The 5th International Symposium on Robotics in Construction June 6-8, 1988 Tokyo, Japan

CONSTRUCTION NETWORK GENERATION WITH APPROXIMATE REASONING

C. WILLIAM IBBS, T.C. CHANG & DIEGO ECHEVERRY

215A McLaughlin Hall
Department of Civil Engineering
University of California
Berkeley, CA 94720

ABSTRACT

An important aspect of construction project control is accurate development of a baseline schedule. It becomes the standard against which all progress and performance is measured. Knowledge-Based Expert Systems (KBES's) offer promise of helping managers control projects better. Research at U.C. Berkeley is being conducted to utilize this tool.

Concepts necessary for an Automatic Construction Schedule Generator are described in this article. The prototype outlined in this paper works from CAD-type information to form a milestone schedule. The goal is to design and implement a system that prototypes a planning system for construction. In addition, the argument is made here that understanding and interpreting qualitative information is crucial to a realistic system. We then go on to describe work aimed at incorporating an approximate reasoning facility into the underlying KBES.

1.0 INTRODUCTION

Numerous attempts at developing construction schedule Knowledge-Based Expert Systems (KBES's) have been reported. Among the more successful have been the efforts by Levitt (1), Gray (2), and De La Garza (3,4). Though each has a somewhat different focus, they generally address the processes of developing and analyzing schedules.

Levitt's work with Platform demonstrates the power of an interactive, graphically-based system for conducting "what if" planning. Work by Gray is useful for constructability analysis which, in a way, touches on scheduling. Previous work by this investigator utilizes a programming environment and its concepts to facilitate the critical analysis of original and in-progress network diagrams. The point of view is a project owner's.

The costs of inaccurate building schedules justify the development of intelligence-simulation tools like KBES's. These previous attempts are limited by their demand for: (1) deterministic, rather than stochastic, information; and (2) information, whether input or output, that is precisely and quantitatively represented. Human knowledge is not always like that though. The next generation of KBES tools are beginning to address these deficiencies.

2.0 CURRENT RESEARCH

Prior research (3,4) forms the basis for this current project. That effort addressed the development of an expert system that conducted both initial and in-progress schedule analysis. The goal was to aid project owners about the reasonableness of a contractor's initial schedule submission and the subsequent updates. It was sponsored by the U.S. Corps of Engineers who are very concerned about the legitimacy of contract schedules, for two reasons. The obvious concern is that the plans be accurate so the Corps can plan and occupy the completed project with minimal cost and dislocation. A second important reason is claims defense.

A rule-based system was developed for the Corps by this research team. It can be considered an intelligent decision assistant for resident engineers in the field, and is currently in a field testing. Preliminary signals suggest that it is indeed a valuable tool.

Four different rule directories constitute the system: cost, schedule, logic and general requirements. The cost concepts pertain largely to overall cash flow details, while those in the schedule directory are more productivity and environmentally related. Logic rules are normally construction logic factors. General requirements pertain to project management necessities like submittals and approvals, building code permits, and compliance with start-finish dates.

To continue this earlier research and to take advantage of more powerful programming concepts, this team has begun using the Automated Reasoning Tool^{tm} (ART) from Inference Corp. Frame and object-oriented data representation and rule-based logic facilities are available. The object-oriented and frame representation ideas are especially important in modeling the Work Breakdown Structure.

3.0 EXTENSIONS: AUTOMATIC NETWORK GENERATION

The objective of this part of our work is to automate some of the network generation task. The result will be a system that assists the project manager in generating an initial schedule. The tools used are taken from the field of Artificial Intelligence. The scope is mid-rise office buildings.

In the first phase of the present work, the KBES described in section 2.0 (with no learning capabilities) is to be advanced. In a second phase, plans call for incorporating a module that will contain a record of the solution process for previous similar problems. By comparing the current problem to existing records, analogies can be found and used. From those records that are more similar to the problem in hand, a solution to the current problem can be obtained by adequately transforming the existing solution.

Flexibility, expandability, and efficient interaction between the different types of knowledge are crucial. For those reasons, a blackboard architecture is employed. Information inside the knowledge modules is stored by an object-oriented approach (frames, inheritance, message passing). Rules are also used.

As mentioned, the prototype consists of an underlying KBES. It encompasses an Inference Engine and a User Interface dictated by the Goldworks shell. It will receive as input a description of the project for which a schedule is required. This input is presently manually performed, but a future interface will allow electronic transmission, possibly from a CAD file.

Important components of the system are the knowledge base and a procedural module in charge of supporting network related computations (critical paths, activity floats, etc.). With phase II, the relevant additions consist of another knowledge module for storing previous scheduling tasks. There is also an analogical reasoning module for identifying relevant past solutions and transforming them to match the current condition. Knowledge is stored in a variety of modules depending on characteristic and use.

3.0.1 Knowledge Modules

<u>Construction Knowledge Module</u>: This knowledge module should store the different systems that constitute a mid-rise building, and the major construction methods that can be used to build them. For example, a building can have

either a shallow (e.g. footings) or a deep (piles) foundation. Piles can be steel, or precast or cast in place concrete, or even timber. Our prototype is restricted to a few such construction methods for a group of representative systems. Foundations are one such group explored in detail.

The knowledge is represented in this module mainly with frames, and using inheritance features.

Assumption Handling Knowledge Module: When a schedule of is needed, not all the required information may be available. In these cases, schedulers use experience and intuition to assume information not present.

This module makes the necessary assumptions when the description of the project is missing. Knowledge representation is mainly rule based.

Scheduling Knowledge Module: This module contains the knowledge used to: (1) Perform the breakdown of the project into activities, (2) interconnect activities (logic generation), (3) generate activity durations, and (4) comply with imposed constraints and milestones.

This knowledge is represented with rules. They access the other knowledge modules by using demons.

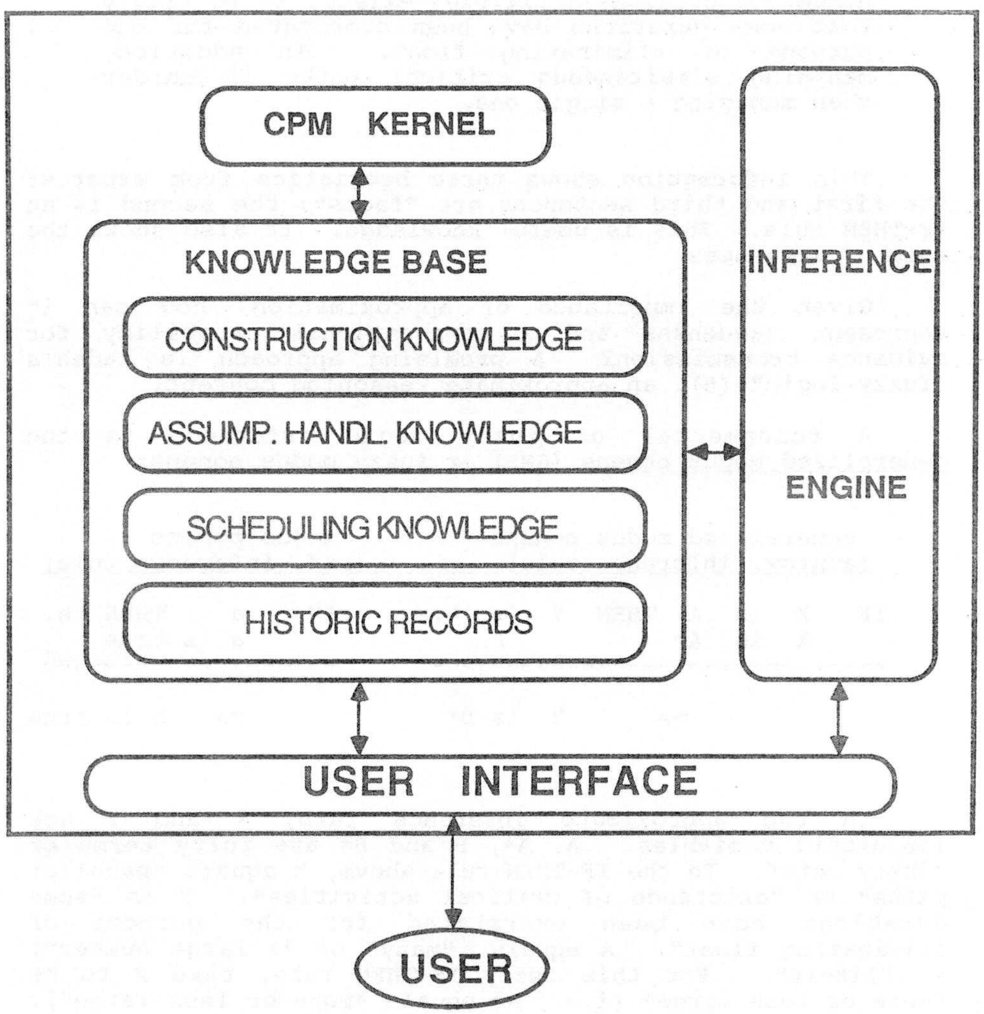
Historic Records Module: This module is part of the second phase of this research work. It contains previously generated schedules, with a description of the generation process.

3.0.2 CPM Kernel

This part of the prototype is procedural in nature. It is one of the demons used by the Scheduling Knowledge Module, charged with performing CPM calculations for the generated schedules. It returns critical paths, floats, start and finish dates. This portion of the computer code is useful for a number of purposes, some beyond the scope of this particular project. Any reader interested in obtaining a copy of the source code can write to the authors.

The entire system is just beginning to take shape. The schematic diagram of Figure 1 shows generally how these different modules come together to make up the entire system. Small case examples are under study at this time to test the system and improve it. Working, beta-level routines are anticipated to be running by late this year.

THE CONSTRUCTION SCHEDULE GENERATOR



3.1 EXTENSIONS: APPROXIMATE REASONING

An expert system knowledge base is facts, rules or heuristics (FRH's) collected from human experts. Because much human knowledge is vague in nature, it would be quicker, economical and easier to describe FRH's with vagueness. Or some degree of certainty. This is true for Those FRH's inherent with uncertainty. The following information regarding construction CPM scheduling from De La Garza (4) is not unusual:

> (Critical path) usually consists of relatively few activities. If many parallel paths and/or a large number of critical activities exists, it is likely that some durations have been overstated for the purpose of eliminating float. In addition, managing simultaneous critical paths is harder than managing a single one.

This information shows three heuristics from experts: the first and third sentences are "facts"; the second is an IF-THEN rule. This is useful knowledge. It also shows the role of vagueness.

Given the importance of approximation, how can it represent vagueness and a computational capability for evidence transmission? A promising approach is Zadeh's "fuzzy logic" (5), an approximate reasoning concept.

A fundamental of fuzzy logic inference is the <u>generalized modus ponens</u> (GMP) or <u>fuzzy modus ponens</u>:

generalized modus ponens (approx. inference rule)	modus ponens (std. inference rule)
IF X is A THEN Y is B.	IF a THEN b.
X is A*	a is true
=> Y is B*	=> b is true

In the approximate inference rule, X and Y are linguistic variables. A, A*, B and B* are fuzzy terms or "fuzzy sets". In the IF-THEN rule above, X equals "parallel paths" or "existence of critical activities". Y is "some durations have been overstated for the purpose of eliminating float". A equals "many" or "a large number"; B, "likely". For this fuzzy IF-THEN rule, take X to be "more or less large" (i.e., A* equals "more or less large").

A conclusion like "Y is more or less likely" (i.e., B* equals "more or less likely") can then be drawn. Chang (6) has algorithms for the GMP.

The above example also shows that use of fuzzy IF-THEN or production rules does not require an exact reasoning match. This eases the user inputs considerably.

To use GMP, all rules should be translated to fuzzy production rule form: "if X is A then Y is B". The following examples are from Chang's PRIORITY RANKING system:

Find degree of susceptibility to rain

rule 1: For an activity j, IF the degree of exposure to rain is **high**, THEN the degree of impact due to rain is **very high.**

rule 2: For an activity j, IF the degree of exposure to rain is **moderate**, THEN the degree of impact due to rain is **high.**

rule 3: For an activity j, IF the degree of exposure to rain is **low**, THEN the degree of impact due to rain is **moderate.**

rule 4: For an activity j, IF the degree of exposure to rain is **zero** (i.e. activity j is not exposed to rain), THEN the degree of impact due to rain is **low.**

rule 5: For an activity j, if the degree of impact due to rain is **high**, then the susceptibility of the activity under the criterion of rain is **high**.

rule 6: For an activity j, if the degree of impact due to rain is **moderate**, then the susceptibility of the activity under the criterion of rain is **moderate.**

rule 7: For an activity j, if the degree of impact due to rain is **low**, then the susceptibility of the activity under the criterion of rain is **low.**

Rules 1 through 4 are relations between degrees of exposure and impact; rules 5 through 7 establish relations between impact and susceptibility. If users input a degree of exposure for an activity, the inference engine will initiate a forward chaining to estimate susceptibility. During that forward chaining GMP algorithms are called to infer a conclusion from each individual rule.

GMP can only be used for knowledge formulable as fuzzy production rules. For other information, such as semantic networks/schemata and facts, different inference mechanisms are needed. That will be part of our future research.

4.0 CONCLUSIONS AND ACKNOWLEDGEMENT

Long-range plans are to take advantage of this previous schedule analysis work and build other program modules essential to construction project control. Clearly, both schedule and cost control need to be on the agenda, as well as schedule generation, perhaps cost estimating and time-cost trade-off analysis. The entire system is called EX-PROCON, an acronym for Expert Project Control.

This work is supported by the U.S. National Science Foundation under Grants No. 8451561 and 8796356. That sponsorship is appreciated.

REFERENCES

1. Levitt, R.E. and J.C. Kunz, "Using Knowledge of Construction and Project Management for Automated Schedule Updating", Project Management Journal, Vol. 16, No. 5, Dec 1983.

2. Gray, C., "Intelligent Construction Time and Cost Analysis" Construction Management and Economics, Vol. 4, 1986. pp. 135-150.

3. De La Garza, J.M. and C.W. Ibbs, "Issues in Construction Scheduling Knowledge", CIB Proceedings, London, 1987. pp. 543-553.

4. De La Garza, J.M., A Knowledge Engineering Approach to the Analysis and Evaluation of Vertical Construction Schedules. Ph.D. thesis, Univ. of Illinois, July 1988.

5. El-Sersy, A. and C.W. Ibbs, "Project Control System Graphics: Their Importance & State", UCB Working Paper.

6. Zadeh, L., "Fuzzy Logic and Approximate Reasoning", Synthese, Vol. 30, Nos. 3/4, April-May 1975. pp. 407-428.

7. Chang, T.C., Network Resource Allocation Using an Expert System with Fuzzy Logic Reasoning. Ph.D. thesis, U.C. Berkeley, Aug. 1987.