AUTOMATED SCHEDULE ADVISORY SYSTEM

Ren-Jye Dzeng and Hsin-Yun Lee

Department of Civil Engineering
National Chiao-Tung University, Taiwan

Planning and scheduling is one of the key factors to the success of a construction project. Because of the increasing complexity and size of construction projects nowadays, pre-construction schedules tend to comprise a large number of activities. These schedules often contain unintentional errors and conflicts, and those due to the lack of experience. With the increasing power and decreasing cost, personal computers have become a common tool used by construction professionals to facilitate their daily decisions. This research applies case-based reasoning and rule-based reasoning techniques, and develops a computer system called ScheduleCoach, which can analyze a computerized schedule and provide corrective advises.

Keywords: Planning, Scheduling, Case-Based Reasoning, Expert System, Critique.

1. INTRODUCTION

Planning and scheduling is one of the key factors to the success of a construction project. Because of the increasing complexity and size of construction projects nowadays, pre-construction schedules tend to comprise a large number of activities and much related information. These schedules often contain unintentional errors and conflicts, and those due to the lack of experience. These mistakes need to be corrected, and often they can be corrected easily by an experience scheduler. However, due to the number of activities involved, sometimes these mistakes are hard to be found.

With the increasing power and decreasing cost, personal computers have become a common tool used by construction experts to facilitate their daily decisions such as scheduling, cost estimation, and cash flow analysis. This research applies artificial intelligence techniques (i.e., case-based reasoning and rule-based reasoning) and develops a computer system called ScheduleCoach, which can analyze a computerized schedule and provide corrective advises. The current knowledge base of the system focuses on high-rise building construction.

2. LITERATURE REVIEW

Various research subjects related to planning and scheduling can be found in the construction related literature and texts. These include activity coding, activity sequential relationship, schedule representation, resource leveling, etc. As more artificial intelligence techniques are being developed, many automatic construction planner have also been developed. These planners define activities and their sequential relationships; some also estimate activity duration. The examples include BUILDER, Construction Planex, GHOST, Know-Plan, OARPLAN, SIPEC, ATOP, ConsPlans, CASCH, TCIS, HISCHED, and CasePlan. Except for CasePlan, the planners apply the rule-based reasoning, and create an activity network for a given project described using a predefined set of component hierarchy. The planners do not have the learning ability; i.e. given the same project, the output network are the same over the time. CasePlan focuses on the learning ability of a planner as the case in human planners, and recognizes that part of planning work cannot be predetermined by a set of rules, and is dependent on the planner’s experience and preference.

De la Garza and William Ibbs (De La Garza and Ibbs 1990) developed a schedule critique system, called CRITEX, which applies the rule-based reasoning, and evaluates construction schedules of high-rise buildings based on a set of critique rules obtained through interviews with several human schedulers. The output of CRITEX is a set of critique statement. However, the report does not include the suggestions regarding how the schedules should be revised. Our research extends the critique knowledge with a different knowledge representation. The output of the system is a list of critique and modification suggestions based on the integrated the rule-based reasoning and case-based reasoning (CBR).

3. CONSTRUCTION PLANNING AND SCHEDULING KNOWLEDGE

We focus on high-rise building construction and acquire the knowledge required to develop a schedule advisory system. The knowledge acquisition is carried out by reviewing the related literature such as texts, papers, and schedules of successful projects, interviewing 12 experts including experienced scheduler and project managers from 3 top-20 contractors and a consulting firm, and on-site observation of 3 job sites in the north area of Taiwan.

This collected knowledge includes:

(1) mandatory principles and logics, e.g., no negative floats, lower level being built before higher floor;
(2) improvments for better scheduling practice, e.g., adjustment of activity duration, more readable schedule representation format.

The knowledge can be further categorized based on their causes and objectives, perspectives, and critique types. Based on the causes and objectives, the knowledge is divided into 9 groups including contract requirement, schedule management, cost management, quality management, purchase, site, safety and environmental protection, regulations, and scheduler's preference. Based on the perspectives, the knowledge is divided into 5 group including government, owner, A/E, and contractor/ subcontractor. Based on the critique types, the knowledge is divided into 7 groups including the number of activities, and activity's naming, duration, precedence relationships, start and finish dates, cost, and resources.

4. REPRESENTATION OF KNOWLEDGE

ScheduleCoach is developed based on the object-oriented concept. The primary objects defined in the system are as follows with their primary attributes listed in the parenthese.

(1) *Project* (Name, Descripition, StartDate, Activities, Links, etc.)
(2) *Activity* (Name, Location, PlannedDuration, ActualDuration, PlannedStartDate, ActualStartDate, WBS, Resources, etc.)
(3) *Link* (Type, LeadTime, Predecessor, Successor)
(4) *Resource* (Name, Type, Quantity, UnitPrice, etc.)

The *Project* may comprise many *Link* and *Activity* objects, and the *Activity* may comprise many *Resource* objects.

This paper refers multiple objects in the collection bracket {*Object1*, *Object2*, ...}*t*, where *t* describes how the collection should be applied as detailed in the follows.

(1) $\{\ldots\}_{All}$ means all elements in the collection are applicable. E.g., {Concrete-Pouring-FL(1), Concrete-Pouring-FL(2)", Concrete-Pouring-FL(3)$\}_{All}$ represents all concrete pouring activities on three floors.
(2) $\{\ldots\}_{AtLeastOne}$ means at least one element in the collection is applicable. E.g., {Concrete-Pouring-FL(1), Concrete-Pouring-FL(2)", Concrete-Pouring-FL(3)$\}_{AtLeastOne}$ represents at least one concrete pouring activity on some floor.
(3) $\{\ldots\}_{In(i)}$ means each element in *i* is applicable. E.g., {Concret-Pouring.planned-duration$\}_{In(Floor)}$ represents the planned durations of all concrete pouring activities in each floor.

The primary part of the collected knowledge is the scheduling principles based on which experienced schedulers criticize and correct a schedule. Each principle can be represented by one or several critique rules.. A general form of a **critique rule** comprises four parts: (1) rule application conditions, (2) object application condition, (3) critique statement, and (4) critique reason.

(1) The **rule application conditions** describe the conditions of a project where the rule can be applied. For example, the rule "Avoid scheduling structure lifting activities during typhoon season" is only applicable for steel-structured building projects.
(2) The **object application conditions** describe the conditions of objects where the rule can be applied. For example, one rule may be applicable only for activities with a name string that contains "install".
(3) The **critique statement** is an IfNot-Then statement, where if the applicable objects do not meet the IfNot condition then proceeds the Then statement. For example, one critiuqe statement may specify that the planned start dates of weather sensitive activities should avoid July and August when typhoons are common in Taiwan. The Then part of statement can be a general suggesting or warning text, or a specific corrective advice of values for certain object attributes. The values can be predetermined and embedded in the rules, or dynamic values calculated based on case-based reasoing (CBR) results.
(4) The **critique reason** provides explanations for the critique statement. For example, one critique reason may be "The construction duration for each underground floor should not exceed 3 months according to Kaoshiung City Building Codes."
(5) The **rule control settings** determine the behavior of the rule during the run time. The choices include:
   - *Mandatory?* All the cases or objects that are used to suggest approparaite values for attributes

that are criticized need to conform to mandatory rules.

● *Activated?* The rule will be deactivated if this setting if off.

● *ShowGeneralSuggestion?* When the *Then* part of the rule is executed, precorded text will be displayed or recorded in the log if this setting is on.

● *CBRValueSuggestion?* When the *Then* part of the rule is executed, a CBR process will be performed to obtain an appropriate suggested value for the examed object attribute.

The follows are two examples describes how critique rules are represented in ScheduleCoach.

Example 1: All activity durations should be greater than 5 days and not exceed 25 days [De La Garza et al.1990].

Rule application conditions

Null

Object application condition

Null

Critique statement: IfNot part

{Project.Activity.PlannedDuration | Null} Between 5 And 25

Critique statement: Then part

MsgBox (
"Activities should be broken down so their durations are between 5 and 25 days.")

Critique reason

"Activities should be broken down so their durations can be controlled within a reasonable range. Between 5 and 25 days is a reasonable range suggested in [De La Garza et al.1990]."

Example 2: All activity costs should be greater than 0.1% and not exceed 2.5% of total project costs [De La Garza et al.1990].

Rule application conditions

Null

Object application condition

Null

Critique statement: IfNot part

{Project.Activity.PlannedCost | Null} Between (Project.PlannedProjectCost *0.1%) And (Project.PlannedProjectCost *2.5%)

Critique statement: Then part

MsgBox (
"Activities should be broken down so their costs are between 0.1% and 2.5% of total project cost.")

Critique reason

"Activities should be broken down so their costs can be controlled within a reasonable range. Between 0.1% and 2.5% of total project cost is a reasonable range suggested in [De La Garza et al.1990]."

Example 3: The total duration for a building project whose number of floors is less or equal to 5 should not exceed 12 months.

Rule application conditions

Project.Location = Kaoshiung*

Object application condition

Null

Critique statement: IfNot part

{Project.PlannedDuration | Null} < 365

Critique statement: Then part

MsgBox (
"The total project duration should not exceed 365 days.")

Critique reason:

"According to Kaoshiung City Building Codes, the total duration for a building project whose number of floors is less or equla to 5 should not exceed 12 months."

5. SYSTEM ARCHITECTURE

As shown in Figure 1, ScheduleCoach comprises two sets of domain knowledge. One is represented by rules (i.e., rule library), and the other is represented by cases (i.e., case library). Rules represent our result from interviews with experts, literature review, and on-site observation. Cases are computerized representation of successful projects. Given a computerized schedule and its project descriptions, the screening mechanism performs a rule-based reasoning process based on the rule library. All the rules whose application conditions are satisfied are activated, and activities or their sequential relationships that violate the rules are collected and sent into the correcting mechanism. For each unfit activities and sequential relationships, the the correcting mechanismmechanism performs a CBR process based on the case library, and tries to suggest appropriate values based on cases that are similar to the project at hand. The output of the system comprises a list of critiques and modification suggestions.

The user may change the parameters of the correcting mechanism (e.g., change the weights of the function that determine the similarity of projects). The user may also modifies, adds, or deletes the rules in the rule library and the cases in the case library.

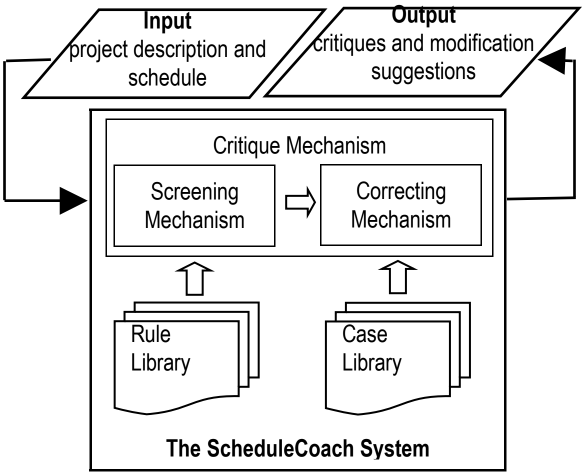

Figure. 1 The System Architecture

Figure 2 describes the reasoning process of ScheduleCoach given a project with a computerized schedule. Give a project schedule, for all critique rules whose application conditons satisfy, exam the attribute values of each applicable object (i.e., activity, link, resource). If the attribute value satisfies the *If*-condition of the rule, execute the *Then*-action of the rule, which can be a general suggesting or warning text, or/and a specific value suggested based on the corresponding attribute value of the most similar object of the most similar case.

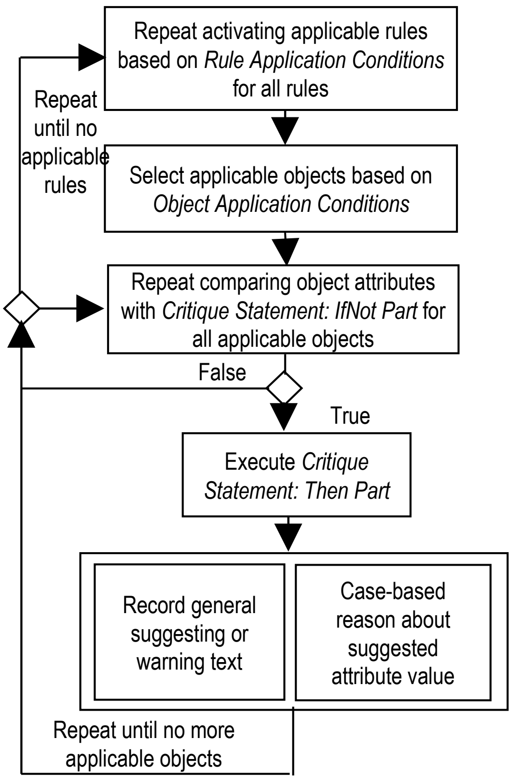

Figure 2 The Reasoning process of ScheduleCoach

When the *CBRValueSuggestion?* option of the rule control settings is turned on, ScheduleCoach will try to suggest an appropriate value for those attributes that are criticized based on CBR. The CBR process determines the most similar object from the most similar case respective to the new project at hand. The corresponding attribute value of the most similar object will be used as the suggested value for the criticized attribute of the new project.

SchecheCoach adopts CasePlan's way of calculating similarity values [Dzeng]. The similarity value of two compared attributes is determined based on the types of attributes, which include logic, numeric, string, and keyword values. The similarity value of two compared objects is determined based on the weighted average of their attribute similarity values. The similarity values of two compared cases is determined based on the weighted average of their object similarity values.

Figure 3 shows the process of suggesting values for citicized attributes. Given an attribute value being criticized by a critique rule, ScheduleCoach first searches for the most similar case based on the calculated similarity values. Because there is no guarantee all previous cases conform to critique rules, ScheduleCoach makes sure that only the case that conforms to the rule can be used. From the selected case, ScheduleCoach searches for the object that is most similar to the object whose attribute is being criticized. For the object to be used, its similarity value should be greater than the threshold determined by the user, and conform to the critique rule and all the mandatory rules. If such attribute value exists, it will be used as the suggested value for the criticized attribute. Otherwise, a message of "unable to suggest" will be reported.

6. CONTRIBUTIONS OF RESEARCH

This research collected critique knowledge through interviews with scheduling experts, literature reviews, and on-site observations. The knowledge is represented as rules with 3 types of indexes (i.e., cause/objective, perspective, and critique type). A general form to represent critique knowledge in a computable form is also presented in this paper. A hybrid rule-based and case-based system, named ScheduleCoach, is developed to facilitate human schedulers find the errors of schedules and suggest appropriate correction.

Acknowledgements

This project was partially funded by National Science Foundation (NSC 89-2211-E-009-046). We thanks participating site managers and engieers at Continent Construction Co., Ta-Hsin Construction Co., Kendge Construction Co., and China Engineering Resources Consultants. We also appreciate the help and the data offered by the professionals at the job sites of Government Human Resource Development Center and Fu-Bund-Tun-Nan building.

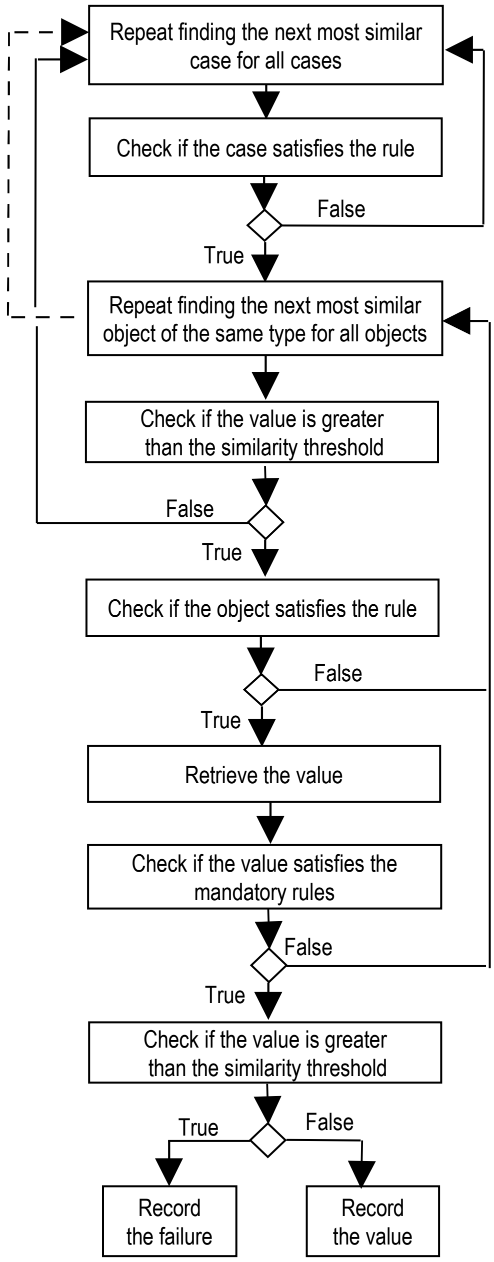

Figure 3 The Process of Suggesting Values for Criticized Attributes

REFERENCES

[1] Asimov, W., 1962, Introduction to Design, Prentice-Hall, Englewood Cliffs, NJ.

[2] Baudin, C., Uderwood, J., Baya, V., Mabogunje, A., 1993, Using Device Models to Facilitate the Retrieval of Multimedia Design Information, Proc. of 13th IJCAI, pp. 1237-1243.

[3] Bicharra Garcia, A.C., Howard, C., and Stefik, M.J., 1993, Active Design Documentats: A New Approach for Supporting Documentation in Preliminary Routine Design, CIFE Tech. Report Nr. 82, CIFE, Stanford University, Stanford.

[4] Coyne, R.D., Rosenman, M.A., Radford, A.D., Balachandran, M., and Gero, J.S., 1990, Knowledge-based Design Systems, Addison-Wesley Publishing Company.

[5] Cutkosky, M., Engelmore, R., Fikes, R., Gruber, T.R., Genesereth, M., Mark, W., Tenenbaum, J.M., Weber, J.C, 1993, PACT: An Experiment in Integrating Concurrent Engineering Systems, IEEE Computer, special issue on Computer Support for Concurrent Engineering, pp. 28-37.

[6] Chung, P.W.H. and Goodwin, R., 1994, Representing Design History, ed. J. Gero, Lausanne, Switzerland, Kluwer Publishing Inc., pp. 735-751.

[7] De La Garza, J.M. and Ibbs,C.W., "Knowledge Elicitation Strategies and Experiments Applied to Construction," Journal of Computing in Civil Engineering,4(2), ASCE (1990)

[8] Dzeng, R.J. and Tommelein I.D. (1997). "Boiler Erection Scheduling Using Product Models and Case-Based Reasoning." J. of Construction Engineering and Management, ASCE, 338-347, New York, NY.

[9] Fenves, S. J., Fleming, U., Hendrickson, C., Maher, M. L. and Schmitt, G., 1990, Integrated software environment for building design and construction, Computer-Aided Design, 22(1), Butterworth Scientific, Ltd., London, pp. 27-36.

[10] Fruchter, R. and Krawinkler, H., 1995, A/E/C Teamwork, Proc. Second ASCE Congress of Computing in Civil Engineering, Atlanta,, pp. 441-448.

[11] Fruchter, R., Reiner, K., Toye, G., Leifer, L., Collaborative Mechatronic System Design, CERA Journal, December 1996, Vol 4, pp. 401-412.

[12] Gruber, T.R., 1989, The Acquisition of Strategic Knowledge, Boston Academic Press,

[13] Khedro, T., 1995, AgentCAD for Cooperative Design, to be published in CAAD Futures 95 Proc., Singapore.

[14] Krishnamurthy, K. and Law, K.H., 1995, Configuration Management in a CAD Paradigm, to appear in Proc. of 1995 International Mechanical Engineering Congress, San Francisco.

[15] Kunz, W. and Rittel, H., 1970, Issues of Elements of Information Systems, Center for Planning an Development Research, UC. Berkeley.

[16] Lakin, F., Wambaug, H., Leifer, L., Cannon, D. and Sivard, C., 1989, The Electronic Design Notebook: performing medium and processing medium, Visual Computer: International Journal of Computer Graphics, 5(4), pp. 214-226.

[17] McCall, R., 1987, PHIBIS:Procedurally Hierarchical Issue-based Information Systems, Planning and Design Acquisition, Boston, MA.

[18] Pohl, J. and Chapman, A., 1990, Expert System for Architectural Design, Journal of Real Estate Construction, 1, pp. 29-45.

[19] Saad, M., 1995, Shared Understanding in Synchronous Collaobrative Building Design, Ph.D. Thesis, University of Sydney.

[20] Schon, D.: 1983, The Reflective Practitioner, Basic Books, Inc., New York.

[21] Sriram, D., Logcher, R., Wong, A. and Ahmed, S.: 1991, An Object-Oriented Framework for Collaborative Engineering Design, in D. Sriram, R. Logcher, and S. Fukada, (eds), Computer-Aided Cooperative Product Development, Springer-Verlag, Berlin.