Boiler Erection Scheduling Using Product Models and Case-Based Reasoning

By Ren-Jye Dzeng[1] and Iris D. Tommelein,[2] Associate Member, ASCE

Abstract: Contractors who repeatedly construct facilities designed by copying major parts from one project to the next find that previously developed schedules associated with those designs could be reused to schedule new work. To facilitate such reuse, project characteristics must be articulated and associated schedules described to include not only traditional, numerical scheduling data, but also scheduling constraints. In addition, knowledge about how to reuse schedules must be available. The CasePlan system, presented here, supports and augments the scheduling activity of people who reason about cases—each case describing the design and schedule of a completed project—to generate new project schedules. Specifically, CasePlan reuses annotated cases to automatically schedule the erection of power plant boilers. Because such boilers have a more-or-less standardized design, a generic boiler product model can serve as the basis for assessing similarities between designs. On this basis CasePlan selects schedules for reuse. A user can also interact with CasePlan to isolate fragments of case schedules and adapt them to better suit the variables of the new project at hand.

INTRODUCTION

Architects/engineers (A/Es) who repeatedly design facilities of the same kind may reuse designs as a whole or in part. This is the case, for instance, in cookie-cutter design of industrial facilities such as power plants. When parts of a design are copied from one project to the next, contractors who have experience in constructing those facilities find that (parts of) schedules can also be reused to schedule new construction work. This saves time, which is of the essence during bid preparation, but, more importantly perhaps, it enables them to reuse some of their company's proven field expertise that is reflected in those schedules.

A scheduler intent on reusing an existing schedule will try to recall major parts from a past project's design and schedule (including components and details, construction resource availability, contractual agreements, environmental factors, etc.) that recur in or in some way resemble the new project at hand, in order to determine which (parts of) schedules could be reused. The scheduler may then choose a single existing schedule for adaption to the new design, contract, and construction needs, or choose parts from several existing schedules, link them, and adapt them to better suit those needs.

This problem-solving method—namely, reasoning about past cases then retrieving and adapting them to solve a new problem—is termed case-based reasoning (CBR). It is a method used by schedulers who can remember salient features about past projects and who have enough such projects to draw upon. CBR works well when cases are organized so that the search for and adaption of relevant cases can proceed in a systematic way. Unfortunately, most valuable project knowledge gets buried in vast amounts of documentation of a paper-based archival system. Moreover, novice schedulers rarely have access to relevant, historic case data because it is in the head of their seniors and not accessible in any other way than through verbal communication. A lot of knowledge is lost when senior schedulers are tied up with other work, get promoted, leave the organization to take on other duties, or retire.

[1]Assoc. Prof., Civ. Engrg. Dept., National Chiao Tung Univ., Hsinchu, 30050 Taiwan.

[2]Acting Assoc. Prof., Civ. and Envir. Engrg. Dept., 215-A McLaughlin Hall, Univ. of California, Berkeley, Berkeley, CA 94720-1712.

Note. Discussion open until February 1, 1998. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on September 9, 1996. This paper is part of the *Journal of Construction Engineering and Management*, Vol. 123, No. 3, September, 1997. ©ASCE, ISSN 0733-9364/97/0003-0338–0347/ $4.00 + $.50 per page. Paper No. 14048.

The aim of the research described in this paper was to address (1) how cases could be described in a systematic way to facilitate classification and reuse; and (2) how a computer-based system could automate the generation of construction schedules by reusing cases that reflect design and construction experience.

RELATED WORK

Automated Planning Systems

Construction planning consists of defining activities with their durations, precedence relationships, and resources, whereas scheduling involves applying the critical-path method (CPM) to calculate early and late activity start and finish times, and floats. Computer tools that perform CPM calculations are widely used but few exist that address the planning task. Plan generation has been automated using artificial intelligence (AI) programming techniques. Examples of such AI-based planning systems (AI-planners, in short) are BUILDER (Cherneff et al. 1991), PLANEX (Zozaya-Gorostiza et al. 1989), GHOST (Navinchandra et al. 1988), Know-Plan (Morad and Beliveau 1991), OARPLAN (Darwiche et al. 1989), and SIPEC (Kartam and Levitt 1990). These are constructive planning systems: they always generate a plan from scratch for each new project. Their knowledge typically comprises (1) a (usually functional) hierarchy of components for a particular type of project (e.g., high-rise steel construction) in which each type of component has a construction activity associated with it (though in principle possible, few models present alternatives); and (2) planning rules or constraints acquired from expert practitioners, based on rules-of-thumb, or stated in the literature. These systems use this knowledge to pick a network of activities for each product's component based on its association in the hierarchy, and to test the preconditions of each activity and determine its precedence relationship relative to other activities in the plan.

Many of these AI-planners were developed to provide designers with constructibility feedback before award of the construction contract (Gray 1986). Their knowledge includes general construction principles but rarely project characteristics and resources such as contractor equipment, material availability, or site constraints (PLANEX is an exception). In the process of generating solution plans, these AI-planners keep track of all possible alternatives while pruning out only those that do not meet the constraints. Because relatively few constraints are known at design time, they typically can output many plans that satisfy all constraints. By leaving the user (contractor) with the largest possible set of satisfying plans,

J. Constr. Eng. Manage., 1997, 123(3): 338-347

they follow a least-commitment approach, but consequently, the plans they generate are not "field-executable": they need to be refined with site-specific data before they can be used to guide field work.

In contrast, people scheduling work typically investigate only a few alternatives but in greater detail, i.e., they follow an early-commitment approach. For instance, after investigating two or three alternative sequences or durations of activities, a person may commit to one of them, discard the others, and develop the remainder of the schedule based on that choice. We are not arguing here that either least commitment or early commitment is better. This depends on the state of information, objectives, and tools available to the problem solver. Tommelein et al. (1991) discuss trade-offs between the two. However, early commitment is the approach for people to pursue because scheduling knowledge is so varied, difficult to spell out exhaustively, and it is project- and organization-dependent. Accordingly, people tend to become good schedulers by gaining hands-on experience on a project-by-project basis.

Some automated early-commitment AI-planners exist, though they tend to focus on only one specific resource, e.g., Thabet and Beliveau (1993) sequence activities based on space availability. CasePlan can accommodate sequencing knowledge based on any one or several resources. Constructive AI-planners have no means of recognizing that they planned similar projects previously or that plans can be reused. CasePlan recognizes that one can develop cases from previous projects and schedules, and reuse this knowledge. Our work is most similar in approach to Miresco's (1992), but CasePlan has many more features and capabilities than Miresco describes.

Case-Based Reasoning

The knowledge base for a CBR system comprises a set of cases plus mechanisms for retrieving cases and adapting their solutions to suit the new problem at hand, rather than a set of rules and facts that make up a traditional rule-based system. CBR has played an important role in the development of AI and expert systems (Kolodner 1993) addressing tasks such as planning, design, explanation, diagnosis, classification, and natural-language parsing. To our knowledge, however, there exist no CBR systems for generating construction schedules.

A CBR system adopts three steps in solving a problem (Dzeng 1995):

1. Identifying and retrieving useful case(s). This requires access to cases and knowledge to assess the appropriateness for retrieval of one case over another. It is typically assumed that the most useful case is the one most similar to the new problem, though the notion of similarity varies; e.g., it can be based on the match between goals or, instead, between design attributes in the new problem versus those in an archival case. Nevertheless, matching and ranking requires three steps: (1) determining the correspondence of matching attributes, (2) assigning an importance weight to attributes used in assessing similarity, and (3) determining the degree of similarity between corresponding attribute values based on some similarity measurement (e.g., semantic hierarchies, quantitative or qualitative ranges of values, and functional roles the values play in problem solving).
2. Reusing and adapting the retrieved case(s). Solutions from retrieved cases seldom perfectly match but need to be adapted to solve the new problem. One may (1) reinstantiate a partial solution from an old solution, (2) adjust some parameters, (3) transform the solution's structure (e.g., delete old or add new parts) before applying it to the new problem, or (4) replay the reasoning steps used in the old case but using data and constraints in the new problem.
3. Categorizing and storing the new problem and solution as a case. When a new problem's solution has been validated, a new case can be created and stored for reuse. Organizing stored cases is not mandatory, but using some indexing scheme (e.g., articulating goals, salient attributes of the problem, or factors that caused failure during problem solving) may improve the efficiency and success rate of subsequent searches.

Most CBR systems use similar concepts and approaches, but vary in their combination thereof to suit their domain of application. Like other systems, CasePlan uses CBR as a means to reuse knowledge specific to individual projects (which is lacking from existing construction planning systems) but it is unique in that it tackles planning problems in construction and uses product models as the basis for case organization.

Product Modeling

A case-based scheduling system should not be a stand-alone tool, but tie into existing modeling practices in design and construction. Establishing a representation to integrate and exchange data spanning the lifetime of a facility that is of use to all participants involved with a common architectural/engineering/construction (AEC) project is a widely recognized problem, exacerbated by the use of heterogeneous computer systems. It has driven the development of an international standardization effort that involves people in AEC practice as well as academia, e.g., the *Special Issue on Data and Product Modeling* in the *Journal of Computing in Engineering* (*Special* 1996).

Several standardization efforts were investigated for this research, namely:

1. Plant information network (PIN). The Electrical Power Research Institute developed a comprehensive power plant model as a guide to specifying integrated computer-aided applications ("Guidelines" 1987). Twelve volumes describing this model were reviewed (unfortunately, no electronic copy could be obtained) but the PIN model was not adopted because it appears to have been superseded by PlantSTEP described in the following.
2. Standard for the exchange of product model data (STEP) and general AEC reference model (GARM). The most significant product-model standardization effort to date has resulted in the International Standards Organization (ISO) draft standard 10303, named STEP (*ISO* 1994a). STEP is an abstract model to be refined with domain-specific details in order to reflect each industry's ontology. For instance, PlantSTEP's application protocol AP-277 serves the power plant industry (*ISO* 1994b) and is currently under development. PlantSTEP is most relevant to the present research, but, unfortunately, insufficient detail had been formalized at the onset of this research for it to be useful. However, our modeling methodology is compatible with PlantSTEP's.

   GARM, developed specifically for AEC applications, was the next best available STEP standard, though it also is an abstract reference model under which product-specific models must be developed. GARM entities describe a product in terms of product definition units (PDUs), which can be the entire product, its components and subcomponents, or relationships between PDUs.
3. Work breakdown structures (WBSs). To elicit the terminology of boiler components used in industry practice, WBSs were obtained from boiler manufacturers Asea

J. Constr. Eng. Manage., 1997, 123(3): 338-347

Brown Boveri Combustion Engineering Services, Inc. (ABB-CE) and Zurn Industries, Inc. (Zurn). WBSs help organize work and serve scheduling and cost-engineering functions. However, they provide no framework for industrywide AEC data exchange. First, they do not capture all data relevant to all AEC participants. Second, many firms use only their own, in-house WBS, which differs from competitors', and not all use it consistently. Nonetheless, given their availability, CasePlan's boiler product model was based on the WBS of one boiler manufacturer and made consistent with GARM's entity representation (Dzeng and Tommelein 1993).

SCHEDULING ERECTION OF POWER PLANT BOILER

Industry Practice: Case Data Collection

Studying the erection of power plant boilers was well suited for this research because:

1. Power plant construction is reasonably complex, though the construction methods used are routine so (parts of) schedules are likely to be reused from one project to the next.
2. Boiler erection is possibly the most critical activity for timely completion of construction of a power plant, so optimizing erection schedules is important.
3. Boiler erection includes the installation of large, heavy, custom-made mechanical equipment. This requires expensive lifting equipment, complex rigging procedures, and highly skilled personnel. Identifying these as part of the scheduling process, before construction starts, is a key to success.
4. Boiler components require protection from inclement weather and must be installed in a timely fashion upon their arrival on site so they can be tested before their warranty expires.
5. Power plant projects are capital intensive. Because of the time value of money, it is important that the project duration be kept short. Many power plants are therefore developed on a fast-track schedule. Components with a long lead time must be procured early and their arrival on site often drives the schedule of other work.
6. Contracts typically include financial incentives to finish the project early as this means the early start of revenue-making operation.

Therefore, contractors are more competitive in constructing these kinds of projects when they invest a reasonable amount of effort in scheduling their work.

Boiler erection was also chosen for practical reasons:

1. A fair number of boilers exist whose designs (general arrangement of components) are similar. Thus, it was anticipated that project schedules would be similar and therefore reusable.
2. The erection schedules we obtained from industry referred to approximately 20 major boiler components (though there may be on the order of 15,000 subcomponents), and they comprised from 60 to 90 activities. This degree of complexity suited this research.
3. Professionals in industry were interested in this work and

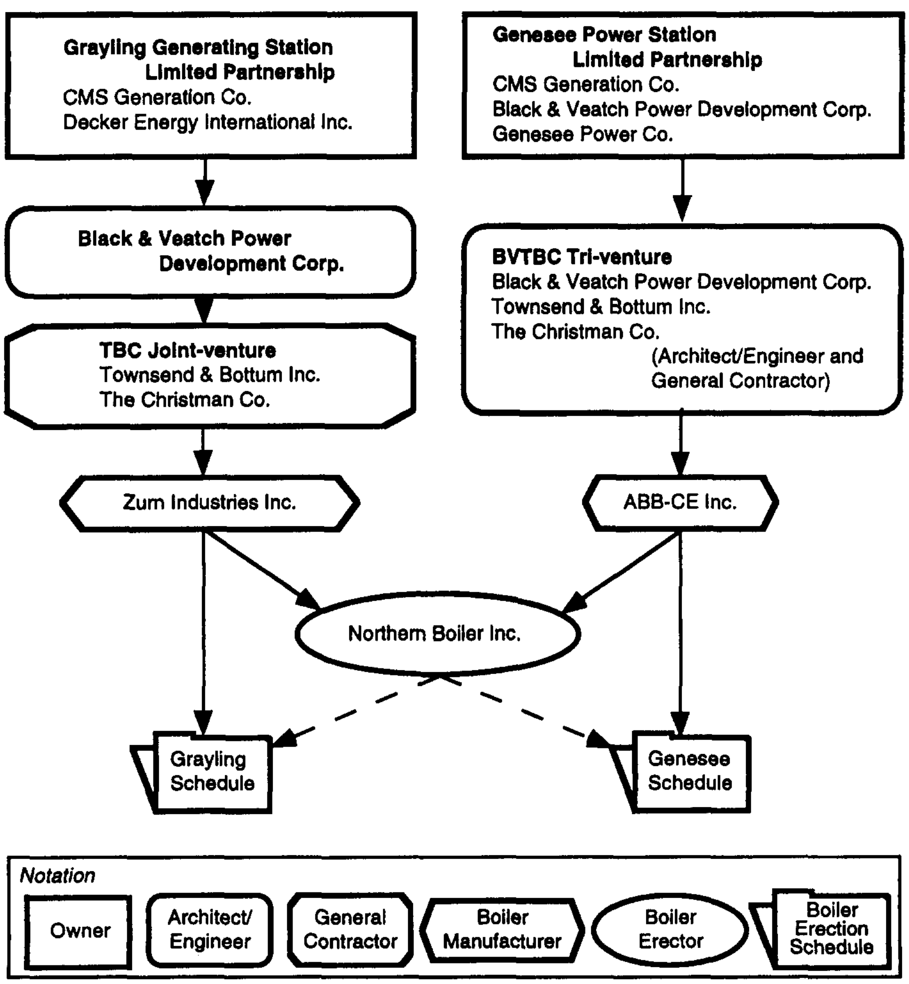

FIG. 1. Organization of Grayling and Genesee Projects

J. Constr. Eng. Manage., 1997, 123(3): 338-347

were willing to provide engineering expertise and data. In fact, boiler erection was suggested as a subject of study for this research by professionals from Black & Veatch.

Two boiler projects were studied in detail: the Grayling Generating Station (Grayling) (''Biomass-fired'' 1993) and the Genesee Power Station (Genesee) (Desmond 1995). Both were turn-key projects located in southeast Michigan and owned, designed, and built by the same parties but under different joint-venture agreements. Fig. 1 shows their owner (represented by a rectangle), A/Es (rounded rectangle), general contractor (octagon), boiler manufacturer (hexagon), and boiler erector (oval). Their similar project organization allowed us to concentrate on schedule differences caused by differences in facility design, site environment, and project specifications.

Both plants are waste-wood fueled. They generate nearly the same power (Grayling 34 MW and Genesee 35 MW). To lower the design cost the A/E reused Grayling's design for Genesee, so the structures of the two plants are similar (e.g., most bays have the same length) albeit not identical (e.g., the foundations are different because of different soil conditions). In addition, they would have liked to reuse Zurn's boiler design at Genesee, but the contract was awarded through an open-bid process and a competitor, ABB-CE, had the lowest bid. Despite some differences in manufacturer design, resulting among other things in differing requirements on hold-out steel during construction, the boiler sizes and configurations are similar because they were designed to meet the same, owner-specified constraints (including fuel and plant capacity). Grayling's boiler erection started in May 1991 and was completed in March 1992; Genesee's started in December 1994 and was completed in October 1995.

As-built boiler erection schedules were obtained from Zurn and ABB-CE. As is common practice, the manufacturers were responsible not only for designing the boiler and manufacturing its components, but they also were substantially involved with its erection. Their schedules had therefore been generated with a considerable amount of input from the boiler erector.

Both companies used the Primavera Project Planner (P3) software (*Primavera* 1993). They saved routine networks—parts of larger schedules—as reusable ''fragnets'' as they are called in P3. P3 did not suit our research needs, however. It is impossible in P3 to describe relationships between product components (e.g., economizer, drums) and activities or subnetworks (unless one treats components as resources of activities). Yet, doing so is useful when searching through a collection of schedules to find reusable pieces. Also, P3's built-in batch language to write macros is limited. Sophisticated computation must be performed through C-language routines external to P3. Yet, annotating a schedule with formulas or other computable expressions to express scheduling knowledge eases schedule adaption for reuse.

Scheduling Knowledge

Knowledge for creating a field-executable schedule is expressed by means of constraints, which fall into six categories (Dzeng and Tommelein 1994, after Gray 1986 and Echeverry et al. 1991):

1. Facility components and their relationships. Components determine which construction alternatives are feasible, e.g., the degree of prefabrication of an economizer affects the duration of construction and the installation equipment required. Physical and structural relationships between design elements (support, embedment, coverage, enclosure, etc.) often govern construction sequencing, e.g., drums of a top-supported boiler cannot be installed until the supporting frame is in place, but may require that some steel be held out for access.
2. Resources (time, money, people, equipment, material, and space). The availability of resources determines activity durations, imposes sequencing of otherwise concurrent activities, and affects lead times, e.g., all waterwalls of a boiler may be installed simultaneously or parts of it concurrently depending on the availability of labor, equipment, material, and space to conduct the work safely and without undue interference. Similarly, trade interaction will prevent the structural contractor from closing the roof until the boiler erector has finished hoisting equipment through it.
3. Procurement and mobilization of equipment and materials, and hiring of labor. The lead time for delivery of custom-designed mechanical equipment may dictate the earliest project completion time, e.g., a turbine generator may take more than a year to be manufactured and delivered to site, so its installation tends to define the critical path. A scheduler will work backward from this delivery date to set the pace for erecting the power plant building and boiler.
4. Site environment. The site environment (climate, access, topography, etc.) determines the timing and nature of construction activities, e.g., most contractors need to clear and grub their site and build temporary facilities; Michigan contractors may not start foundation work before early spring to avoid frozen ground, soggy access roads, and problems associated with placing concrete in cold weather.
5. Regulations and specifications. Contracts specify that testing and inspection be performed or permits obtained at given times. These introduce schedule delays or mark milestones. The Occupational Safety and Health Administration (OSHA) (*Construction* 1985) regulates that ''safety nets shall be installed and maintained whenever the potential fall distance exceeds two stories or 8 m [25 ft].'' The erection of steel therefore awaits installation of the floors two stories below, though a contractor may ''install safety net'' and ''remove safety net'' to speed up frame erection.
6. Scheduling detail and format. The type of information, level of detail, and format of a schedule is a function of who generates the schedule, the information available at that time, contractual arrangements, schedule use, and the timing and frequency of schedule generation and updating, e.g., a contractor's schedule may detail activities for work performed with its own forces and only summarize those for subcontracted work.

Many AI-planners use ''Facility components and their relationships'' as the primary constraints and they encode some ''Regulations and specifications'' into their heuristic rules for sequencing activities. Most plan with different ''Scheduling detail and format'' in that they generate hierarchical plans, but the level of detail is predetermined and not for the user to change. Only PLANEX considers ''Resources'' and ''Site environment'' constraints. None consider ''Procurement'' constraints. In contrast, CasePlan allows practitioners to account for all these constraints so schedules can be better custom tailored.

CASEPLAN ARCHITECTURE AND COMPONENTS

Project Modeling Knowledge

The CasePlan architecture builds on the premise that a generic product model, comprising a hierarchy of classes that represent abstract and physical components with has-a-com-

J. Constr. Eng. Manage., 1997, 123(3): 338-347

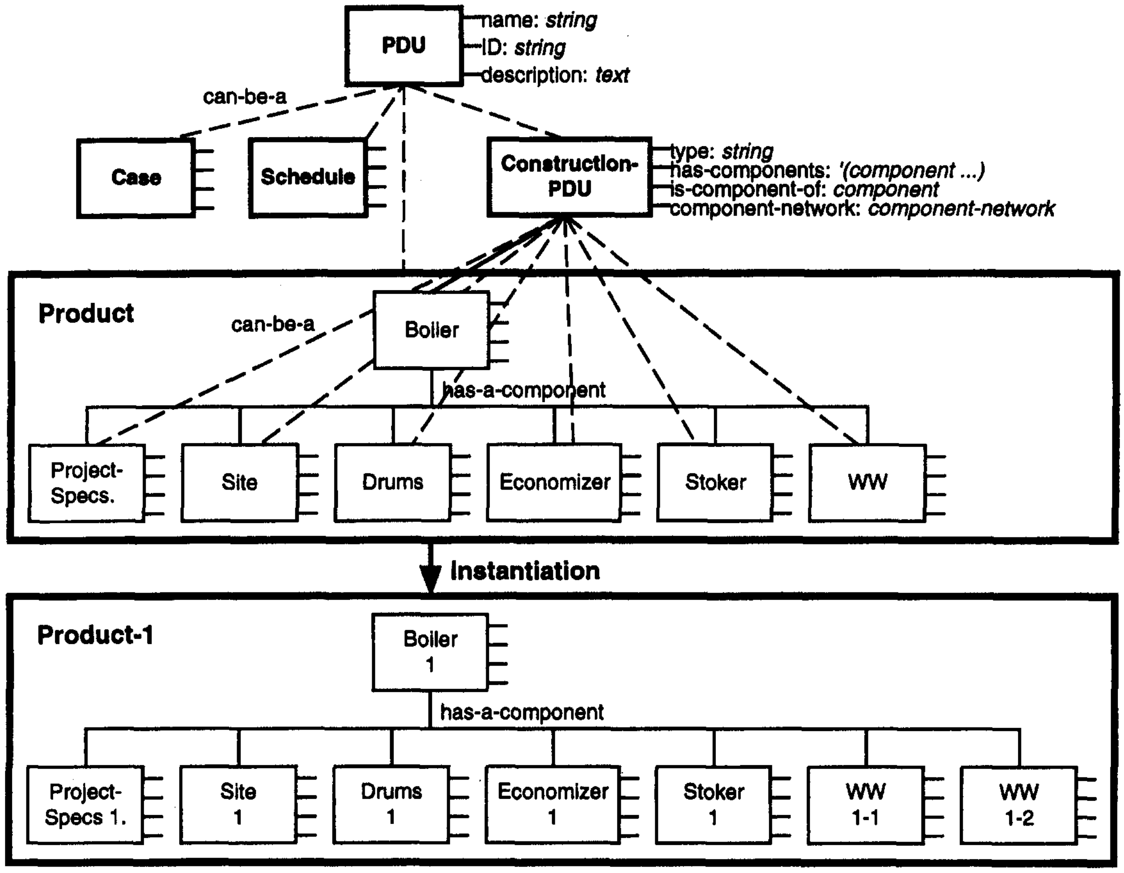

FIG. 2. Class Inheritance

ponent links between them, can describe the type of project of interest (here, power plant boilers). Specifically, a generic 'Case' for a boiler comprises a 'Product' and a 'Schedule' (Fig. 2). The product (here, 'Boiler') in turn comprises components such as 'Drums', 'Economizer', and 'Waterwall' ('WW'), but also 'Project Specifications' and 'Site'.

The Project Specifications class describes constraints of the type "Procurement, mobilization, and hiring" and "Regulations and specifications." Similarly, 'Site' describes "Site environment" and 'Product' describes "Facility components and their relationships" constraints. 'Schedule' comprises activity subnetworks and precedence links between them, which controls "Scheduling detail and format." "Resources" can be assigned to activities and their productivity used to estimate activity durations.

The 'Case' for an individual, new project is created by instantiating the 'Project' class hierarchy, so it may comprise zero, one, or several instances of each component (e.g., 'WW 1-1' and 'WW 1-2' both instantiate 'WW'). At first, a new 'Case' will not include a 'Schedule' instance but it will be created by CasePlan.

In terms of implementation, 'Product', 'Schedule', 'Case', and 'Construction PDU' are classes specializing the class 'PDU' (see Fig. 2 and section "Related Work"). A 'Construction-PDU' has attributes: (1) *type*, naming the class type (e.g., the product 'Boiler' that describes the hierarchy as a whole, or the component 'Economizer') of the item in the product model; (2) *has-components*, listing the components of the 'Construction-PDU'; (3) *is-component-of*, referring to the object of which the 'Construction-PDU' is a component; (4) *component-network*, referring to the sequence of activities that represents the process by which to construct the 'Construction-PDU'.

Again, a number of classes specialize 'Construction-PDU'. They inherit its attributes plus add their own. 'Site' may have attributes: *location*, *soil type*, *ground-water level*, etc. and 'Project Specifications' may *require activities* (e.g., to include the "Hydro test" activity that cannot be directly related to any one boiler component, but which is to be performed when the boiler has been substantially completed) and *impose constraints* (e.g., to specify that some inspection activity must occur before a certain date, i.e., that it is subject to a timing constraint).

Schedule Modeling Knowledge

A CPM schedule's activities and sequential links are instances of the classes 'Activity' and 'Link', both of which specialize the 'PDU' (not shown in Fig. 2).

Component Network

CasePlan constructs a schedule by determining a network of activities (termed a "component network") describing the construction process for each component in the product model and then combining them into a single large network. A sequential link in a component network is of type start-to-start (SS), start-to-finish (SF), finish-to-start (FS), or finish-to-finish (FF). Unless marked otherwise, a single line represents the default FS link. Links that have no arrow imply that they go from left to right.

Fig. 3(a) shows the component network for a typical erection sequence for 'Drums'. First, the steam drum (upper drum) is raised to a height approximately equal to the length of the generating bank while the mud drum (lower drum) is raised just above the ground. With the drums temporarily secured, pipefitters assemble the generating bank, which connects the drums, near the ground. When the bank assembly (including both drums and therefore referred to as the component 'Drums', in the plural) is completed, it is further raised. After the drums have been erected, pipefitters finish their piping hookups. Fig.

J. Constr. Eng. Manage., 1997, 123(3): 338-347

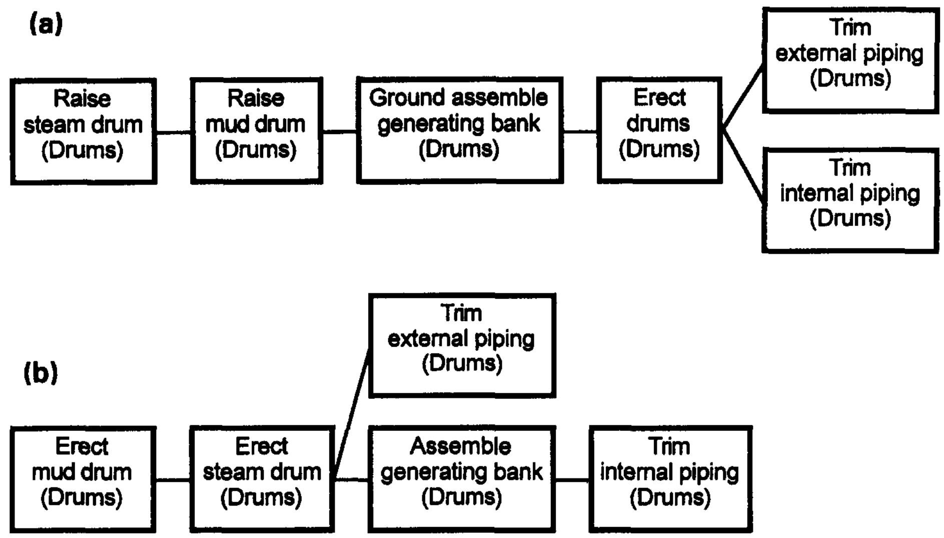

FIG. 3. Alternative Component Networks for 'Drums'

3(b) shows another possible network for Drums, but it takes longer to complete as pipefitters must work about 20 m [60 ft] in the air to assemble the generating bank and fit external piping to the steam drum. Yet following this procedure may be necessary if the length of the steam drum is greater than the bay width of the boiler cavity (i.e., the drum exceeds the dimensions assumed for it by the designers of the structure), so the drum must be lifted at an angle.

Product Network

A product network shows the sequence of construction of all components in the product. It interlinks their component networks. An ''interlink'' sequences activities of different component networks, specifically:

1. A ''network interlink'' is a sequential link between two component networks. It defaults to having all activities of the preceding network (the network from which the link emanates) precede with a FS link all activities of the succeeding network (the network to which the link points) but this can be customized by the user.
2. An ''activity interlink'' is a sequential link between two activities of different component networks.

Activities, designated in CasePlan by a *name-verb* and a *name-noun* (described later), e.g., ''Erect Drums'' can be consolidated if they are of the same type, i.e., if they use the same verb or verb phrase. For example, project specifications usually require conducting hydro tests for 'Drums', 'WW's, 'Superheater', and 'Economizer' when they are installed. Each component must be tested, but testing each individually and separately is meaningless as it does not provide any feedback on the operation of the system as a whole. Therefore, all activities ''Hydro test'' are consolidated into a single one, to be performed after all these components have been installed. Similarly, activities can be consolidated based on the type of their associated components.

Activity

An activity has attributes to express the following:

1. Identification, namely: *ID*, identification number; name-verb: (e.g., ''Erect'') or verb phrase (''Trim external piping for'') that describes the work to be performed; *name-noun*: usually the component on which work is to be done (e.g., 'Drums'); and *name*, which can be specified by the user directly or derived by CasePlan by combining the activity's attributes name-verb and name-noun (e.g., ''Trim external piping for Drums''). Derived names are more amenable to reuse than user-specified ones.
2. Sequencing, namely: *predecessors*, which lists the activities that immediately precede the activity, and *successors*, those that immediately succeed it.
3. Timing, namely scheduled times: the early and late start and finish time, and four floats [Harris (1978) gives definitions and formulas]; and actual times: *AS*: actual start; *AF*: actual finish; *RD*: remaining duration = duration − (current date − AS); *AD*: actual duration = AF − AS. Scheduled times are calculated using CPM based on the sequencing and timing constraint attributes (described next); they cannot be changed directly by the user. As a minimum, the project start time, all activity durations, and sequencing attributes need to be known for CasePlan to calculate a schedule. The actual times become available only after the activity has started and possibly finished.
4. Timing constraints, namely: *SNE* (start no earlier than), *FNE* (finish no earlier than), *SNL* (start no later than), *FNL* (finish no later than), and others (Dzeng 1995). They specify constraints that a contractor may impose on an activity for a reason unrelated to network logic per se, e.g., to reflect material delivery schedules or contract milestones.
5. Resource allocation, namely to refer to materials (e.g., piping), laborers or crews (e.g., four pipefitters and a foreman), or equipment (e.g., 30-ton PCSA class 12-105 crane) that are required to perform the activity. CasePlan groups those resources into a 'Construction Method', which is a class with attributes: *crew*, a list of individual trades or crews and their number specified in the form: [(crew-name crew-size) . . .]; *equipment*, a list of equipment and its quantity specified in the form: [(equipment-name number-of-pieces-of-equipment) . . .]; and *productivity*, the amount of work the specified crew and equipment accomplish per day. Upon reuse of an activity, the associated construction method is also reused, specifically, to estimate the activity duration based on the productivity and possibly based on attributes of the associated component.
6. *For-component* and *Use-condition* (described later).

J. Constr. Eng. Manage., 1997, 123(3): 338-347

Activity Link and Interlink between Component Networks

Each link or interlink specializes the Link class, which has the following attributes:

1. *Preceding-activity.*
2. *Succeeding-activity.*
3. *Type*: SS, SF, FS, or FF.
4. *Lead-time*, the amount of time between the link's head and tail.
5. *Strictness*, quantified as a number between 0 and 1, with a default value of 0.5. A ''strictness threshold'' (set at 0.8 but changeable by the user) determines whether a *link is* strict (strictness ≥ threshold) or not (strictness < *threshold*). CasePlan uses strictness to gauge whether it should reuse a link automatically (strict) or only after user confirmation (not strict). A link that is strict reflects a sequencing principle that applies to all projects, e.g., the sequential link between activities ''Install columns on floor 1'' and ''Install beams on floor 1'' is quite strict because one supports the other and is thus built first. Conversely, a link that is not strict reflects that it was introduced for a project-specific reason, e.g., a sequential link between ''Install east wall'' and ''Install north wall'' is not necessarily reusable because this ordering may be the result of the availability of a specific crane.
6. *Use-condition* (described later).

Relationships between Project and Schedule

CasePlan's user must annotate cases for archival by representing project knowledge and relating projects to their schedules. This is done using specific attributes and a language for describing relationships.

For-Component and Use-Condition Attributes

Each part of a product network must have been included for one of two reasons, which are specified using the following attributes:

1. *For-component*, specified on classes 'Component-network', 'Activity', or 'Link', describes the relationship between e.g., an activity (''Install drums'') or network and the component ('Drums') for which it represents a construction alternative.
2. *Use-condition*, specified on classes 'Component-network', 'Activity', 'Link', 'Product-network', or 'Construction Method', expresses in a value specification language (described later) the conditions under which an activity, link, network, or method can be reused. It relates the class to constraints specified in the product of the 'Case', 'Site', or 'Project-Specifications'; e.g., the network for a 'Front WW' cannot start until the network for 'Drums' is completed because the WW is connected to the drums. Erecting the WW earlier would make the connection work more difficult and the WW may get damaged in the process. The *Use-condition* specifies that the interlink between the two networks will be imposed only if 'Front WW' has an is-connected-to relation with 'Drums'.

Value Specification Language

A ''value specification'' (VS), stored as value of an attribute, expresses how that value is to be computed and on what project-specific data (instances of other attributes, activities, links, components, methods, or products) it depends. It is written in CasePlan's VS language that includes functions, global variables, and numerical values. Functions are those specific to CasePlan, attribute names to retrieve values from attributes, and all Common Lisp functions (*Allegro* 1994) except for constructive ones (i.e., functions that build data structures, such as defun or defclass). Scheduling knowledge is also expressed using CasePlan's VS language. CasePlan recalculates each VS each time the associated instance from an old case is reused in a new project or when the new project is updated. Dzeng (1995) provides examples.

A VS should be computable. It should avoid circular references and CasePlan must be able to evaluate it and yield a result of the intended type (string, logic value, number, etc.) for the associated attribute. A VS should also ease CasePlan's reuse of activities and links that need adaption from one project to another.

AUTOMATED SCHEDULING

Fig. 4 illustrates the steps CasePlan takes to automate the planning and scheduling tasks: a double-edge rectangle represents a step for which CasePlan reuses cases; a single-edge rectangle one for which it does not; a rounded rectangle represents information available before or after a step, and in it, underlined text represents information generated or changed by the preceding step.

Step 1: Determine Component Network

Given a new project, CasePlan first determines a network for each product component by reusing the corresponding network (i.e., the network used by a component of the same type) of the best available case. CasePlan determines the best match based on the component's similarity function. This yields a ''component similarity value'' (CSV) with value between 0 and 1, representing the similarity between a component in the case being considered for reuse and the corresponding component in the new product. A CSV is an average with user-assigned weights of the component's ''attribute similarity values'' (ASVs). An ASV is also a number between 0 and 1 but it represents the similarity of a component's attribute in a case to a corresponding attribute in the new product; e.g., ''To what extent is the Drums' total-mass in a Case similar to the Drums' total-mass in the new product?'' CasePlan uses several schemes to measure the similarity of attribute values and to normalize them according to the type of value that is being compared (an example follows, but for more detail see Dzeng 1995).

A corresponding network will be reused if its use-condition—if specified—is satisfied in the new product setting. If so, the derived attributes of its activities and links (e.g., activity-duration) will be reused, but their values are not calculated at this time (they may depend on the network as a whole). The associated construction method is also reused, though it can be changed later. After completing this step, information about component networks and construction methods and activity duration specifications are available (Fig. 4).

Suppose that CasePlan is to select a component network for 'Drums' of a new project (Project-1) based on two previous cases (Case-A and Case-B). In Table 1, the Attribute column lists the 'Drums' attributes with their user-assigned weights (shown in parentheses) for finding the best case (higher weights reflect greater importance). Columns ''Project-1/AV,'' ''Case-A/AV,'' and ''Case-B/AV'' show the attribute values for 'Drums' of Project-1, Case-A, and Case-B, respectively.

CasePlan computes the ASVs as indicated in columns ''Case-A/ASV'' and ''Case-B/ASV'' using the user-specified quantitative difference range for the attribute max-unit-mass:

[1.0 (0 4,000)], [0.8 (4,000 9,000)], [0.6 (9,000 20,000)],

[0.2 (20,000 30,000)], [0 (30,000 max.)]

J. Constr. Eng. Manage., 1997, 123(3): 338-347

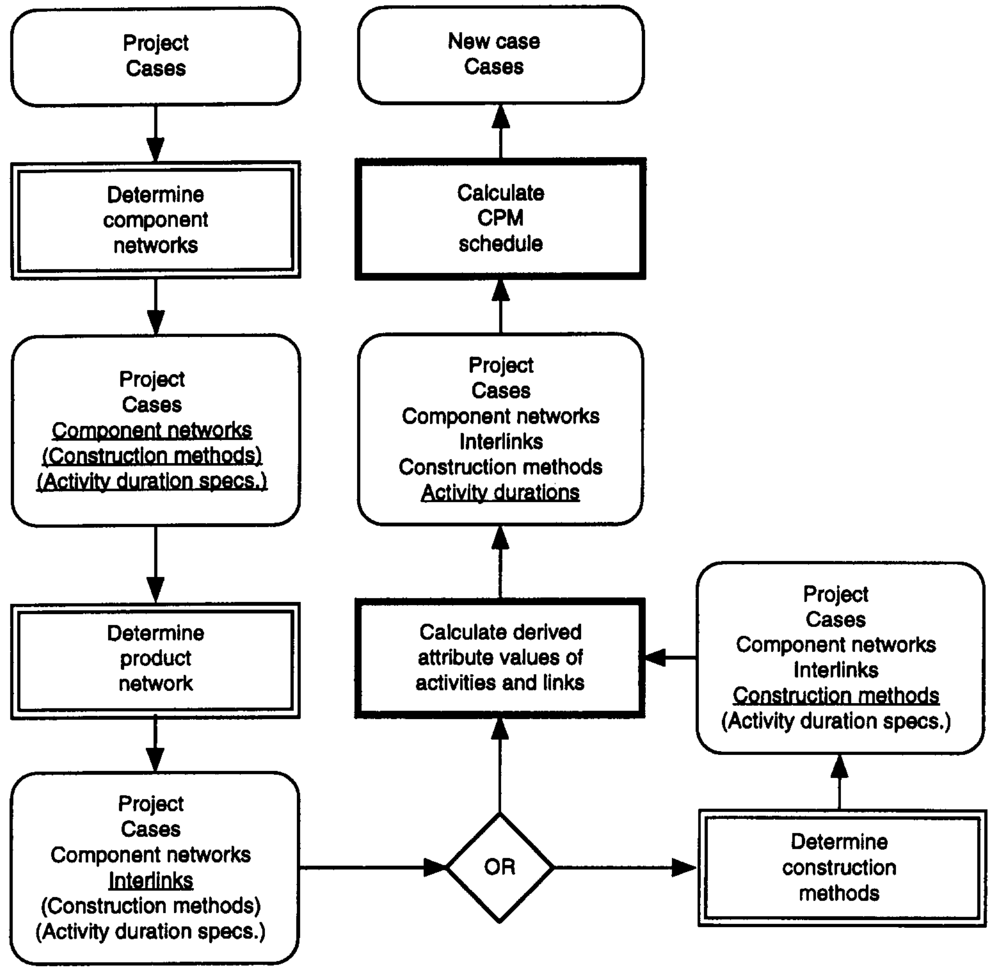

FIG. 4. CasePlan Subtasks

TABLE 1. Example Comparison or Project-1 and Two Cases

Attribute (1)	Project-1 AV[a] (2)	Case-A AV (3)	Case-A ASV[b] (4)	Case-B AV (5)	Case-B ASV (6)
total-mass (0.2)	125,000 kg	90,000 kg	0.6	148,000 kg	0.6
max-unit-mass (0.2)	34,000 kg	30,000 kg	0.8	45,000 kg	0.6
number-of-generating-tubes (0.4)	15,000	12,000	1.0	20,000	0.8
is-drum-length-greater-than-bay-width? (1.0)	false	false	1.0	true	0.0

[a]Attribute values.
[b]Attribute similarity values.

This specifies that the ASV is 1.0, 0.8, 0.6, 0.2, and 0, respectively, if the difference of max-unit-mass of the project and case being compared is between 0 (inclusive) and 4,000 kg, between 4,000 and 9,000 kg, between 9,000 and 20,000 kg, between 20,000 and 30,000 kg, and greater than 30,000 kg, respectively. Because the difference between the project and Case-A is (34,000 kg − 30,000 kg) = 4,000 kg, Case-A's ASV is 0.8; similarly, Case-B's ASV calculation yields 0.6. Other ASVs are determined using different schemes but these are not discussed here due to space limitations. The CSV of 'Drums' in each case is then calculated in the following two equations:

CSV (Case-A's 'Drums')

$$= \frac{0.6 \times 0.2 + 0.8 \times 0.2 + 1.0 \times 0.4 + 1.0 \times 1.0}{0.2 + 0.2 + 0.4 + 1.0} = 0.93 \quad (1)$$

CSV (Case-B's 'Drums')

$$= \frac{0.6 \times 0.2 + 0.6 \times 0.2 + 0.8 \times 0.4 + 0.0 \times 1.0}{0.2 + 0.2 + 0.4 + 1.0} = 0.31 \quad (2)$$

Because Case-A has the higher CSV (0.93), its 'Drums' network will be used for Project-1's 'Drums', provided that no use-condition has been violated in the new project setting. If all use-conditions are satisfied, CasePlan next examines each individual activity and link of the network, and removes those for which the use-condition has been violated. After each removal the user may have to relink disjoint activities. Networks for other components are determined similarly.

Step 2: Determine Product Network

Second, CasePlan determines a product network by interlinking component networks and consolidating activities. CasePlan chooses the best case for doing this based on the product similarity value (PSV), which is an average with user-assigned weights of the product's CSVs.

Suppose that the weights for determining a product network are 1.0, 0.8, 0.8, 0.6, and 0.6 for components 'Boiler', 'Drums', 'Economizer', 'Front WW', and 'Rear WW', respectively, and Case-A's CSVs are 1.0, 0.93, 0.8, 1.0, and 1.0 for those same components. Then

PSV (Case-A)

$$= \frac{1.0 \times 1.0 + 0.8 \times 0.93 + 0.8 \times 0.8 + 0.6 \times 1.0 + 0.6 \times 1.0}{1.0 + 0.8 + 0.8 + 0.6 + 0.6}$$

$$= 0.943 \quad (3)$$

J. Constr. Eng. Manage., 1997, 123(3): 338-347

As a result of this step, activities of different component networks are now interlinked.

Step 3: Determine Construction Method

CasePlan can now reselect methods by reusing cases based on component similarity functions with weights different from those used for determining component networks. Alternatively, it can do nothing and calculate activity durations based on their known methods. Not all activities must have methods specified; they need to only if their derived attributes refer to the methods in one way or another, e.g., when activity duration is a function of the method's productivity.

Step 4: Compute Schedule

CasePlan then calculates activity durations by evaluating the VS of each activity's duration attribute (e.g., a heuristic factor multiplied by the size of the activity's associated component, or a formula based on the productivity of the selected method). VSs for other derived attributes such as timing constraints are also evaluated at this time, e.g., an activity that started no earlier than May 20, 1991, being the completion date of the steel structure of a boiler house, will start no earlier than December 1, 1994, if it is reused in a new project where the completion of the structure is on December 1, 1994.

Finally, CasePlan computes the remainder of the schedule data using CPM. The schedule is then stored with the product instance as a new case.

DECISION-SUPPORT SCHEDULING

CasePlan can function as a stand-alone or as a decision-support tool. Because CasePlan relates a product (and its components) to its schedule (and its component networks), a user can browse and manually copy parts of existing schedules on a component-by-component basis after choosing cases that are similar to the project at hand. The user may modify component networks or construction methods from selected cases or create new ones from scratch. This allows for much more intelligent assembly than is possible using ''fragnets.''

IMPLEMENTATION

CasePlan has been implemented in object-oriented lisp, Allegro CL/PC 2.0 (*Allegro* 1994), that runs in the Microsoft Windows™ 3.1 environment. The model includes several graphical display windows, examples of which are shown by Dzeng (1995). The system includes a generic product model for boilers of fossil-fueled power plants with a capacity ranging from 30 to 90 MW. The product model is the result of a survey sent out to industry practitioners knowledgeable about boiler erection and subsequently discussed with them face-to-face or over the telephone. CasePlan's library currently includes seven realistic cases, including Grayling and Genesee, with design details and preproject schedules obtained from two boiler manufacturers.

CASEPLAN CAPABILITIES AND LIMITATIONS, AND FUTURE RESEARCH

CasePlan makes selected knowledge from a project's paper trail available to users by putting it on line. When successful projects representing the best company practices are stored as cases, CasePlan can help train schedulers new to those projects by letting them browse through historic data, so they may study why certain choices were made and experiment with scheduling alternatives. Bad examples could also be stored but should be annotated to warn the user when CasePlan retrieves them.

CasePlan's use of a single standard to define a product and its schedule eases the association between product components and schedule subnetworks. It also allows CasePlan to develop a new schedule by reusing subnetworks from different cases instead of the entire network from a single case. Integrating CasePlan with PlantSTEP and expanding the model describing construction methods deserve further research attention.

Clearly, annotating cases and describing the relationships between projects and schedules requires a considerable amount of work. VSs are not unique and documenting the rationale that was used to derive a certain schedule may be hard to come by. Also, schedules pieced together from various sources and ''refreshed'' with a new project's data may include contradictions that must be resolved manually before they can be evaluated by CasePlan. Despite the effort it requires from the user, CasePlan's VS language makes it possible to describe schedule constraints, which could not be done using other scheduling tools.

The requirement that scheduling knowledge be expressed in a computable form may prevent CasePlan from using some knowledge that human planners take into account. Nonetheless, it is worthwhile encoding what can be articulated as such, since quite a few pieces of knowledge are computable, and if there are many some may get overlooked by human planners.

CasePlan assumes that construction methods and activity sequences in existing cases are satisfactory, that they can be reused on an activity-by-activity basis, and that such reuse is adequate for estimating activity durations. At this time CasePlan has no way of accounting for the desirability of sharing or reusing resources in concurrent or consecutive activities. Performing resource leveling for the schedule as a whole is left to the user. Also, the question arose whether CasePlan should use bid, preproject, or as-built schedules, but we found pros and cons for each. Again, this issue is to be investigated further.

The aim of CasePlan was to produce schedules that are field executable. While CasePlan's architecture makes this possible, we have been only moderately successful at capturing scheduling knowledge considered by the boiler erector. Boiler erectors derive their competitive advantage in part from knowing their labor force well, and taking skill levels, motivation, and learning-curve effects into account. The erector we interviewed declined to give out his expertise on this. This avenue for research may be the most interesting one to pursue, as it would increase our knowledge of how construction schedules are broken down and used on site.

CasePlan follows an early-commitment approach, though it has most of the mechanisms needed to generate multiple alternatives. We have not investigated how to keep track of multiple alternatives in a computationally efficient way, that is, without having to spell out each individual alternative's details. An interesting research issue here is, ''Which cases should be stored as individuals and which ones as parametrized variations of a prototype case?''

CasePlan uses the best it can find in the cases available to it, so it will yield different results when supplied with different cases. Users can add new cases and modify existing ones as they see fit and expand the case base. This approach is fundamentally different from the one taken by existing AI-planners in which the planning knowledge is defined by the system designer instead of the individual user of the system.

CONCLUSIONS

This paper presented CasePlan, a tool that uses case-based reasoning to automate construction scheduling. It illustrates a methodology to relate construction schedules to individual projects and contractors' practice using product modeling, and to reuse that knowledge in scheduling new projects. Cases are

J. Constr. Eng. Manage., 1997, 123(3): 338-347

annotated with scheduling knowledge to reflect key factors, such as facility design considerations, site conditions, contractual requirements, contractor's practice, etc., that determined why a schedule ended up being one way and not another.

CasePlan is a graphical decision-support tool that allows schedulers to apply their own judgment in browsing and copying reusable parts from schedules of previous cases whose products are similar to the new one. This not only assists contractors in generating bid schedules when they are short of time, but it also provides them with a means of documenting and reapplying their company's best practices.

ACKNOWLEDGMENTS

We would like to acknowledge the kindest support from and participation in this research by industry practitioners at Asea Brown Boveri Combustion Engineering Services, Inc., Black & Veatch, Inc., The Christman Co., Genesee Power Station, Grayling Generating Station, Northern Boiler Mechanical Contractors, Inc., Townsend & Bottum, Inc., and Zurn Industries, Inc.

This research was funded by grant MSS-9215935 from the National Science Foundation (NSF), whose support we gratefully acknowledge. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

APPENDIX. REFERENCES

Allegro CL/PC 2.0, object oriented development system for Windows. (1994). Franz, Inc., Berkeley, Calif.

"Biomass-fired plant practices total resource management." (1993). *Power Engrg.*, (April), 33–36.

Cherneff, J., Logcher, R., and Sriram, D. (1991). "Integrating CAD with construction-schedule generation." *J. Comp. in Civ. Engrg.*, ASCE, 5(1), 64–84.

"Construction industry: OSHA safety and health standards." (1985). *29 CFR 1926/1910, OSHA 2207*, Occupational Safety and Health Administration, Washington, D.C.

Darwiche, A., Levitt, R. E., and Hayes-Roth, B. (1989). "OARPLAN: generating project plans by reasoning about objects, actions and resources." *AI EDAM*, 2(3), 169–181.

Desmond, C. (1995). "New Genesee Twp. power plant to be fueled by clean waste wood." *Mich. Contractor & Builder: The Weekly Jobs Book*, Contractor Publishing Co., Detroit, Mich., (Jan. 21), 6–8.

Dzeng, R. J. (1995). "CasePlan: a case-based planner and scheduler for construction using product modeling," PhD Dissertation, Civ. and Envir. Engrg. Dept., Univ. of Michigan, Ann Arbor, Mich.

Dzeng, R. J., and Tommelein, I. D. (1993). "Using product models to plan construction." *Proc., 5th Int. Conf. Comp. in Civ. and Build. Engrg.*, ASCE, New York, N.Y., 1778–1785.

Dzeng, R. J., and Tommelein, I. D. (1994). "Case storage of planning knowledge for power plant construction." *Proc., 1st Congr. Comp. in Civ. Engrg.*, ASCE, New York, N.Y., 293–300.

Echeverry, D., Ibbs, W., and Simon, K. (1991). "Sequencing knowledge for construction scheduling." *J. Constr. Engrg. and Mgmt.*, ASCE, 117(1), 118–130.

Gray, C. (1986). "Intelligent construction and cost analysis." *Constr. mgmt. and economics*, London, U.K., 4, 135–150.

"Guidelines for specifying integrated computer-aided engineering applications for electric power plants." (1987). *EPRI NP-5159M*, Electric Power Res. Inst., Palo Alto, Calif.

Harris, R. B. (1978). *Precedence and arrow networking techniques in construction.* John Wiley & Sons, New York, N.Y.

ISO Standard 10303-1. (1994a). "Product data representation and exchange. Part 1: overview and fundamental principles." Int. Standards Org.

ISO Standard 10303 AP. (1994b). "Project proposal for plant spatial configuration AP." Int. Standards Org.

Kartam, N. A., and Levitt, R. E. (1990). "Intelligent planning of construction projects." *J. Comp. in Civ. Engrg.*, ASCE, 4(2), 155–176.

Kolodner, J. (1993). *Case-based reasoning.* Morgan Kaufmann, San Mateo, Calif.

Miresco, E. (1992). "Expert system for scheduling building projects." *Proc., CIB 92 Vol. 1*, Nat. Res. Council Canada, 258–259.

Morad, A. A., and Beliveau, Y. J. (1991). "Knowledge-based planning system." *J. Constr. Engrg. and Mgmt.*, ASCE, 117(1), 1–12.

Navinchandra, D., Sriram, D., and Logcher, R. (1988). "GHOST: project network generator." *J. Comp. in Civ. Engrg.*, ASCE, 2(3), 239–254.

Primavera project planner for Windows 1.0. (1993). Primavera Systems Inc., Bala Cynwyd, Pa.

Special issue on data and product modeling. (1996). *J. Comp. in Civ. Engrg.*, ASCE, 10(3).

Thabet, W. Y., and Beliveau, Y. J. (1993). "A model to quantify work space availability for space constrained scheduling within a CAD environment." *Proc., 5th Int. Conf. Comp. Civ. and Build. Engrg.*, ASCE, New York, N.Y., 110–116.

Tommelein, I. D., Levitt, R. E., Hayes-Roth, B., and Confrey, T. (1991). "SightPlan experiments: alternate strategies for site layout design." *J. Comp. in Civ. Engrg.*, ASCE, 5(1), 42–63.

Zozaya-Gorostiza, C., Hendrickson, C., and Rehak, D. R. (1989). *Knowledge-based process planning for construction and manufacturing.* Academic Press, Boston, Mass.

J. Constr. Eng. Manage., 1997, 123(3): 338-347