A LEAST-COMMITMENT APPROACH TO PLANNING CONSTRUCTION PROJECTS WITH REPEATED CYCLES OF OPERATION

Nabil A. Kartam and Raymond E. Levitt
Department of Civil Engineering
Stanford University
Stanford, California 94305-4020

ABSTRACT

Knowledge representation and reasoning techniques derived from Artificial Intelligence (AI) research permit computers to generate plans, not merely analyze plans generated by human planners. They explicitly represent knowledge about how to generate plans in the form of initial and goal states, descriptions of actions along with their preconditions and effects, and a control structure for selecting new actions to insert into a project plan.

From the more than two dozen AI planners developed and published since the 1960's, we have chosen SIPE (System for Interactive Planning and Execution monitoring) to investigate the utility of AI planners for construction project planning. This paper presents our experience implementing SIPE to plan a multi-story office building construction project with repeated cycles of operation. A least-commitment approach, which aims to delay decisions concerning ordering links and variable instantiations until the system has as much useful information as possible for making them, has proven to be very useful for planning this kind of construction projects. With the use of a frame hierarchy, generic operators, and a least-commitment approach, SIPE is able to generate logically correct activity networks for multi-story building construction from a description of components of a facility. To model such construction projects in a concise and uniform framework, we show the usefulness of some underlying principles for establishing ordering relationships among the project components involved in construction activities.

1. INTRODUCTION

Traditional network-based tools employing CPM and PERT algorithms can help in analyzing a plan, not in generating it. The definition of activities and their predecessors must be provided to such systems by a human planner. Because of this, plan generation has been a challenging area of study for AI researchers since the 1960's. [Tate 85], [Levitt 88], and [Zozaya-Gorostiza 89] provide a thorough review and analysis of previous research on AI planning.

This paper demonstrates the utility of AI techniques for generating construction project plans by presenting our experience implementing SIPE to plan a multi-story office building project with repeated cycles of operation. In the course of this research, we address several issues in automated planning. **First**, we present the notion of employing general principles that can be extracted from experienced human planners and physical laws. Modeling such general principles using proper representation techniques allows the generation of specific construction activities and their sequence. **Second**, we examine the use of generic operators, each of which can be applied to several objects to define specific construction activities. These operators make use of the constraint formulation and satisfaction technique in describing and handling constraints and relationships among the objects that are organized in a frame hierarchy. **Third**, we encounter the problem of developing the least-constrained correct plan, which is a very important problem in construction planning. The desire here is to be able to develop plans that reflect actual ordering constraints among the activities and only those constraints. This is essential to allow scheduling and optimization procedures to achieve the best results. **Finally**, we show how SIPE successfully generates plans for an office building project, and conclude by highlighting the contributions of this research along with a future extension that focuses on generating project-specific knowledge necessary as input for SIPE from an AutoCAD database containing a description of the project and its components.

2. Principles of Ordering Relationships among Activities

Traditional network-based tools that require a complete sequence of activities to develop the project plan can help only in analyzing a plan, not in generating it [Levitt 88]. However, AI provides a variety of knowledge representation techniques that can be used to deduce specific activities and their sequence from more general, abstract knowledge. It is a requirement of knowledge-based plan generation, therefore, to obtain such general knowledge and then to find an appropriate representation technique to model the planning process.

Most ordering relationships among activities are based on underlying principles or physical laws that can be elucidated by an experienced human planner. For example, project components must generally be installed after the elements which provide them with *gravity support.* A series of "dependency principles" of this kind enabled us to capture the causal logic dictating the sequence of the structural activities of the Office Building Project. Another "dependency principle" is that project components, which are *enclosed by* other project components, need to be installed first. We have to make those placements before the enclosure part because we would be unable to enter the enclosed area due to space-time collision. In the Office Building project, we encounter several activities that can be sequenced using this general principle; for example, "under-slab services" must be installed before pouring the slab, and "rough mechanical and electrical services in partitions" must be completed before closing the drywalls, etc. We explored the mechanism of dealing with interaction among activities in SIPE to capture and obtain the correct sequence for the activities.

The above two "dependency principles" illustrate the concept and can be used to generate much of the needed sequence logic for our example project. For those activities not covered by one of these two principles, we have given explicit predecessor activities, such as would be input to conventional sequencing methods based upon CPM or PERT algorithms. Acquiring additional general principles requires more knowledge engineering with experienced human construction planners.

3. Modeling Multi-Story Building Construction

Our example office building is a two story L-shaped concrete building with three bays in the long direction and two bays in the short direction. In this section, we explain how the office building project provides several opportunities for compact and explicit representation using the above general principles within the architecture of SIPE. First, the repetition of elements over bays and floors in the structure provides an opportunity to capture the essential requirements for constructing structural members using very few operators. Second, the similarities of the various structural members allow the use of a frame-based hierarchy to organize such knowledge. Third, the need for gravity support by all structural members provide an opportunity to model and reason about such topological knowledge using a constraint-based approach. Finally, the space-time collision among some of the finishing activities allow the modeling of these activities so that such interaction can be easily detected.

3.1. Tessellation in Structural Components

We know that a multi-story building commonly contains structural tessellations or geometrical repetitions. As a result, many parts of the structure are built using the same core sequence of activities. Looking at most construction buildings, we can see that beams, columns, walls, and decks appear to be laid out in a similar manner across many floors. In a typical multi-story rectangular block structure, each floor of the structure differs only slightly from the other floors with the exception perhaps of the ground floor, mechanical floors, and the topmost floor. In essence if we develop the network of activities required to build a single floor, it might be quite possible, with some qualification, to use similar networks for the construction of the other floors of the building. This obviously saves considerable effort in planning and is exploited by human planners. Even within a single floor, structural components exhibit great uniformity and similarity. We do not have to define a different operator for every beam, column, wall, and deck in the structure such as "construct_beam12_of_floor_1". We want to be able to define generic operators such as "do-beam" and have a planner complete the description with the particular beam and determine which columns have to be constructed before that beam and so forth.

Beams can be constructed any time the columns that support them have been constructed. Here we shall assume that beams have to be supported at both ends rather than cantilevered. Columns can be constructed when immediately supporting lower column and decks have been constructed. This condition is the result of two general principles: (1) the need to provide *gravity support* for structural members (the columns below) and (2) the safety requirement to provide protection against falls (the decks below). Clearly, we are relying on the permanent structural support for the development of these ordering relationships. Under atypical circumstances, which we will not worry about here, it is possible to use temporary structural supports to overcome these construction limitations while still satisfying the *gravity support* principle.

In actual building construction, however, we do not construct in such a piecemeal manner. We prefer to construct large areas at the same time for reasons of economies of scale. We typically construct the scaffolding and formwork and put in the rebar for all of the concrete columns, beams and slabs of a given floor all at the same time. Then we pour the concrete for these structural members area by area. Construction knowledge and experience tell us that we can use the scaffolding for the slab of the next floor to support moving workers and equipment. We recognize these considerations but they are beyond the scope of this particular research. Related projects at Stanford are currently addressing these scheduling issues [Axworthy 89].

3.2. Generic Operators and their Sources of Knowledge

In formulating the input for the structural elements of this project, we can define just five generic operators, as follows:

- Do-CF : to build a column and its associated footing at the ground level.
- Do-Column : to build a column at the first and subsequent floors.

- Do-Beam : to build a beam.
- Do-Deck : to build a deck.
- Do-EW : to build an exterior wall.

All the activities for the construction of the building structure can be described by the application of these five operators to specific elements of the structure. These construction operators are generic in that they can be applied to different individual elements of the structure. When they have been parameterized with the actual structural element we get well defined unique construction activities.

We have chosen these five operators because of the underlying similarities of the activities required for the construction of various elements. For example, we differentiated the construction of ground floor columns from those of the first and second floor because the requirements for the construction of the ground floor are different. The ground floor columns do not need supporting columns beneath them; instead there is a need for the construction of footings to support these ground floor columns. Now since all ground floor columns have this requirement it is possible to define a generic operator "do-CF" that can be applied to the column elements of the ground floor and not those of the other floors. There is no need for a similar differentiation in the construction of beams, decks or walls. Furthermore, this single operator "Do-CF" combines the construction of a ground column with its corresponding supporting footing. We could have broken down this single operator into two operators, namely, "do-Footing" and "do-Ground-Column". However, such breakdown is only enlightening if there are activities we can perform after "Do-Footing" that do not have to wait until "Do-Ground-Column". In this case we have no such activities; so as a convenience we combine the construction of the footing with that of the ground floor column.

Clearly, in order to take advantage of these generic operators, we not only need to be able to instantiate them with different objects, we also need more information. The information comes from the object hierarchy, the world model, and the constraints that we specify in the description of operators. **First**, the object hierarchy contains information about various structural members organized in classes, sub-classes, and instances. Each class and instance can have its own "property" attributes and can inherit further properties from nodes higher in the hierarchy. **Second**, the world model contains information about the topology of the structural members, i.e. their location, encoded as predicates, e.g., the predicate (LOCB B54-1 L5-1 L4-1) describes the location of beam B54-1 by specifying its end points L5-1 and L4-1.

Such project-specific knowledge about structural members and their location are currently provided to the system manually by a user. However, recent research in Computer Aided Design (CAD) and AI [Cherneff 88] looks at ways of generating this knowledge directly from a CAD system and making it readily accessible to an AI planner such as SIPE. A CAD system can capture information at the relationship level that can be used for both design and construction. A number of modern CAD systems can exploit an inheritance and part-of hierarchy. For such systems, only the form of the knowledge has to be made to conform to that needed by a planner or *vice-versa*. If this can be done successfully then it is possible to have even more complete automation of the entire planning process. A user can design the structure on a CAD system and request that a plan be produced automatically for its construction.

Finally, the operators contain information about constraints that should be satisfied for each specific activity. The arguments of these operators are variables that are constrained in two ways. First, by virtue of the variable *name* to ensure that it belongs to the right class or subclass, e.g., beam1 is a variable that belongs to the sub-class of BEAMS. Second, by virtue of the variable *location* to ensure the topology of the structural members. This is achieved by using predicates that specify the location of the structural elements. These building constraints allow the planner to determine which elements should be constructed first and which ones should be constructed later.

3.3. Dealing with Interaction among activities in the Planning Process

In construction, there are many activities in which the preconditions are *thought of* negatively. In order to perform such activities, some preconditions (or "predicates") should be true and some predicates should **not** be true beforehand. The latter are called negative preconditions. Figure 1 illustrates this characteristic with several "coupled" activities from the Office Building Project. Although this notation is seldom used in CPM scheduling, the conceptual grouping of negative preconditions with activities follows human intuition and makes modeling these activities easier to understand and represent. It is also the case that all of these activity couplings result from the space-time collision principle, i.e., the enclosed-by principle. For example, "rough mechanical and electrical services" are enclosed by "drywalls"; therefore, an ordering link is introduced to avoid otherwise harmful interaction between the two activities. Note that we do not need to go down to the level of reasoning about space-time explicitly.

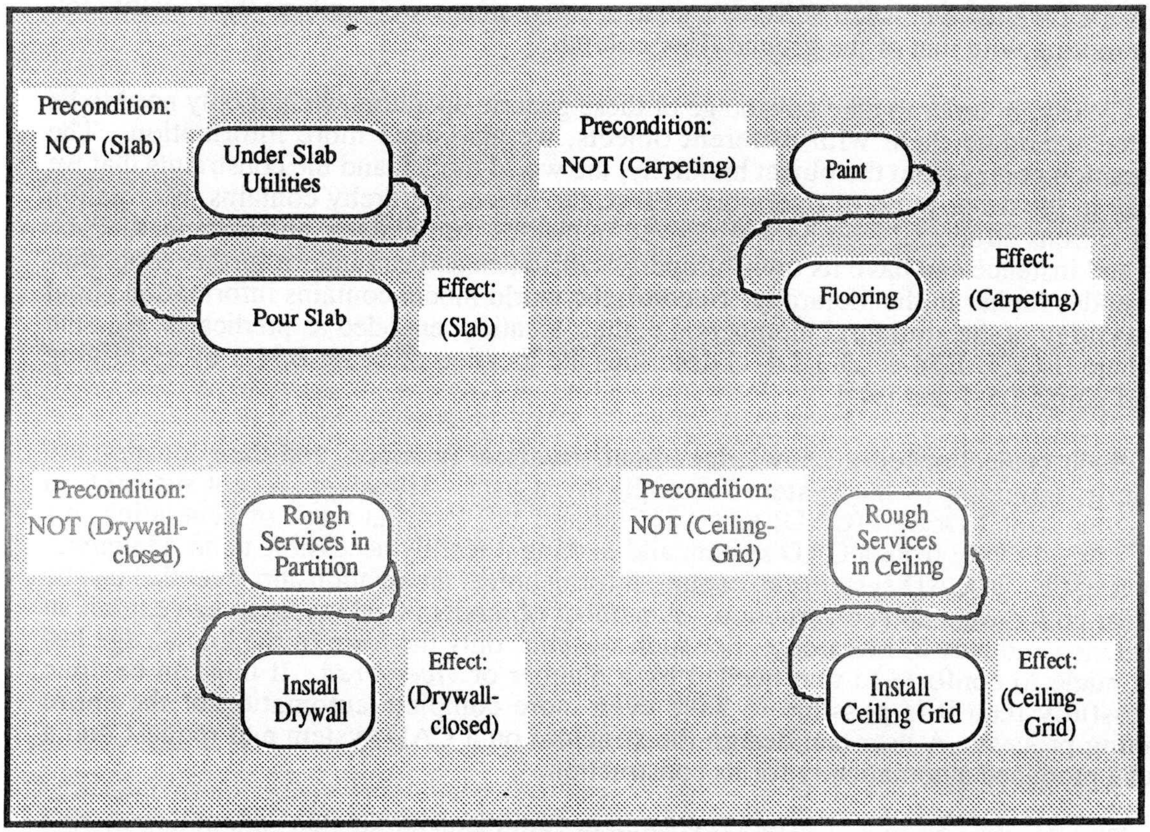

Figure 1: Interaction among activities in the Office Building Project

Furthermore, each of these "coupled" activities is characterized by:

(1) the latter activity does not actually require the former activity as an imperative prerequisite, i.e., the latter activity has no dependency at all on the former one. For example, a slab can be poured regardless of whether under-slab utilities will be installed. This is different from other construction activities that have strong logical dependency, such as the activities that follow one another base on the gravity support principle.

(2) the former activity normally must precede the latter activity; otherwise the former activity would be much more difficult to perform. For example, to install under-slab utilities after the slab has been installed would require extra tasks, e.g., destroying part of the slab, then installing the utilities, and finally pouring concrete to cover gaps. Clearly, this additional work not only would result in extra time and a higher cost for the project, but it also might adversely affect the quality of work performed. To avoid these detrimental effects, the planner should recognize such interaction among activities and consequently introduce the proper ordering sequence.

The most natural semantics for these two types of interaction is this: If a negative precondition is required for an activity then the condition must not hold when the activity is scheduled to be performed; on the other hand, if a positive precondition is required for an activity then the condition must hold when the activity is scheduled to be performed. Since we do not represent negative information in the world model we use the closed world assumption of negation as failure. If we cannot find its positive assertion in the database we assume that the desired negative precondition is satisfied. It is important to understand the consequences of this assumption because we may need the same proposition as a positive precondition for one activity and as a negative precondition for a different activity. For example, the predicate "Ceiling Grid Installed" is a positive precondition for the action "Install Suspended Ceiling" but a negative precondition for the action "Install Mechanical and Electrical Services in Ceiling".

4. Planning the Office Building Project

Now that we have seen how this problem is modeled and the rationale for it we move on to specifying the goals that we would like the planning system to achieve. When dealing with SIPE we can specify our goal to be the completion of any part of the project, i.e., not necessarily the whole project. This added flexibility allows the different parties in a project to obtain plans of how to achieve their own tasks. In this project, we have specified several goals to be solved, such as:

- construct a specific deck
- construct the first floor structure
- construct the structure of the project
- complete the finishes of the first floor
- construct the whole project (structure and finishes)

SIPE produced plans in their least-constrained, correct form for each of the goals specified in the Office Building project. We solved these goals using both the automatic and interactive search options in SIPE along with graphical interface.

5. Conclusion

In general, construction activities result in the accumulation of project components. Occasionally, activities result in result in removal of objects or components, e.g., removing formwork after pouring concrete, removing scaffolding, demolishing a pre-existing facility, etc. This generally additive nature of construction makes the process of formulating and manipulating construction planning easily handled by an AI planner since it is intuitive and natural to represent operators in this formalism.

A multi-story structure such as an office building has dozens of beams, columns, walls, and decks. We have illustrated how few operators are needed to plan for such a problem, if we can use constraints and take advantage of object hierarchies already defined on drawings or in CAD data files. We do not have to define an operator for each construction activity but can use generic operators. However, we still need means to differentiate among the activities. The tasks of differentiating among, e.g., the activities of constructing the different beams, and of resolving the ordering relationships among construction activities for the elements of the structure, depend on the use of constraints. SIPE has a powerful constraint language that allows us to do this.

In this project, only five operators are used to generate the plan for the structure, and another fifteen operators for the mechanical, electrical, and finishing activities of the building. Which beam and which column should be constructed first or next have been resolved by the planner with the help of constraints. The construction of a five-story or a ten-story building could be planned with the same operator information although we would of course have to update the object-hierarchies, in particular, the part-of hierarchy. Note that only the leaf nodes of this inheritance hierarchy would need to change from project to project — the types or classes do not need to change much. Much of the information can be reused from project to project. This demonstrates clearly the power of using a small number of generic operators in conjunction with constraints and the project-specific component data stored in frame hierarchies.

Furthermore, research at Stanford and MIT in CAD and AI is focusing on generating such project-specific knowledge directly from a CAD system and making it readily accessible to an AI planner. Presently, the authors are involved in such a research effort aimed at generating the frame-hierarchy and the world model knowledge necessary as input to SIPE from an AutoCAD system. In fact, it might conceivably be possible to automate the construction processes further. One might even add methods for facilitating scheduling and project control allowing for the full integration of all four main processes: design, planning, scheduling, and control. The advantages are widespread availability of valuable specialized knowledge, more efficient construction performance, early design decision making informed by construction impacts, and fewer errors in design and construction planning.

ACKNOWLEDGMENT

This research has been funded by the Stanford Construction Institute, the Stanford Center for Integrated Facility Engineering, and the National Science Foundation, grant # MSME-87-16608. Their support is gratefully acknowledged.

The authors wish to acknowledge the contribution of Dr. David Wilkins of SRI International for his extensive support in extending SIPE to meet the requirements of the construction domain. The authors also gratefully acknowledge the valuable inputs that they have received from several colleagues in computer science and civil engineering: Boyd Paulson, Paul Teicholz, John Fondahl, Edward Feigenbaum, Earl Sacerdoti, Austin

Tate, and Mark Drummond. Special thanks also to Professors Thomas Binford and Jean-Claude Latombe of the Robotics Laboratory at Stanford for providing us with access to their Symbolics machines.

REFERENCES

Axworthy, A. and Levitt, R. 1989. "An Object-Oriented Approach for Scheduling of Construction Projects," in the *Proceedings of the Sixth International Symposium on Automation and Robotics*.

Bremdal, B. 1988. *An Investigation of Marine Installation Process - A Knowledge-Based Planning Approach,* Institute for Marine Prosjektering, Trondheim.

Carr, R. and Meyer, W. 1974. Planning Construction of Repetitive Building Units. ASCE *Journal of the Construction Division.* **100** (CO3), pp. 403-412.

Chan, W. and Paulson, B. 1988. Exploratory design using constraints. *Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing* **1**(3), 59-71.

Chapman, D. 1987. Planning for Conjunctive Goals. *Artificial Intelligence* **32**, 333-377.

Cherneff, J., Logcher, R., and Sriram, D. 1988. *Automatic Generation of Construction Schedules from Architectural Drawings.* MIT, unpublished paper.

Currie, K. and Tate, A. 1988. *O-Plan: the Open Planning Architecture*, Artificial Intelligence Applications Institute, University of Edinburgh, UK.

DeLaGarza, J. and Ibbs, W. 1987. Issues in Construction Scheduling Knowledge Representation. *Proceedings of the CIB W-65 Symposium*, London, 543-553.

Kartam, N. 1989. *"Towards a Foundation for Evaluating Artificial Intelligence Planners,"* T.R. 8, Center for Integrated Facility Engineering, Stanford University, California.

Kartam, N. and Levitt, R. 1988. "An Analysis of Artificial Intelligence Planning Systems," in the *Proceedings of the Third IASTED International Conference*, Expert Systems Theory and Applications.

Levitt, R., Kunz, J., and Kartam, N. 1987. Using Artificial Intelligence Techniques for Automated Planning and Scheduling. In the Proceedings of *The Fourth International Symposium on Robotics and Artificial Intelligence in Building Construction,* 27-52.

Levitt, R., Kartam, N. and Kunz, J. 1988. Artificial Intelligence Techniques for Generating Construction Project Plans. ASCE *Journal of Constr. Engineering and Mgmt..* **114** (3), pp. 329-343.

Logcher, R. 1987. *Adding Knowledge Based Systems Technology to Project Control Systems.* Department of Civil Engineering, M.I.T., unpublished paper.

Navinchandra, D., Sriram, D., and Logcher, R. 1988. GHOST: Project Network Generator. ASCE *Journal of Computing in Civil Engineering.* **2** (3), pp. 239- 254.

Sacerdoti, E. 1975. The Non-linear Nature of Plans. *IJCAI*, 206-214, Tbilisi, USSR.

Stefik, M. 1981. Planning with Constraints. *Artificial Intelligence* **16** (2), 111-140.

Tate, A. 1977. Generating Project Networks. *IJCAI*, 888-893, Boston, Massachusetts.

Wilkins, D. 1988. *Practical Planning: Extending the Classical AI planning Paradigm.* Morgan Kaufmann, Los Altos, Calif.

Zozaya-Gorostiza, C., Hendrickson, C., and Rehak, D. 1989. *A Knowledge-Based Process Planning for Construction and Manufacturing,* Academic Press.