Advanced Engineering Informatics 17 (2003) 23–39

ADVANCED ENGINEERING INFORMATICS

www.elsevier.com/locate/aei

A generic feature-driven activity-based cost estimation process

Sheryl Staub-French[a,*], Martin Fischer[b,1], John Kunz[c,2], Boyd Paulson[d,3]

[a]Department of Civil Engineering, University of British Columbia, Vancouver, BC, Canada V6T 1Z4
[b]Department of Civil and Environmental Engineering and (by Courtesy) Computer Science, Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, USA
[c]Center for Integrated Facility Engineering, Stanford University, Stanford, CA 94305, USA
[d]Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305, USA

Received 6 August 2002; revised 5 May 2003; accepted 10 May 2003

Abstract

Understanding how the building design influences construction costs is a challenging task for estimators. Estimators must recognize the design conditions that affect construction costs and customize the cost estimate accordingly. Estimators have different preferences for how and when to adjust a project's activities, resources, and resource productivity rates that form the basis of a cost estimate. Current tools and methodologies lack ways to help estimators customize construction cost information according to their preferences and the particular features in a given design. This paper describes the process we formalized to customize a project's activities, resources, and resource productivity rates based on a project-independent representation of estimators' rationale and a project-specific feature-based product model. The formal process creates an integrated model that explicitly relates features, activities, resources, costs and the estimator's rationale. Our tests show that this process enables a software prototype to generate and maintain cost estimates quickly, consistently, and accurately for feature-based product models.

Keywords: Activity-based cost estimating; Product features; Computer-integrated design and construction; Symbolic modeling; Knowledge-based systems; Derivation tasks

1. Introduction

Construction cost estimators are confronted with the challenging task of having to estimate the cost of constructing one-of-a-kind facilities. Estimators must recognize the design conditions of the facility design that are important (i.e. incur a cost) and how they affect construction costs. Estimators have different preferences for what design conditions they want to consider and how the cost estimate should be adjusted to account for them. Today, estimators account for the cost impact of many design conditions by manually adjusting the project's activities, resources, and resource productivity rates that form the basis of a cost estimate for a specific design. Then, if the design changes, estimators have to manually identify the cost information affected and adjust the activities and resources accordingly so that the project's design and cost estimate remain in balance. Estimators make these project-specific adjustments manually because current tools and methodologies are unable to customize the project's activities and resources based on estimator's varied preferences and the particular design conditions in a given product model. Without automated support to customize construction cost information, cost estimators often employ ad hoc methods that are prone to error and result in inconsistencies and inefficiencies in the cost estimating process.

Cost estimation is a typical example of a knowledge-intensive engineering task. We classify the cost estimation process as a derivation task within knowledge-based systems design. Companies often hire professionally educated engineers to do the task, and those engineers typically require several years to develop expertise in performing this task. In spite of this training, owners and

* Corresponding author. Tel.: +1-604-827-5118; fax: +1-604-822-6901.
E-mail addresses: sherylsf@civil.ubc.ca (S. Staub-French), fischer@stanford.edu (M. Fischer), kunz@stanford.edu (J. Kunz), paulson@stanford.edu (B. Paulson).
[1] Tel.: +1-650-725-4649; fax: +1-650-725-6014.
[2] Tel.: +1-650-725-1546; fax: +1-650-723-4806.
[3] Tel.: +1-650-723-2235; fax: +1-650-725-6014.

1474-0346/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S1474-0346(03)00017-X

builders of facilities find that there is wide variability in construction cost estimates of different estimators for the same project, and that the lack of consistency in the current manual process often leads to overestimating or underestimating construction costs, resulting in lost opportunities or unexpected expenses, respectively.

This paper presents the generic process we formalized to customize construction costs for a given product model. It describes the implementation of this generic process in the computer and demonstrates the power of this process to perform the knowledge-intensive cost estimation task. We refer to the *customization* of construction costs as the adjustment of a project's activities, resources, and resource productivity rates to reflect the cost impact of specific design features and design conditions in a given product model. The generic process leverages the project-independent representation of estimators' rationale to create project-specific resource-loaded and cost-loaded activities for project-specific feature-based product models. This five-step reasoning process identifies cost-incurring features in a given product model, creates the necessary activities for constructing the project-specific features, assigns the appropriate resources for executing the activities, and adjusts the resources' productivity rates to calculate construction costs. The implementation is interactive, enabling estimators to calculate the costs of the product and process design based on their preferences, and to estimate the change in costs following specific changes in either the product or construction process. The result is an integrated model consisting of features, activities, resources, resource productivity rates, costs and the estimator's rationale that supports the maintenance of construction cost information. The main contributions of the paper are the formalization of the feature-driven activity and resource customization process and the specification of the resulting integrated model.

1.1. Motivating case: current practice

This section describes a use case to illustrate the requirements for automated support of the cost estimating process. The case study is based on a drywall estimator's process for estimating the labor costs for one of the rooms in an office project shown in Fig. 1. The building components in the room are annotated in Fig. 1.

Drywall estimators must identify the design conditions that affect the cost of constructing the walls and adjust the cost estimate accordingly. Estimators have different preferences for adjusting the activities, resources, and resource productivity rates to account for the cost impact of different design conditions. Fig. 2 shows how the estimator customizes the activities, resources, and resource productivity rates to account for specific design conditions in the motivating case and explains the estimator's rationale for making the adjustments.

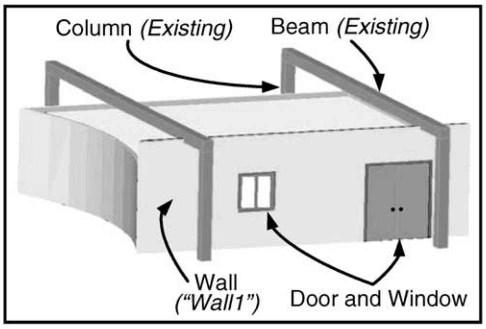

Fig. 1. Building components in the office project case study. The drywall estimator is estimating the costs for constructing the four walls shown.

The motivating case shows how estimators customize a project's activities, resources, and resource productivity rates to account for the cost impact of the different design conditions. Estimators identify relevant design conditions and adjust the activities, resources, and resource productivity rates regardless of the type of component they are estimating. Consequently, it is possible to formalize and generalize the estimator's process and provide automated support of the cost estimating process for a variety of construction domains. To provide a formal and general process, we abstracted the design information estimators consider, the different ways estimators adjust the cost information to account for different design conditions, and the different steps estimators perform to customize activities and resources.

We use the concept of *features* to describe the design information estimators care about. 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'. *Design conditions* describe when a particular feature affects construction costs. Design conditions can be based on properties of component features (e.g. the curvature and height of the wall), groupings of component features (e.g. the grouping of walls based on component similarity), the existence of intersection features (e.g. the existence of turns and openings), and properties of intersection features (e.g. the orientation of wall turns). We use *activities* to describe the impact features have on construction execution to predict the corresponding cost of construction.

The motivating case demonstrates that component and intersection features drive the requirement for activities for constructing a particular component. Activities are required to install the subcomponents of a component (e.g. 'Install Metal Studs' activity for wall's decomposition), to account for component properties (e.g. 'Layout Wall' activity for wall curvature), to perform supporting activities for constructing a component (e.g. the Layout Wall activity supports the Install Metal Studs activity), and to construct

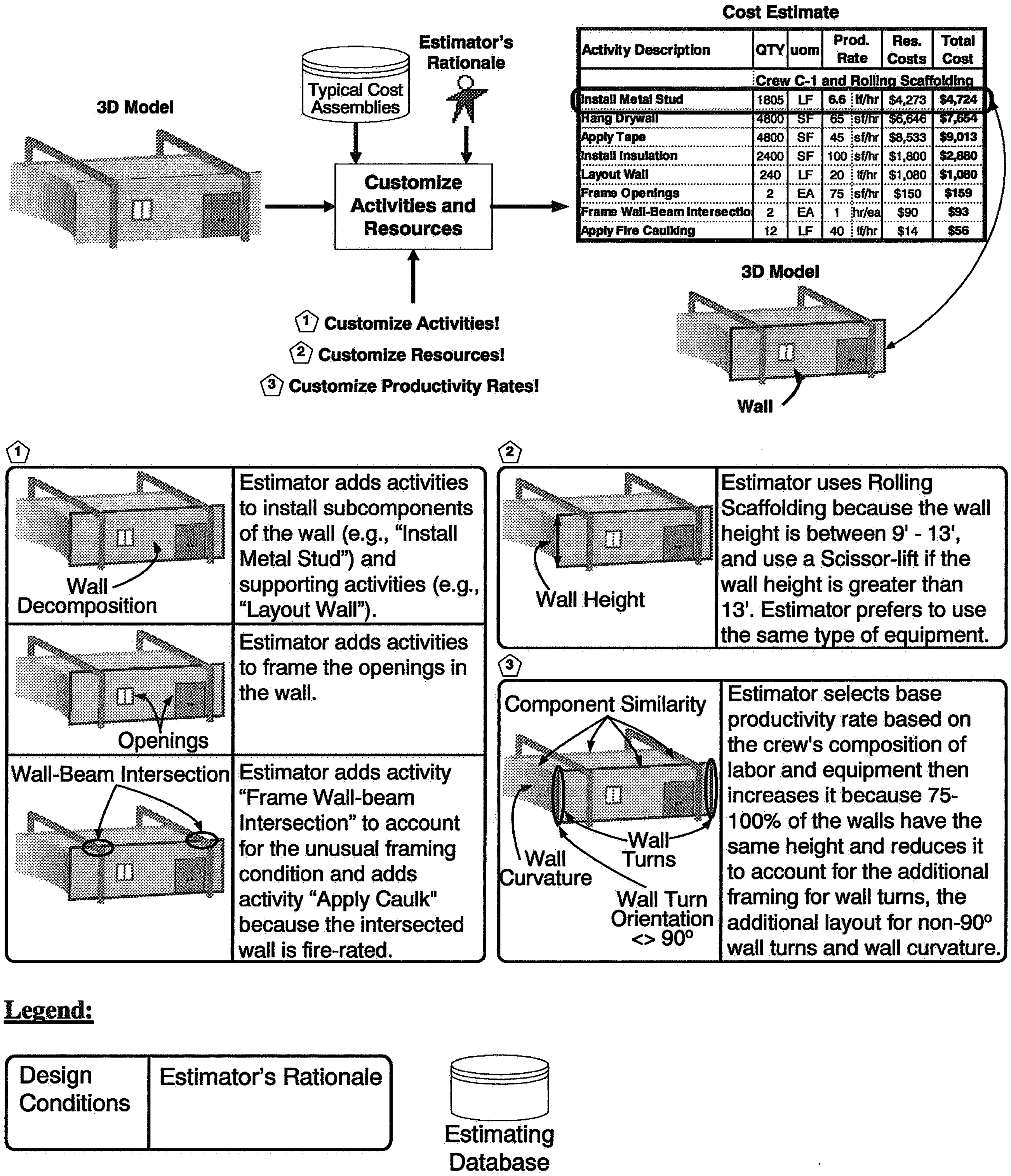

Fig. 2. Estimators customize the activities, (1) resources (2) and (3) resources productivity rates to account for different design conditions when configuring a cost estimate for a particular design.

intersection features resulting from a component's intersection with other components (e.g. 'Frame Wall-Beam Intersection' for *Wall1's* intersection with the beam). The motivating case also illustrates the estimator's process for customizing the resources required to execute the activities. For the Install Metal Studs activity, the estimator assigns equipment based on the height of the wall. Then, if multiple instances of the activity calls for different types of equipment (e.g. Rolling Scaffolding and Scissor-lifts), the estimator adjusts the equipment assignments so the activities are using the same type of equipment. Finally, the estimator selects the base productivity rate for each activity based on the crew composition, and adjusts the base crew productivity rate to reflect the production impact of specific design conditions (e.g. component similarity).

For a large project, it is typically too time-consuming to perform all the project-specific adjustments of activities, resources, and resource productivity rates for the different design conditions in a given product model manually. Consequently, estimators often employ ad hoc methods (e.g. reduce the crew productivity to account for wall turns and non-90° wall turns) and overlook the cost impact of features and feature properties (e.g. overlook the wall-beam intersection and its cost impact). Moreover, if the design changes, estimators must manually identify the cost information affected because current cost estimates do not explicitly represent the estimator's rationale for customizing activities and resources. Hence, the lack of automated support leads to inconsistencies and inefficiencies in the cost estimating process and the corresponding cost estimate.

Estimators need a formal process for customizing the activities and resources to create an integrated model that explicitly relates the project-specific features, activities, resources, resource productivity rates, costs, and the project-independent estimator's rationale. The formal process needs to:

- Customize the activities for constructing a specific component based on the component's decomposition, the component's properties, the intersection features of the component, and the requirement for supporting activities based on an estimator's preferences and the design conditions in a given product model.
- Customize the crew composition of labor and equipment based on the specific design conditions in a given product model and an estimator's preferences for when and how to make resource assignments.
- Customize the crew productivity rate based on the crew composition, the specific design conditions in a given product model, and an estimator's preferences.
- Identify the cost information affected by design changes and adjust the cost estimate according to the estimator's preferences.

Estimators need automated support to customize activities and resources according to their preferences when generating and maintaining cost estimates from 3D product models. Our research addresses this need by formalizing a feature-driven activity and resource customization process that can be implemented in the computer.

1.2. Research goals

The use case illustrates the needs of the formal process:

(1) *Formal*. Estimators need a formal process for customizing activities, resources, and resource productivity rates to help them generate and maintain construction costs and prevent them from using implicit or ad hoc methods to account for the cost impact of different design conditions. By formalizing a process to provide this functionality, estimators should be able to account for the cost impact of features explicitly and consistently.

(2) *General*. Estimators need a process that is general enough to account for different estimator preferences when generating and maintaining construction cost estimates from 3D product models. It must also be general enough to support cost estimating of different product models, product model changes, and construction domains.

In summary, cost estimation must enable an estimator to recognize unique design conditions and estimate the cost of the project's activities and corresponding resources, and generate and maintain an integrated cost model based on the estimators' rationale. To provide this type of functionality, we developed, implemented and tested a formal process that builds on previous research in knowledge-based systems, and combines and extends previous research in construction cost estimating and activity modeling.

2. Related research background

Previous research efforts demonstrate that construction costs can be generated from 3D models. Researchers also identify the design conditions that affect the applicability of resources and impact a resource's productivity rate when executing an activity. However, they do not provide a formal process that customizes a project's activities, resources, and resource productivity rates based on an estimator's preferences and the relevant features and design conditions in a given product model. Moreover, they do not create an integrated model that represents the many relationships between features, activities, resources, costs, and the estimator's rationale.

2.1. Prior research on knowledge-based systems in engineering

Dym and Levitt [1] use two broad classes of problem-solving to provide a basis for discussing all engineering tasks when designing knowledge-based systems in engineering: derivation and formation tasks. They define tasks that derive solutions from given facts and data as derivation or classification tasks, which includes diagnosis, interpretation, and monitoring. In formation or synthesis tasks, problems are solved by forming or creating an object or a set of plans for making an object, which includes planning and design. Cost estimation can be classified as a derivation task because this process exhibits many of the characteristics of classification problems. For example, it requires the *selection* of appropriate construction methods, the *evaluation* of alternative crew configurations, the *interpretation* of design features that impact construction execution, the *prediction* of costs for alternative designs, and the *monitoring* of costs to account for design changes. The activity and resource customization process we formalized executes many of these tasks to generate and maintain cost estimates for a particular design and a specific estimator's preferences.

2.2. Prior research on construction cost estimating

Many research efforts focus on generating cost estimates from 3D models and creating integrated models that explicitly relate the components, activities, resources, and costs similar to our research [2–6]. They customize construction costs by identifying the typical activities, resources, and resource productivity rates for a specific component in a product model. However, they do not account for different design conditions or different estimator preferences when configuring a cost estimate for a particular design.

In contrast, previous research efforts on method modeling customize an activity's resources based on the unique design conditions in a given product model [7–10]. For example, Fischer [7] identifies the applicable formwork methods for reinforced concrete structures based on a given 3D product model. However, these research efforts do not customize the methods assigned to activities based on estimators' varied preferences. Moreover, the project-specific resource assignments do not explicitly represent the design conditions that determined the assignment.

Similarly, researchers focusing on productivity modeling customize resource productivity rates for unique design conditions in a given product model [11–15]. For example, Thomas and Sackrakan [12] formalize a process to forecast labor productivity based on the work environment, which includes different design conditions and construction methods. However, they do not account for different estimator preferences when customizing the resource productivity rates for a particular design. Moreover, they account for the production impact of some features in an ad hoc way by adjusting the productivity rate for features that actually require additional activities to be constructed (e.g. openings and wall turns).

2.3. Prior research on activity modeling

Prior research efforts in activity modeling generate resource-loaded activities for the component feature being installed and the method being used [9,16–19]. They generate activities that know what object ⟨O⟩ they act on, what action ⟨A⟩ is being performed, and what resources ⟨R⟩ are being used. However, prior research efforts do not identify the required resources or adjust the resource productivity rates to account for the production impact of different features and different estimators' rationale. Our research extends the activity definition by generating activities that also know what feature ⟨F⟩ requires the activity's execution and how much the activity costs ⟨C⟩ to create an integrated ⟨FARC⟩ model that is explicitly related to the estimator's rationale.

3. A generic feature-driven activity-based cost estimation process

So far this paper has presented the need for a formal process to customize activities and resources for a given product model and to create an integrated model that supports the maintenance of construction cost information. The research challenge is that different features and design conditions exist in any given product model, that different features and design conditions affect construction costs in different ways, and that estimators have different preferences for how to adjust construction costs to account for different features and design conditions. The process we formalized addresses these challenges because it is general enough to customize activities and resources for different estimators' preferences and for different features and design conditions in a given product model. The formal process assembles a project-specific integrated model consisting of features, activities, resources, resource productivity rates, and costs based on the project-independent representation of estimator's rationale and a project-specific feature-based product model.

We describe this general process in subsequent sections, but first we describe how we represent estimators' rationale in a project-independent way.

3.1. Representing cost estimators' rationale generically

To capture estimators' rationale and represent it generically (i.e. independent of a particular project), we abstracted the common attributes of estimators' rationale for how and when different design conditions affect construction costs and developed templates to capture this estimating knowledge from estimators [20]. The motivating case illustrated the different ways estimators adjust construction costs to account for different types of design conditions (Fig. 2). Estimators adjust the project's activities, resources, and resource productivity rates to account for the cost impact of different design conditions. We created different templates that allow estimators to specify the design conditions that affect the requirement for activities (Activity Specifications), the allocation of resources (Resource Specifications), and the execution of resources (Resource Productivity Specifications), as shown in Fig. 3. The templates represent the estimators' rationale in a project-independent way so that this knowledge can be reused from project to project.

Activity Specification templates capture estimators' rationale about how and when activities are required for

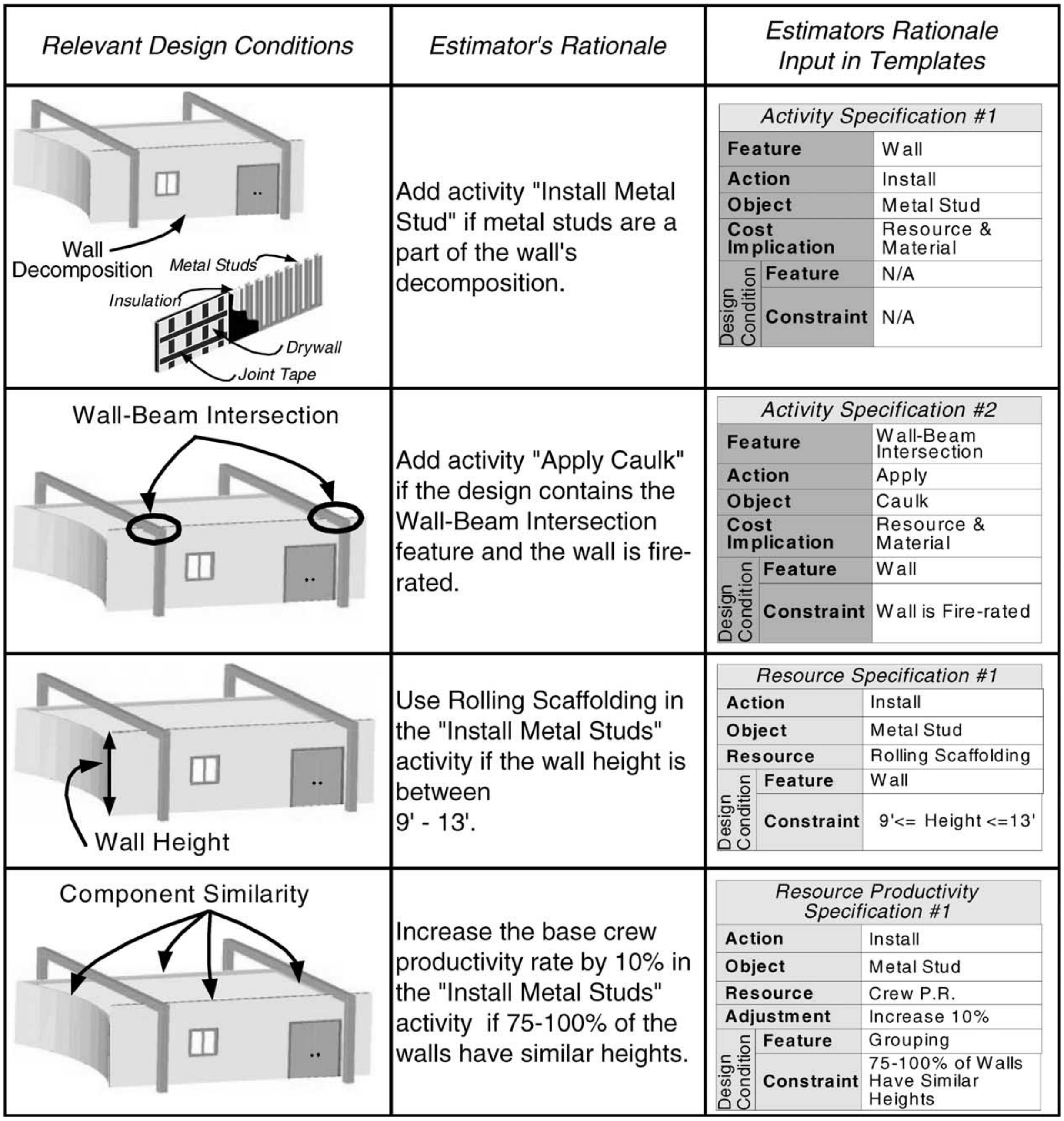

Fig. 3. Estimators fill out Activity, Resource, and Resource Productivity Specification templates to represent their preferences for how activities, resources, and resource productivity rates should be adjusted for various design conditions. This figure shows example templates for different design conditions from the motivating case.

different features. Estimators specify 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. Fig. 3 shows the attributes of Activity Specifications and two instances from the motivating case. The estimator specifies the activity to add for constructing the feature by identifying the object to install (e.g. ‘Metal Stud’) and the action to perform on the object (e.g. ‘Install’). The action–object pair points to a generic activity that contains attributes for specifying the *formula* for calculating the activity’s quantities and the need for *supporting activities*. If the design condition is satisfied, the activity specified in the Activity Specification is instantiated for each instance of the specified feature in a given project-specific product model.

Resource Specification templates capture estimators’ rationale about when resources are required for a given activity. Estimators specify the *activity* (represented as an action–object pair), the *resource*, and the *design condition* that dictates when the feature requires the resource.

The action–object pair points to a generic activity that contains attributes to represent estimators' preferences for the *possible labor* and *possible equipment resources* to execute the activity and for using the *same equipment* for all instances of the activity. If the design condition is satisfied, the resource specified in the Resource Specification is assigned to each project-specific activity in accordance with the estimator's preferences for using the same equipment.

Resource Productivity Specification templates capture estimators' rationale about when and how to adjust resource productivity rates for different design conditions. To represent estimators' rationale about resource productivity rates, estimators specify the *activity*, the *resource* whose productivity rate to adjust, the *adjustment* of the productivity rate, and the *design condition* that dictates when the feature requires the productivity rate adjustment. The action–object pair points to a generic activity that contains attributes to represent estimators' preferences for the possible *base productivity rates*. If the design condition is satisfied, the resource productivity rate specified in the Resource Productivity Specification is adjusted according to the estimator's preferences for each project-specific activity.

The main contributions of the generic representation of estimators' rationale lie in the different templates, the structure of the templates, and the attributes of the templates. The process we formalized creates activities, assigns resources to activities, and adjusts resource productivity rates according to the estimator's preferences in Activity and Resource Specifications and based on the specific design conditions in the feature-based product model, which we discuss next.

3.2. A formal process for customizing activities, resources, and resource productivity rates

We developed the feature-driven activity and resource customization process by abstracting estimating tasks and associated knowledge about how to do them. We reviewed previous research in this area, supported a design-build team with state-of-the-art tools to generate estimating quantities from a 3D model of the design [21], and interviewed fourteen different cost estimators. We interviewed two general contractors and twelve subcontractors who self-perform construction works on drywall, structural concrete, ductwork, process piping, and electrical systems. We performed three case studies on two drywall construction projects and one concrete column construction project. We formalized the different vocabularies used by estimators to describe the design conditions that affect construction costs and formalized a process to customize a project's activities, resources, and resources' productivity rates for specific design conditions in a given product model.

We implemented the feature-driven activity and resource customization process and the corresponding mechanisms in a software prototype called activity-based cost estimating (ACE) to validate this process. We use ACE in the following sections to describe the specific mechanisms executed in the activity and resource customization process.

Fig. 4 shows an overview of the five-step feature-driven activity and resource customization process:

(1) *Create feature-based product model*. Create a feature-based product model that instantiates the cost-driving features that are important to the estimator.
(2) *Customize activities*. Customize the activities for each component feature being estimated based on the estimator's rationale in Activity Specifications and the features in the product model.
(3) *Customize resources*. Customize each activity's resources based on the estimator's rationale in Resource Specifications and the particular features in the product model.
(4) *Customize resource productivity rates*. Customize each activity's resource productivity rates based on the resource composition, the estimator's rationale in Resource Productivity Specifications, and the particular features in the product model.
(5) *Generate and maintain construction costs*. Calculate each activity's quantities and duration to determine the activity's cost. If the estimate is based on a revised design, identify the cost information affected and reconcile the activities and resources so that the design and estimate remain in balance.

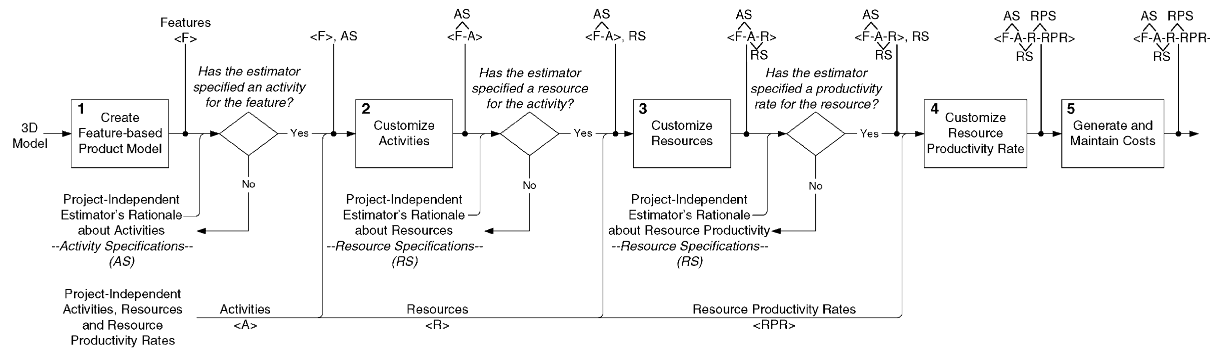

Fig. 4. Overview of the formal feature-driven activity and resource customization process. This process assembles features ⟨F⟩, activities ⟨A⟩, resources ⟨R⟩, resource productivity rates ⟨RPR⟩, and costs ⟨C⟩ based on the estimator's rationale (AS, RS, RPS) and the particular features in the input 3D model.

Based on the abstracted project-independent estimating knowledge, the formal process transforms the project-specific feature-based product model and the generic activities and resources into project-specific resource-loaded and cost-loaded activities. To represent estimators' rationale about the influence of features on activities, resources, and resource productivity rates, we created Activity Specifications (AS), Resource Specifications (RS), and Resource Productivity Specifications (RPS), respectively. The process creates an integrated model consisting of activities ⟨A⟩ that explicitly represent what feature ⟨F⟩ requires their execution and why (AS), what resources ⟨R⟩ are assigned to the activities and why (RS), what the resources' productivity rates ⟨RPR⟩ are and how they were adjusted (RPS), and how much the execution of the activities cost ⟨C⟩. The explicit relationships between features, activities, resources, resource productivity rates, and the estimator's rationale provide the foundation for generating and maintaining construction costs for feature-based product models.

3.2.1. Representing the features that are important to cost estimators

The motivating case demonstrated that different design conditions influence construction costs: the "wall-beam intersection" created an unusual framing condition and the *orientation* of the 'wall turn' affected the wall lay out (Figure 2). To help estimators generate and maintain cost estimates, cost estimating software must represent the design conditions that are important to cost estimators. To represent the design conditions that affect construction costs, we abstracted the different vocabularies used by estimators to describe different design conditions and formalized a feature-based product model to support construction cost estimating [22].

Different types of design information affect construction costs. Estimators consider properties of component features, groupings of component features, intersections of component features, and properties of component intersections when creating cost estimates (Fig. 2). We formalized a feature ontology that classifies features and represents the sets of features and properties that affect costs for a specific construction domain. Using the feature ontology, estimators can create features and specify the features and properties that affect a particular component's construction costs. We created a prototype application called feature generator (FeaGen) that leverages the feature ontology to create a project-specific feature-based product model that represents the features and properties that are important to estimators. FeaGen creates a feature-based product model in a two-step process: (1) infers the existence of features by analyzing the geometry and topological relationships between components, and then (2) instantiates the features in a project-specific feature-based product model. Fig. 5 shows the symbolic representation of the features and properties instantiated for the drywall estimator estimating the wall component feature in the motivating case. In the resulting feature-based product model, each component feature instantiated explicitly represents its decomposition and

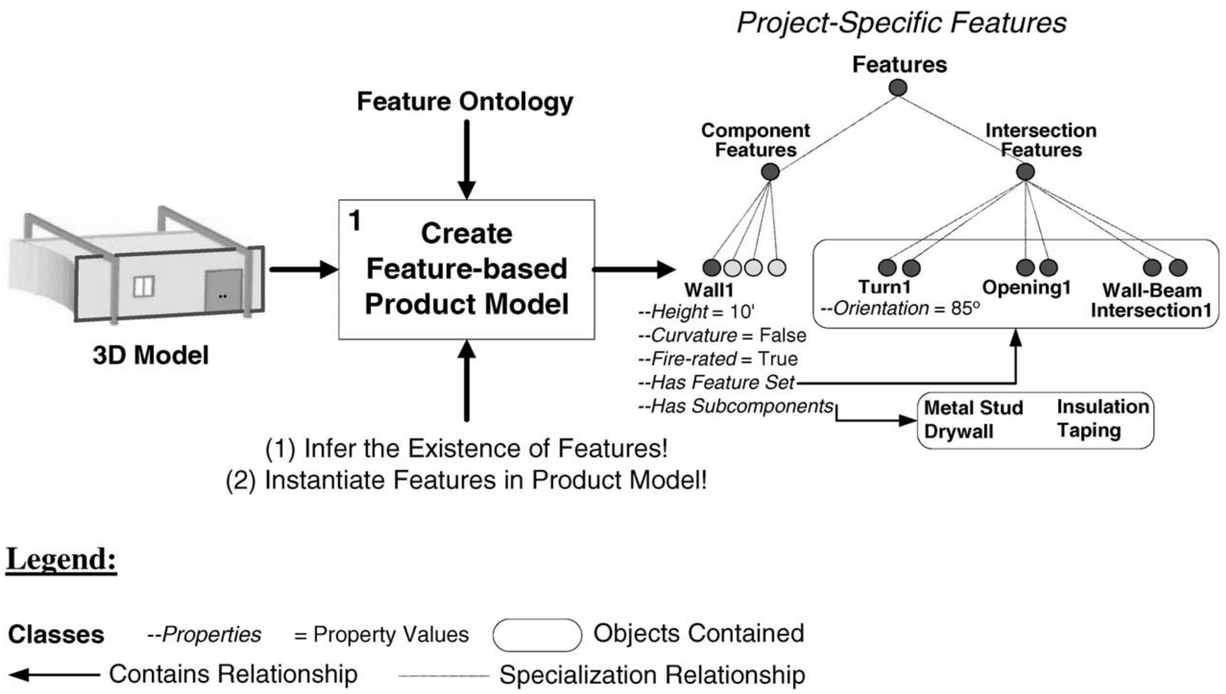

Fig. 5. Symbolic representation of the project-specific feature-based product model instantiated for Wall1 for the estimator in the motivating case. The project-specific feature-based product model represents the features and properties that are important to the estimator. The properties of Wall1 that affect construction costs are instantiated (e.g. height, curvature, and fire-rated), the intersection features of Wall1 are instantiated in the 'Has Feature Set' attribute (e.g. Turn1, Opening1, and Wall-beam Intersection1), the properties of intersection features affecting Wall1 are instantiated (e.g. the orientation of Turn1), and the subcomponents of Wall1 are instantiated (e.g. Metal Stud and Drywall).

the properties, intersection features, and intersection feature properties that influence its construction. Staub-French et al. [22] describes the specific details of the feature ontology and the mechanisms implemented to generate project-specific feature-based product models. The main contributions lie in the formalization of the feature ontology and the framework developed to capture this knowledge from estimators.

The project-specific feature-based product model drives the activity and resource customization process, which we discuss next.

3.2.2. Customize the activities to the features and design conditions in a product model

The motivating case demonstrated that component and intersection features drive the requirement for activities. The construction of the wall required the execution of activities based on the wall's decomposition (e.g. Install Metal Studs), based on the wall's properties (e.g. the curved wall requires the Layout Wall activity), based on the intersection features that result from the wall's intersection with other components (e.g. Frame Wall-Beam Intersection), and based on the need for supporting activities (e.g. Layout Wall supports the Install Metal Studs activity). Hence, the activity customization process must formally consider the component's decomposition, the properties of the component, the intersection features of the component, and the requirement for supporting activities when customizing the activities needed to construct a specific component.

The project-specific feature-based product model and the project-independent Activity Specifications enable the activity customization process. The activity customisation process creates project-based on the estimator's preferences captured in Activity Specifications for each of the project-specific features and properties instantiated in the input feature-based product model (Fig. 6).

(1) *Identify activity specifications*. For each feature instance in the input feature-based product model, identify the appropriate Activity Specifications.

(a) *Component features*. To identify the Activity Specification for the wall's subcomponents and component properties, ACE identifies the Activity Specifications that have the component feature in the 'Feature' attribute. If the Activity Specification is for a subcomponent, ACE also matches the particular subcomponent in the 'Object' attribute. For example, ACE identifies Activity Specification #1 because metal stud is a subcomponent of the wall and because metal stud and wall are specified in the Feature and Object

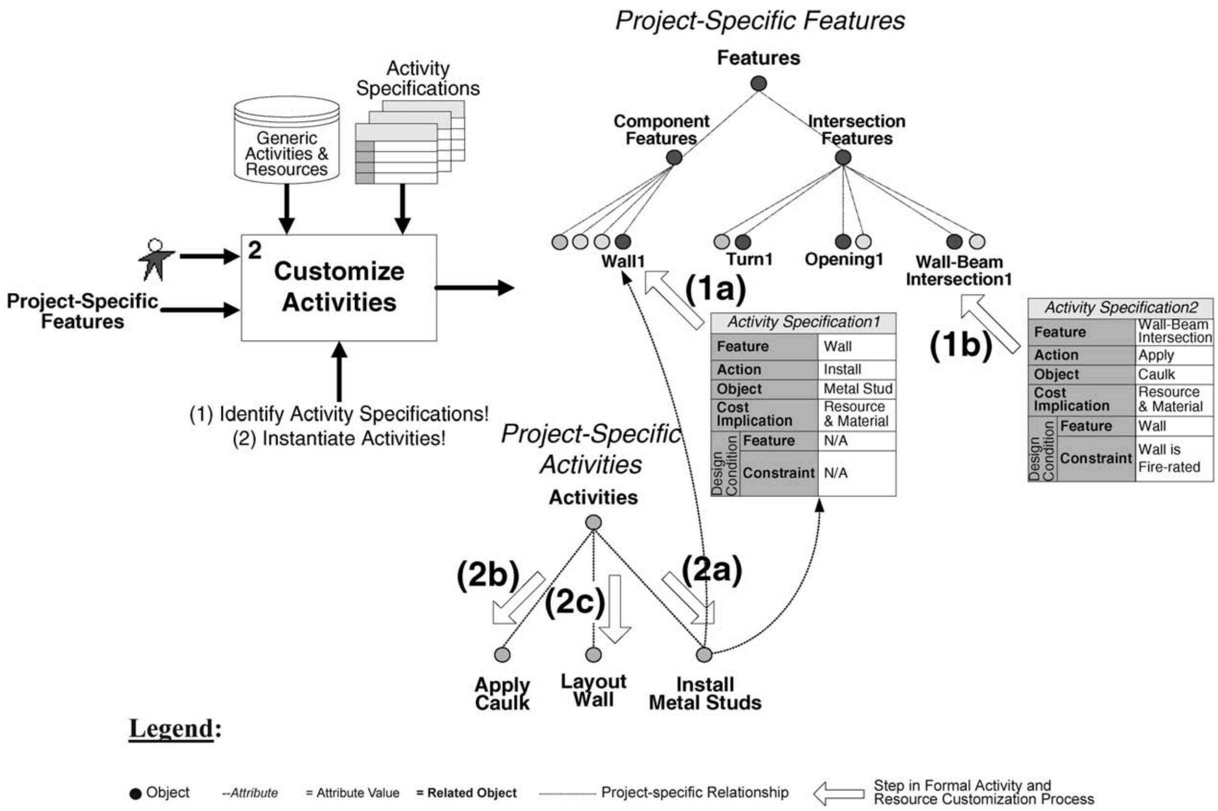

Fig. 6. The second step customizes the activities based on the features and properties in the input product model and the estimator's rationale in Activity Specifications. To customize activities, two reasoning mechanisms are implemented that (1) identify the appropriate Activity Specifications, and (2) instantiate the necessary activities to construct the features in the input product model.

attributes of the Activity Specification. If there are several matches, ACE asks the estimator to select the most appropriate Activity Specification.

(b) *Intersection features*. Identify the Activity Specification for the intersection features of each component by matching the intersection feature in the Feature attribute. Then, analyze the 'Design Condition' attribute of the Activity Specification because the applicability of the activity may be constrained to certain design conditions. For example, in the motivating case, the intersection feature Wall-Beam Intersection requires the activity 'Apply Caulk' if the intersecting wall is fire-rated. Analyze the intersection feature and related intersecting components in the estimator's feature-based product model to determine if the design condition is satisfied.

(2) *Instantiate activities*. Instantiate the activities specified in the Activity Specifications for the component and intersection features.

(a) *Component features*. Instantiate the activities specified in the Activity Specifications for subcomponents of the component and properties of the component. Then, create relationships to the component feature that requires the activity and to the Activity Specification that represents the estimator's rationale for adding the activity. Finally, find the corresponding generic activity in the database and copy the generic activity's formula for calculating the activity's quantities, the possible labor and possible equipment resources that the activity can use to execute the activity, the requirement for using the same equipment for all instances of the activity, and the need for supporting activities to the activity instance.

(b) *Intersection features*. Instantiate the activities specified in the Activity Specifications for the intersection features of the component. For example, creates the Apply Caulk activity for the Wall-beam Intersection feature based on Activity Specification #2 shown in Fig. 3. Then, create relationships to the intersection feature that requires the activity and to the Activity Specification that represents the estimator's rationale for adding the activity and copy the corresponding generic activity's attributes to the activity instance as described in Step 2b.

(c) *Supporting activities*. For each newly instantiated activity, assess whether the activity requires supporting activities by checking the supporting activities attribute of the activity. If the activity requires supporting activities, instantiate those activities as described in Steps 2a and 2b. For example, ACE creates the Layout Wall activity because it is a supporting activity of the Install Metal Studs activity.

The activity customization process creates project-specific activities for each component in a given product model based on the component's decomposition, the intersection features of the component, and the requirement for supporting activities. The process creates activities formally and systematically and prevents estimators from using implicit or ad hoc methods to account for the cost impact of features. For example, estimators cannot account for the production impact of wall turns by fudging the crew's productivity (Fig. 2). Rather, estimators must account for intersection features explicitly by representing the activities that need to be executed. Moreover, if the design changes, the formal process re-assembles the activities for each component based on the new features in the revised design. At this stage in the reasoning process, each activity knows what feature requires its execution, the material and resource cost implications of the activity, the possible resources for executing the activity, and the estimator's rationale for adding the activity. The resource customization process leverages this integrated model to identify the specific resources needed to execute each activity.

3.2.3. Customize the resources in an activity to the features and design conditions in a product model

Estimators often have preferences for when a specific resource should be used in an activity, which is often based on the particular conditions in a given design. For example, the estimator in the motivating case preferred to use Rolling Scaffolding in the Install Metal Studs activity when the wall height is greater than 9. and less than 13., and use a Scissor-lift when the wall height is greater than 13. The estimator also preferred to use the same piece of equipment for constructing all instances of the Install Metal Studs activities rather than switching equipment to execute the activities (Fig. 2).

The project-specific feature-based product model, the project-specific activities, and the project-independent Resource Specifications enable the resource customization process. With the three reasoning mechanisms, the resource customization process assigns resources to activities and identifies the resource's base productivity rate based on the estimator's preferences for the project-specific features instantiated in the input product model (Fig. 7).

(1) *Identify resource specifications*. For each activity, analyze the Resource Specifications of the possible equipment and labor resources to determine the appropriate crew composition of labor and equipment for executing the activity. An activity can have one or many resources assigned to it. Analyze the design condition of the Resource Specification to determine the applicability of the resource for executing the activity. For example, ACE analyzes the design condition of Resource Specification #1 to assess whether rolling scaffolding is needed to execute the Install Metal Studs activity as shown in Fig. 7. If more than one or no labor crew is appropriate or if more than

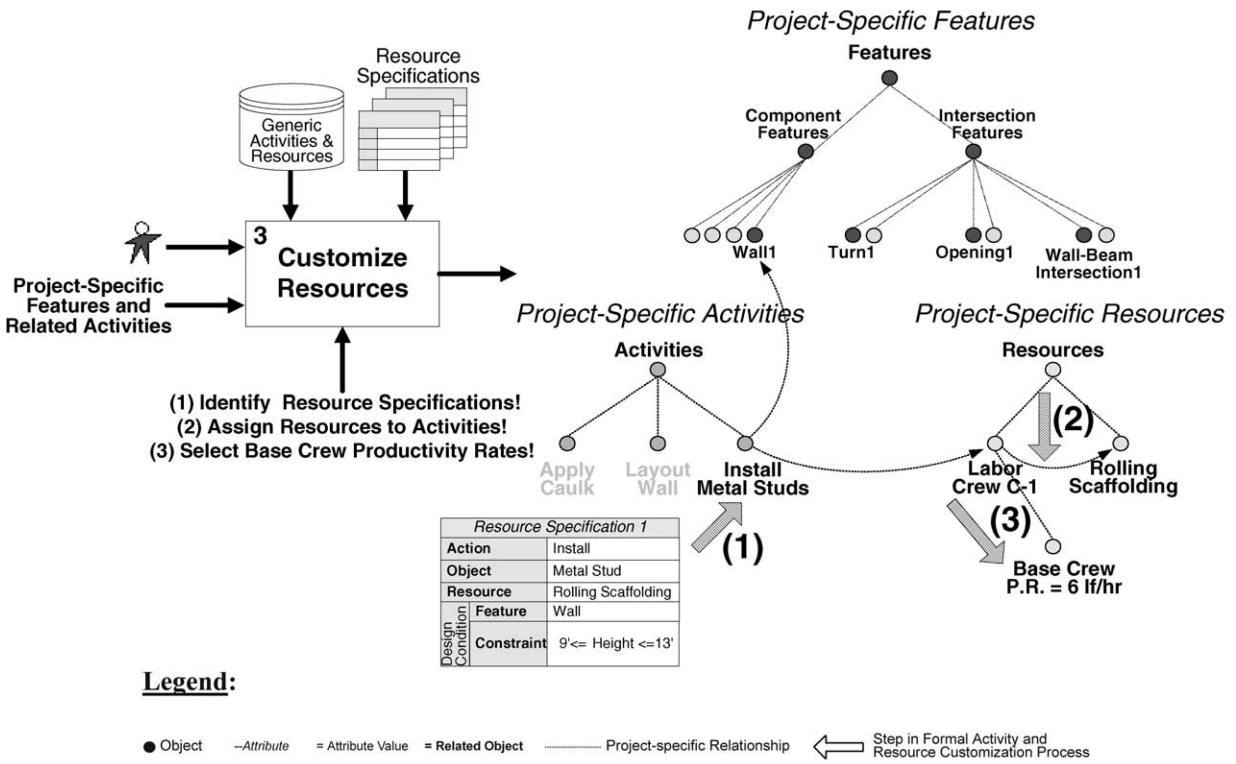

Fig. 7. The third step customizes the resources in an activity for each component based on the particular features in a given design. The process generates resource-loaded activities customized for each feature in a given product model.

one equipment resource is appropriate, ACE asks the estimator to select the most appropriate resources to execute the activity.

(2) *Assign resources to activities*. If the design condition in the Resource Specifications is satisfied, the specific resource is appropriate for executing the activity. For the Install Metal Studs activity for Wall1, ACE identifies Labor Crew C-1 and Rolling Scaffolding as the appropriate resources for executing the activity. Then, adjust the crew composition based on the estimator's preference to *use the same equipment* for all instances of an activity. For example, if different instances of the Install Metal Studs activity require Rolling Scaffolding and Scissor-lifts, ACE adjusts the crew composition so that all Install Metal Studs activities use Scissor-lifts. Then, formally relate the resources to the activity and relate the Resource Specifications to the resources assigned.

(3) *Select base crew productivity rates*. Identify the appropriate base crew productivity rate based on the equipment and labor resources assigned to the activity. For example, according to the estimator from the motivating case, the base productivity rate for Labor Crew C-1 using Rolling Scaffolding is 6 lf/h whereas the base crew productivity rate for Labor Crew C-1 without the requirement for equipment is 8 lf/h. Hence, select the base crew productivity rate according to the customization of the resources required to execute the activity.

3.2.4. Customize the resource productivity rates based on the resources assigned to activities and the features and design conditions in a product model

Estimators have different preferences for when a crew's productivity rate is appropriate in a given activity and how it should be adjusted for different design conditions. In the motivating case, the drywall estimator selected the crew's base productivity rate for the Install Metal Stud activity based on the resources required (Crew C-1 using Rolling Scaffolding) and then adjusted the productivity to account for wall curvature and component similarity (Fig. 2).

The project-specific features, activities, and resources, and the project-independent Resource Productivity Specifications enable the refinement of the resources' productivity rates. With the three reasoning mechanisms, the resource's base productivity rate is adjusted based on the estimator's preferences for the project-specific features instantiated in the input product model (Fig. 8).

(1) *Identify resource productivity specifications*. Identify the relevant Resource Productivity Specifications that specify how and when to adjust the crews productivity rate. The Resource Productivity Specification that represents this preference is shown in Fig. 3. Analyze the estimator's feature-based product model to determine if the design condition in the Resource Specification is satisfied. If the design condition is satisfied, adjust the base crew productivity rate accordingly.

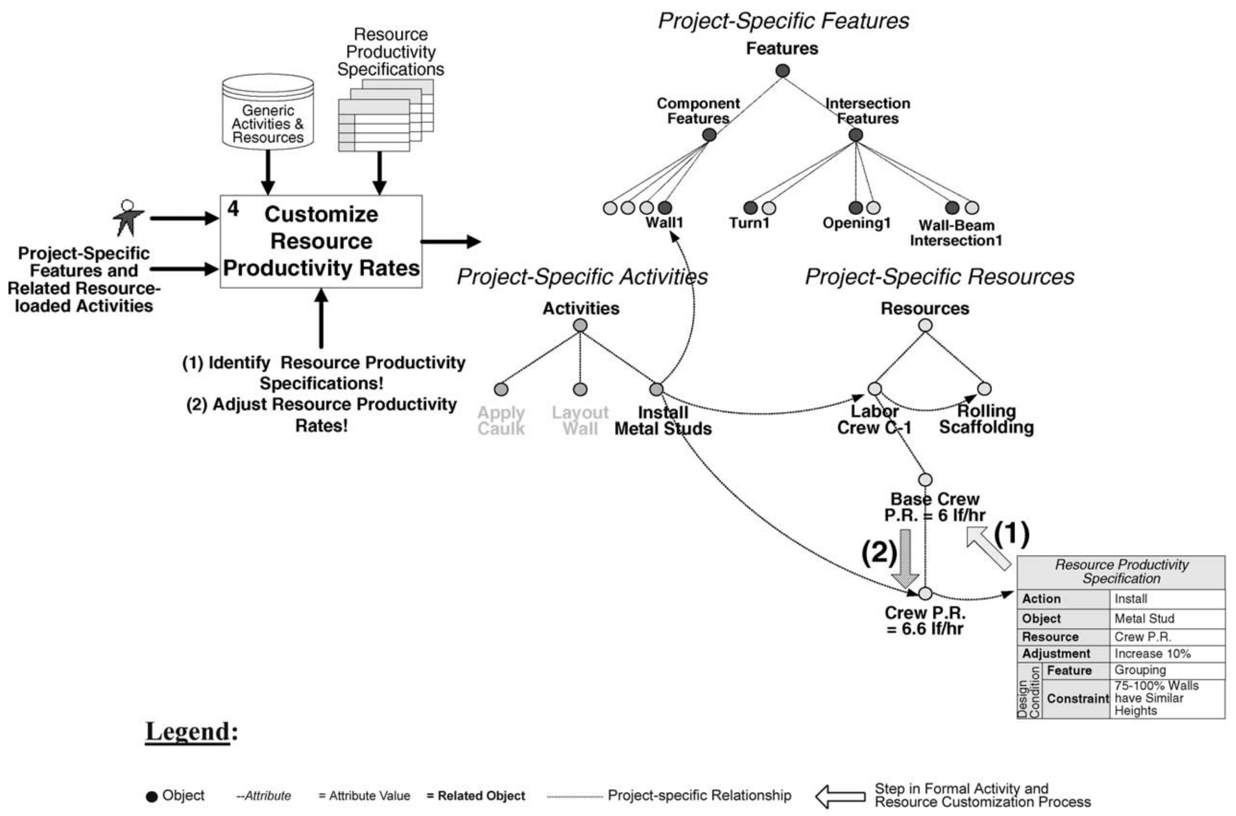

Fig. 8. The fourth step customizes the resource productivity rates for each activity based on the particular features in a given design and the estimator's preferences in Resource Productivity Specifications.

(2) *Adjust the resource productivity rates*. Adjust the resource productivity rates based on the estimator's preference specified in the Resource Productivity Specification. For example, the estimator in the motivating case prefers to increase the base crew productivity rate 10% when 75–100% of the walls have the same height (Fig. 2). Relate the project-specific crew productivity to the activity and relate the Resource Productivity Specifications to the project-specific crew productivity rate.

At this stage in the reasoning process, the activity and resource customization process generates resource-loaded activities that are explicitly related to the estimator's feature-based product model and the estimator's rationale in Activity and Resource Specifications, which is leveraged to generate and maintain cost estimates.

3.2.5. Generate and maintain cost estimates with resource-loaded activities and related features

The resource-loaded activities and related feature-based product model provide the basis for generating and maintaining cost estimates. Calculating the construction costs for resource-loaded activities is a straightforward process. However, the explicit relationships between features, activities, resources, costs, and Activity and Resource Specifications enable the maintenance of cost estimates. Using the explicit relationships, the formal process identifies the cost information affected by design changes and calculates the corresponding cost impact of design changes. Fig. 9 shows the three reasoning mechanisms that calculate costs and reconcile costs if the design changes.

(1) *Calculate quantities and resource durations*. Based on the activity's resources and cost implication, calculate the activities' quantities and resource durations to determine the cost implications of each activity. If an activity has a material cost implication, calculate the activity's material costs using the quantity calculation. Calculate the resource costs using the resource duration.
(2) *Calculate costs*. Calculate the costs for activities depending on the material and resource cost implications of the activity.
(3) *Reconcile costs (if the design has changed)*. Compare the revised cost estimate to the previous estimate to assess the cost impact of a revised design. Identify the affected activities, resources, and resource productivity rates and ask the estimator to accept or reject the specific changes resulting from the design change. Based on the estimator's input, generate the corresponding cost estimate for the revised design. This functionality is possible because the formal process is able to leverage the explicit relationships between features, activities, resources, resource productivity rates, and Activity and

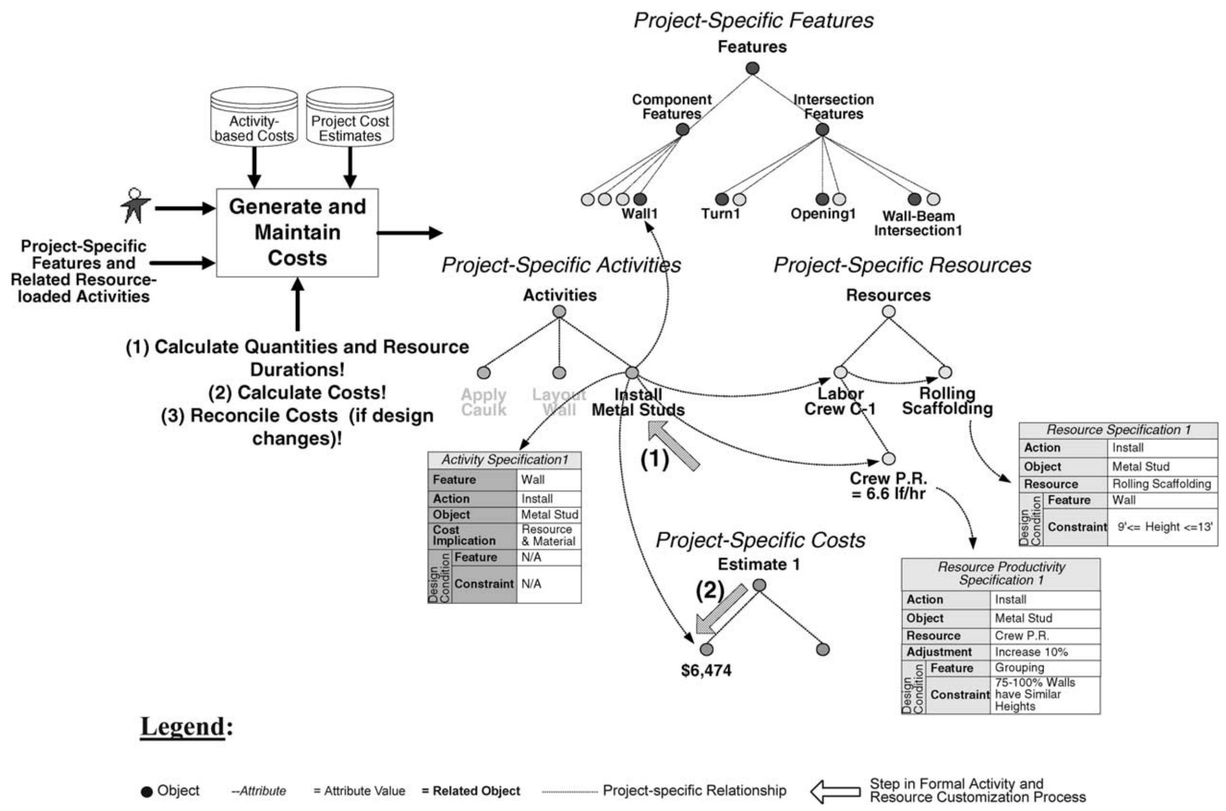

Fig. 9. The fifth step generates and maintains costs for each resource-loaded activity. The process calculates the material quantity and duration of resource use to calculate each activity's material and resource costs. The resulting resource-loaded, cost-loaded, and quantity-loaded activities are explicitly related to the feature-based product model and the estimator's rationale.

Resource Specifications to identify the affected cost information. Reconcile the cost estimate by executing the following steps:

(a) *Identify the cost estimate to compare with the revised estimate*. Query the Project Estimate database (Fig. 9) to identify the previous estimate that the estimator wants to compare with the revised estimate.

(b) *Compare activities in estimates*. Compare the activities in each estimate to determine if any activities in the revised estimate are new, deleted, or changed. For each activity in the revised estimate, identify the corresponding activity in the previous estimate. The activity is matched based on the *feature* requiring the activity, the *object* the activity is acting on, and the *action* being performed. If the activity does not exist in the previous estimate, then the activity is considered to be a 'new' activity required for the revised design. If an activity does not exist in the revised estimate, the activity potentially needs to be 'deleted' for the revised estimate. Then, if the activity in the previous estimate matches the feature, object, and action, compare the activity's equipment, labor, and productivity rate to determine the cost information affected by the revised design. Identify the activity's equipment, labor, and productivity rate using the explicit relationships to this information, as shown in Fig. 9. If the activity's equipment, labor, or productivity rate in the previous estimate is different in the revised estimate, then the activity has 'changed'.

(c) *Estimator input*. List the new, deleted, and changed activities and ask the estimator to accept or reject the changes.

(d) *Create revised estimate*. Generate the revised estimate that includes the new, deleted, or changed activities accepted by the estimator. In the revised estimate, highlight the specific cost information that was affected by the design change. The estimator can also query the estimate to identify why the design change resulted in changes to activities and related resources by using the relationships to the Activity and Resource Specifications (Fig. 9).

This paper described a feature-driven activity and resource customization process that we formalized to help estimators generate and maintain cost estimates from feature-based product models. This process is unique because it assembles activities, resources and resource productivity rates for various features in a given product model and for different estimators' preferences. The result of the process is also unique because it assembles an integrated model that explicitly relates features, activities resources, resource productivity rates, costs and the estimator's rationale. Our tests show that the formal process and resulting integrated model enables estimators to generate and maintain construction cost estimates from feature-based product models more completely (i.e. less ad hoc and with fewer omissions), consistently, and quickly.

4. Validation

We performed a Charrette test [23] and three retrospective tests to demonstrate the power and generality of the formal feature-driven activity and resource customization process [24]. Because ACE implements the formal process, we used ACE to perform each of the four validation tests.

To demonstrate the power of the formal process, we wanted to show that the structure of the formal process enabled estimators to account for the cost impact of features explicitly, consistently, and quickly. 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 level of 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 cost impacts explicitly and excludes ad hoc methods normally used by estimators in practice. 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 [25]. 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 repeated changes to relevant features and feature properties and their associated cost impacts based on the estimators' rationale.

The results of the validation tests demonstrate that the process we formalized enabled practitioners to generate and maintain more complete cost estimates than the state-of-the-art process. Estimators could generate and maintain cost estimates that are less ad hoc and contain fewer omissions than estimators using state-of-the-art tools. The Charrette test demonstrates 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 the state-of-the-art process. Fig. 10 shows the completeness results for the four validation tests. Therefore, the four validation tests demonstrate the power of the formal process by showing that practitioners could account for the cost impact of features more explicitly (completely), consistently, and quickly using ACE than the same practitioners using state-of-the-art tools.

The four validation tests also demonstrate that the formal process is sufficiently general to generate and maintain cost estimates using different estimators' rationale and considering different design conditions for different component types. Two of the retrospective tests evaluated estimators from different companies estimating the same component type. The different estimators for drywall construction were able to represent their preferences in ACE on both projects. In addition, the eight practitioners in the Charrette test were able to represent their preferences in ACE. To demonstrate generality across construction component types, we modeled costs for two different component types in three retrospective test cases. The construction of these two different component types required different activities, methods, and equipment. Moreover, different features and feature properties impacted costs for these two component types. These tests demonstrate the generality of the formal process across component types and user types.

5. Possible uses of resource-loaded and cost-loaded activities and related features

The formal estimating process integrates resource-loaded and cost-loaded activities with feature-based product models for different estimator preferences. This integrated model supports a variety of additional project management functions, such as value engineering, scheduling, and project control.

(1) *Value engineering*. Estimators using a system like ACE can provide specific cost feedback to designers on the particular design conditions that impact construction cost, which is a critical part of the value engineering process. For example, drywall estimators can provide specific cost feedback for the cost implications of different wall heights or wall curvature. On the drywall test cases I studied, the wall height significantly increased the cost of drywall construction because it required additional equipment, it impacted worker productivity, and it required additional activities for cutting drywall. Had the owners and designers known about the cost implications of that design decision they might have chosen a different wall height. The organization of project teams need to change to facilitate this type of team-oriented design process and incentive structures need to change so that designers and builders of facilities are rewarded for more cost-effective designs [21]. By providing prompt and detailed cost feedback about the specific design information that affects construction costs, estimators can help designers to better understand the cost implications of their design decisions and develop more cost-effective designs.

(2) *Scheduling*. The resource-loaded and cost-loaded activities generated from the formal estimating process

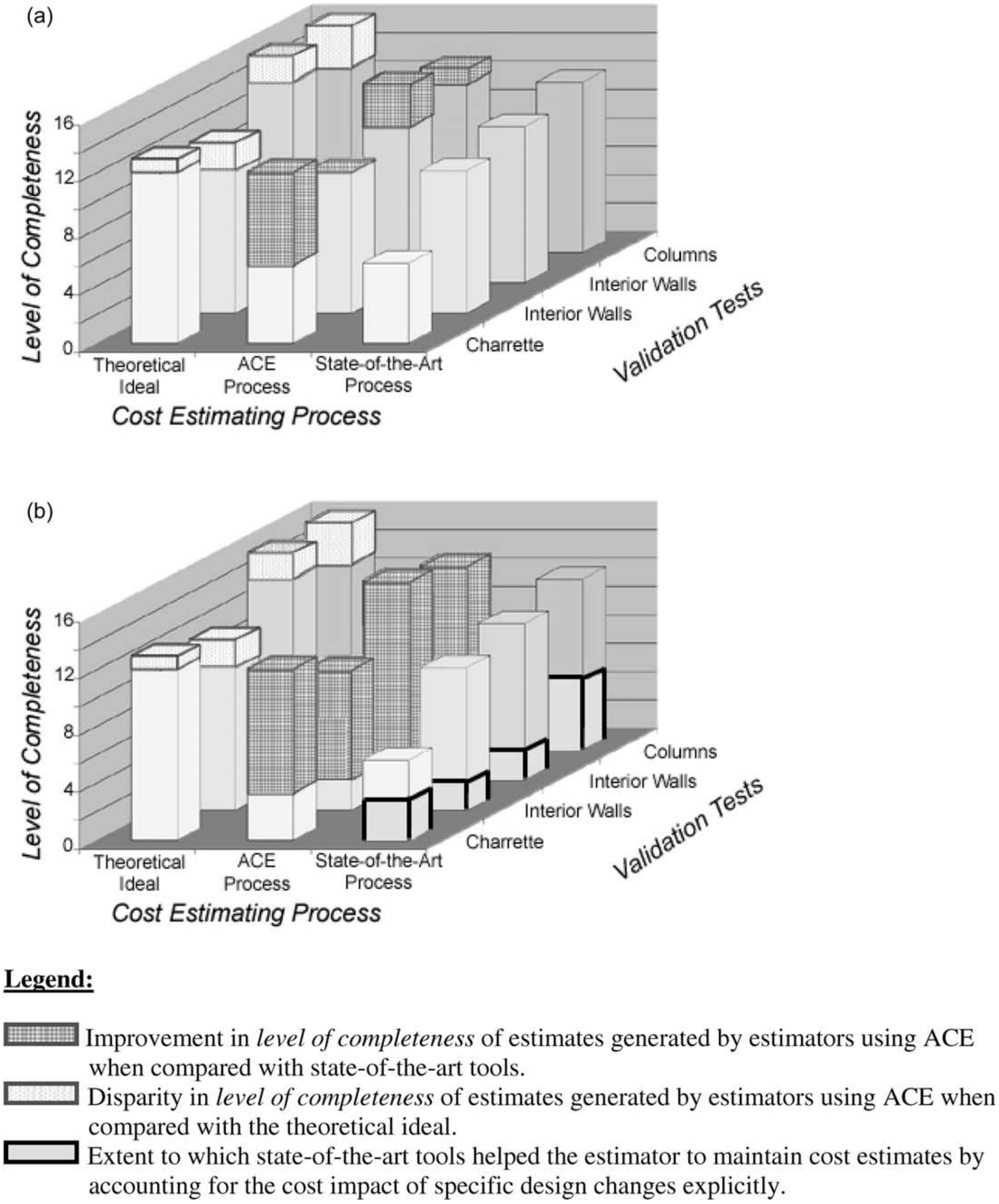

Fig. 10. The results of the four validation tests show that the level of completeness of estimates generated and maintained by practitioners using ACE is substantially greater than practitioners using state-of-the-art tools and approaches the theoretical ideal. (a) The level of completeness of estimates generated by estimators using ACE and state-of-the-art software tools relative to the theoretical ideal. (b) The level of completeness of revised estimates generated by estimators for specific design changes using ACE and state-of-the-art software tools relative to the theoretical ideal.

could provide the input to a scheduling program. A scheduler would need to add precedence relationships to create a resource-loaded and cost-loaded schedule. Resource-loaded and cost-loaded schedules are rarely used in practice because of the time it takes to create and maintain them. The formal estimating process addresses this problem by quickly and consistently generating resource-loaded and cost-loaded activities.

(3) *Project control*. Today, time and cost control is performed using separate systems with limited information sharing between them, which leads to inconsistencies between the information and inaccuracies when determining the cost and schedule status. Moreover, cost control is tracked at such a high level that it is nearly impossible to relate each cost control account back to the cost items in the cost estimate that provide the basis for evaluating the cost status. This research enables the generation of resource-loaded and cost-loaded activities and related features that could provide the foundation for an integrated project control system. Project teams would need to add methods to update the cost and

schedule information based on actual progress. Mechanisms would also need to be added so that the aggregated cost control accounts know what cost items in the estimate they relate to and how the information was aggregated so that cost and schedule information could be shared across the different levels of detail. By linking the product, schedule, and cost information, the project's cost and schedule status could be easily obtained if the cost and schedule information is consistently updated. An integrated project control system can facilitate the early detection of problem activities that are running over budget or falling behind scheduled progress.

6. Conclusions

This paper contributes a formal feature-driven activity and resource customization process for predicting construction costs based on a generic representation of estimators' rationale and a project-specific feature-based product model. This process helps estimators to avoid ad hoc and error-prone methods that lead to inconsistencies and inefficiencies in the cost estimation process. The result is an integrated model consisting of features, activities, resources, and costs that supports the generation and maintenance of construction cost information for different estimators' preferences, design features, and construction processes.

We limited the scope of our research in many ways. We excluded cost-incurring features that result from dissimilar component types that are not connected and factors exogenous to a design, such as site characteristics and resource skill and availability, that ACE cannot estimate. The activity and resource customization process assumes that the need for activities precedes the need for resources. There may be instances where the types of resources available drive the requirement for activities. We speculate that cost estimating software should allow both sequences for generating activities and their corresponding costs. Finally, we excluded cost impacts associated with design changes that affect activity sequencing and hinder activity progress. Future extensions should represent other types of features and consider activity sequencing and progress when assessing the cost impact of design changes.

We believe that our approach is scalable, although we have not yet tested the limits on its scalability. Current practice clearly is scalable in the sense that it estimates cost of large and small facilities. In general, current practice uses the simple heuristic of estimating both material and production costs by linear distance or area, applying simple factors to account for special factors such as unusual complexity. Standard industry sources, e.g. R.S. Means, now provide baseline production rates, and companies develop their own unit item costs in their local markets.

The scalability of our method depends on a similar two-part process: developing generic Activity and Resource Specifications and applying those specifications in practice for individual jobs. Thus, our method requires some 'up-front' creation of specification types to account for different design conditions, including component properties (e.g. curvature), intersection features (e.g. wall turns), properties of intersection features (e.g. orientation of wall turns), and macro features (e.g. component similarity). While creating these structured specifications requires effort, it can apply for all the similar work of a company or, potentially, even the field of construction. Once created, it was our experience that an estimator can apply specifications quickly and easily.

We tried to make the specifications as simple and intuitive as possible so that estimators could easily fill out the templates without the support of IT professionals. The number of specifications needed will vary with the estimator and the type of work being estimated. Based on our interviews with drywall estimators and typical building construction, we believe drywall estimators would need to create approximately five Activity Specifications to account for intersection features, five Activity Specifications to account for typical wall subcomponents, three Resource Specifications to account for the different labor and equipment configurations required for different wall heights, and five different Resource Productivity Specifications to account for productivity rate impacts resulting from component properties and component similarity. For design conditions that are not recurring from project to project, estimators may not need to create the necessary Activity and Resource Specifications. However, they would only need to create the specification once and the system would check for that design condition without any additional effort from the estimator. The effort it takes to create these specifications needs to be weighed against the effort it currently takes estimators to create and maintain their cost assemblies and to make the necessary project-specific adjustments manually. Although we have not tested for the time requirements to create and maintain the specifications, we believe that our approach is much more efficient than the current process. Additional effort will be required initially, but this effort is scalable, and even within a small office, it should more than pay for itself throughout the life of a project and from project to project.

This paper introduces a generic method to estimate the cost of constructing building systems that considers the material and manufacturing process costs. It describes the implementation of that method in the computer and demonstrates the power of the method to perform the knowledge-intensive cost estimation task. The method builds on explicit symbolic representation of features in the product model and presents a symbolic process model that estimates the costs. The implementation is interactive, enabling users to estimate the costs of the product and process design based on their preferences, and to estimate the change in costs following specific interactive changes in either the product or the construction process design. While the specific motivating test cases and validation examples are drawn from the architecture–engineering–construction industry, the methodological framework seems to apply to many types of manufactured products.

Our research has advanced the current state of knowledge about the representation and reasoning mechanisms required to generate and maintain the relationships between scope, schedule, and cost information. We addressed the limitations in current research by providing a process that is general enough to consider different estimator preferences and formal enough to account for different features explicitly. We also extended the current activity definition by generating activities that know what feature requires the activity's execution, what resources are executing the activity and why, how much the activity costs, and the estimator's rationale for relating this information. The integrated model provides a foundation for representing a project's scope (features), schedule (activities, resources, and durations), and corresponding cost.

Software tools that explicitly represent the relationships between a project's scope, schedule, and corresponding cost can help project teams to test different design, cost, and schedule alternatives, and maintain integrated models as the project evolves. Hence, through better modeling and analysis prior to construction, project teams can avoid many of the inefficiencies that result in cost overruns and schedule delays, and better manage and control the design and construction process.

Acknowledgements

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

[1] Dym CL, Levitt RL. Knowledge-based systems in engineering. New York: McGraw-Hill; 1991.

[2] Laitinen J. Model based construction process management. PhD Thesis, Royal Institute of Technology, Stockholm, Sweden; 1998.

[3] Aouad G, Betts M, Brandon P, Brown F, Child T, Cooper G, Ford S, Kirkham J, Oxma R, Sarshar M, Young B. ICON: integration of construction information. Department of Surveying and Information Technology Institute, University of Salford; 1994.

[4] Froese T, Yu K, Shahid S. Project modeling in construction applications. Computing in Civil Engineering: Proceedings of the Third Congress, ASCE, Anaheim; June 1996:572–78.

[5] Marir F, Aouad G, Child T, Brandon P. OSCONCAD: a model-based CAD system integrated with computer applications. ITCON 1998;3:25–44.

[6] Clayton M, Kunz J, Fischer M. Rapid conceptual design evaluation using a virtual product model. Engng Appl Artif Intell 1996;9(4): 439–51.

[7] Fischer M. Constructibility input to preliminary design of reinforced concrete structures. Technical Report #64, Center for Integrated Facility Engineering, Stanford; 1991.

[8] Hanna AS, Willenbrock JH, Sanvido VE. Knowledge acquisition and development for formwork selection system. J Construct Engng Mgmt, ASCE 1992;118(1):179–98.

[9] Udaipurwala A, Russell AD. Reasoning about construction methods. Proceedings of Construction Congress VI, Orlando Florida; February 2000:386–95.

[10] Froese T, Rankin J. Representation of construction methods in total project systems. Computing in Civil Engineering. Proceedings of the International Computing Congress, ASCE, Boston, MA; October 19–21 1998:383–94.

[11] Thomas HR, Zavrski I. Construction baseline productivity: theory and practice. J Construct Engng Mgmt, Am Soc Civil Engrs 1999; 125(5):295–303.

[12] Thomas HR, Sackrakan A. Forecasting labor productivity using the factor model. J Construct Engng Mgmt, Am Soc Civil Engrs 1994; 120(1):229–39.

[13] Smith GR, Hanna AS. Factors influencing formwork productivity. Can J Civil Engng 1993;20(1):144–53.

[14] Sanders S, Thomas R. Factors affecting masonry-labor productivity. J Construct Engng Mgmt 1991;117(4):626–44.

[15] Akbas R, Fischer M. Examples of product model transformations in construction. 8dbmc, Durability of building materials and components. Information technology in construction. CIB W78 Workshop, Vancouver, BC, Canada; 1999:4:2737–46.

[16] Aalami F. Using method models to generate 4D production models. PhD Thesis, Stanford University, Stanford; 1998.

[17] Darwiche A, Levitt R, Hayes-Roth B. OARPLAN: generating project plans by reasoning about objects, actions and resources. AI EDAM 1989;2(3):169–81.

[18] Jagbeck A. MDA planner: interactive planning tool using product models and construction methods. J Comput Civil Engng 1994;8(4):536–54.

[19] Stumpf A, Ganeshan R, Chin S, Liu L. Object-oriented model for integrating construction product and process information. J Comput Civil Engng 1996;10(3):204–12.

[20] Staub-French S, Fischer M, Kunz J, Paulson B. An ontology for relating features with activities to calculate costs. Accepted for publication Journal of Computing in Civil Engineering.

[21] Staub-French S, Fischer M. Industrial case study of electronic design, cost, and schedule integration. Technical Report #122, Center for Integrated Facility Engineering, Stanford; 2001.

[22] Staub-French S, Fischer M, Kunz J, Paulson B, Ishii K. A feature ontology to support construction cost estimating. Accepted for publication in AIEDEAM: Artifical Intelligence for Engineering Design, Analysis and Manufacturing.

[23] Clayton M, Kunz J, Fischer M. The Charrette test method. Technical Report 120, Center for Integrated Facility Engineering, Stanford; 1998.

[24] Staub-French S. Feature-driven activity-based cost estimating. PhD Thesis, Stanford University; 2002.

[25] Timberline Software Company. Precision estimating extended and CAD integrator. Users Documentation, Beaverton, Oregon; 2001.