Automation and Robotics in Construction XVI © 1999 by UC3M

Knowledge Learning System for Slurry Wall Construction

Ren-Jye Dzeng[1] and Yi-Shan Ho[2]

[1]*Associate Professor, National Chiao-Tung University, Taiwan*
[2]*Former Graduate Student, National Chiao-Tung University, Taiwan*

rjdzeng@cc.nctu.edu.tw, 8516501@cc.nctu.edu.tw

Abstract: This research proposes an inductive learning model for acquiring the knowledge about the pre-construction planning of a slurry wall system. The model is suitable for the engineering problems where human experts rely on past experience, rule-of-thumbs, or subjective judgement. The system can be used to facilitate human experts to solve a new problem at hand. It can also be used to generalize the knowledge of existing cases and so the knowledge can be kept and shared by other engineers.

Keywords: slurry wall construction, planning, induction, artificial intelligence, generalization

INTRODUCTION

Knowledge acquisition is the transformation of problem solving expertise from knowledge sources such as human expert, text, data and documents to a human or computer program (Buchanan et al 1983). People are interested in acquiring knowledge because (1) it may help solve major problems of today and tomorrow with technological solutions; and (2) it may satisfy certain needs by filling the gaps between what currently is and what should be tomorrow (Modesitt 1992).

Knowledge acquisition has also been identified as a key bottleneck in the development of a knowledge-based system (KBS) (Barr and Feigenbaum 1981). Knowledge acquisition is an expensive process, and good knowledge engineers are hard to find. In addition, engineering knowledge sometimes has the nature of being dynamic, unstable, subjective, incomplete, and conflicting.

Slurry wall technology has been used as an independent construction approach or in conjunction with ground control techniques for the temporary support of deep excavations or/and as part of the permanent structure. Slurry walls have been characterized as a water resistant and high strength design, and their construction as a low noise and low vibration construction method. The increased building density in the urban areas has led to a proliferation of using slurry wall systems as temporary support of deep excavations or part of a permanent foundation structure. The advantages of slurry walls include feasibility of deeper construction, reduced and more controllable risk of disturbance and damage to buildings, absence of noise and vibrations, reduced disruption of surface activities and minimum surface restoration, and often fastest construction time (Xanthakos 1994).

The pre-construction planning of a slurry wall system requires the determination of trenching equipment, panel length, etc. Optimal engineering decisions involve complex geotechnical (e.g., groundwater and soil chemistry), design (e.g., wall bracing, settlement, anisotropy), and site (space availability, existing utilities, transportation) considerations, complex calculations (e.g., finite element), and trial-and-error repeated processes.

This research proposes an inductive learning model for acquiring the knowledge about the pre-construction planning of a slurry wall system. The model is suitable for the engineering problems where human experts rely on past experience, rule-of-thumbs, or subjective judgement. The model is implemented on a Windows 95 PC platform and comprises several design cases of slurry wall panels. The system can be used to facilitate human experts to solve a new problem at hand. It can also be used to generalize the knowledge of existing cases and so the knowledge can be kept and shared by other engineers.

TYPICAL CONSTRUCTION PROCEDURE FOR SLURRY WALLS

Slurry walls are prepared from the surface by excavating a vertical trench supported by a slurry instead of bracing to the required depth of a wall. The structure is constructed in the trench by the simultaneous extrusion of the supporting slurry. This slurry not only provides the stability of the trench, but might also become the final structure. A typical construction procedure of a slurry wall comprises four execution stages: (1) excavation, (2) insertion of steel tubes, (3) placement of reinforcement cage, and (4) concrete placement.

Excavation

The first stage is to excavate a linear trench using one or more excavating equipment passes. Fig. 1 shows a typical three-pass excavated trench using clamshells. Numbers indicate the excavation sequence. The first pass begins away from the last concreted panel to give extra time for the concrete to set. As the trench is excavated, the excavated soil is replaced immediately by a suitable bentonite slurry to provide trench stability. The length of trench prepared in one cycle of operation is called a **panel**. The average panel length is about 5 m for three passes, 7.5 m for five passes, and 10.5 m for seven passes (Xanthakos 1994).

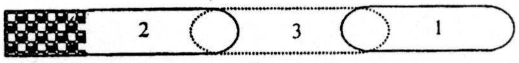

Figure 1 Typical excavation sequence of the slurry wall construction

Insertion of Steel Tubes

On completion of the excavation, the next stage is to insert a round steel tube, called a stop-end tube, to form the panel joint with the adjacent panel. By connecting several panels, a continuous diaphragm wall can be made.

Placement of Reinforcement Cage

The third stage is to fabricate a reinforcement cage and assemble on the ground according to the structural requirements, including stiffness necessary for its lifting. The cage is then lowered into the trench by suitable equipment and fastened.

Concrete Placement

Fresh concrete is continuously poured into the trench using tremie pipes and the supporting slurry is simultaneously forced out of the trench due to concrete's gravity. The slurry is pumped into a storage area for reconditioning and reuse, and the stop-end tube is gradually withdrawn after a suitable concrete setting time.

The determination of panel length is a fundamental decision for an optimal slurry wall system. Efficiency is maximized if the panel length is optimized in terms of equipment passes. In practice, a tentative panel length is first selected to accommodate trench stability and concreting requirements, and then is compared with the range of excavating systems and checked for other considerations that may govern. The panel length is not finalized until the construction phase and sequence has been established (Xanthakos 1994). In general, it is advantageous to have a wall constructed in long units, which reduce the number of vertical construction joints and may result in less water seepage.

LITERATURE REVIEW

This paper proposes a system named Inductive Knowledge Acquisition System (IKAS), which is an expert system that solves problems and learn based on related experience. This section reviews expert systems and machine learning.

Expert System

A typical expert system comprises a knowledge base and an inference engine. Some expert systems may comprise a knowledge acquisition and/or an explanation facility. Knowledge engineering is a critical process in the development of an expert system. It includes knowledge acquisition, knowledge representation, and knowledge inference.

Knowledge acquisition is a process of acquiring problem-solving knowledge from human experts, literature, computer files, and other knowledge sources. Knowledge acquisition has been recognized by researchers as the kry bottleneck in the development of expert systems [Feigenbaum 1981]. Knowledge acquisition is an expensive process and good knowledge engineers are hard to find. In addition, engineering knowledge sometimes has nature of being dynamic, unstable, subjective, incomplete, and conflicting. Machine knowledge can be acquired from interviews with human experts, literature, and observations. Knowledge acquisition can also be automated with or without interactions with human.

To be understood by a computer, knowledge can be represented as rules, semantic networks, frames, logic statements, objects, scripts (which includes actor, action, and objects), or cases.

There are two types of knowledge inference approaches: forward chaining (i.e., event-driven reasoning) and backward chaining (i.e., goal-driven reasoning). To reduce the search space and increase the reasoning efficiency, several search strategies such as depth-first, breadth-first, hill-climbing, and best-first are available.

Machine Learning

Several learning strategies are available. Examples are rote learning, deductive learning, inductive learning, learning from observation, learning from analogy, case-based reasoning, and neural networks-parametric learning (Arcisaewski 1992). Inductive learning is a suitable approach for this research because cases with applicable solutions are available and the problem solution can be described by a simple representation.

ID3 (Quinlan 1986, 1987) and STAR (Michalski 1983) are two inductive learning algorithms that are commonly used in the field of artificial intelligence. ID3 algorithm generalizes a set of examples and represents the result as a decision tree, where each branch layer represents a problem attribute, each node an attribute value, and each leaf a solution node.

The STAR algorithm generalizes a set of positive examples (i.e., examples whose solutions match the goal of interest) and negative examples (i.e., examples whose solutions do not match the goal of interest). It tries to create STAR statements, which can cover all positive examples but cannot cover any of negative examples. These statements describe the conditions of examples that lead to the target solution.

SYSTEM ARCHITECTURE

Overview

The architecture of the IKAS is illustrated in Fig. 2. During a regular operation, the user specifies a new problem using the IKAS graphic user interface and determines appropriate settings for the problem solving mechanism such as the correctness threshold and cases screening criteria for the induction process. Based on the new problem and the setting, the system selects only the cases that meet the specified criteria and performs the induction reasoning. The induction result represents the dynamic knowledge and is expressed by rules. During this stage, the user may inspect, modify, or delete the rules. If necessary, the dynamic rules can also be integrated into static knowledge base. The problem solving mechanism combines the static knowledge and dynamic knowledge, and presents the final recommendation. If the system cannot reach a final recommendation, rules that lead to different recommendations are noted so that the user may make the decision. For the maintenance purpose, the system manager may periodically expand the case library, and review the static knowledge base and make modifications.

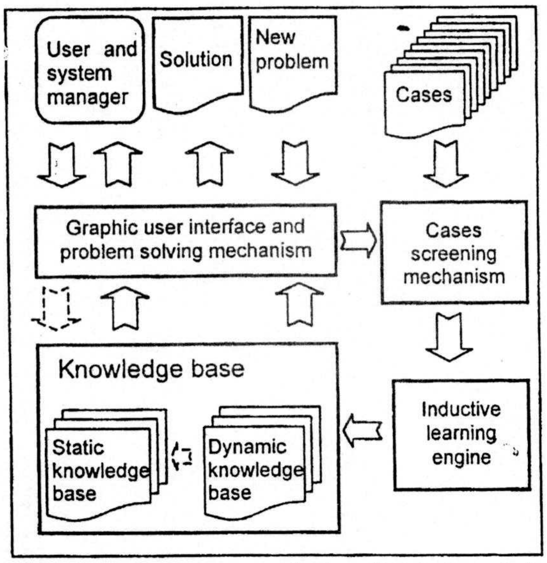

Figure 2 Architecture of the IKAS knowledge learning system

Graphic User Interface

The IKAS is implemented on the Microsoft Windows 95 platform using Microsoft Visual Basic 5.0 and Microsoft Access 97. The main menu comprises 6 choices: *File*, *Case Library*, *Knowledge Base*, *Setting*, *Window*, and *Help*. *File* allows a user to create a new problem and exit the system. *Case Library* provides functions to browse, search, add, delete, and modify the cases. *Knowledge Base* provides the functions to browse, add, delete, and modify the static rules. *Setting* allows the change of options such as searching with or without goal, correctness threshold, and popularity threshold. *Window* switches between available windows. *Help* provides on-line user manual.

To determine appropriate panel lengths for a slurry wall construction project, the IKAS user inputs the information by following a series of dialogs. The first dialog is the Project Dialog, which includes 16 attributes (e.g., *project name*, *project location*, *space availability for reinforcement assembly*). These attributes are used to represent a case and are detailed in the next section.

The second dialog allows the user to set up the query for searching appropriate cases based on from which induction rules will later be derived. As shown by the two file tabs of the dialog, the user may input the query for each attribute one by one, or using SQL (Structured Query Language). The SQL is a standard database query language and is suitable for the user who is familiar with the language or intends to specify a complicated query. The query shown in the dialog asks the system to search for the cases whose type of soil is either 'clay' or 'sand', and soil bearing capacity value is below 20.

The system starts the induction process once the user sets up the query and presses the 'Induce' button in the dialog. The induction result is represented by the rules that are grouped into 4 categories: static knowledge base for panel unit, and dynamic knowledge base for panel unit.

Case

Each slurry wall construction project is represented by a case, which is a set of attributes that describe the project and the solution, i.e., the panel length. The following lists these attributes, and their value types (i.e., string, keyword, and value) and descriptions.

Project-ID-number (string)

The unique indexing number assigned to the project, e.g., "A0001".

Project-name (string)

The name of the project, e.g., "ABC Office Building".

Project-location (keyword)

The site location of the project, which can be chosen from a list of cities and counties in Taiwan; e.g., 'Taipei City'.

Number-of-floors-of-superstructure (number)

Number of floors above the ground; e.g., 16.

Number-of-floors-of-substructure (number)

Number of floors under the ground; e.g., 4.

Basement-area (number)

The size of the area enclosed by the slurry walls, expressed in m^2; e.g., 200.

Space-availability-for-reinforcement-assembly (keyword)

Space availability for assembling reinforcement cage, which can be 'ample' and 'constrained'. Within the capacity of lifting equipment, it is more efficient to assemble the whole cage on the ground and place it with a single lift and without splicing.

Capacity-of-cleaning-plant (number)

Capacity of the cleaning plant that separates the slurry from the soil particles mixed with it during the excavation operations, expressed in m^3/hr. The cleaning of slurry can be effected by sedimentation, a vibrating screen, or a cyclone.

Storage-capacity-of-dumped-soil (number)

The size of the storage area of excavated soil, expressed in m^3.

Soil-type (keyword)

The type of soil, which can be 'clay', 'sand', and 'gravel'.

Soil- bearing-capacity-factor (number)

The soil bearing capacity factor.

Ground-water-level (number)

The location of the level of ground water, expressed in m.

Wall-depth (number)

The depth of slurry wall panel, in m.

Wall-width (number)

The width of slurry wall panel, in cm.

Required-splice-length (number)

The required splice length between male and female reinforcement cage, expressed in m.

Type-of-excavating-equipment (keywords)

Type of equipment used to excavate the trench for slurry walls; e.g., 'MHL 60100' and 'MHL 80120'

Male-panel-length (number)

The length of male panel, expressed in m.

Female-panel-length (number)

The length of male panel, expressed in m.

KNOWLEDGE BASE

The knowledge base stores the knowledge that can be used to solve a new problem. The IKAS chooses to use rules to represent such knowledge because they are easy to understand and modify. Based on whether the rules will change or not each time the system is initiated, the knowledge is divided into two categories: static and dynamic knowledge bases. The static knowledge base stores the rules created by the end user or the system manager, which can be updated by them. The dynamic knowledge base contains the rules created by the system during the run time. Each time the system is initiated, dynamic knowledge is created from the scratch based on the cases that are pertaining to the new problem at hand. Thus, this part of knowledge may be different each time if the cases in the system and the new problem at hand are different.

Inductive Learning

Traditional inductive learning applied in the common sense learning (e.g., identify geometric shape) usually performs generalization based on the entire examples that are available. In this research, an 'example', termed a 'case' here, comprises more attributes. To reduce the search space and reasoning time, the IKAS allows its user to screen cases before they are generalized by specifying cases screening criteria. For example, one may ask the system to select cases whose *soil-type* is either 'clay' or 'sand', and *soil-bearing-capacity-factor* is smaller than 20.

The foundation of the IKAS's inductive learning is based on the STAR algorithm (Michalski 1983). The following uses simplified cases to describe the algorithm. Assume each case comprises only five attributes: *project-ID-number*, *soil-type*, *wall-width*, *wall-depth*, and *panel-length*, where the last attribute represents the solution to the problem. Table 1 lists the attribute values of five cases.

Table 1 Attribute values of five simplified cases

project-ID-number	***soil-type***	***wall-width* (cm)**	***wall-depth* (m)**	***panel-length* (m)**
A001	clay	80	30	4.0
A002	clay	100	45	3.6
A003	clay	100	43.5	3.6
A004	sand	80	40	3.6
A005	clay	80	38	4.0

The algorithm represents the knowledge of each case as a statement (called STAR statement). For example, case A001 can be represented as a STAR statement, which comprises an IF statement and a THEN statement as follow:

(*soil-type* = 'clay') and (*wall-width* = 80) and (*wall-depth* = 30) ⇨ (*-panel-length* = 4.0), which can be simplified as follow:

('clay', 80, 30) ⇨ (4.0) STAR-1

STAR-1 states that for a site whose soil type is clay, the designed panel width for the slurry wall is 80 cm, and the slurry wall length is 30 m, the recommended panel length is 4.0 m.

Suppose we try to find all case conditions that will lead to a final decision of 4.0-m long panel. Cases whose panel lengths are 4.0 m are called **positive cases** (e.g., cases A001 and A005), and others are **called negative cases** (e.g., cases A002, A003, and A004). The goal of the reasoning is to find a STAR statement that includes all positive cases (called **completeness condition**), but does not include any of negative cases (called **consistency condition**).

If an initial STAR statement cannot satisfy both completeness and consistency conditions, it has to be either generalized or specialized. For example, STAR-2 and STAR-3 are two statements generalized from STAR-1. STAR-2 states that for a site whose soil type is clay, and the designed panel width for the slurry wall is 80 cm, the recommended panel length is 4.0 m. STAR-3 states that for a site whose soil type is either clay or sand, the recommended panel length is 4.0 m.

('clay', 80, all) ⇨ (4.0) STAR-2

({'clay', 'sand'}, all, all) ⇨ (4.0) STAR-3

If a case satisfies a STAR statement, it must satisfy all of its generalized statements. However, a case that satisfies a generalized statement does not necessarily satisfy the original statement from which the statement is generalized. Thus, if a STAR statement cannot satisfy the completeness condition, it must be further generalized until it includes all positive cases.

Specialization is a reverse process of generalization. For example, STAR-2 is a specialized statement of STAR-3; STAR-1 is a specialized statement of STAR-2. If a case satisfies a statement, the case does not necessarily satisfy its specialized statement. However, if a case satisfies a specialized statement, it must satisfy the original statement from which the statement is specialized.

Following the previous example, the most generalized statement is (all, all, all), which cannot be further generalized. STAR-1 is an example of the most specific statement, which cannot be further specialized. The STAR algorithm attempts to find the most generalized statement that includes all positive cases but exclude all negative cases.

Search Strategy

Inductive learning is a search problem. Based on (Mitchell 1982), search strategies can be classified into two categories: data driven and goal driven. Depth-first, breadth-first, and version space search strategies are examples of data driven search strategies. General-to-specific and specific-to-general are examples of goal driven search strategies. Table 2 compares these five search strategies.

Based on Table 2, the version space strategy seems to be a good strategy. However, the search path of the strategy is longer in general, and the collections of specific and general cases tend to be large. Thus, the reasoning process will require more time and computing memory. In addition, the case in this research comprises attributes whose values are continuous, which make the strategy difficult to reach the convergence of the specific and general spaces.

Table 2: Comparisons of five search strategies

Strategy / Item	Depth-first	Breadth-first	Version Space	G to S	S to G
Backtrack	Yes	No	No	Yes	Yes
Re-testing	Yes	Yes	No	Yes	Yes
Required memory	little	medium	much	little	little
Search starting point	most specific	most specific	both	most general	most specific
Search space	smaller	larger	larger	smaller	smaller

The IKAS uses both general-to-specific and breadth-first search strategies. The system starts from the most general STAR statement and continues to specialize the statement(s) until all negative cases are excluded. When all negative cases are excluded from the statements in memory, check each statement and remove statements that do not include all positive cases.

Our strategy has the following characteristics:

1. Compared to the version space strategy, the search path is shorter and the number of STAR statements kept in memory is fewer in general.
2. Backtrack is not necessary and each case is only checked once.
3. All STAR statements that satisfies the search goal will be found.
4. The strategy removes a statement only when at least one negative case proves it is wrong. Thus, the resulting statements may include incorrect statements but will not miss any correct statements. It has the advantage of possibly providing good information that human experts never thought of. Incorrect statements can be removed through user's inspection.

Noise

Noise in this research represents cases whose solutions are not satisfactory. Because the cases collected in the real life often contain noises, a useful result is hard to come out if the STAR algorithm must rigidly meet the completeness and consistency conditions. To remedy this problem, the completeness condition is considered to be met if some threshold conditions are met. However, the consistency condition still has to be rigidly met. We use the correctness rate and coverage rate as two thresholds, which are commonly used in the field of signal detection. The following defines two terms.

For any STAR statement S:

$$\text{Correctness Rate} = \frac{NP_{(S)} + NN_{(S)}}{NP + NN} \qquad \text{Formula-1}$$

$$\text{Coverage Rate} = \frac{NP_{(S)}}{NP} \qquad \text{Formula-2}$$

$NP_{(S)}$: the number of positive cases included in S
$NN_{(S)}$: the number of negative cases excluded from S
NP: the number of positive cases
NN: the number of negative cases

The IKAS allows the user to adjust the thresholds for the correctness and coverage rates. For any STAR statement to be legitimate, both rates must exceed the thresholds (i.e., the completeness condition is met) and the consistency condition is met.

Induction Goal

The IKAS allows induction learning with or without a specific goal. The purpose of learning with a specific goal is to solve a particular problem at hand. On the other hand, learning without a specific goal is to induce a set of general knowledge about each type of panels.

Use the five cases shown in Table 1 as an example. Without a specific goal, the resulting STAR statements are:

For the 3.6m long panel: (all, 80< ≤100, all), (all, all, 38< ≤45).

For the 4.0m long panel: ({sand, gravel}, 80< ≤100, all), ({sand, gravel}, all, 30< ≤43.5), (all, all, 30≤ <40).

Assume the goal is to solve a new problem: ('gravel', 80, 40), and the thresholds for both correctness rate and coverage rate are 0.4. A STAR statement (all, all, 38< ≤45) is found for the 3.6 m long panel. No STAR statement is found for the 4.0m long panel. Therefore, the recommend length for the new problem is 4.0 m.

EXPERIMENT

The IKAS' database currently includes more than 20 real life cases about slurry wall construction. These projects are located in the northern part of Taiwan (i.e., Taipei and Hsinchu areas).

Based on the experiment result, the following is our initial findings:

1. The learning time required by the IKAS is an exponential function of the number of negative cases.
2. The learning time of the IKAS decreases as the number of positive cases increases.
3. The number of rules induced by the IKAS increases as the required learning time increases.

CONCLUSION

This research proposes an inductive learning model for construction knowledge. The model is suitable for the engineering problems where human experts rely on past experience, rule-of-thumbs, or subjective judgement. The design of slurry wall panels is one of such example. The model is implemented on the Windows 95 PC platform and currently includes more than 20 cases of slurry wall construction. The system can be used to facilitate human experts to solve a new problem at hand. It can also be used to generalize the knowledge of existing cases and so the knowledge can be kept and shared by other engineers.

REFERENCES

Arcisaewski, T. and Ziarko, W. 1992. "Machine learning in knowledge acquisition", *Knowledge Acquisition in Civil Engineering*, ed. Arciszewski, T. and Rossman, L.A., ASCE, pp.50-68.

Barr A. and Feigenbaum, E.A. 1981. *The Handbook of Artificial Intelligence*, Vol. 1, William Kaufman Inc., Los Altos, California.

Buchanan, B.G., et al. 1983. "Constructing an expert system", *Building Expert Systems*, Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., ed., Addison-Wesley.

Dzeng, R. J. and Tommelein, I. D., "Case-Based Scheduling Using Product Models", *Proc. of 2nd Computing Congress on Computing in Civil Engr.*, ASCE, Washington, D.C. , 1995.

Feigenbaum, E.A. 1981. "Expert systems in the 1980s", *State of the Art Report on Machine Intelligence*, Bond, A. ed., Pergamon-Infotech.

Forsythe, D.E. and Buchanan, B.G. 1989. "Knowledge acquisition for expert systems: some pitfalls and suggestions", *IEEE Transaction on System, Man, and Cybernetics*, Vol. 19(3), pp. 435-442.

Hoffman, R.R. 1987. "The problem of extracting the knowledge of experts from the perspective of experimental psychology", *AI Magazine*, Vol. 8(2), pp.53-67.

Kenneth L. M., 1992. "Basic Principles and Techniques in Knowledge Acquisition", *Knowledge Acquisition in Civil Engineering*, American Society of Civil Engineers, pp.11-49.

Michalski, R.S. 1983. "A Theory and Methodology of Inductive Learning", *Machine Learning: An Artificial Intelligence Approach*, Vol. I.

Mitchell, T.M. 1982. "Generalization as search", *Artificial Intelligence*, Vol. 18, pp.203-226.

Quinlan, J.R. 1986. "Induction of decision trees", *Machine Learning*, Vol. 1, No. 1, pp.81-106.

Quinlan, J.R. 1987. "Simplying decision trees," *Inter. J. of Man-Machine Studies*, No. 27, pp.221-234.

Sestito, S. and Dillon, T. S., 1994. *Automated Knowledge Acquisition*, Prentice Hall, New York.

Xanthakos, P.P. 1994. *Slurry Walls as Structural Systems*, 2nd ed., McGraw-Hill, New York.