



ELSEVIER

Automation in Construction 13 (2004) 341–360

AUTOMATION IN
CONSTRUCTION

www.elsevier.com/locate/autcon

Product modeling to support case-based construction planning and scheduling

Ren-Jye Dzung^{a,*}, Iris D. Tommelein^b

^aDepartment of Civil Engineering, National Chiao-Tung University, 1001 Ta-Hsueh Road, Hsin-chu, Taiwan

^bCivil and Environmental Engineering Department, University of California, Berkeley, CA 94720-1712, USA

Accepted 28 October 2003

Abstract

Many human schedulers create schedules by reusing past similar schedules. The retrieval and reuse of similar schedules are subjective and experience-based. This paper explores different notions of similarity required when performing different scheduling tasks. It describes the CasePlan system that helps schedulers retrieve and reuse parts of existing schedules based on a generic product model, and apply case-based reasoning to generate new schedules. The validation experiment demonstrated CasePlan's accuracy in determining individual subnetworks and activity durations, but weak performance in determining interlinks between subnetworks, which highly depends upon the availability of pertinent cases and the level of detail of project information.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Case-based reasoning; Scheduling; Product modeling; Power plant construction; Similarity assessment

1. Introduction

Many contractors, especially those who repeatedly build the same kind of facilities, have been reusing fragments of past project schedules to construct new schedules. For them, constructing a new schedule is a matter of finding suitable network fragments from previous schedules and adapting these for reuse.

Traditional scheduling tools provide only limited features to facilitate such reuse. For example, Primavera Project Planner (P3) [26] allows the user to define and reuse the fragnet, a subnetwork composed

of activities with fixed names, duration, and sequential relationships between each other. A limitation of P3 is that the fragnet has no relationships with the data pertaining to the project it was initially created for, and can be retrieved only by its name. Modification of the fragnet may be required before it is used. P3 provides features that facilitate the addition of a prefix or a suffix to all activity numbers, or specific text or a number to all activity names. However, the added values have to be specified manually even though many of them could have been figured out by a computer had the relationships between the fragnet and the project data been available.

We have developed a generic product model that may be used in power plant construction to describe a boiler erection project by specifying the boiler's components to be installed and their installation

* Corresponding author. Tel.: +886-35733529; fax: +886-35716257.

E-mail address: rjdzeng@mail.nctu.edu.tw (R.-J. Dzung).

procedures. Existing projects and schedules can be described using this model and reused to generate the schedule for a new project.

A computer system, called CasePlan, was developed to facilitate schedule reuse. CasePlan supports multiple similarity metrics so that different scheduling tasks (e.g., determining networks and activity durations) may use different weight distributions in determining the similarity. Similarity can also be computed at the activity, component, and project levels, and on a component-pair basis. CasePlan's case library currently consists of boiler erection projects for fossil-fueled power plants that were built in and around Michigan.

This paper presents the generic boiler product model and illustrates its use by the similarity assessment and the retrieval mechanism of CasePlan. The paper also presents a survey of professional schedulers that was conducted to obtain their similarity assessment of boiler erection projects in order to evaluate CasePlan's performance. Finally, CasePlan's performance was assessed.

2. Related work

2.1. Automated planning systems

Construction planning consists of defining activities and precedence relationships. Scheduling involves determining resources and activity durations, then 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 tasks of plan generation. Textbooks (e.g., [23]) provide guidelines for breaking down a project into activities, sequencing them, and assigning resources to them. OSHA spells out safety rules [24] that can be used in these efforts. Gray [13] and Echeverry et al. [7] articulate constraints that pertain to schedule generation and develop rules by which to sequence activities. While the task model for planning can be generic, the domain knowledge is specific to each type of project and further customized for each individual contractor.

Plan generation has been automated using Artificial-Intelligence (AI) programming techniques. Dzung and Tommelein reviewed such AI-based planning systems (AI-planners, in short) in detail [5]. Most of

these planners use a least-commitment approach in generating their plan(s). The resulting least-commitment plans subsequently need substantial augmentation with resource data before they can be used in the field. In contrast, human schedulers typically follow an early-commitment approach and investigate only a few alternatives but in greater detail. In developing CasePlan, we adopted the early commitment approach because knowledge to generate field-executable schedules is difficult to spell out exhaustively due to variations in project and organizational conditions. The reviewed AI-planners have no means of recognizing that they planned similar projects previously or that plans can be reused. Our research 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 [21], but CasePlan has appeared to be the result of a much more comprehensive research effort than Miresco describes.

CBR (Case-Based Reasoning) has played an important role in the development of artificial intelligence and expert systems [17] addressing tasks such as planning, design, explanation, and diagnosis. In the architectural/engineering/construction (AEC) domain, CBR was applied to solving design problems such as building design [1,19,20], bidding [3], and contractor prequalification [22]. Various techniques may be applied to assigning importance weights, measuring similarity, and adapting cases. Most CBR systems use similar concepts and approaches, but vary in their combination and modification of techniques thereof to suit their domain of application. Like other CBR systems, CasePlan uses CBR as a means to reuse knowledge specific to individual projects (which is lacking from existing construction planners), but CasePlan is unique in that it tackles planning problems in construction and uses product models as the basis for case organization.

2.2. Product modeling and standards

The technical and economical complexity of AEC project has increased over the past decades, and is likely to continue to increase in the next several decades [11]. The delivery of most AEC projects nowadays requires the cooperation of multiple participants with different specialization [30]. With wide use of computer applications in the AEC industry,

exchanging electronic data is essential to ensure proper delivery of the project. Researchers and practitioners alike are investigating how data, knowledge, and goals can be shared among all project participants throughout the engineering lifecycle.

Many AEC firms use work breakdown structures (WBS) as the foundation for sharing project data among interdisciplinary engineering processes. Researchers have also shared project models. For example, COKE links a constructability expert system with a project model and allows design and construction professionals to share knowledge [9]. Others have established project modeling structures to be shared among interdisciplinary organizations. For example, EPRI developed the Plant Information Network (PIN), which is a comprehensive power-plant model as a guide to specifying integrated computer-aided applications [8].

The logical extension of shared product models is to develop a modeling standard—a neutral representation of digital product data that can be exchanged among heterogeneous systems. The most significant product-model standardization effort to date has resulted in the International Standards Organization's (ISO) standard 10303 STEP, developed by ISO TC184/SC4 task force [15]. The General AEC Reference Model (GARM) described a building project model as a progression of design procedures [12]. The process involves the definition of the product as a functional unit, associated with functional requirements, followed by the selection of an appropriate technical solution from a set of options. This concept has been used in the early development of STEP. STEP has several layers consisting of a series of definition parts [16] such as the Description Methods (Parts 11–13) and the Application Protocol (AP, Parts 201–233).

Several ongoing standardization efforts follow STEP [16]. ISO 18876 (“IIDEAS”) emphasizes sharing and integration of data from multiple, heterogeneous sources, and the interoperability between applications and organizations that implement different standards. ISO 15531 Manufacturing Management Data (MANDATE) defines methods to standardize management activity data, which express information exchanged within an industrial manufacturing company. ISO 13584 is a series of standards for the computer-sensible representation and exchange of part

library data. The objective is to provide a mechanism capable of transferring parts library data independent of any application.

Like product data, process data is also used throughout the life cycle of a product. ISO 15926 Integration of Life-cycle Data for Oil and Gas Production Facilities emphasizes such sharing and integration of data associated with engineering, construction, and operation of oil and gas production facilities. ISO 18629 Process Specification Language (PSL) defines a neutral representation (a language) that may be used for sharing process data. For example, Cheng et al. [2] show how the PSL language maps to XML.

In the process industries, major US companies formed PlantSTEP, in 1994 to work with STEP. PlantSTEP vendor members represent more than 90% of the plant design systems market [25]. International efforts that include consortia of PISTEP (UK) [31], PlantSTEP (US), and P-CALS (Japan) started in 1996 to work under the umbrella of PIEBASE (Process Industries Executive for achieving Business Advantage using Standards for data Exchange) [4]. PIEBASE has developed an activity model, documented using the IDEF0 modeling technique, through collaboration among national and international consortia.

Another effort, albeit one that is indirectly related to process plants, is the standardization of Industry Foundation Classes (IFCs), undertaken by Industry Alliance for Interoperability [14]. IFCs define data structures for the exchange of intelligent AEC-related objects (e.g., objects for architecture such as walls and windows, HVAC, and construction management) among CAD systems. While the IFCs development is driven by leading CAD vendors and therefore is more technology-oriented, STEP is more user-driven and results-oriented [29]. However, because of the significant progress of the IFCs development in recent years, many researchers in AEC have shifted their focus from STEP to IFCs [6].

STEP must cover geometry, topology, tolerances, relationships, attributes, assemblies, configuration and more in order for the product data to be shared through a product's entire life cycle. The basic, generic parts (e.g., testing procedures, file formats and programming interfaces) are complete and published, while others (e.g., the industry-specific APs) are still under development.

The case representation proposed in this research consists of descriptions of both the product and its construction processes. The corresponding data exchange standards thus required for such a case to be sharable in the industry. At the time of this research, standards were not complete or detailed enough for us to build our application. Among the APs specific to the process plant industry, AP 212 (Electro-technical Design and Installation) and AP 227 (Plant Spatial Configuration) have since March, 2001 been formally published; AP 221 (Functional Data and Schematic Representation for Process Plant) is still being developed; and AP 231 (Process Engineering Data) has been withdrawn. PIEBASE's activity model focuses only on high-level business activities, and is too abstract for this research. Thus, our model conforms to STEP only at the abstract level, namely the entity representation of General AEC Reference Model (GARM). Nevertheless, the case representation introduced in this paper applies equally to such data models.

3. Components of power plant boilers

An integral part of product-model development is to agree upon English terms referring to modeling elements. A typical boiler comprises components to support the circulation systems of water/steam, fuel, and air as shown in Fig. 1.

Each component may require a sequence of activities to assemble and erect. For example, the upper subnetwork in Fig. 2 conceptually describes the construction procedure of the drum as follows. First, the steam drum (upper drum) is raised to a height approximately equal to the length of the generating bank. The mud drum (lower drum) is raised just above the ground. With the drums temporarily secured, steamfitters assemble the generating bank, which connects the drums, near the ground. When the bank assembly is completed, the entire drum set (including two drums and the bank) is further raised with a single lift. After the drums have been erected, steamfitters finish the internal and external piping for the drums.

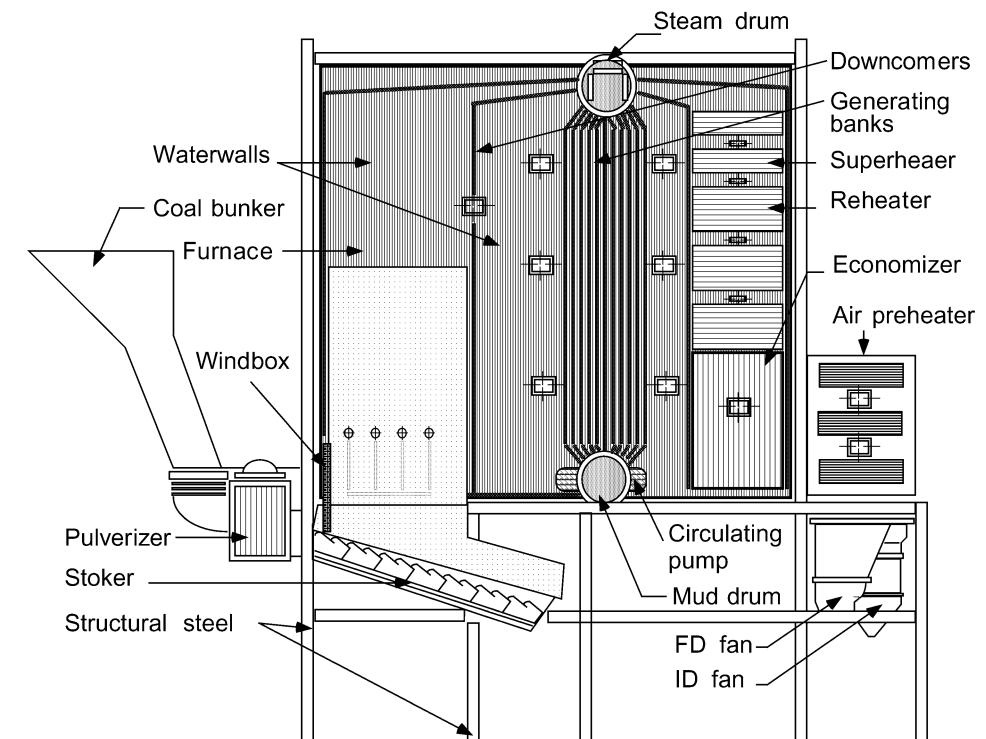


Fig. 1. General arrangement of a boiler.

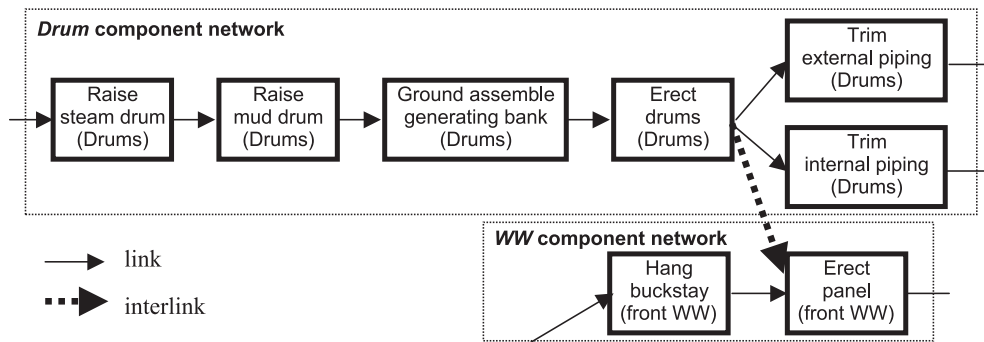


Fig. 2. Drum and front *Waterwall* component networks with their interlink.

The erection sequence of components depends on the design of the boiler and configuration of components. Most modern boilers are top-supported, allowing their components to move downward due to thermal expansion during operation. The erection of a top-supported boiler usually starts after the steel erector has installed the roof steel for the boiler house. Coordination among the boiler erector, steel erector, and cladding subcontractor is important in that many boiler components are large, and some steel and enclosure sections may need to be held out until these components are installed. Readers unfamiliar with power plant components may refer to [18,28] for more information.

4. Caseplan architecture and modeling knowledge

Each construction project schedule includes the knowledge about what needs to be built and how they are built. Thus, the project can be viewed as a product that comprises physical and abstract components to be built. The network of the schedule can also be divided into subnetworks that can be associated with the components. For example, a typical boiler erection project needs to assemble and erect components such as steam drum, superheater, waterwalls, and stokers. A subnetwork may conceptually describe the construction procedure of the drum as follows. First, the steam drum (upper drum) is raised to a height approximately equal to the length of the generating bank. The mud drum (lower drum) is raised just above the ground. With the drums temporarily secured, steamfitters assemble the generating

bank, which connects the drums, near the ground. When the bank assembly is completed, the entire drum set (including two drums and the bank) is further raised with a single lift. After the drums have been erected, steamfitters finish the internal and external piping for the drums.

The CasePlan architecture builds on the premise that a generic product model, comprising a hierarchy of classes which represent abstract and physical components with *has-a-component* links between them, can describe the type of product of interest (here, power plant boilers) and be associated with the construction processes used to build the product. CasePlan's user creates objects from predefined classes to model a construction project. A class determines the attributes and mechanisms of objects (called instances) that are created based on the class. A class that is inherited from its superclass has the attributes and mechanisms of the superclass, and may also have its own attributes and mechanisms. CasePlan predefines the following classes: *PDU* (representing the Product Definition Unit in STEP's GARM), *Product*, *Schedule*, *Case*, *Construction-PDU*, *Project-Spec.*, *Site*, *Boiler*, *Economizer*, etc. *Product*, *Schedule*, *Case*, and *Construction-PDU* are the subclasses inherited from *PDU*. *Project-Spec.*, *Site*, *Boiler*, and classes representing boiler's components such as *Drums* are the subclasses of *Construction-PDU*.

A generic *Case* for a boiler comprises a *Product* and a *Schedule*. The *Product* in turn comprises physical components such as *Boiler*, *Economizer*, *Drums*, *Waterwall (WW)*, and *Site*, but also an abstract component *Project-Specs.* (Fig. 3). Each component can be marked by attribute *construction-component-p* as a

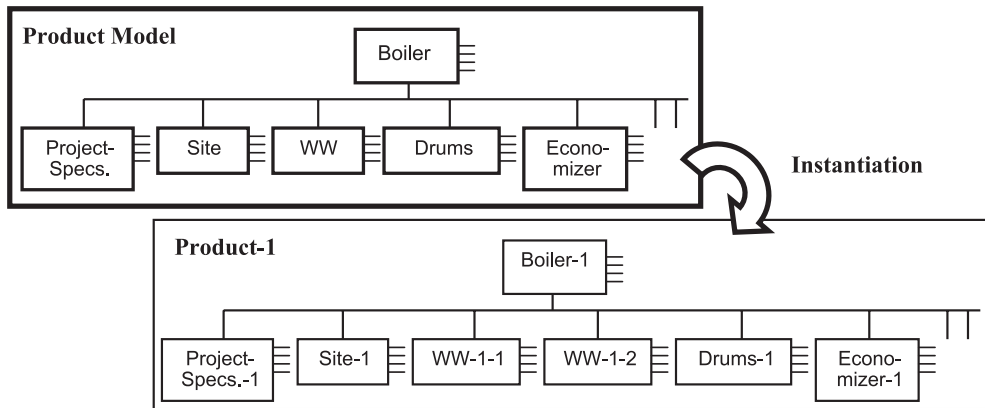


Fig. 3. Product model and a product of a specific project.

construction component or a non-construction component depending on whether it may require a construction network. CasePlan attempts to create activities only for the construction components.

CasePlan's user creates a *Case* for a new project by instantiating the generic *Case*, and its *Product* may comprise zero, one, or several instances of each component (e.g., *WW* 1-1 and *WW* 1-2 both instantiate *WW*). The values for the attributes of each component instance are also specified at this time. Each component may have topological relationships with other components, including nonconstruction component. *Construction-PDU* comprises an attribute, named *relationships*, which is a list of relationship types (i.e., *is-connected-to*, *is-embedded-in*, *is-on-top-of*, and *is-supported-by*) and their importance values (defined later). The user can edit or augment this list.

CasePlan constructs a schedule by determining a network of activities (termed a component network) that describes the construction process for each component, and then combining them into a single large network (termed a product network). A precedence 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). A link may also have a lead time. Unless marked otherwise, a single line represents the default FS link with zero lead time. Links that have no arrow imply that they go from left to right.

Fig. 4 (top) shows a component network for a *Stoker* that removes ashes with overfeed cooling design through movable traveling grates. Stokers that

are designed to remove ashes with a different mechanism require a different component network. For example, Fig. 4 (bottom) shows an alternative network for a *Stoker* that removes ashes through hydrogrates. The network requires an additional activity, and thus more time, to install the cooling water tubes and connect them through the modules and the boiler circulation system.

The attributes of the components are described using fixed values; unless the user modifies them, these values will not change in the course of CasePlan's 'reasoning.' However, most attributes of the *Schedule's* objects (i.e., *Product-Network*, *Component-Network*, *Activity*, and *Link*) can be described by value specifications using CasePlan's specification language, an extended version of CLOS (Common Lisp Object System) [10]. Value specifications

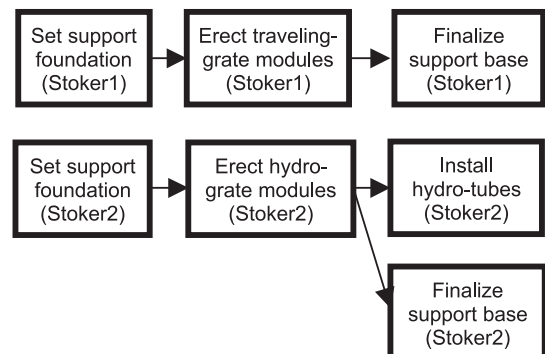


Fig. 4. *Stoker* network examples 1 (top) and 2 (bottom).

will be evaluated to determine their values at run time and when their associated objects are reused. They allow the user to describe how the values are derived and what project-specific data they depend on. This functionality facilitates CasePlan's adaptation of reused objects to the new project setting. The following is a value specification example for the attribute *duration* of *Activity* "Assemble generating tubes for drums".

```
Activity-1:
name: "Assemble generating tubes for drums"
duration: /(number-of-generating-tubes *COMPONENT*)
          (productivity *METHOD*)
```

The specification states that the activity duration should be derived by dividing the value of the *number-of-generating-tubes* of the activity's associated component by the value of the *productivity* of the activity's construction method.

A product network describes the construction sequence of components by defining the relationships of component networks. For distinction, "link" is used to describe the relationship between two activities within a component network, while "interlink" is used to describe the relationship between two activities of different component networks. Thus, a product network is a set of interlinks that sequence all component networks of a project. With information about the product network and the associated component networks, an entire project schedule can be built.

For example, Fig. 2 shows a part of a schedule that includes two component networks for drum and waterwall (WW). Each link (e.g., the one connecting activity "Raise steam drum" and "Raise mud drum"), represented by a regular arrow, sequences activities of a component network. The interlink, represented by a dashed bold arrow (e.g., the one connecting activity "Erect drums" and "Erect panel"), sequences activities of different component networks.

An interlink defaults to having all the last activities of the preceding network precede with a FS link all the start activities of the succeeding network, but this can be customized by the user. Both the interlink and the link are represented by the class *Link*. Note that CasePlan allows redundant links (e.g., if A precedes B

and B precedes C, then expressing that A precedes C is redundant) to remain in a network because redundancy may facilitate later schedule reuse. Many CPM algorithms eliminate redundant links in order to simplify schedule computations.

A product network also defines consolidation—specification, which is a list of criteria that are used to consolidate activities that have common characteristics. Activities may be consolidated if the verb phrases of their names are the same. For example, boiler projects usually require conducting hydro-tests for *Drums*, *WWs*, *Superheater*, and *Economizer* when they are installed. Each component must be tested, but testing each individually and independently is impractical and does not provide any feedback on the operation of the system as a whole. Therefore, the activities "Hydro test" may be consolidated into a single activity.

Activities can also be consolidated if they are associated with the same type of components. For example, a typical configuration includes four side panels plus a roof panel, but sometimes a boiler with a particularly long configuration may use two panels on two sides, resulting in a total of six side panels plus a roof panel. Thus, the boiler comprises seven *WWs* (e.g., front *WW*, rear *WW*, roof *WW*, and four panels of side *WWs*). All, except for the roof *WW*, are of the type "Side-WW" and require hanging buckstay steel (vertical steel that provides lateral support to allow for heat expansion of the waterwalls) before their erection. One may choose to represent the six "Hang buckstays" activities as a single one if they are executed consecutively. The following is an example of such a value specification.

```
Product-Network-1:
consolidation-specification:
```

```
((activity-type "Hydro-test")
```

```
(and (activity-type "Hang buckstays") (component-type "Side-WW"))))
```

Execution of activities requires resources, including materials (e.g., piping), crews (e.g., four pipe fitters and a foreman), or equipment (e.g., 30-ton PCSA class 12–105 crane). CasePlan groups these resources into a construction method, which has

attributes: *crew*, *equipment*, *productivity*, *for-activity* (the activity to which the method is applied), and *use-condition*. The *use-condition* specifies the condition under which the object (here, the method) can be reused. *Component-Network*, *Activity*, and *Link* also have a *use-condition* attribute. In addition, they have an attribute called *for-component*, which specifies their associated component.

For example, the activities “Hang buckstays” and “Erect panel” are required to install a *WW*. A *WW* that is connected to the drums requires an additional activity, namely “Install feeders between drums and panel”. Thus, one may specify the *use-condition* for the additional activity so that the activity is reused only when its component (*WW*) has an *is-connected-to* relationship with *Drums*.

5. Task model for case-based planning and scheduling

CasePlan’s model to perform case-based planning and scheduling comprises five tasks (Fig. 5): (1) determine component networks, (2) determine a product network, (3) determine construction methods, (4) adapt activities and links, (5) calculate the CPM schedule. While tasks (4) and (5), shown by single-edge rectangles, are accomplished based on predefined algorithms, tasks (1), (2), and (3), shown by double-edge rectangles, are based on CBR. Information passed between tasks is represented by rounded boxes. Underlined text represents the new information or the information that may have changed after completing the preceding task. Texts in parentheses represent information that may be modified in later tasks.

First, given a new project, CasePlan determines a component network for each construction component of the product by reusing the corresponding networks (i.e., networks used by components of the same type) of similar cases. When a network is reused, its activities’ construction methods and duration specifications are also reused by default. Information about component networks, and methods and duration specifications are available after completing this task. For a network to be reused for a new component, the network’s *for-component* should be of the same type, and its *use-condition*, if specified, should be satisfied

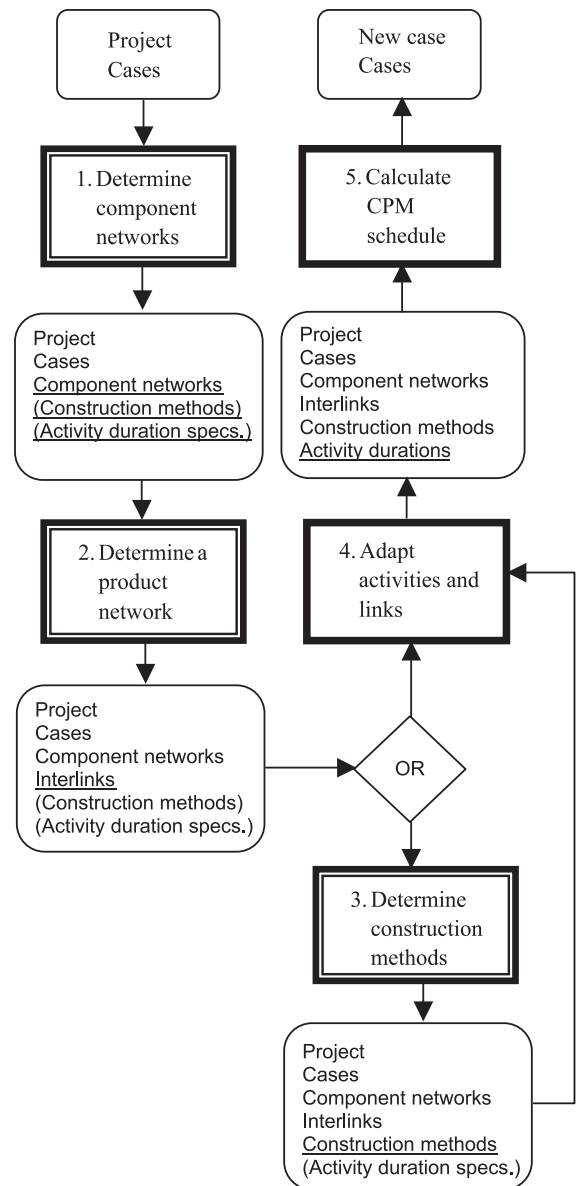


Fig. 5. CasePlan’s task model.

in the new project setting. CasePlan adapts the reused activities and links by evaluating their attributes based on the data of the new product.

Second, CasePlan determines a product network, which interlinks the component networks, and consolidates activities that require consolidation. These may be the default interlinks or customized interlinks depending on the interlink specifications of the re-

trieved case. Customized interlinks are applicable only when the corresponding pair of activities exists in the new plan. The interlinks can be determined based on the reuse of a single case or multiple cases. In the single-case situation, the case that has the highest product similarity value (described later) is chosen. In the multiple-cases situation, the case that has the highest average similarity value for the pair of components involved is chosen for each set of interlinks connecting any two component networks.

Third, CasePlan can calculate activity durations if the user chooses to use the methods of reused activities. Not all activities are required to have methods specified; they need to only if their duration specifications refer to the method (e.g., see example *Activity-1*). Alternatively, CasePlan can re-select methods from a different set of cases. A best case is chosen for each component to determine the methods of all activities of its component network.

Fourth, CasePlan calculates activity durations based on the value specification of the *duration* attributes. For example, the specification could be a heuristic factor multiplied by the size of the activity's associated component. It could also be a formula that is based on the productivity of the selected method for the activity.

Last, CasePlan calculates the early and late schedules, and the activities' total and free floats.

CasePlan maintains a similarity metric for each of the three CBR tasks because different tasks may perceive the similarity differently. In practice, experts assign similar weights for most components and attributes regardless of the task at hand. However, some may be different. For example, *Economizer* may have attributes such as *type*, *size*, and *weight*. The attribute *type* is more important than *size* and *weight* for determining an *Economizer* network because different *types* of *Economizers* require different erection sequences. In contrast, the attributes *size* and *weight* are more important than *type* for determining construction methods because they determine the hoisting equipment to be used.

The similarity metric is used to rank the cases and determine the best one for each CBR task. The best case is always reused for the task unless the specification of the *use-condition* attribute is not satisfied. If that occurs, CasePlan tries to reuse the next best case available for the unsolved part until the reuse succeeds

or no other case exceeding the user-defined similarity threshold is available.

More than one case may be reused during the execution of any CBR task. CasePlan maintains a grouping scheme for each CBR task. The grouping scheme allows the user to group components so that only a single case is reused for that group of components. The user may determine that the selection of interlinks, networks, or methods of activities of certain components should be based on the reuse of a single case so that the selected items can be consistent and compatible. For example, a boiler typically consists of four sides and a roof of waterwall panels. There is no reason to use different variations of crew-and-equipment (or method) to erect the panels except for the roof panels, which typically require an altogether different method. Thus, one may group the four sides of panels so that only one type of method and component network is reused, of course, assuming that only one type was used in the reused case.

6. Similarity assessment for products and components

Similarity can be assessed between any two products, components, or attributes of the same type. Two products or components are considered of the same type if they are instantiated from the same class. Two attributes are considered of the same type if they have the same name and belong to the same type of component. CasePlan represents this similarity by a value between 0 and 1, termed similarity value (S). An entity (i.e., product, component, or attribute) with a lower similarity value is less similar to the corresponding entity of the new project compared to that with a higher similarity value.

In the following discussion, let $S_{t,i}$ = product similarity value (PSV) of product i for task t ; $S_{t,i,j}$ = component similarity value (CSV) of component j of product i for task t ; $S_{t,i,j,k}$ = attribute similarity value (ASV) of attribute k of component j of product i for task t ; $W_{t,j}$ = component importance value (CIV) of component j for task t ; $W_{t,j,k}$ = attribute importance value (AIV) of attribute k of component j for task t .

The component similarity value ($S_{t,i,j}$) of a component j of an existing product i to a component j of a

new product for task t is calculated by the component similarity function:

$$S_{t,i,j} = \frac{\sum_{k=1}^n W_{t,j,k} \times S_{t,i,j,k}}{\sum_{k=1}^n W_{t,j,k}}, \quad (1)$$

where n is the number of the attributes of the component j that are evaluated.

The product similarity value ($S_{t,i}$) of a product i in a case to a product i in a new project for subtask t is calculated by the product similarity function:

$$S_{t,i} = \frac{\sum_{j=1}^m W_{t,j} \times S_{t,i,j}}{\sum_{j=1}^m W_{t,j}} = \frac{\sum_{j=1}^m W_{t,j} \times \frac{\sum_{k=1}^n W_{t,j,k} \times S_{t,i,j,k}}{\sum_{k=1}^n W_{t,j,k}}}{\sum_{j=1}^m W_{t,j}}, \quad (2)$$

Product, *component*, and *method* are used in specifying task t to represent the three CBR tasks of determining a product network, determining component networks, and determining construction methods, respectively. For example, $S_{\text{component}, \text{Stoker-1}, \text{type}}$ represents the similarity value of the *type* attribute of *Stoker-1* to the new *Stodfker's type* for determining the component network.

A component has only one importance value for determining a product network. An attribute has three importance values, indicated by the numbers in parentheses in the order of $W_{\text{product}, \text{component}, \text{method}}$. A component's similarity function may include other components if the network or method selection should consider the similarity of those components. Different component importance values can also be assigned in the function. This assignment is independent of the assignment in the product similarity function. For example, one may want to include the *Drums* in the *Waterwall's* similarity function because the erection procedure for a waterwall depends on its relationship with *Drums*. Waterwalls that are not directly connected to the drums may be erected before or after lifting the drums, provided that the boiler cavity is

large enough. However, waterwalls that are connected to the drums can be erected only after the drums have been lifted and secured, and require an additional activity to connect the waterwall tubes to the drums. The user may also include other components in the default similarity function for determining methods.

7. Similarity assessment for attribute values

The attribute similarity value factored into the similarity functions are determined based on the similarity assessment schemes set up by CasePlan's user. CasePlan recognizes five types of attribute values: string, logical value, keyword, number, and relationship. Table 1 shows the attribute values of three projects based on a simplified product model to illustrate similarity assessment. The *Product* here consists of only three components, namely *Boiler*, *Stoker*, and *Economizer*, shown under column 1 with an importance value in the parentheses. Column 2 lists attributes for each component and their importance values in the order of $W_{\text{product}, \text{component}, \text{method}}$. *Product-New* (column 3) is the new project to be planned, and *Product-A* and *Product-B* (columns 4 and 8) are the only two existing cases.

7.1. Strings and logical values

Strings are assigned attribute similarity values of either 1 or 0. Strings that match each other perfectly have a similarity value of 1. Otherwise, their similarity value is 0. For example, the value for the attribute *project-name* of *Boiler* is of the string type. Thus, in Table 1, *Product-A*, whose *project-name* is the same as *Product-New's*, has a similarity value of 1 for *project-name*. *Product-B's* similarity value is 0. The same similarity assessment is applied to the logic values "true" and "false".

7.2. Keywords

Keywords are sets of strings that are predefined by the user for specific attributes in the generic product model. Common practice is to use an abstraction hierarchy to measure the similarity between two keywords. After all legitimate keywords have been predefined in such a hierarchy, the most specific common

Table 1
Components and attributes of exemplar projects

Component ($W_{i,j}$) (1)	Attribute ($W_{i,j,k}$) (2)	Product-New (3)	Product-A (4)	S1 (5)	S2 (6)	S3 (7)	Product-B (8)	S4 (9)	S5 (10)	S6 (11)
<i>Boiler</i> (0.9)	<i>project-name</i> (0, 0, 0)	Detroit Power Station	Detroit Power Station	1			Pittsburgh Power Station	0		
	<i>manufacturer</i> (0.5, 0.1, 0.1)	Company A	Company B	1			Company A	0.6		
	<i>operating-pressure</i> (0.8, 0.8, 0.6)	17 MPa	22.5 MPa	0	0.42		11 MPa	1	0.88	
	<i>steam-output</i> (0.8, 0.8, 0.6)	900 ton/hr	2,000 ton/hr	0.4		0.54	700 ton/hr	0.8		0.72
<i>Stoker</i> (0.4)	<i>type</i> (0.6, 0.6, 0.6)	chain-grate	Vibrating-grate	0.8	0.8		spreader-grate	0.4	0.4	
	<i>Economizer</i> (0.6)	<i>type</i> (0.2, 0.7, 0.2)	bare-tube	Extended-surface	0.2			bare-tube	1.0	
<i>is-on-top-of</i> (0.8, 0.2, 0.2)		Dust-collector-N Air-heater-N	Air-heater-A Dust-collector-A Stoker-A	1			Air-heater-B Stoker-B	0.5		0.91
<i>is-connected-to</i> (0.8, 0.2, 0.2)		Air-heater-N WW-N-1 WW-N-2	Air-heater-A		0.36		Air-heater-B WW-B-east WW-B-west WW-B-rear			

S1: $S_{Product, Product-A, Component, Attribute}$; S2: $S_{Product, Product-A, Component}$; S3: $S_{Product, Product-A}$; S4: $S_{Product, Product-B, Component, Attribute}$; S5: $S_{Product, Product-B, Component}$; S6: $S_{Product, Product-B}$.

abstraction (MSCA) of each pair of keywords can be computed. Fig. 6 shows an example of an abstraction hierarchy for different stoker types.

The user may assign a “specificity value” (SP) to each node in the hierarchy as shown by the numbers in parentheses in Fig. 6. The similarity value of two nodes is determined by the SP of their MSCA. For example, the similarity value is 0.8 (SP of “overfeed”) for “chain-grate” and “vibrating-grate,” and 0.4 (SP of *Stoker: type*) for “chain-grate” and “spreader-grate”. Thus, a “chain-grate” is more similar to a “vibrating-grate” than to a “spreader-grate” type of stoker. These two similarity values are shown

in Table 1. The attributes *Boiler’s manufacturer* and *Economizer’s type* are also described using keywords with an abstraction hierarchy (not shown), and their similarity values are also shown in the table.

7.3. Numbers

The similarity between numbers is computed based on a quantitative or qualitative range. A quantitative range is suitable for numbers that can be compared based on quantitative differences. A qualitative range is suitable for numbers that can be compared based on threshold values separating one interval from the next.

For attributes where a quantitative measurement works better, ranges for possible differences are specified to ensure consistency on the similarity scale. A quantitative “difference range” is a list of similarity value and value range pairs. For example, the difference range for the attribute *steam-output* (in metric-ton/hr) of a *Boiler* may be specified as follows:

((1(0 100)) (0.8(100 500)) (0.6(500 1000))
(0.4(1000 2000)) (0.2(2000 3000)) (0(3000 max)))

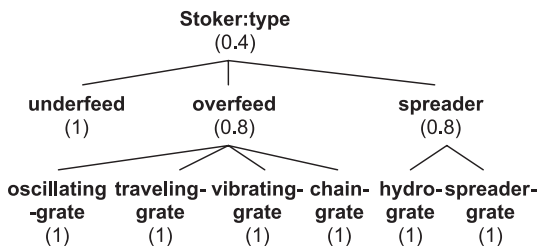


Fig. 6. Abstraction hierarchy for *Stoker: type*.

Thus, boilers with a steam output difference less than 100 ton/h are considered no different. Those with a difference between 100 and 500 ton/h are considered quite similar with an attribute similarity value of 0.8, and so on. In Table 1, the steam output is 900 ton/hr for *Product-New*, 2000 ton/hr for *Product-A*, and 700 ton/h for *Product-B*. Thus, the attribute similarity value is 0.4 for *Product-A* (with a difference of 1100 ton/h) and 0.8 for *Product-B* (with a difference of 200 ton/h).

Now consider another *Boiler's* attribute, namely *operating-pressure*. The operating pressure is 17 MPa for *Product-New*, 22.5 MPa for *Product-A*, and 11 MPa for *Product-B*. If only the quantitative difference is measured, *Product-A* appears to be more similar to *Product-New* than to *Product-B* because it has a smaller quantitative difference with *Product-New*. However, this conclusion about similarity is incorrect. *Product-B* and *Product-New* are both designed to operate below the critical pressure of water, which is 22.1 MPa, and they have a similar configuration. In contrast, *Product-A* is designed to operate at a super-critical pressure (i.e., a pressure above the critical pressure) and has a quite different configuration. A boiler that operates above the critical pressure is usually called a once-through boiler because there is no recirculation of the working fluid in any part of the unit, and it therefore does not require a condenser. Such a boiler does not require drums either, as their primary function is to separate steam from water, and water above its critical pressure can exist only in the form of vapor. Without the installation of drums, which takes a long time and is on the critical path of the erection schedule for a boiler that operates below the critical pressure, the schedule will have quite different activities and durations.

The previous example shows a situation where a qualitative comparison is more relevant than a quantitative one. CasePlan allows the user to specify “qualitative ranges” for such attributes. Each range specification is a list of qualifier and value range lists. A specification example for *operating-pressure* is:

```
((sub – critical(022.1)) (super – critical(22.1 max)))
```

This means that a pressure under 22.1 MPa will be classified as “sub-critical” and one that is equal to or

greater than 22.1 MPa as “super-critical”. Note that a number next to a left parenthesis is inclusive, and a number next to a right parenthesis is inclusive. If no abstraction hierarchy is defined for “sub-critical” and “super-critical”, the attribute similarity value of *Boiler's operating-pressure* for *Product-A* is 0 and for *Product-B* is 1.

7.4. Relationships

The similarity of a relationship attribute is determined based on the number of components (percentage wise) in the new product whose corresponding components can be found in the case. For example, in Table 1, the *Economizer* of *Product-New* has the following relationships:

```
is-on-top-of: (Dust-collector-N Air-heater-N)
is-connected-to: (Air-heater-N WW-N-1 WW-N-2)
```

The *Economizer* of *Product-A* has the following relationships:

```
is-on-top-of: (Air-heater-A Dust-collector-A
Stoker-A)
is-connected-to: (Air-heater-A)
```

For the relationship *is-on-top-of*, *Product-New* has a *Dust-collector* and an *Air-heater*, for both of which corresponding matches in the case (i.e., *Dust-collector-A* and *Air-heater-A*) can be found. Thus, the attribute similarity value of the case with respect to *is-on-top-of* is 1 (=2/2). For the relationship *is-connected-to*, the project has an *Air-heater*, for which a corresponding match in the case can be found, but also has two *WWs*, for which corresponding matches cannot be found. Thus, the attribute similarity value of the case with respect to *is-connected-to* is 0.3 (=1/3).

8. Implementation

CasePlan 1.0, was implemented on a personal computer platform with Microsoft Windows NT 4.0. It was implemented using Allegro Common Lisp 3.0 (ACL 3.0) (Franz 1996). ACL 3.0 is a 32-bit compiler compatible with the ANSI Common Lisp Object

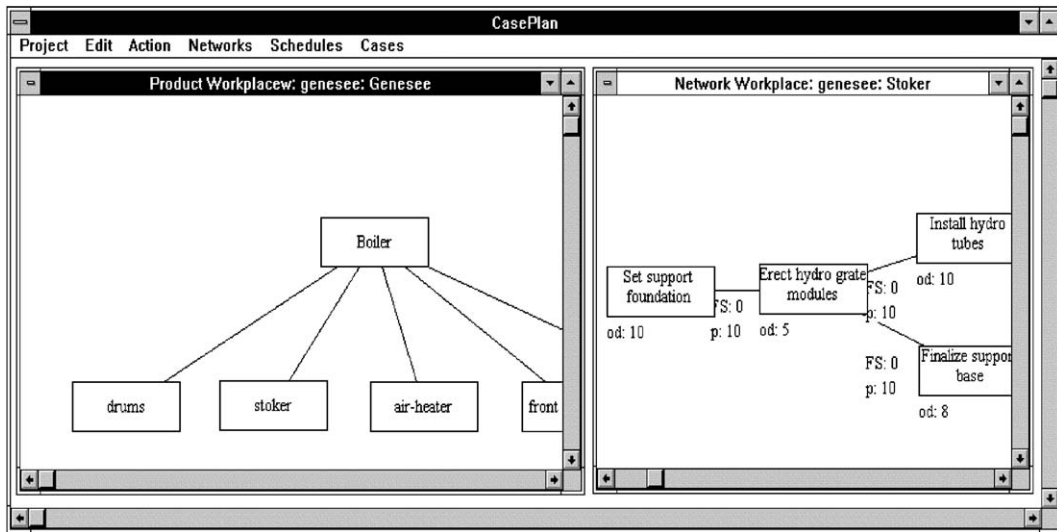


Fig. 7. Case defined in the planner.

System (CLOS), which is an object-oriented Lisp language standard.

CasePlan 1.0 includes two modules: (1) The Model Generator is an application to define or edit generic product models. (2) The Planner allows the user to refine the generic model to describe cases and a new project. It also plans and schedules the new project by reusing cases.

CasePlan’s modules are implemented in an object-oriented fashion. Objects (e.g., windows, dialogs, components, activities, links, and schedules) that the user can see, define, or drag around on the screen are instances of classes that have their own attributes and functions for their intended behaviors. Different windows respond differently to user interaction and pertain to objects of different CasePlan classes such as *Activity* and *Component* discussed in this paper. This allows for multiple combinations of window behaviors.

The Workplace windows display the information about the new project, which includes product definition, represented as a component hierarchy in the Product Workplace window (left window in Fig. 7), component networks, represented by boxes and lines in the Network Workplace window (right window in Fig. 7), and a product network, represented by boxes and lines in the Schedule Workplace window. The Base windows browse the same information about a single case in the case library.

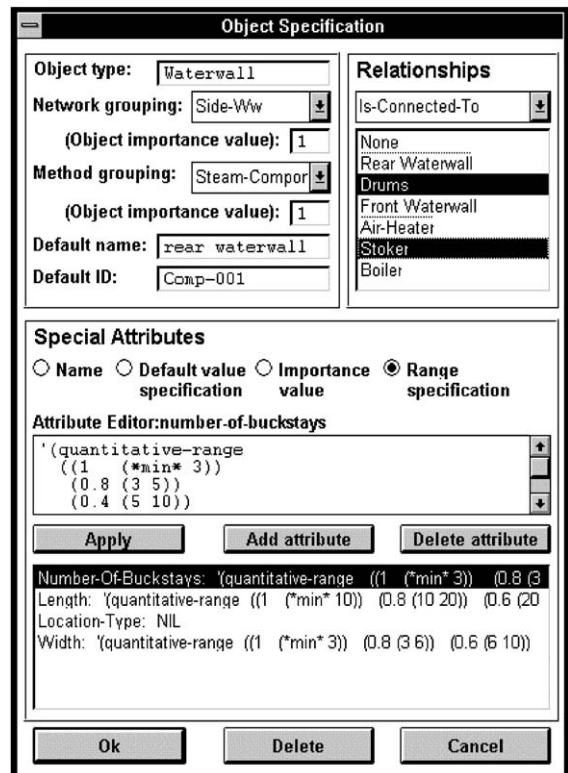


Fig. 8. Object specification dialog viewing range specifications of attributes.

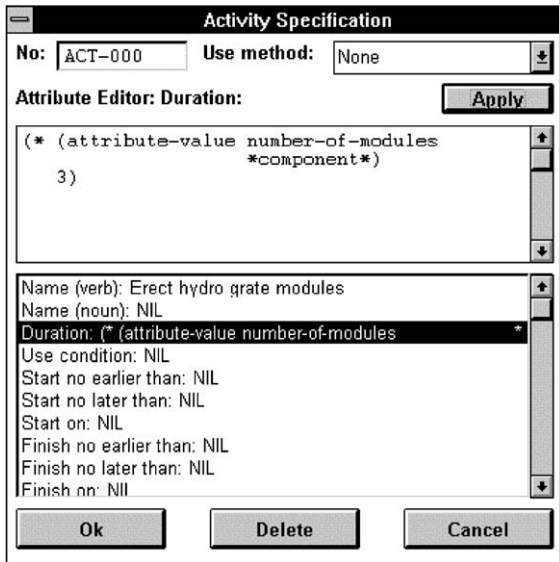


Fig. 9. Activity specification dialog viewing range specifications of attributes.

Double-clicking on an object (i.e., box or line) activates an input dialog for that object. For example, Fig. 8 shows the pop-up dialog for specifying a component when double-clicking on a box in the Product Workplace window. Figs. 9 and 10 show the dialogs for specifying an activity and a link when double-clicking on a box and a line, respectively, in the Network Workplace window.

The drag-down menu items in Fig. 11 show the steps for creating a new schedule. Selecting Reused Cases allows the user to modify CasePlan's prioritization in reusing cases that are ranked based on their similarity functions. The user may ask CasePlan to automatically create a schedule directly by choosing

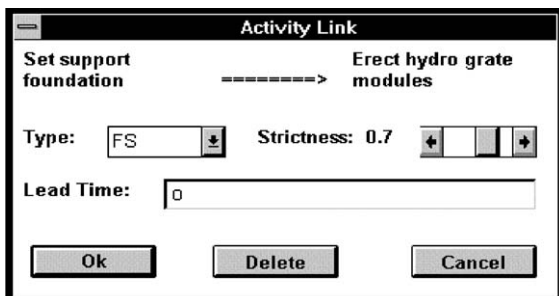


Fig. 10. Activity link dialog.

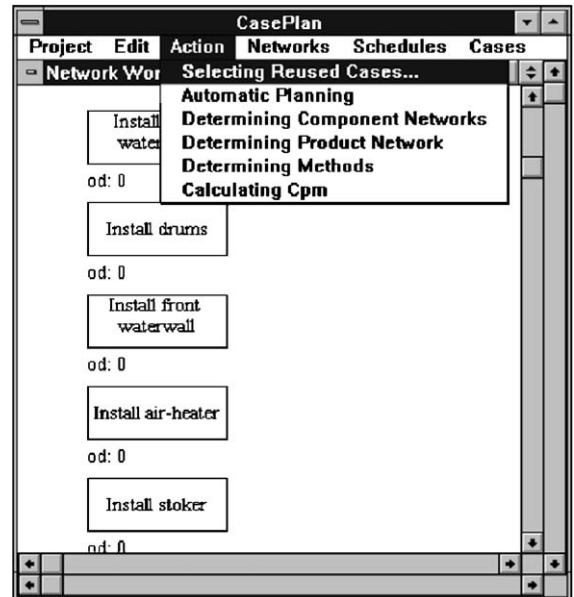


Fig. 11. Actions for creating a new schedule.

Automatic Planning, or do it step-by-step and have the opportunity to modify CasePlan's intermediate outcomes by choosing consecutively Determining Component Networks, Determining Product Network, Determining Methods, and Calculating CPM. The window in Fig. 11 shows an initial set of component-level activities that CasePlan generates based on

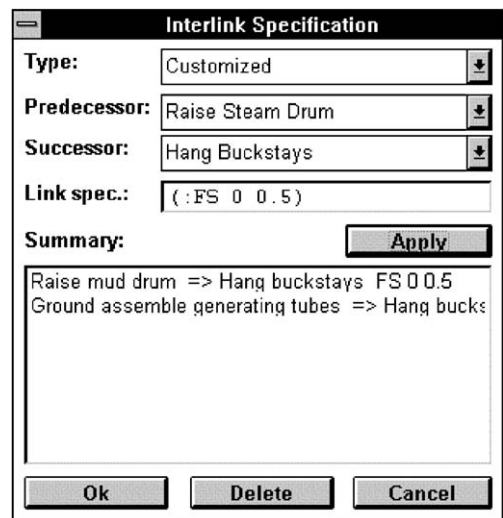


Fig. 12. Customized interlink specification.

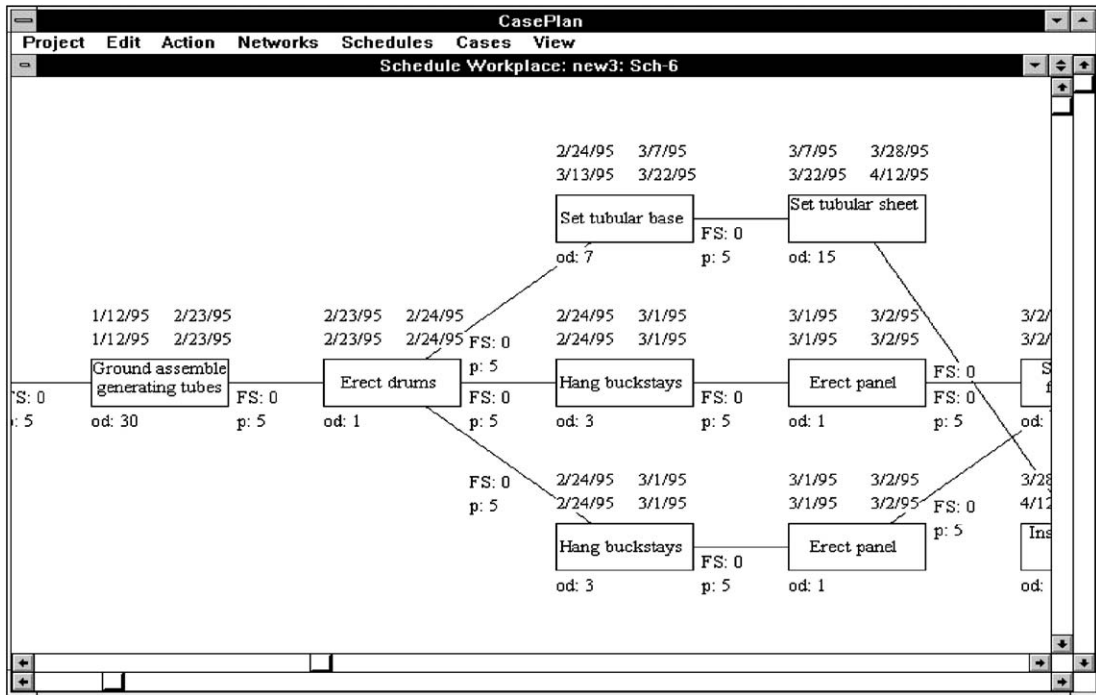


Fig. 13. Schedule generated by the planner.

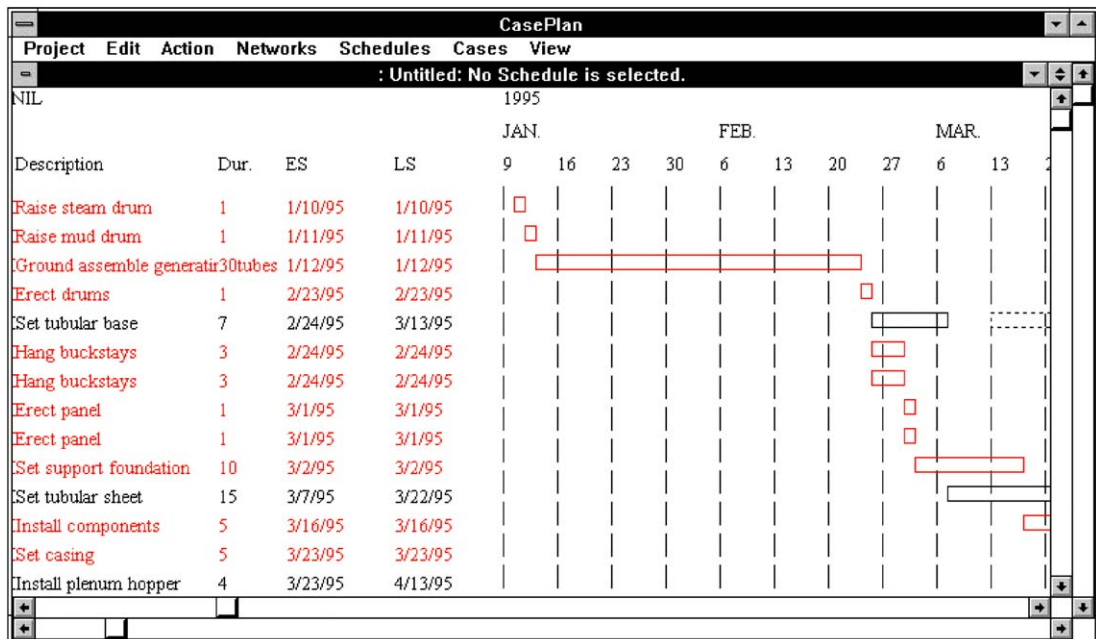


Fig. 14. Bar-chart schedule generated by the planner.

The product similarity value for Grayling case is 0.74.

PROJECT'S COMPONENT	CASE'S COMPONENT	CSV	CIV	CSVxCIV
rear waterwall	rear waterwall	0.97	1.00	0.97
rear waterwall	front waterwall	0.91	1.00	0.91
front waterwall	rear waterwall	0.91	1.00	0.91
front waterwall	front waterwall	0.97	1.00	0.97
air-heater	air-heater	0.00	1.00	0.00
stoker	stoker	1.00	1.00	1.00
drums	drums	0.20	1.00	0.20
boiler	boiler	0.94	1.00	0.94

...

The Genesee case has the highest similarity value of 0.85.
The product similarity values for the evaluated cases are:
0.85 for Genesee case
0.74 for Grayling case

...

The component similarity value for Boiler is 0.94.

ATTRIBUTE-NAME	PROJECT	CASE	ASV	AIV	ASVxAIV
support-type	top-support	top-support	1	1	1
steam-output (1000 lb/hr)	2200	2500	0.8	0.8	0.64
operating-pressure (psi)	2500	2600	1	0.8	0.8

...

NETWORK REUSE SUMMARY:
The rear waterwall component reused the Waterwall network of Genesee case.
The drums component reused the Drums (up) network of Genesee case.
The front waterwall component reused the Waterwall network of Grayling case.
The air-heater component reused the Air-heater network of Genesee case.
...

Fig. 15. Excerpt from sample report.

the product specification. The next step is to determine a component network for each of the activity. Double-clicking on an activity will show the detailed component network for the activity. Dragging a line between any two component-level activities will pop-up the Interlink Specification dialog (Fig. 12).

The Graphic Report Window displays the schedule generated by CasePlan with CPM times and floats calculated as shown in Fig. 13. It may also display the schedule in a bar-chart format as shown in Fig. 14. The Text Report Window shows the intermediate reasoning actions performed by CasePlan, and summaries of product similarity values, component similarity values, and network uses. Fig. 15 shows a sample report.

9. Determination of importance values and evaluation of CasePlan

CasePlan's performance depends on the availability of cases that, as a whole or on a component basis,

are similar enough to the project to be planned for, and on the ability of retrieving and reusing them. Retrieving good cases, in turn, depends on good similarity metrics. We sent surveys to professional schedulers and followed up with interviews to determine the appropriate importance values, assess the feasibility and consistency of professionals' assignment of importance values, and evaluate CasePlan's performance.

We first invited twelve professionals to participate our survey. Each professional had more than 5 years of experience in power plant construction. Their affiliations included Black and Veatch, Townsend and Bottum, ABB-CE, and Zurn Energy Eight professionals agreed to participate in the survey. The survey was conducted in two rounds. The first round was a structured interview with an open-answer questionnaire to gather potential criteria. The second round was a structured interview with the re-designed questionnaire that comprised all significant criteria gathered in the first round of interview and from a literature review.

The interview started with a scenario where the participant was assumed to be an experienced project scheduler who just started working for a boiler manufacturer with years of experience in the design and erection of hundreds of boilers. All the boiler designs and their erection schedules were well organized and stored as cases in the participant's computer but they were unfamiliar to the participant. The participant was given a description of a new project, and was asked to complete the new schedule by using the cases so the schedule would conform to the company's practice. The assumed computer allowed the participant to specify any search criteria and would retrieve the cases that met the criteria.

Two sets of questions were asked in the first round of interviews. The first set of questions asked the participant what criteria and how much weight (from 0 to 10) he would use for the search when completing each of scheduling subtask (e.g., determining a subnetwork for the drums). The second set of questions asked the participant how much weight (from 0 to 10) he would assign to each component to determine the case similarity when he could only view one case at a time and ranking the cases is necessary. The second round of interview repeated the first set of questions with specific criteria for the participant to choose and assign weight (from 0 to 10) to. The average weights, with the highest and lowest answers removed, were used in CasePlan for the evaluation of its performance.

These professionals provided a consistent assessment of what components are important (e.g., generating bank, waterwalls) or not important (e.g., burner, windbox). However, their rankings were different. Similarly, the consensus was pronounced for attributes that received the highest (e.g., size of waterwall) or lowest (e.g., water flow direction of waterwall) importance values. However, there were some discrepancies for attributes (e.g., tube arrangement of waterwall) whose importance values were in the middle range.

The follow-up interviews found some factors contributing to the discrepancies. Experts perceive the schedule at different levels of detail (e.g., a scheduler's schedule is more detailed than a project manager's), and this necessarily affects what components they consider important when reusing schedules. Some components and attributes may have an impact

on the detailed schedule, but may not appear to be of relevance in an abstract schedule. Importance values assigned based on an abstract schedule were ignored when we compute their average.

Construction professionals with less design background might overlook some design attributes that indirectly affect the schedule. This is the case for components whose selection is highly correlated with that of other components. For example, the stoker type is important in determining the stoker's erection procedure. The ash content a boiler's fuel affects the choice of the stoker type (e.g., a traveling-grate type is suitable for a boiler with high ash content, but a vibrating-grate type is suitable for one with low ash content). Thus, the ash content is indirectly an important attribute. Whether or not such indirect attributes are considered has a limited effect on CasePlan's performance if these attributes are highly correlated with other implemented attributes.

In addition to what we learned about professionals' opinions on importance values, we were also glad to learn that they were able to consistently assign importance values. The averages of their assignments were used to perform a test to evaluate the generated schedule and the usability of CasePlan. A project manager and an engineer at Zurn, a boiler manufacturer, created four imaginary boiler designs for fossil-fueled power plants with a capacity ranging from 30 to 90 MW. CasePlan generated schedules for the boilers based on a library of seven actual cases with similar capacities provided by the same manufacturer. The results were also evaluated by the same professionals.

The professionals considered the accuracy of activity sequencing and duration to be the most important criteria in evaluating a schedule. CasePlan was able to produce satisfactory schedules that met both criteria. This success is based on the provision that existing cases have correct networks and formulae for estimating durations. CasePlan was accurate in determining component networks, but not in determining a product network. Projects that are considerably different in configuration often use the same component networks if the associated components are of the same type. However, even for boilers that are very similar to each other (e.g., same capacity, same fuel, and similar size of boiler house), the erection sequence for some components may be different if their configurations or delivery schedules are different.

CasePlan's weak performance in determining a product network can be attributed to the following factors. CasePlan ignores overall project schedule optimization such as time cost tradeoff and equipment selection. One may forget to specify the necessary *use-conditions* for each interlink. One may also forget to specify redundant links and the spatial relationships between components. Inadequate specifications may result in an incorrect or under-constrained schedule. The professionals all agreed that accurate determination of a product network would require many more cases and much more project input, including complete topological relationships between components and other information such as delivery times that are dynamic and uncertain. They would rather inspect interlinks themselves than specify the input and hope for an accurate result.

In determining a product network, the multiple-case approach, which reused the best case for each component pair, resulted in more accurate interlinks than the single-case approach. In determining methods, no significant difference was found between the single-case and multiple-case approaches due to the fact that only major equipment was specified in the previous cases and their types were limited.

The professionals that used CasePlan found it time-consuming to define the generic product model and annotate cases (e.g., specifying the redundant *Link*, *use-condition*, *Method's productivity*). Specifying the *use-condition* and formulae for estimating activity durations using the Lisp language also discouraged them. Nevertheless, specifying the generic product model and annotating cases is a one-time job. The professionals were comfortable with the amount of effort involved in instantiating the model to describe a new project. They all liked CasePlan's ability to generate a preliminary schedule and represent it both in the CPM logic format and in bar charts. They also appreciated CasePlan's component-by-component and attribute-by-attribute report on the comparison of cases.

10. Future research

CasePlan's development can be steered in several directions that are worthwhile of future research efforts. One direction is to integrate CasePlan with

applications in other engineering phase (e.g., design, simulation, maintenance). For example, the AP-227 protocol supports spatial and physical information for the design and construction of piping systems, architectural, electrical, HVAC, instrumentation and control, and mechanical systems. Instead of asking a user to specify the spatial relationships (e.g., *is-connected-to* or *on-top-of*) between components, CasePlan could be extended to receive such spatial information from CAD or 3D product modeling software that support AP-227. CasePlan can also be extended to simulate its generated schedule to avoid interferences during the erection sequence. One could also extend the specification of construction method in CasePlan to include a process simulation network so part of its generated schedule can be sent to construction simulation system such as ABC-Sim [27]. Another direction is to extend CasePlan's applicability. One can study CasePlan's generality by applying it to other types of construction projects. Currently CasePlan has only been applied to the construction of power plant boilers and US post offices. We believe that CasePlan will have a satisfactory performance for the project that has distinct components and whose construction sequences mainly depend on the configuration of the components instead of other factors such as geography, climate, site condition, and resource availability. For example, CasePlan may not be applicable for tunnels (which have no distinct components) or highway construction (whose construction sequences are significantly affected by the surrounding geography and equipment availability).

The last direction is to improve CasePlan's capabilities. One can investigate other similarity schemes that may produce better schedules for some types of projects. For instance, CasePlan's value specification language can be extended to allow a user to specify a set of criteria for searching cases to be reused for each scheduling subtask. As an example, one could ask CasePlan to reuse cases whose drums' weight and number of generating tubes fall in certain ranges. These criteria can be gradually relaxed if no case is found that meets them. This approach seems to be more similar to how people retrieve cases in real life. However, automation of this approach is not easy. One can also develop an application to convert a Primavera schedule into CasePlan component networks and interlinks based on the schedule's activity

numbering or work breakdown structure. The networks and interlinks should be annotated to the extent possible (e.g., *activity duration*, *link type* and *for-component*). However, annotating value specifications and *use-condition(s)* cannot be automated because they cannot be described in Primavera.

One can further incorporate a learning ability for CasePlan to find better similarity functions. When the initial similarity functions do not produce satisfactory schedules because they lead to what the user perceives to be wrong cases, the user will need to adjust the function's parameters. This adjustment is tedious and might be automated provided that the user can identify correct cases. For example, a component similarity function is an aggregation of a multiplication of pairs of attributes and importance values. In planning a new project, assume that CasePlan suggests Case-1 as the best case based on the similarity function but the user identifies Case-2 as the best one instead, and wishes to adjust importance values so that Case-2 would be retrieved the next time. One way to automate the adjustment is to increase, with a specified increment, the importance value of the multiplication pair that favored Case-2, and decrease the importance value of the multiplication pair that favored Case-1.

One may also investigate the usefulness and possibility of generalizing planning knowledge stored in cases through reasoning by induction. The planning knowledge in a case includes the product network, component networks, methods, formulae for estimating activity durations, etc., that are stored in the case. One could create for each type of project a generalized case that is a generalized representation of the individual cases. For example, the generalized case may show what component networks have been used for one type of component and in what conditions they should be reused. This type of generalization often happens in human schedulers' minds as they learn project by project. Nevertheless, it is not known if the generalization can improve CasePlan in terms of quality or efficiency.

11. Conclusions

CBR provides a method to custom tailor a construction planning and scheduling system to individual projects and a contractor's practice. This paper pre-

sented CasePlan's generic product model and its use in a CBR process. The model allows for the association of a facility design with its schedule, and the retrieval of reusable schedule parts based on their similarity. CasePlan includes several similarity assessment schemes that allow its user to express notions of similarity for different planning tasks and types of attribute values within each generic product model. This approach is fundamentally different from the approaches taken by existing computer-based planning systems.

The survey conducted by the researchers showed that professionals are able to consistently assess importance values, and that there is consensus on which components and attributes are the most (least) important. These attributes dominate CasePlan's result for case retrieval. The discrepancies in professionals' opinion on the attributes whose importance values are in the middle range were explained. Experimentation showed CasePlan's accuracy in determining component networks and activity durations, but weak performance in determining interlinks between component networks. Performance may be improved by adding more cases and detailing project information, as noted, with redundant links in order to facilitate reuse. In addition to schedule generation, CasePlan's systematic report on the comparison of cases also helps professionals review existing cases.

Acknowledgements

We like to acknowledge the support from and participation in this research by professionals at ABB Combustion Engineering Services, Black and Veatch, The Christman, Genesee Power Station, Graying Generating Station, Northern Boiler Mechanical Contractors, Townsend and Bottum, and Zurn Industries. 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.

References

- [1] S.F. Bailey, F.C. Ian, Case-based preliminary building design, A.S.C.E. J. Comput. Civ. Eng. 8 (4) (1994) 454–468.

- [2] J. Cheng, P. Trivedi, K.H. Law, Ontology mapping between PSL and XML-based standards for project scheduling, in: I.D. Tommelein (Ed.), Proc. the 3rd Intl. Conference on Concurrent Engr. in Construction, U.C. Berkeley, Berkeley, 2002, pp. 143–155.
- [3] D.K.H. Chua, D.Z. Li, W.T. Chan, Case-based reasoning approach in bid decision making, A.S.C.E. J. Constr. Eng. Manag. 127 (1) (2001) 35–45.
- [4] Computer-Integrated Construction Group, National Institute of Standards and Technology, <http://cic.nist.gov/piebase> (Feb. 1, 2001).
- [5] R.J. Dzeng, I.D. Tommelein, Boiler erection scheduling using product models and case-based reasoning, A.S.C.E. J. Constr. Eng. Manag. 123 (3) (1997) 338–347.
- [6] C.M. Eastman, Building Product Models: Computer Environments Supporting Design and Construction, CRC Press, Boca Raton, USA, 1999, pp. 279–318.
- [7] D. Echevery, W. Ibbs, K. Simon, Sequencing knowledge for construction scheduling, A.S.C.E. J. Constr. Eng. Manag. 117 (1) (1991) 118–130.
- [8] Electric Power Research Institute, Guidelines for Specifying Integrated Computer-Aided Engineering Applications for Electric Power Plants, EPRI NP-5159M. (EPRI, Palo Alto, CA, 1987) 75–121.
- [9] M. Fischer, Linking CAD and expert systems for constructibility reasoning, Proc. the 5th Intl. Conf. on Comp. in Civil and Building Engr, ASCE, Anaheim, CA, 1993, pp. 1563–1570.
- [10] Franz, Allegro, CL/PC 3.0 Reference Manual, vol. 1(13), Franz, Berkeley, CA, 1996, pp. 1–12.
- [11] T. Froese, Information Standards in the AEC Industry, Can. J. Civ. Eng., vol. 11(6), NRC Research Press, Ottawa, 1994, pp. 3–5.
- [12] W. Gielingh, The General AEC Reference Model (GARM), Working Paper ISO TC184/SC4/WG1 N158, CADLAB IBBC-TNO, Netherlands, 1987.
- [13] C. Gray, Intelligent Construction and Cost Analysis, Constr. Manag. Econ., vol. 4, 1986, pp. 135–150.
- [14] Industry Alliance for Interoperability, http://www.iai-international.org/iai_international (May 24, 2003).
- [15] International Organization for Standardization (ISO), <http://www.iso.ch/iso/en/ISOOnline.frontpage> (Jan. 11, 2003).
- [16] ISO TC184/SC4, <http://www.tc184-sc4.org> (Dec. 16, 2002).
- [17] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann, San Mateo, CA, 1993.
- [18] K.W. Li, A.P. Priddy, Power Plant System Design, Wiley, New York, 1985, pp. 311–330.
- [19] M.L. Maher, A.G. de Silva Garza, Design case adaptation using genetic algorithms, Proc. the 8th Intl. Conf. on Comp. in Civil and Building Engr., ASCE, Anaheim, CA, 1996, pp. 294–300.
- [20] M.L. Maher, Creative design using a genetic algorithm, Proc. the 6th Intl. Conf. on Comp. in Civil and Building Engr., ASCE, New York, 1994, pp. 2014–2021.
- [21] E. Miresco, Expert system for scheduling building projects, Proc. CIB 92, vol. 1, National Research Council, Canada, 1992, pp. 258–259.
- [22] S.T. Ng, N.J. Smith, Verification and validation of case-based prequalification system, A.S.C.E. J. Comput. Civ. Eng. 12 (4) (1998) 215–226.
- [23] H.N. Ahuja, S.P. Dozzi, S.M. Abourizk, Project Management: Techniques in Planning and Controlling Construction Projects (Wiley, New York, 1994) 20–25, 73–75.
- [24] OSHA, Construction Industry: OSHA Safety and Health Standards, Occupational Safety and Health Administration, Washington, DC, 1985, 29 CFR 1926/1910, OSHA 2207.
- [25] PlantSTEP, <http://cic.nist.gov/plantstep> (May 8, 2003).
- [26] Primavera Systems, Primavera Project Planner Reference Manual-Version 3.0, Primavera Systems, Bala Cynwyd, PA, 1999, pp. 58–60.
- [27] J.J. Shi, Activity-based construction (ABC) modeling and simulation method, A.S.C.E. J. Constr. Eng. Manag. 125 (5) (1999) 354–360.
- [28] H.M. Spring, A.L. Kohan, Boiler Operator's Guide: Boiler Construction, Operation, Inspection, Maintenance, and Repair, with Typical Boiler Questions and Answers for Plant, Operating, and Maintenance Engineers, McGraw-Hill, New York, 1981.
- [29] STEP Tools, <http://www.steptools.com/library/standard> (2003).
- [30] I.D. Tommelein, G. Ballard, G. Coordinating Specialists, Technical Report No. 97-8, Dept. of Civil and Environment Engr, U.C. Berkeley, CA, 1997.
- [31] UK Council for Electronic Business (UKCEB), <http://www.ukceb.org/step> (July. 18, 2003).