a) The building components in the office project case study. The drywall estimator estimates the costs for constructing the four walls shown. (b) The building components in the retail store case study. The concrete estimator estimates the costs for constructing the columns shown.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

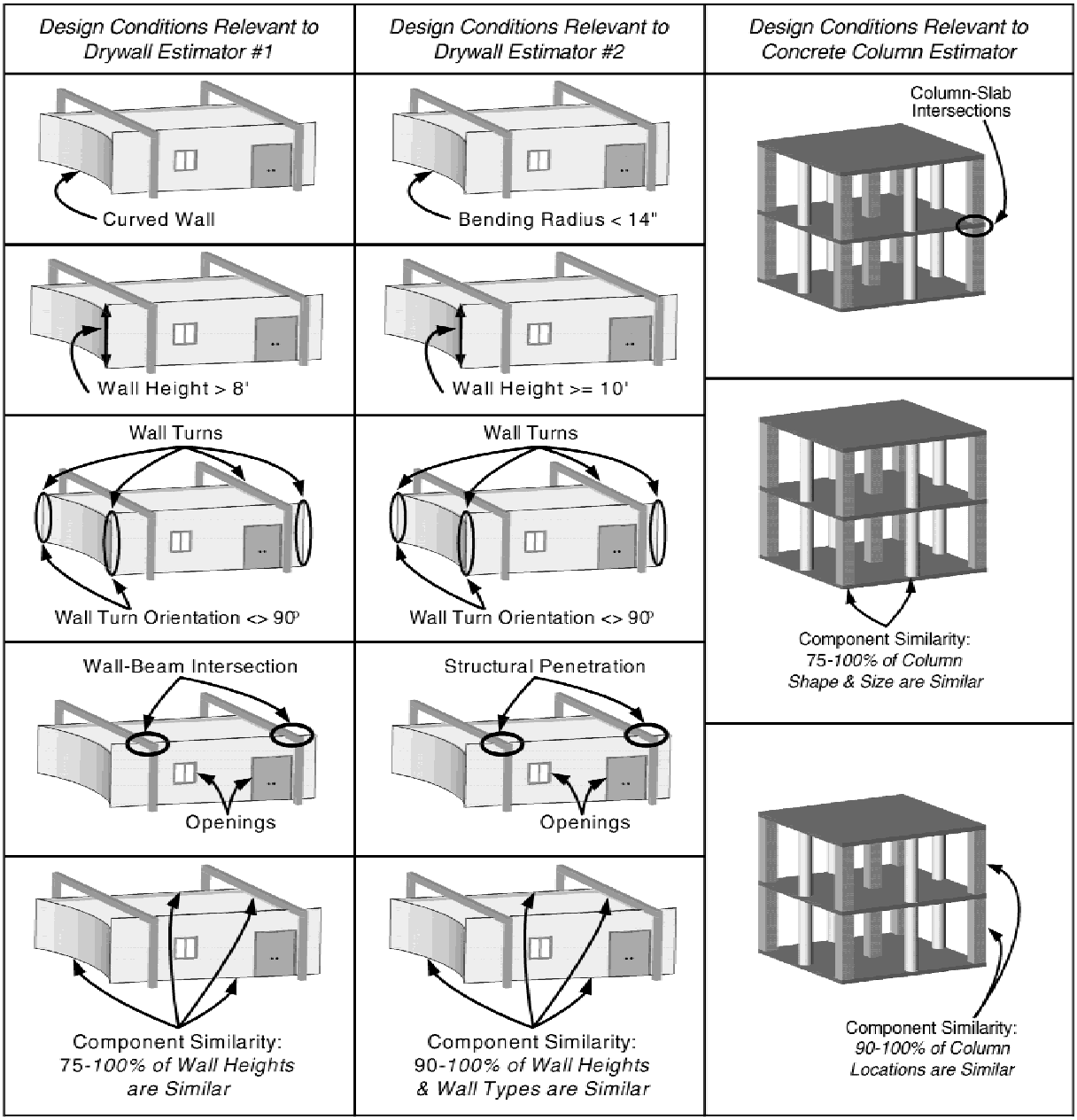


Fig. 2. The design conditions that are important to drywall and concrete estimators for the two case examples.

drywall estimators consider and the design conditions that the concrete column estimator considers. The two drywall estimators consider the same design conditions (e.g., 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).

Current cost estimating software, such as Timberline Software Company's Precision Estimating (2001), and concepts found in the literature (Aouad et al., 1994; Froese et al., 1996; Stumpf et al., 1996; Laitinen, 1998; Marir et al., 1998; Slaughter, 2000) allow estimators to represent their preferences for assembling cost items to construct a component based on design conditions that are specific to the properties of the component, such as its curvature. However, as Figure 2 shows, many other design conditions affect construction costs in addition to component properties. Design conditions can be based on:

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

- 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 in a given design and to manually 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 often employ ad hoc methods (e.g., fudge the productivity rate to account for wall turns), overlook the cost impact of design conditions (e.g., overlook structural penetrations and their cost impact), and account for the cost impact of design conditions inconsistently (e.g., define component similarity in different ways or adjust the productivity rate by different factors to account for it). Consequently, the current inefficient and inconsistent cost estimating process often leads to inaccuracies and omissions in the resulting cost estimate and limits the ability of estimators to provide cost feedback to designers about the cost implications of their design decisions. 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 developed a vocabulary using *features* to represent the different design conditions that are important for estimating the cost 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 *macrofeatures*.

1.2. Research goals

The goals of this research were to formalize an ontology to describe the features that affect building construction costs and to provide a computer-interpretable way for estimators to specify their preferences for the features that affect a specific construction domain. The drywall and concrete case examples demonstrate 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 consistency. The formal representation should include all the attributes necessary for estimators to describe the different design conditions that affect building construction costs (Fig. 2).
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 representation of features also has to be general enough to support cost estimating of different construction domains.
3. *Flexible:* Estimators have different preferences for describing the different design conditions that affect construction costs (Fig. 2). 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 feature ontology we formalized meets these criteria. 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 builds on prior research in ontological modeling and combines and extends previous research in construction cost estimating and product modeling.

2.1. Prior research on ontological modeling

Gruber (1995) defines an ontology as an "explicit specification of a conceptualization." He refers to a conceptualization as the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. The two main elements of ontologies are concepts and relations, where concepts are used to define and explain things and relationships order the concepts, often in a hierarchical structure (Weinstein & Birmingham, 1998; Scholten & Beulens, 2002). Ontologies can be viewed as a framework to represent the semantics of data about a certain domain and are used extensively in artificial intelligence research (Scholten & Beulens, 2002). Ontologies are particularly useful for representing domain-specific knowledge and developing knowledge-based technologies because they provide a common vocabulary, explication of what has been often left implicit, systematization of knowledge, standardization, and metamodel functionality, where the concepts and relations among them are used as building blocks for the model (Mizoguchi, 2001).

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

We use ontologies to represent construction knowledge about cost-driving features of building components. We use the concept of features and define the objects, attributes, and relationships of different classes of features ordered in a hierarchical structure to enable feature-based cost estimation. Our feature ontology is formal in the sense that it explicitly defines the terms, relationships, and hierarchical structure to represent the concepts for describing cost-driving features of building components, and in the sense that it is implemented in the computer. The formal feature ontology enables a knowledge-based program to systematically perform feature-based cost estimation.

2.2. Prior research on construction cost estimating

Many researchers identify the design conditions that affect the cost of building construction (Hanna & Sanvido, 1990; Fischer, 1991; Sanders & Thomas, 1991; Smith & Hanna, 1993; Thomas & Zavrski, 1999). For example, Hanna and Sanvido (1990) recognize that component similarity limits the applicability of different formwork systems, and Thomas and Zavrski (1999) 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.

2.3. 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 formalized by the IFCs but extend building product models using the concept of features to represent the design conditions that affect construction costs. 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 & 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 wide range of specializations of firms serving the building construction industry heightens the need for user customizability. In summary, our research applies the manufacturing concept of features to building construction and extends it to represent the features that are useful to construction cost estimators.

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

Cunningham and Dixon (1988) formalize the classification of intersection features and macrofeatures, but they do not formalize attributes of each feature type to enable practitioners to create or customize instances of intersection and macrofeatures. We formalize the different attributes of intersection and macrofeatures 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 IFCs explicitly represent the connectivity between components using the RelatingElement and RelatedElements attributes, which we use to represent intersection features. However, the IFCs 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 whereas the wall's connections to other walls (wall turns) and to the beam (structural penetration) were important because these connections impact drywall 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. The results from our research provide 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 (Cunningham & Dixon, 1988; Hyer & Wemmerlov, 1984; Shah, 1991). For example, Cunningham and Dixon (1988) determined that "feature sets" could be deduced by a process-activity

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

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 IFCs represent property sets for different component types to represent the properties that are important to designers. However, the IFCs 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 THE FEATURES THAT AFFECT CONSTRUCTION COSTS

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 2 general contractors and 12 subcontractors that self-perform construction work on drywall, structural concrete, mechanical 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, Fig. 3). 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 user-customizable 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, 2002).

The next sections describe the feature ontology (Section 3.1), the Feature Specification templates and mechanisms developed to represent cost-driving features generically in the computer (Section 3.2), and the mechanisms implemented to create project-specific feature-based product models (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, different types of design conditions affect construction costs, estimators have different preferences for representing design conditions, and 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 and implementing the generic feature descriptions in user-customizable templates.

3.1.1. Classifying features

The case examples demonstrated that features can be components (e.g., a wall), features can emerge from intersections of components (e.g., the feature *turn* emerges from intersections of walls), and features can emerge from groupings of components based on their similarity (e.g., wall similarity based on similar wall heights). Consequently, we classify features into the following types:

1. *Component Features:* Features that result from components in an IFC-based building product model, such as walls and columns.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

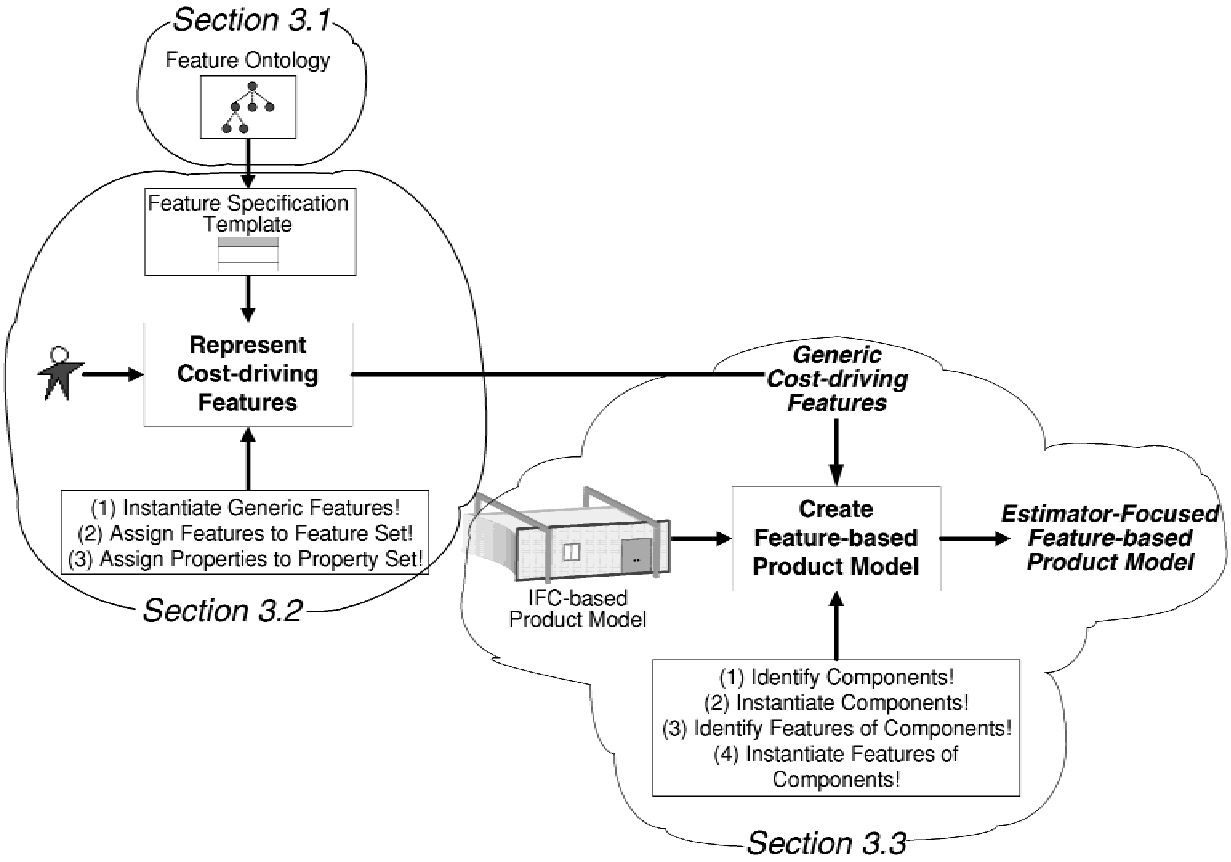


Fig. 3. FeaGen uses the attributes of the feature ontology to capture estimators' preferences for the features and properties that affect construction costs in Feature Specification templates. FeaGen represents the cost-driving features and properties generically specified by the estimator so that they 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 an estimator-focused feature-based product model given an IFC-based product model.

2. *Intersection Features:* Features that result from intersections of components, such as penetrations and turns.
3. *Macrofeatures:* Features that result from prespecified combinations of other features. We focused on macrofeatures that emerge based on the similarity of components of the same type.

Figure 4 shows the three feature types formalized in this research and examples of each feature type from the drywall case example. Each feature type has different attributes that help estimators represent feature instances according to their preferences.

3.1.2. Common attributes of features

The attributes of the three feature types 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 5 shows the common attributes of each feature type. Some attributes are common to all features, and other attributes are specific to a feature type. The common attributes of all features are as follows:

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 drywall case example, one drywall estimator called the intersection of the wall and the beam a wall–beam intersection and another called it a structural penetration (Fig. 2).
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 drywall case example represented orientation in the property set of the feature *turn*. We use properties to describe the attributes of features to be consistent with the terminology used in the IFCs.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

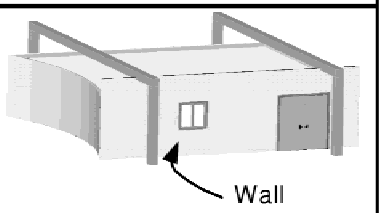
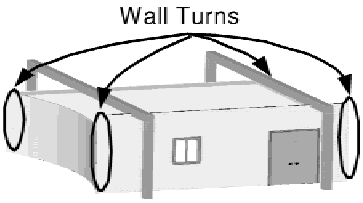
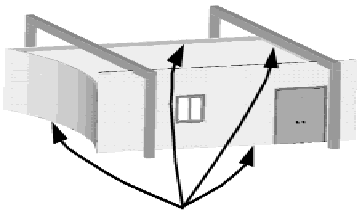
Feature Type	Feature Type Definition	Example from Case
Component Features	Features that result from components in an IFC-based building product model.	
Intersection Features	Features that result from intersections of components.	
Macro Features	Features that result from pre-specified combinations of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

Fig. 4. The ontology classifies features into three types: component features, intersection features, and macrofeatures.

3.1.3. 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 drywall case example represented wall–beam intersections, turns, and openings in the feature set for walls.

3.1.4. Attributes of intersection features

The two attributes of intersection features are based on the attributes used by the IFCs 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 intersections of components. The two attributes of intersection features are:

1. *Relating Component:* The component class that the estimator is estimating. From the case examples, the estimators considered 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 specified wall for the relating component and beam for the related components. To represent the opening feature, the estimator specified wall for the relating component and door and window for the related components.

3.1.5. Attributes of macrofeatures

The attributes of macrofeatures 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 drywall case example, the estimator specified 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.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

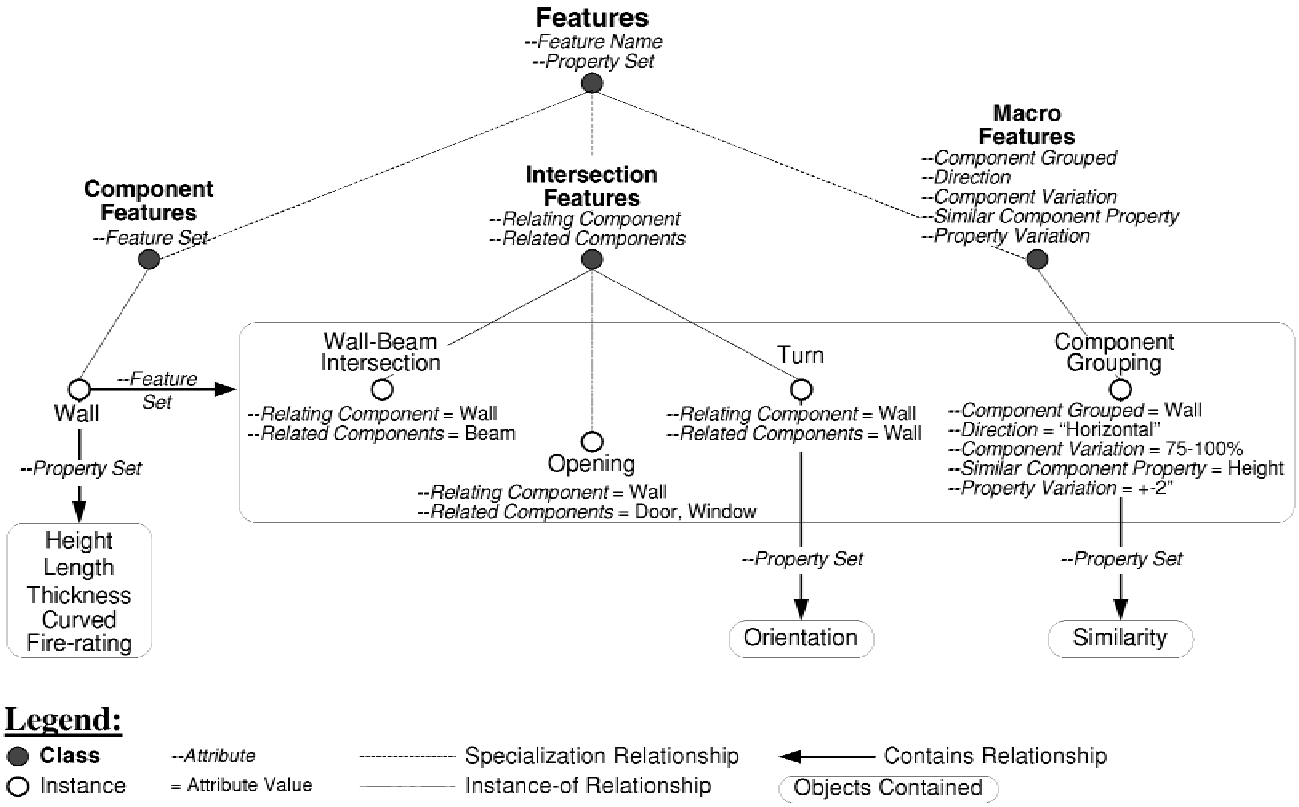


Fig. 5. The feature ontology that represents the attributes of the three feature types and example feature instances from the drywall case example. Each component feature knows the estimator's preference for what features and properties influence the cost of its construction using the feature set and property set attributes. Each intersection feature knows the estimator's preference for the properties of the feature that affect a component feature's construction cost using the property set attribute and the component intersections that are important using the related component and relating components attributes. Each macrofeature 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 permitted.

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 drywall case example, the drywall estimator evaluated the similarity of the wall components in the horizontal direction, and the concrete estimator evaluated the similarity of the column components in the vertical direction.
3. *Component Variation:* The overall variation of the components (see item 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 drywall case example, the estimator preferred that 75–100% of the walls have similar heights (see item 4) for component similarity to exist.
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 drywall case example, one drywall estimator analyzed the property height and the other drywall estimator analyzed 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 in. for the property variation, then the estimator views wall #1 as similar to wall #2 if its height is at most 2 in. shorter or taller than wall #2.

3.2. A framework for estimators to represent cost-driving features

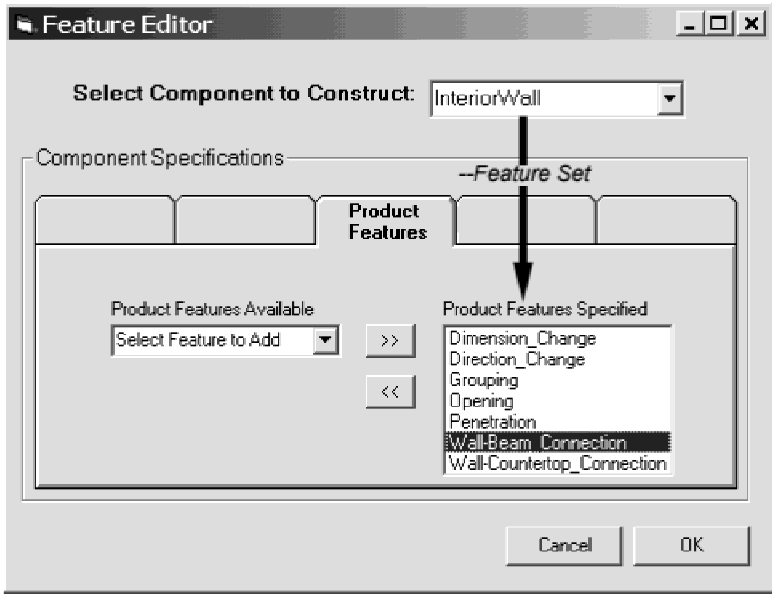
FeaGen implements the objects and attributes of the feature ontology to provide a formal, flexible, and computer-interpretable way for estimators to specify the design conditions that are important to them. Estimators use Feature Specification templates to specify the features and properties that affect a component's construction costs, to represent the cost-driving component intersections as intersection

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

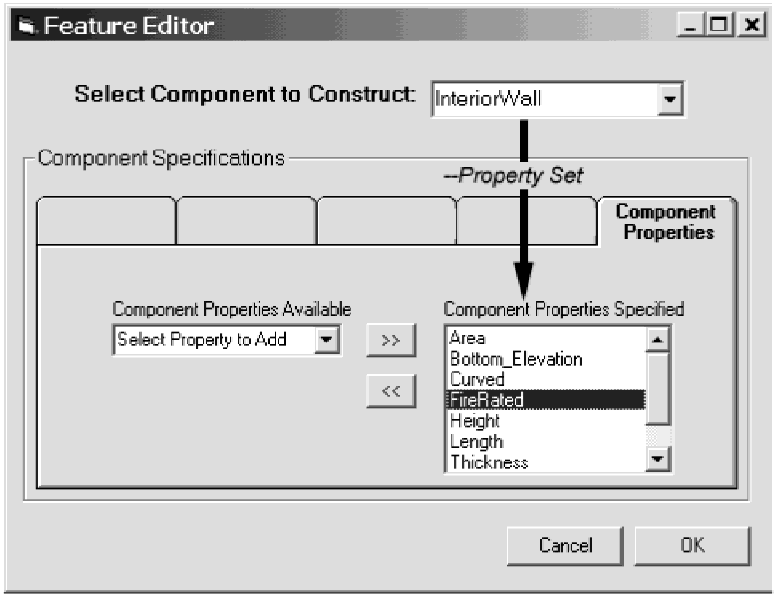
features, and to define component similarity according to their preferences.

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 IFCs. Figure 6 shows the user interface from FeaGen and example feature and property sets specified by the estimator from the drywall case example 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 macrofeature is not available, estimators need to first create the feature before adding



(a)



(b)

Fig. 6. 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. (a) An example feature set for walls specified by the estimator from the drywall case example. (b) An example property set for walls specified by the estimator from the drywall case example.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

it to the feature set, which is described in the next section. FeaGen represents this knowledge independent of a project 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.

3.2.2. Represent generic instances of intersection feature

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 5 shows instances of intersection features from the drywall case example and the attribute values for each feature. If estimators want to create a new intersection feature, they 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 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 estimators' 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 macrofeatures (component similarity)

FeaGen leverages the attributes of the ontology to provide a framework that allows estimators to represent their preferences for defining component similarity (Fig. 7). 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 7 shows an estimator's preference that 90–100% of the wall heights be similar 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 using the generic cost-driving features specified by the estimator (Fig. 8). In FeaGen, the input 3-D model is created using Bricsnet Architecturals (Bricsnet, 2001). The representation of product models in Bricsnet is very similar to product models represented using the IFCs. 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.

To create a project-specific feature-based product model, FeaGen executes the four steps shown in Figure 8, which are as follows:

Step 1. *Identify Components:* The estimator selects the component class to estimate. In the drywall case example, the estimator selects the component class wall. FeaGen identifies instances of the component in the input IFC-based product model. In IFC-based

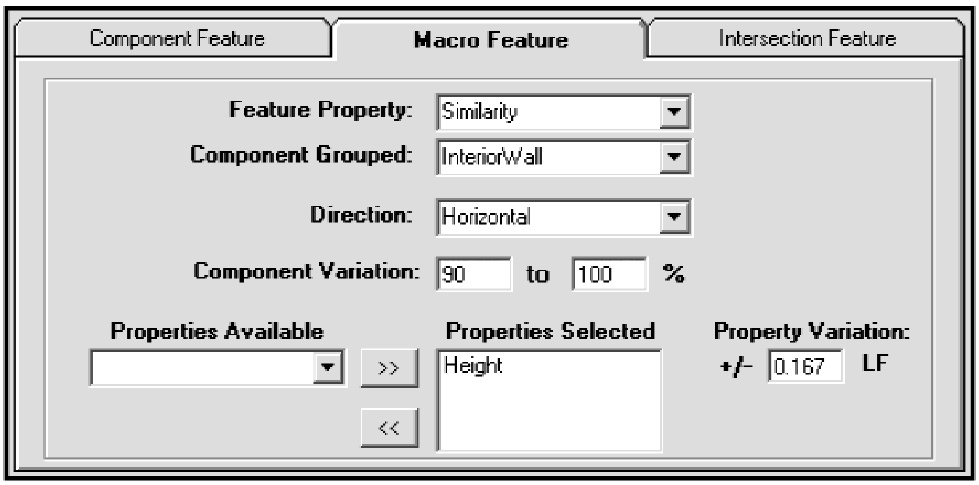


Fig. 7. FeaGen uses the attributes of macrofeatures to provide a framework for estimators to define component similarity. The example from the drywall case example shows an estimator's preference that 90–100% of the walls have similar wall heights for component similarity to exist.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

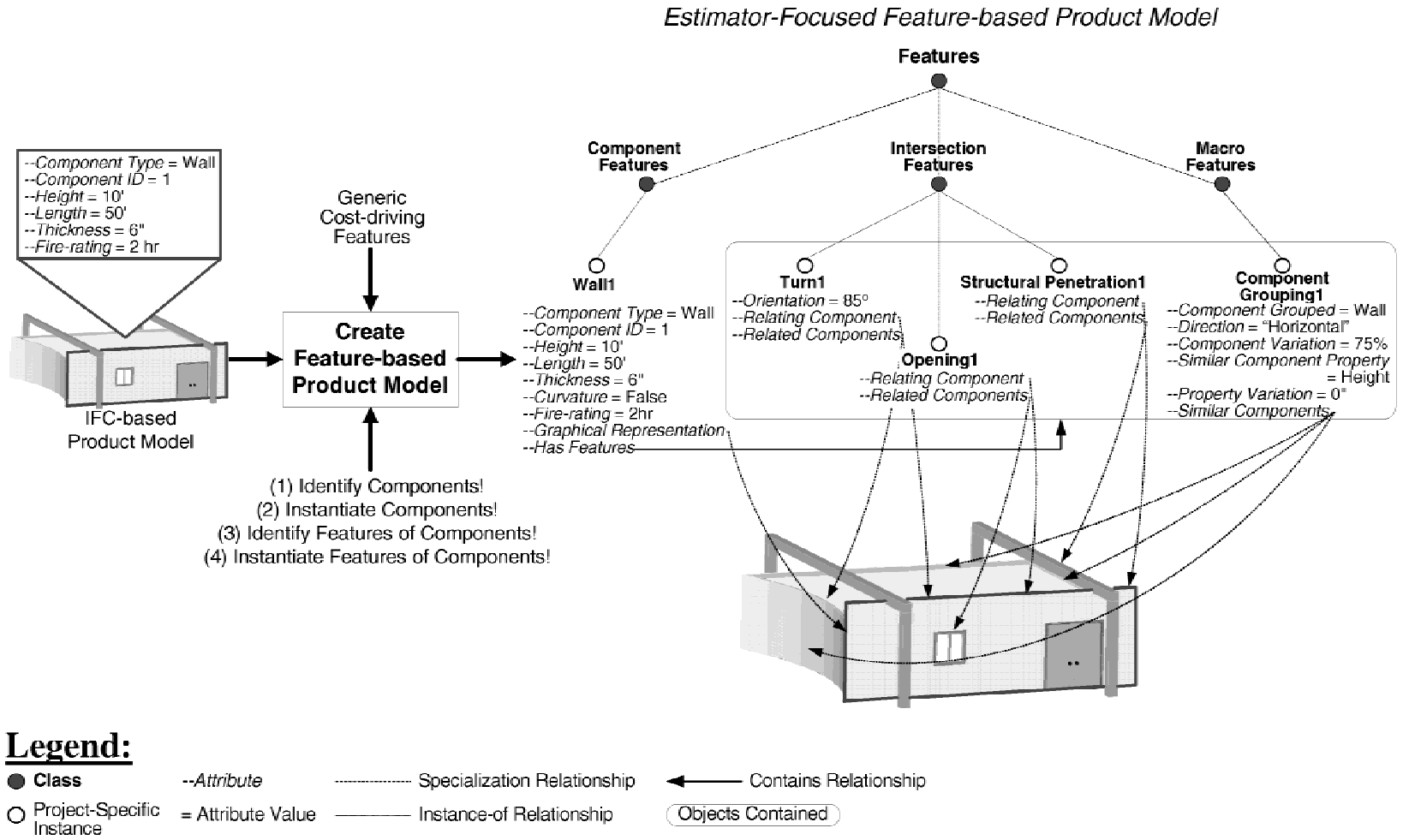


Fig. 8. FeaGen transforms an IFC-based product model into a project-specific feature-based product model that explicitly represents the cost-driving features specified by the estimator.

product models, components are explicitly represented as features so FeaGen simply has to query the input product model to identify them.

Step 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 IFC-based product model's 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.

Step 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 9 shows different representations of the connectivity between components using a Bricsnet 3-D 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 9. 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 currently need to identify these types of component intersections manually.

b. *Identify Macrofeatures (Component Similarity):* To identify component similarity, FeaGen reasons about the properties of building components of the same type to determine whether 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, FeaGen evaluates the building components on a single floor. If the estimator specified vertical in the direction attribute, 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 whether the components are similar. FeaGen compares the property values to determine whether 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, 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, FeaGen considers the components to be similar.

Step 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 macrofeatures, FeaGen adds the instances of similar components to the similar components attribute in the project-specific feature-based product model (Fig. 8).

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 blueprint 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 construc-

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

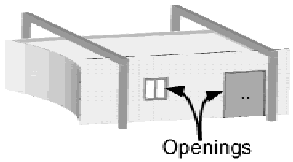
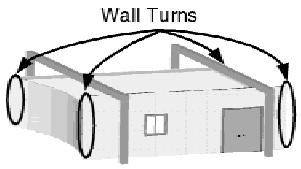
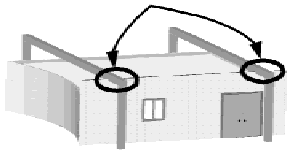
Example Intersection Features from Motivating Case	*Bricsnet Objects and Attributes Analyzed*	*IFC Objects and Attributes Analyzed*	*Description of Reasoning Process*
	<Form.HasOpenings> <Form> <Form.TypeOf>	<IfcBuidlingElement.HasOpenings> <IfcBuildingElement> <IfcBuildingElement.UserDefinedType>	FeaGen 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 to identify the intersecting components that have the same component type.
	<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.

Fig. 9. The 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.

tion. 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 demonstrates the feature ontology enables cost estimators to:

- Represent the design conditions that affect construction costs more explicitly than standard 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. Cost estimators represent the design conditions affecting construction costs more explicitly with the feature ontology than with 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 macrofeatures 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 macrofeatures 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 IFCs 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 1 point for each feature represented explicitly using the feature ontology and the IFCs. Table 1 shows that the feature ontology represents 92% of the cost-driving features explicitly whereas the IFCs represent 2.3% of the cost-driving features explicitly. These results suggest that the feature ontology explicitly represents substantially more cost-driving features than the IFCs and that the features formalized in the feature ontology approach the theoretical ideal for the three feature types shown.

Table 1. *Comparison of features represented in feature ontology and in IFCs to assess ability of feature ontology to support explicit representation of cost-driving features*

Features	Theoretical Ideal		Feature Ontology	IFCs
Component	Walls	1	1	1
	Columns	1	1	1
Intersection	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 (component similarity)	Similarity of heights	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
Represented explicitly		13	12	3

The theoretical ideal represents the component, intersection, and macrofeatures that estimating experts confirmed are important for interior wall and concrete column construction. The results show that the feature ontology represents 92% of the features explicitly whereas the IFCs represent 2.3% of the features explicitly. These results suggest that the feature ontology represents cost-driving features more explicitly than the IFCs and that the features represented in the feature ontology approach the theoretical ideal.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

4.2. Cost estimators represent the design conditions affecting construction costs for two 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.

4.3. Cost estimators specify their preferences for representing the design conditions affecting 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. The different users of FeaGen did not encounter difficulty using the system. These tests demonstrate generality across user types and suggest that the feature ontology is sufficiently flexible to represent different estimators' preferences.

4.4. Cost estimators generate and maintain cost estimates better with feature-based product models than with IFC-based ones

We developed a prototype cost estimating application called ACE to test whether the feature ontology helps 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 (Fig. 10). 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.

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, 2002, chap. 3).

Figure 11 shows the attributes of Activity and Resource Specifications and examples of estimators' rationale for adjusting activities and resources for the features from the drywall case example. 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*, 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.

Table 2. *Features represented in feature ontology for two test cases on walls and columns*

Component Features	Intersection Features	Macrofeatures (Component Similarity)
Walls	Turns	Similarity of height
	Openings	Similarity of width
	Penetrations	Similarity of type
	Wall–beam intersections	
	Wall–countertop intersections	
Columns	Column–slab intersections	Similarity of height
		Similarity of width
		Similarity of length
		Similarity of shape

The ability of practitioners to represent different features for different component types provides evidence for the generality of the feature ontology.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

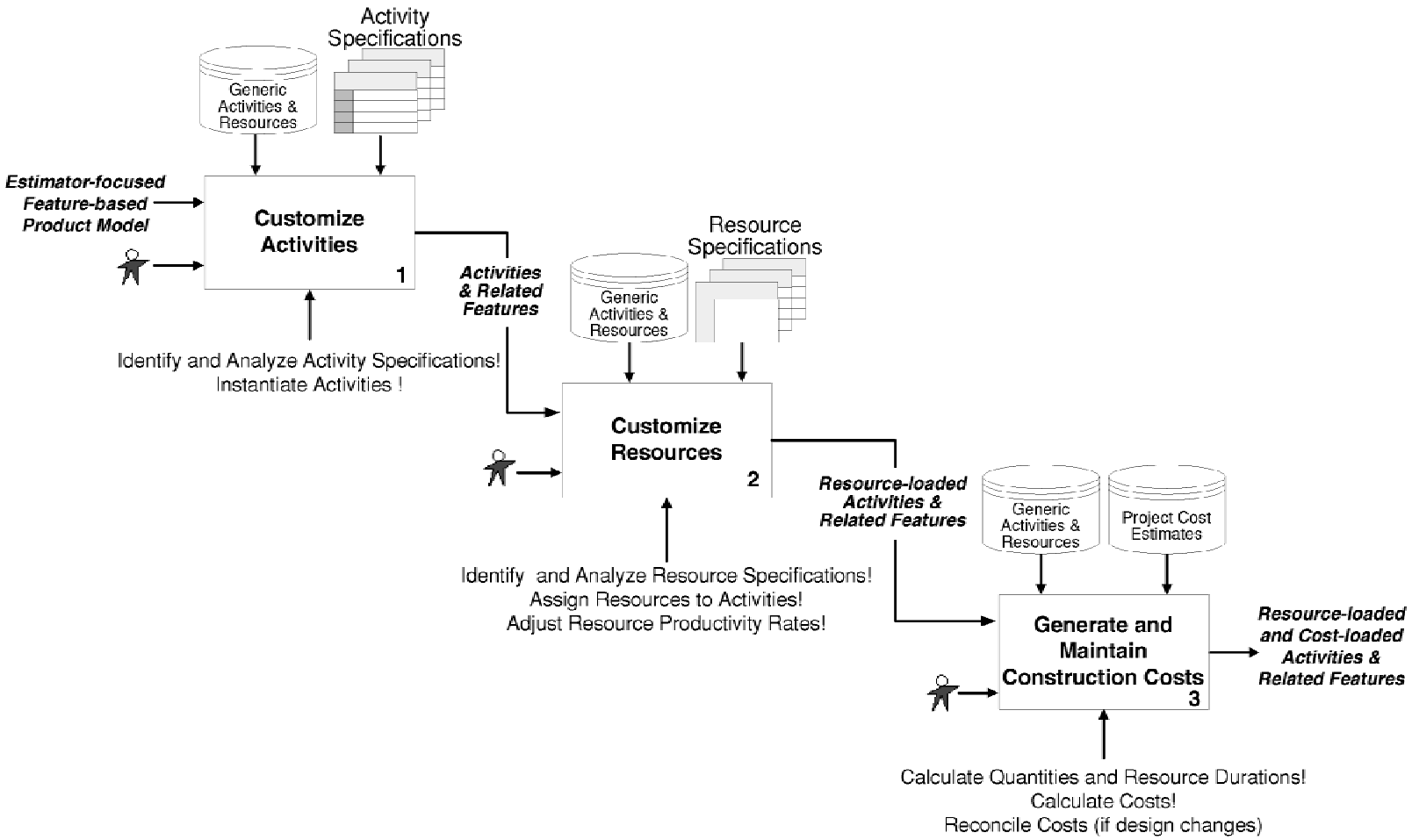


Fig. 10. The different steps of the activity and resource customization process we formalized and implemented in ACE to support the generation and maintenance of construction cost estimates with feature-based product models. We used ACE to validate the feature ontology by showing that estimator-focused feature-based product models enable estimators to generate and maintain cost estimates more accurately, consistently, and quickly than IFC-based product models.

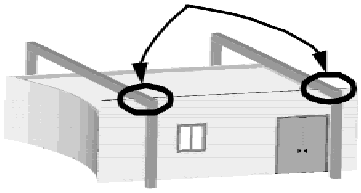
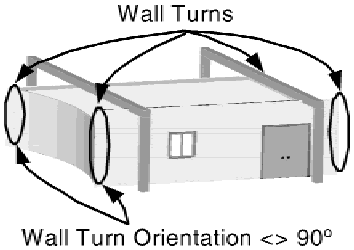
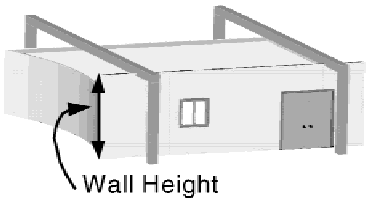
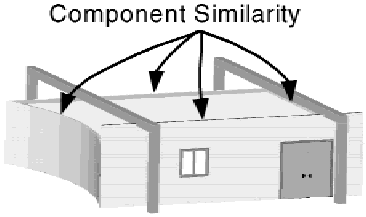
Relevant Design Conditions	*Estimator's Rationale about Activities and Resources Required for Design Conditions*	*Estimators Rationale Input in Activity and Resource Specification Templates*
Wall-Beam Intersection 	Add Activity "Apply Caulk" if the intersecting wall is fire-rated.	*Activity Specification #1* **Feature**: Wall **Action**: Layout **Object**: Wall **Cost Implication**: Resource Design Condition — **Feature**: Wall; **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; **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; **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; **Constraint**: 75-100% Walls have Similar Heights

Fig. 11. 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, 2002, chap. 4). Figure 10 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,

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

Table 3. *Summary of results from four validation tests measuring level of completeness of estimates generated by estimators using ACE and state of the art software tools*

	Theoretical Ideal		ACE Process		State of the Art Process	
Validation Test	Level of Completeness	Complete (%)	Level of Completeness	Complete (%)	Level of Completeness	Complete (%)
Charrette	13	100	12.13	93	5.63	43
Sequus project	12	100	10	83	10	83
DPR office project	16	100	14	88	11	69
Bay street project	16	100	13	81	12	75
Total	57	100	49.13	86	38.63	68

These results show that estimators using ACE generate estimates that are 86% complete whereas the same estimators using state of the art software tools generate estimates that are 68% complete. These results demonstrate that the estimator-focused feature-based product model enabled estimators to generate more complete cost estimates than state of the art software tools that use IFC-based product models.

and quickly than IFC-based product models (Staub–French, 2002, chap. 5). To assess the accuracy of the estimates, 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 Software Company, 2001), which is an industry standard for cost estimating software. 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. We also evaluated the ability of ACE and PE to capture and reuse the estimators' rationale to maintain the cost estimates when the design changes by recognizing changes to relevant features and feature properties and their associated cost impacts based on the estimators' rationale.

Table 3 shows the results of the four validation tests indicating the level of completeness of estimates generated by estimators using ACE and state of the art software tools. Estimators using ACE generated estimates that are 86% complete whereas the same estimators using state of the art software tools generated estimates that are 68% complete. Table 4 shows the results of the four validation tests indicating the level of completeness of revised estimates for specific design changes using ACE and state of the art software tools. These results show that ACE helps estimators to maintain estimates that are 92% complete whereas state of the art software tools help estimators to maintain estimates that are 29% complete. These results are presented graphically in Figure 12. The results of the validation tests demonstrate that the estimator-focused, feature-based product model enabled estimators using ACE to generate (Table 3, Fig. 12) and maintain (Table 4, Fig. 12) more complete cost estimates than the same estimators using state of the art

Table 4. *Summary of results from four validation tests measuring level of completeness of revised estimates for specific design changes using ACE and state of the art software tools*

	Theoretical Ideal		ACE Process		State of the Art Process	
Validation Test	Level of Completeness	Complete (%)	Level of Completeness	Complete (%)	Level of Completeness	Complete (%)
Charrette	13	100	11.5	88	4	31
Sequus Project	12	100	11.25	94	3.75	31
DPR Office Project	16	100	14.75	92	3.75	23
Bay Street Project	16	100	14.75	92	5	31
Total	57	100	52.25	92	16.5	29

These results show that ACE maintains estimates that are 92% complete whereas state of the art software tools maintain estimates that are 29% complete. These results demonstrate that the estimator-focused feature-based product model enabled estimators to maintain substantially more complete cost estimates than state of the art software tools that use IFC-based product models.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

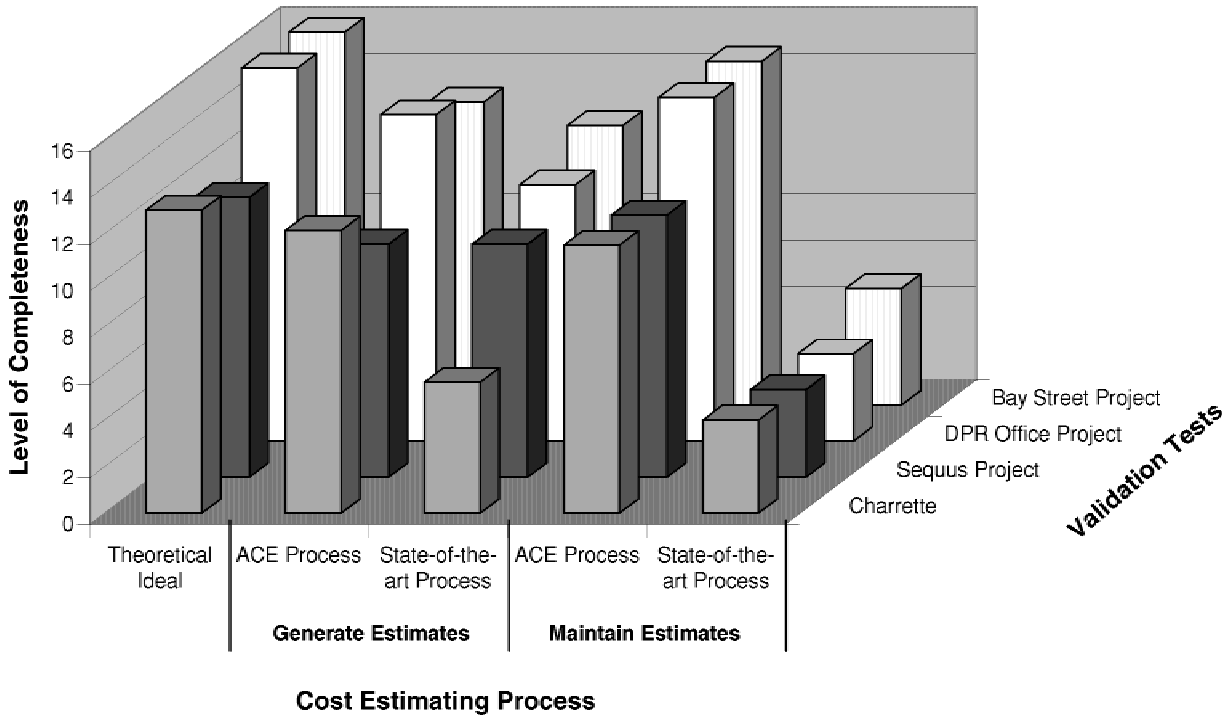


Fig. 12. The results of the four validation tests show that the level of completeness of estimates generated and maintained from the estimator-focused feature-based product model in ACE is significantly greater than using state of the art tools and approaches the theoretical ideal.

software that uses IFC-based product models. These results suggest that estimators using ACE 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. Therefore, the four validation tests demonstrate the power of the feature ontology by showing that the estimator-focused feature-based product model helps 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.

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 formalized a vocabulary that classifies three types of features, defines the attributes and functions of each feature type, and represents the relationships between features explicitly. The descriptive semantics of the model are flexible enough to represent estimators' varied preferences for naming features, specifying the features that result from component intersections and the similarity of components, and grouping features that affect a specific construction domain. We also formalized a framework implemented in a software prototype to represent estimators' preferences about cost-driving features in a project-independent way so that this knowledge can be reused from project to project to create estimator-focused product models that support feature-based cost estimating. The power and generality of the descriptive semantics of the model were validated for certain classes of problems with industrial test cases.

The feature ontology presented in this paper is limited in several ways. It does not represent all the types of features that affect construction costs. For example, material features, such as the workability of concrete, and features that result from the proximity of components, such as the proximity of a duct run to a pipe run, may also affect construction costs. These types of features are currently not represented in the feature ontology. 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, additional research is needed to develop mechanisms that identify intersection features resulting from physical conflicts between components, which require conflict detection mechanisms. We also recommend additional testing to validate the generality of the feature ontology across different construction domains in addition to building construction.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

This research extends the concept of product features that is used extensively in the manufacturing industry to the domain of cost estimating for building construction. The results of this research demonstrate that features provide a theoretical framework for representing the design conditions that are important for estimating the costs of building construction. This research also demonstrates that formal ontological models provide a foundation for representing cost estimating knowledge about the product features that impact construction costs. The validation studies provide evidence for the power of the formal ontological modeling approach in representing knowledge and developing knowledge-based programs for the domain of construction cost estimating.

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 cost estimates, improve the efficiency and consistency of the cost estimating process, and improve the accuracy of 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.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the following organizations: the Center for Integrated Facility Engineering at Stanford University, the National Science Foundation (Grant 9625228), Stanford University's Future Professors of Manufacturing Program, Hathaway–Dinwiddie Construction Company, Flad & Associates, Mazzetti & Associates, Rosendin Electric, Paragon Mechanical, and Rountree Plumbing and Heating.

REFERENCES

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

Bjork, B. C. (1991). A unified approach for modeling construction information. *Building and Environment, 27(2),* 173–194.

Bricsnet. (2001). *Architecturals, Users Documentation.* Portsmouth, NH.

Clayton, M.J., Kunz, J.C., & Fischer, M.A. (1996). Rapid conceptual design evaluation using a virtual product model. *Engineering Applications of Artificial Intelligence, 9(4),* 439–451.

Clayton, M.J., Kunz, J.C., & Fischer, M.A. (1998). *The Charrette Test Method.* Technical Report 120, Stanford, CA: Center for Integrated Facility Engineering.

Cunningham, J.J., & Dixon, J.R. (1988). Designing with features: The origin of features. *Proc. Computers in Engineering Conf.*, pp. 237–243. San Francisco, CA: ASME.

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

Froese, T., Yu, K., & Shahid, S. (1996). Project modeling in construction applications. *Computing in Civil Engineering: Proc. Third Congress*, pp. 572–578. Anaheim, CA: ASCE.

Gielingh, W.F. (1988). *General AEC Reference Model. ISO TC184/SC4/WG1 doc. 3.2.2.1. TNO Report BI-88-150.* New York: ISO.

Gruber, T.R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies, 43(5–6),* 907–928.

Hanna, A.S., & Sanvido, V.E. (1990). Interactive vertical formwork selection. *Concrete International: Design and Construction, 12(4),* 26–32.

Hyer, N.L., & Wemmerlov, U. (1984). Group technology and productivity. *Harvard Business Review, 62(4),* 140–149.

International Alliance of Interoperability. (2001). *IFC 2x Extension Modeling Guide.* Available on-line at http://www.iai.org.uk.

Laitinen, J. (1998). *Model based construction process management.* PhD Thesis, Stockholm: Royal Institute of Technology.

Marir, F., Aouad, G., Child, T., & Brandon, P. (1998). OSCONCAD: A model-based CAD system integrated with computer applications. *ITCON, 3,* 25–44.

Mizoguchi, R. (2001). Ontological engineering: Foundation of the next generation knowledge processing. *Proc. 1st Int. Conf. Web Intelligence and Intelligent Agent Technologies*, Maebashi City, Japan, pp. 44–57.

Sanders, S.R., & Thomas, H.R. (1991). Factors affecting masonry–labor productivity. *Journal of Construction Engineering and Management, ASCE, 117(4),* 626–644.

Scholten, H., & Beulens, A. (2002). *Ontologies to structure models and modeling tasks.* Paper presented at 16th JISR-IIASA Workshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment, July 15–17, Laxenburg, Austria.

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. *Proc. Construction Congress VI*, pp. 1051–1057. Orlando, FL: ASCE.

Smith, G.R., & Hanna, A.S. (1993). Factors influencing formwork productivity. *Canadian Journal of Civil Engineering, 20(1),* 144–153.

Staub–French, S. (2002). *Feature-driven activity-based cost estimating.* PhD Thesis. Stanford, CA: Stanford University.

Stumpf, A., Ganeshan, R., Chin, S., & 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., & Zavrski, I. (1999). Construction baseline productivity: Theory and practice. *Journal of Construction Engineering and Management, ASCE, 125(5),* 295–303.

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

van Leeuwen, J.P. (1999). *Modelling Architectural Design Information by Features.* PhD Thesis. Eindhoven University of Technology, Eindhoven, The Netherlands.

Weinstein, P.C., & Birmingham, W.P. (1998). Creating ontological metadata for digital library content and services. *International Journal on Digital Libraries, 2(1),* 19–36.

Sheryl Staub–French is an Assistant Professor of civil engineering and Coordinator of the Engineering Management Program at the University of British Columbia. She earned her PhD and MS degrees in civil engineering at Stanford University and her BS degree from Santa Clara University. Her research interests focus on developing symbolic models for the design and construction of civil engineering projects, modeling and predicting the cost and constructibility implications of design features, formalizing feature-based product models to represent the construction perspective, and visualizing construction process and constructibility information. Dr. Staub–French's teaching

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press

interests focus on computer applications for construction, symbolic modeling, construction methods, and technology entrepreneurship.

Martin Fischer is an Associate Professor of civil and environmental engineering and (by courtesy) computer science and the Director of the Center for Integrated Facility Engineering at Stanford University. He attained a PhD in construction engineering and management and an MS in industrial engineering and engineering management, both from Stanford University, and a DiplIng Civil from the Swiss Federal Institute of Technology, Lausanne. Dr. Fischer is a leader in the application of object models for design and construction of facilities. He pioneered the application and further development of 4-dimensional modeling for construction process design and constructibility evaluation. He has been published in over 60 refereed publications and has made more than 100 invited and conference presentations. Dr. Fischer is the recipient of several publication, research, and university awards, including a NSF Career Award, the *Engineering News Record* Top 25 Newsmaker Award, and the Best Paper Award at the 6th International Conference on Artificial Intelligence in Design.

John Kunz is Executive Director and Chief Scientist of the Center for Integrated Facility Engineering at Stanford University. His current and recent projects focus on virtual design and construction, including developing models of engineering products and processes, organizational modeling, automated construction planning, visualization, use of business objectives, and automated checking of building codes. He teaches classes in virtual design and construction, as well as symbolic modeling in engineering, and has conducted an engineering seminar series and various professional short courses in knowledge-based systems technology. Dr. Kunz' research interests cross the engineering life cycle from preproject planning through design build, retrofit, and decommissioning. An important research and teaching theme in his work is the use of different strategies to validate new formalized theories that have been implemented as computer models.

Kos Ishii is a Full Professor at Stanford University and serves as the Director of the Manufacturing Modeling Laboratory. He earned his BS in mechanical engineering in 1979 at Sophia University (Tokyo), his MS in mechanical engineering in 1982 at Stanford University, and his MS in control engineering in 1983 at Tokyo Institute of Technology. After serving Toshiba Corporation for 3 years as a Design Engineer, he returned to Stanford and completed his PhD in mechanical design in 1987. He was on the Faculty at The Ohio State University from 1988 to 1994. He directs the Graduate Course Sequence on Design for Manufacturability, which is subscribed to by over 12 companies through the Stanford Instructional Television Network. He authored or coauthored more than 140 refereed articles, served as chair of the ASME Computer and Information in Engineering Division (1998), and was an Associate Editor of the *Journal of Mechanical Design* (ASME, 1995–1998). Dr. Ishii is the recipient of many significant awards, including the NSF Presidential Young Investigator Award (1991), the Pitney Bowes–ASME Award for Excellence in Mechanical Design (1993), and the Japan Society of Mechanical Engineers Achievement Award (Systems and Design Division, 2000). His research focuses on structured product development methods, commonly known as Design for *X*.

Boyd C. Paulson, Jr., is the Charles H. Leavell Professor of Engineering in Stanford's Graduate Program in Construction Engineering and Management. He earned his BS, MS, and PhD degrees in civil engineering from Stanford University. He is the author of two books and over 100 papers. Dr. Paulson has received several national and international awards for his work, including fellowships and grants to support research and teaching in China, Germany, Japan, and the United Kingdom. His research and teaching interests focus on computer applications in construction, construction equipment and methods, international construction, and affordable housing.

https://doi.org/10.1017/S0890060403172034 Published online by Cambridge University Press