CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Selection and Design of Temporary Retaining Walls Using Expert Systems

by

Naoki Ikoma

TECHNICAL REPORT
Number 73

September, 1992

Stanford University

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Summary: CIFE TECHNICAL REPORT Number 73

Title: Selection and Design of Temporary Retaining Walls Using Expert Systems

Author: Naoki Ikoma

Date: September 1992

Funding: Naoki Ikoma was supported by Kumagai Gumi Co., Ltd, Japan.

Abstract:

There are many projects that need Temporary Retaining Walls (TRW). Special techniques are needed to design and select these walls. Specialists (experts) should design and select the Temporary Retaining Walls. There are not so many specialists. They are always very busy. Sometimes they can not work perfectly because of a lack of time. This situation is very dangerous. To design and select TRW speedily, effectively, and correctly is very important. To do so successfully brings success to projects.

Expert systems are absolutely needed for selection and design of TRW. There are many barriers to producing an expert system. There are very few examples. Trying to produce expert systems has just started recently. No concrete concepts exists of a whole system for selection and design of TRW because there are no regular procedures to identify them. It always depends on the conditions of each site.

The concepts of a whole system are considered in making useful expert systems. For wall type selection, the real procedure is flexible, fluid, and ambiguous. It was very hard to create wall type selection expert systems. Some concrete ideas are proposed. A fuzzy theory is applied to realize this idea. A prototype model is based on the ideas. Critical problems to produce expert systems are identified.

Subject:

Retaining walls are structures used to provide stability for earth or other material where conditions disallow the mass to assume its natural slope. They are usually used to hold back or support soil and water. The target of this Expert System is <u>Temporary</u> Retaining Walls (TRW). That is to say, that the wall is not permanent but temporary. When the excavation work is finished, the wall is removed or buried. A strong or expensive wall is not necessary for the temporary structure. An inexpensive yet sufficiently strong wall is required. An expert system, which can design and select the best TRW for each site, is going to be produced.

Objectives/Benefits:

On almost all excavation work, TRW are necessary. There are many unknown factors in the ground. Design and selection of TRW are very difficult. Only specialists can do these jobs. There are so many sites that request TRW especially in Japan. Specialists are always very busy because of high demands. Sometimes such jobs can not be done perfectly because of the lack of specialists. Errors in the design or selection of the proper type of TRW can cause serious problems. After walls are erected and excavation work begins, they can not be changed. If some problems do occur, there is no way to solve them except by adding other assisting construction methods. These normally involve significant cost and time. Much time is necessary for these efforts during which the excavation work must be stopped. These situations can be dangerous and very costly. To avoid such dangerous and ineffective situations, a development of expert systems for design and selection of TRW is strongly needed.

Methodology:

To make a TRW Design Expert System, there are some problems. As there are few general procedures for the design and selection of walls, there are no concepts for the whole systems. There are very few examples of such systems, and especially, there are no concrete ideas for the selection of wall types. It is very hard to fix rules for programming these concepts. Three points are proposed in solving the problems mentioned above: (1) to fix the concept of the whole TRW Design Expert System, (2) to propose an idea for the selection of walls to make a good expert system, and (3) to identify the critical problems to make a TRW Design Expert System. A prototype model for the selection part of walls is produced by using Fuzzy Set Theory.

Results:

The whole system should (1) check necessity of walls for the site, (2) select wall types, (3) design walls, and (4) check the safety, cost, and duration. Design functions are comparatively easy to program because the method to calculate design parameters is known. For the other functions, especially selection of wall types, expert systems are desirable. There are few established rules to select walls. Expert systems can help engineers to select walls. New ideas (Advantage Table Method, Fuzzy System, and so on) to select walls are proposed in this paper. An Advantage Table is formed based on design experience that every specialist has acquired. It is very easy to modify these tables to reflect new experience and knowledge. These ideas will improve progress in future research. Even if the scale of a wall can not be decided easily (i.e., small scale walls or large scale walls), the best wall is selected easily by using Advantage Table and Fuzzy System. The critical problems to make a TRW Expert System is a shortage of specialists' knowledge. There are no formalized knowledge for this. To obtain this information could help to produce an excellent TRW Expert System.

Research Status:

The production of such kinds of expert systems has only just recently started . More research needs to focus on this problem. Therefore this research provides a new chart for future research. In order to find out more practical concepts of the wall system, it is necessary to interview more experts and to coordinate their knowledge. Every expert should have his own knowledge such as an Advantage Table. This table will be able to be modified for real usage to coordinate knowledge from other experts. How to decide weight coefficients and scores of rankings on an Advantage Table is a very important issue. Some rules based on knowledge of experts are necessary to decide weight coefficients and scores of rankings. The way to determine Membership Functions of the Fuzzy system should become clearer to realize this system for real usage. In a case of over 30 meter's excavation depth, this system does not deal with that now. For the future, some expert systems based on new ideas will be able to help us to design such huge walls. There are often special conditions unique to specific sites. There is a possibility of neglecting to consider such conditions in other parts of this system. Such conditions should be checked according to safety, cost, and duration. An expert system is needed for this part. To produce such a part of an expert system should be part of a future study.

TABLE OF CONTENTS

Selection and Design of Temporary Retaining Walls Using Expert Systems

Naoki Ikoma[1]

Abstract

There are many projects that need Temporary Retaining Walls (TRW) [see Index]. Special techniques are needed to design and select these walls. Specialists (experts) should design and select the Temporary Retaining Walls. There are not so many specialists. They are always very busy. Sometimes they can not work perfectly because of a lack of time. This situation is very dangerous. To design and select TRW speedily, effectively, and correctly is very important. To do so successfully brings success to projects.

Expert systems are absolutely needed for selection and design of TRW. There are many barriers to producing an expert system. There are very few examples. Trying to produce expert systems has just started recently. No concrete concepts exists of a whole system for selection and design of TRW because there are no regular procedures to identify them. It always depends on the conditions of each site.

The concepts of a whole system are considered in making useful expert systems. For wall type selection, the real procedure is flexible, fluid, and ambiguous. It was very hard to create wall type selection expert systems. Some concrete ideas are proposed. A fuzzy

[1] Kumagai Gumi Co.,Ltd. Tokyo, Japan. c/o The Center for Integrated Facility Engineering, Stanford University, CA 94305 - 4020

theory is applied to realize this idea. A prototype model is based on the ideas. Critical problems to produce expert systems are identified.

1. Introduction

There is not so much land in Japan. Land space is at a premium. In the construction world, projects to extend living space are very important. One such development project is called GEO-FRONT [see Index]. The GEO-FRONT is becoming more and more popular. On the sites of GEO-FRONT, there is almost always excavation work. This is work digging into the ground to build structures. Work in the ground is very difficult because of the many unknown factors within the ground itself. In order to do excavation work successfully, problems about unknown factors should be solved.

On almost all excavation work, TRW are necessary. As mentioned above, there are many unknown factors in the ground. Design and selection of TRW are very difficult. Only specialists can do these jobs. There are so many sites that request TRW. For example, it is reported in a certain technical jounal for civil engineering (SEKOU 1) that a certain general construction company in Japan showed many examples of sites that requested TRW (SAWADA 1). In this jounal, 76 projects were introduced in only 5 years. All 76 projects were introduced as an example of projects where there are TRW which are difficult to build, design, or select. Specialists are always very busy because of high demands for their expertise. Sometimes such jobs can not be done perfectly because of a lack of specialists. Errors in the design or selection of the proper type of TRW can cause serious problems. After walls are erected and excavation work begins, they can not be changed. If some problems do occur, there is no way to solve them except by adding other assisting construction methods. They normally involve significant cost and time. For example, when a selected TRW is too weak to hold lateral earth pressures, it is necessary to strengthen the wall using additional steel, timber, or other methods. Much time is necessary for these efforts during which the excavation work must be stopped. These situations can be dangerous and very costly.

To avoid such dangerous and ineffective situations, a development of expert systems for the design and selection of TRW is strongly needed.

The target of this Expert System is temporary retaining walls. That is to say, that the wall is not permanent but temporary. When the excavation work is finished, the wall is removed or buried. A strong or expensive wall is not necessary for the temporary structure. An inexpensive yet sufficiently strong wall is required.

In order to make expert systems for TRW, there are many problems. At first these problems should be identified in order to solve them. There are many excellent techniques to make expert systems in computer science. The development of an expert system for TRW should include these excellent techniques.

2. Purpose

In Japan, almost every site has TRW. There are almost always large numbers of temporary retaining wall jobs in the Construction Design Department.

Each wall is unique because of a large combination of factors. There is few simple way to select an appropriate wall for each site. In order to select a wall that is appropriate to the site, extensive information, techniques, and experience of experts are necessary. Many other factors must be considered. These factors should include design, cost, period, labor, equipment, site conditions and so on. Only experts can select the best wall effectively. Experts are few and usually busy. There is a shortage of experts because of an extensive use of retaining walls.

Considering the above mentioned factors, and to minimize the time and work effort, selecting an appropriate TRW is very important at construction sites. Selecting an appropriate temporary retaining wall insures safety and minimal cost.

To do so, an expert system for TRW is needed.

The purpose of such a TRW Design Expert System should be as follows:

(1) Accelated design of TRW,

(2) Accumulate data about TRW designs,

(3) Reduce cost of work,

(4) Make consistently high quality designs, and

(5) Reduce reliance on TRW experts.

To make a TRW Design Expert System, there are some problems. As there are no general procedures for designing and selecting walls, there are no concepts for a whole system. There are very few examples of such systems, and especially, there are no concrete ideas for selection parts. It is very hard to fix rules of programming for these parts.

These three points are important in solving the problems mentioned above:

(1) To fix the concept of the whole TRW Design Expert System,

(2) To propose an idea for the selection part of walls to make a good expert system, and

(3) To identify the critical problems to make a TRW Design Expert System.

3. Temporary Retaining Walls Design Expert System

3.1 Background

Before mentioning the details of TRW Design Expert Systems, examples of designing and selecting of walls by specialists should be mentioned. Such examples are mentioned in this section.

Some TRW examples of projects in Japan (SAWADA 1) are described in Table 3-1

Table 3-1 TRW's project

		Condition of Ground		Scale		
project name	location	soil	ground water	depth	Type of Walls	Timbering
Rinkoukansen High Way	Yokohama	sand, mud rock	GL -1.5m	20m	steel sheet pile, pipe sheet pile	tieback anchor
Hokkaido Newspaper Bld.	Kushiro	alluvial sand	GL -3.0m	7.8m	steel sheet pile	tieback anchor
Sapporo Tokeidai Bld.	Sapporo	sand, gravel	GL -9.0m	12.1m	soldier beam and breastbord	tieback anchor
Mizusawa City Hall	Mizusawa	sand, gravel	GL -1.4m	8.5m	continuous underground wall	tieback anchor
Akita Shoping Center	Akita	poor ground		6.9m	soldier beam and breastbord	brace
Utunomiya East Post Office	Utunomiya	gravel	GL -2.0m	6.0m	steel sheet pile	brace
Urawa Station Bld. A	Urawa	diluvium	GL -6.0m	17.9m	column type under ground wall	brace
Funabashi City Hall	Funabashi	poor ground	GL -2.0m	10.4m	continuous underground wall	tieback anchor
Daiichikangin Head Office	Tokyo	poor ground	GL -2.0m	25.0m	continuous underground wall	brace
N S Building	Tokyo	alluvial sand	GL -7.5m	21.2m	continuous underground wall	brace
Hazama Building	Tokyo	diluvium	GL -7.0m	22.5m	continuous underground wall	brace
Shinjuku Nomura Bld.	Tokyo	diluvium		27.7m	soldier beam and breastbord	brace
Kawasaki Nikkou Hotel	Kawasaki	poor ground	existence	14.1m	steel sheet pile	brace
Okura Hotel Niigata	Niigata	alluvial sand	GL -1.4m	9.5m	continuous underground wall	brace
Kitakyusyu City Hall	Kitakyusyu	alluvial clay	GL -3.0m	12.3m	column type under ground wall	brace
			GL = Ground Level			

From the examples of projects listed above, a detailed example of Number 1 is described below.

(1) Project Number 1

The floor plan of this site is shown in Figure 3-1. This site is located in Yokohama City in Japan. This project was one of the projects for MINATO-MIRAI 21 [see Index]. This project was very huge. On this site, a submarine tunnel was built. It should be built under the sea. The site was surrounded by a huge TRW to protect the site from the sea. The site has approximately two types of TRW: (1) double cofferdam [see Index] by steel pipe sheet pile and (2) steel sheet pile [see Index]. Both walls are supported by tie backs (earth anchors) [see Index]. The site is surrounded by the sea from three directions with a high ground water level. Cutting off the water with the walls is absolutely necessary .

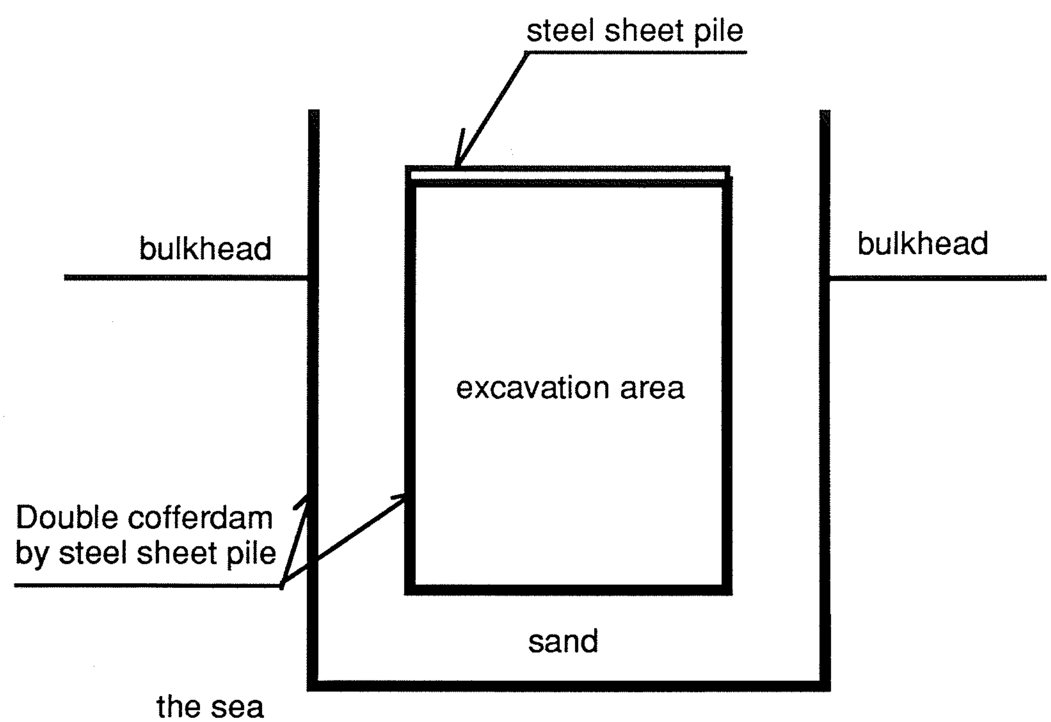


Figure 3-1 The floor plan of the site

The main factors to be considered for designing and selecting of a TRW are as follows:

(1) The site is huge, the area of excavation is wide, and the depth of excavation is deep;

(2) The duration of the project is short;

(3) The site is surrounded by the sea;

(4) Much ground water is in the ground;

(5) The base of the ground is mud rock -- the level is not flat but hard and inflexible; and

(6) There is confined ground water under sand and mud rock layers.

In accordance with these factors as mentioned above, many characteristics of walls are needed to select and design suitable TRW as mentioned below:

(1) Large scale walls are needed. (In this case, the design of walls might be unique).

(2) Quick excavation work is needed. An open space for the excavation area is very useful to dig quickly. Tieback anchors, for supporting the walls, might be useful to keep a wide open space for the excavation area.

(3) Cutting off the water by the walls is absolutely necessary. An additional construction method might be necessary,

(4) Placing walls into the mud rock is difficult. Walls which can be placed into the mud rock should be selected. Additional construction methods might be necessary to place the walls. When placing walls, every wall should be checked for penetration into the ground base with enough penetration depth because of the unflatness of the bases' level.

(5) Boiling [see Index] and heaving [see Index] should be avoided. Additional construction methods might be necessary.

Mainly these factors were considered in designing and selecting TRW on the sample site. There are pros and cons about the results of the designing and selection of a TRW. These walls on the sample site are described in the next paragraph.

Selected walls are (A) and (B) on this sample site.

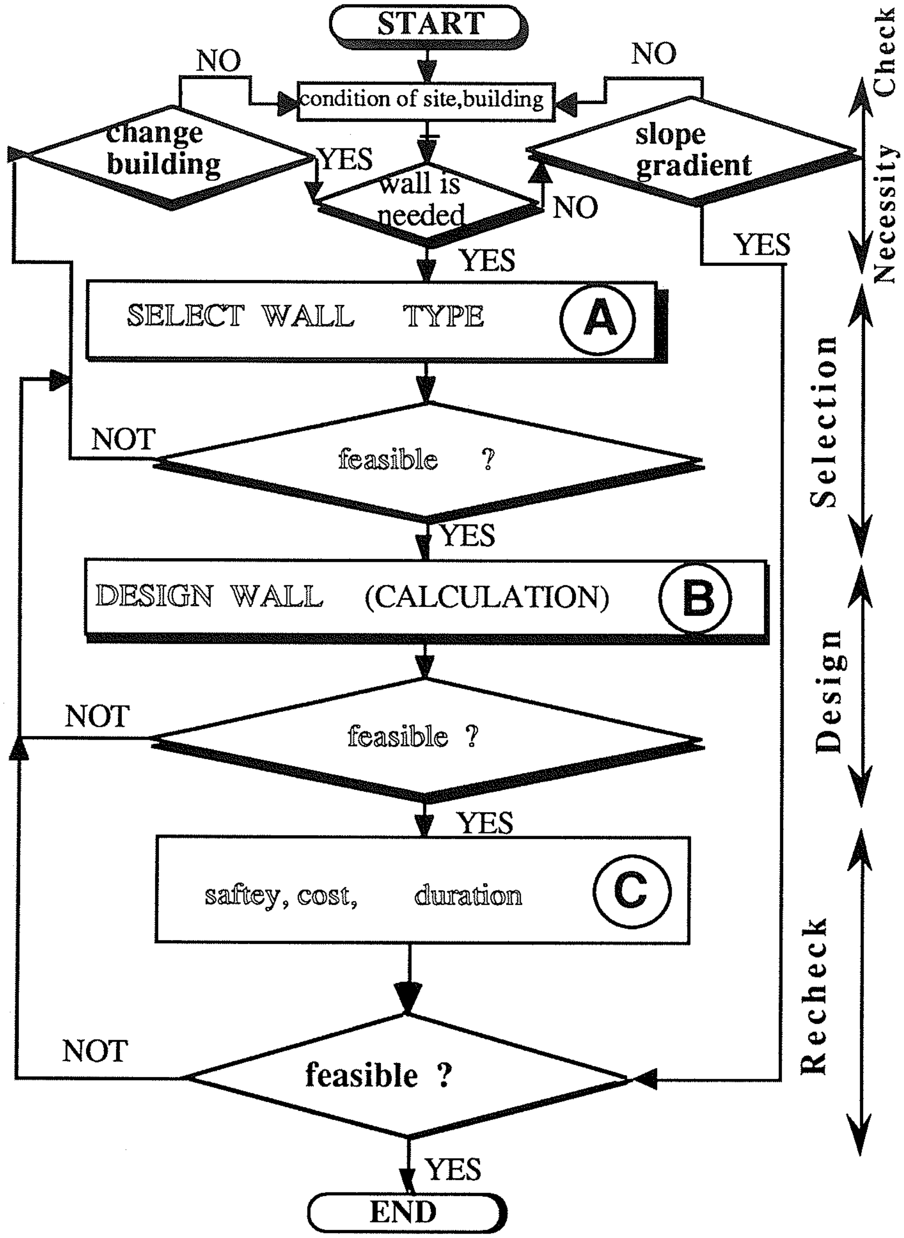
(A) Double cofferdams made by steel pipe sheet piles supported by tieback anchors, with additional methods (Deep Wells):
Pro-side--Double cofferdams are good at cutting water. Steel pipe sheet piles are strong. Tieback anchors are good for quick excavation.
Con-side--There are no examples of large double cofferdams. There are unknown factors for tieback anchors. Steel pipe sheet piles are hard to place into the ground. Many additional construction methods are needed.

(B) Steel sheet piles supporting by tieback anchors, with additional methods (Deep Wells):
Pro-side--Steel sheet piles are easy to place into the ground and easily cut off water. Tieback anchors are good for quick excavation. There are many examples of these walls.
Con-side--Large bending occurs with steel sheet piles. To avoid large bending, many tieback anchors are needed.

By the steps that are described above, many specialists designed and selected these TRW over a long time period without expert systems being employed.

3.2 Concept of the Whole System

The concept of the whole system is shown in a flow chart in Figure 3-2.



(A) For example, <u>a sheet pile</u> is selected as the <u>type</u> of wall.

(B) For example, <u>a type 5L</u> is selected as the <u>class</u> of sheet pile.

(C) After consideration of safety, cost, duration, the best wall is selected.

Figure 3-2 Concept of the Whole System

To consider and arrange a procedure of excavation work, the concept of the whole system for the TRW Design should comply with the flow. The flow consists of four parts:

(1) Necessity check of a wall for the site

In this part, the system considers if a wall is necessary for the excavation work based on the condition of the site or building. If the wall is necessary, the system flows to the next step. If not, the excavation work might be possible without using a temporary retaining wall; and the system flows to the step for the design of a slope in an open cut excavation. For example, slope stability analysis, check of slope erosion, slope protection work and so on.

(2) Selection of wall types

The details are mentioned in Chapter 3.3. Most of this part relies on the experience of experts (specialists). There are no established rules in selecting wall type and method of placing. The selection is dependent on the expert's experience and ability to analyze the site condition. A good selection brings success of design, execution, cost, and duration. Selection methods are vague and are done case by case; selecting the appropriate wall has been difficult and time-consuming. Considering the points mentioned above, a formalization of the procedure of walls selection is necessary. An expert system, based on that formalized procedure, is certainly required to be produced to select good walls type for each site.

(3) Design of wall

After selecting candidates of the best wall type in the selection part of the system, classes of these selected walls types are determined in this part as shown in Figure 3-2. For example, when a steel sheet pile is selected as a candidate for the best wall in the selection part of the system, a class of the steel sheet pile is determined like a Type III in this part of the system. What is called--design of wall, especially structural design--should be done at this step. Except in special cases, the design method is already established in accordance with several public standards.

Normally the calculation procedure is done as shown below:

- By surveying the ground, several parameters of soil are determined (i.e., friction angle, unit weight, cohesion [see Index] and so on);
- Penetration depth is calculated by using active (and passive) earth pressure and stability moment to determine the entire length of the wall (Appendix I); and
- In accordance with lateral pressure, stress and bending moment that occur on the wall are calculated. Class and material of the wall that has reasonable section modulus and arrangement of struts are determined.

We already have sufficient software applications to calculate stress, shear stress, and moment of the wall or struts in accordance with design standards. These are not user friendly. Normally, a user can discover the appropriate wall class only by a trial and error method. The trials continue till the result meets design standards.

Expert systems can help users by assigning weighted values to parameters. If such an expert system can be created, it will be very helpful in establishing rapid designs. The problem with existing software application is that it is too cumbersome. In order to arrive at results quickly, there is a "steep learning curve." If a user friendly system is implemented, the possibility of fast and consistently high quality designs will increase.

The goal is not to develop a user-friendly wall design system. What is required to be done is to construct an expert system that can select candidates of walls to be used for the design system. The details of that system is mentioned in Chapter 3.3.

(4) Recheck of safety, cost, and duration

Safety, cost, and duration of construction--to use the wall candidates that are determined by the result of design parts--are checked in this last step. There are often special conditions unique to specific excavation sites. It is sometimes difficult to check such conditions in the selection steps and so on. There is a possibility of omitting to consider such factors about the special conditions. Checking is required at this stage. Essentially an expert system is needed for this part, but development of such a system is not part of this study. It should be part of a future study.

3.3 Selection of Wall Types

3.3.1 Introduction

Metropolitan Expressway Public Cooperation in Japan uses a standard for temporary structures (MEPC 1). This standard is very popular and useful especially for the design of heavy construction works. It was modified in October 1990. In this standard, there is a flow chart that shows the selection procedure for TRW, shown in Figure 3-3.

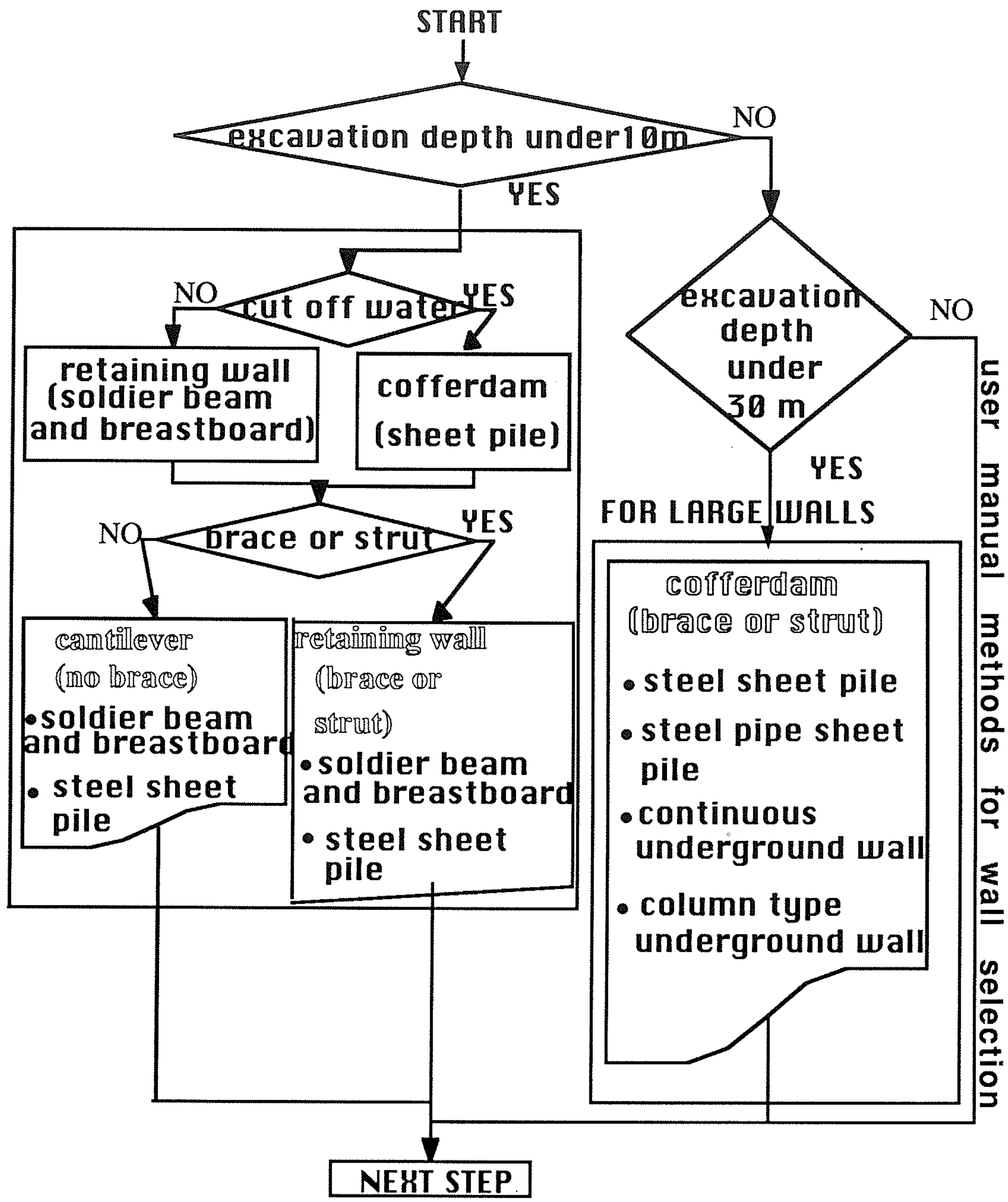


Figure 3-3 Procedure of Walls Selection by Metropolitan Expressway Public Cooperation

Based on the standard for temporary structures (MEPC1), the selection procedures of TRW are distinguished by the scale of excavation depths. Excavation depths are divided into three groups as described:

- Group 1 (Excavation depth less than 10m),
- Group 2 (Excavation depth larger than 10m and less than 30m), and
- Group 3 (Excavation depth larger than 30m).

In Group 1, walls selected used only IF-Then rules as shown in the flow chart of Figure 3-3. These rules are very clear and comparatively easy to be created for an expert system. These walls in this Group 1 are called "small or middle class of walls".

In Group 2, the method of walls' selection depends on the condition of sites. There are various procedures for the selection. These walls in this Group 2 are called "large middle class of walls".

In Group 3, wall's scale is fairly large. A special method is required for each site's condition to design walls.

Basically, the selection of walls type of the system follows this idea. For the selection, it is not necessary to follow a particular standard, but in this stage, this standard is the easiest one to apply to produce the expert system. The idea of this standard is very clear and similar to real selection procedures used by specialists.

Considering the existing state of things, Group 3 involves unique conditions and should be handled directly by specialists. It is too complicated for an expert system. The selection of walls in Group 1 (especially the small class of walls) almost can be done by using only If-Then rules. The key is the selection of walls in Group 2. For this selection, knowledge and experience of specialists are necessary. Selecting walls by If-Then rules is generally too complicated. It can not be used effectively in an expert system.

For the selection in Group 2, a new idea is proposed. A procedure based on the new idea shown in Figure 3-4.

		merits, demerits, and characteristics
wall type	A	
	B	
	C	

		evaluation items				
		ground	cost	duration	–	weight coeffcient
wall type	A	○	×	○	–	1
	B	○○	○	×	–	0
	C	×	○	○○	–	1
Total point		3	5	2	–	

○○ high mark
○ middle mark
× low mark

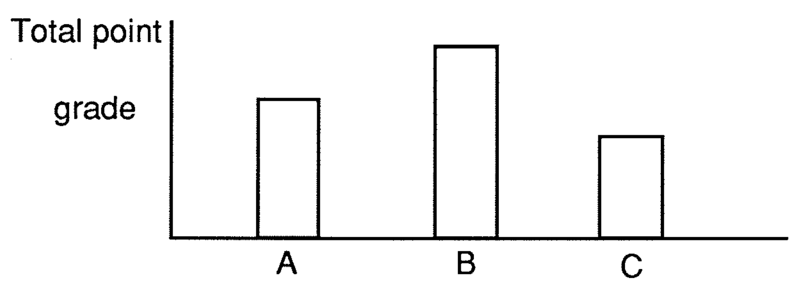


Figure 3-4 Procedure of Walls Selection Parts

Arrangement of wall characteristics, merits, and demerits, for each type of wall is shown in Table 3-2.

Table 3-2 Characteristics of each wall type

condition / type of wall	characteristics (merits and demerits)
soldier beam and breastboard	• no cut off water. • not good for poor ground. • not good for much ground water. • great displacement. • economic.
steel sheet pile	• good for cut off water. • noise and vibration for placing. • settlement. • great displacement. • economic.
steel pipe sheet pile	• good for cut off water. • great stiffness. • little displacement. • good for large walls. • noise and vibration for placing. • no reuse. • impossible to remove.
continuous underground wall (R.concrete)	• good for cut off water. • great stiffness. • little displacement. • good for large walls. • little noise and vibration for placing. • long duration is necesarry. • wide space is necesarry. • no reuse. • impossible to remove.
column type underground wall (R.concrete)	• great stiffness. • little displacement • little noise and vibulation for placing • impossible to remove.

Based on the table shown in Table 3-2, each wall type is ranked to each evaluation characteristic shown in Table 3-3.

Table 3-3 Advantage Table (example)

		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	0
	sandy soil	○○	○○	○	△	△	0
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	10
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	1
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	0
	width (large area)	○○	○○	○	○	○	0
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	1
REUSE		○	○○	×	×	×	1
PERIOD		○○	○○	×	×	○	0
COST		○	○	×	×	×	1
	TOTAL	11	42	25	35	17	
EVALUATION RANK	**SIGN**	**SCORE**	**YOUR CHIOCE**				
BEST	○○	**high mark**	3				
BETTER	○	**middle mark**	2				
Possible but not preferred	△	**low mark**	1				
Disadvantage	×	**negative mark**	0				
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
	TOTAL	11	42	25	35	17	

The table in Table 3-3 is labeled "an Advantage Table". Total points for each wall type are calculated by the Advantage Table. Candidates for the best type of wall are determined by the total points. The details about the Advantage Table are mentioned in Chapter 3.3.2.

Each group (Group 1, Group 2, and Group 3, as mentioned above) should have a unique Advantage Table. In this case, the selection from each Advantage Table is done separately. By comparison, real world selection procedures done by specialists are more dynamic. They are not limited to specific groups, but consider a broad range of alternatives. To incorporate this dynamic, Fuzzy Theory [see Index] is being used. The details will be mentioned in Chapters 3.3.3 and 3.3.4.

3.3.2 Advantage Table Method

For the prototype model, the Advantage Table that is shown in Table 3-3 has been implemented. That table basically follows an example of the Japan Society of Civil Engineers. In the table, weight coefficients are set. By using this table, weight can be put on important evaluation items, and unimportant evaluation items can be ignored according to specific site conditions. Also in that table, scores for evaluation ranks are set. Basically, these weighted coefficients and scores should be determined by specialists. By using the prototype model, these weighted coefficients and scores can be formalized to fix their values comparing results from the system and real examples from actual sites. This will be left for future work.

3.3.3 Fuzzy System

What is Fuzzy Theory? By the references (FUZZY 1, FUZZY 2, FUZZY 3, FUZZY 4), Fuzzy Theory is briefly explained as follows:

"Something ambiguous that is felt by a human's sense can be numerically expressed by using Fuzzy Theory", [FUZZY 1].

This theory is applied to the system. As mentioned in Chapter 3.3, three groups for wall grades [Index] are set in accordance with the depth of excavation. As described in Figure 3-5 and Figure 3-6, using distinct groups makes it easy to design an expert system, and specialists do not separate their knowledge of groups so clearly. When they are selecting the best wall type for large types of walls, actually they are also using their knowledge of small types of walls as well. This is fuzzy logic.

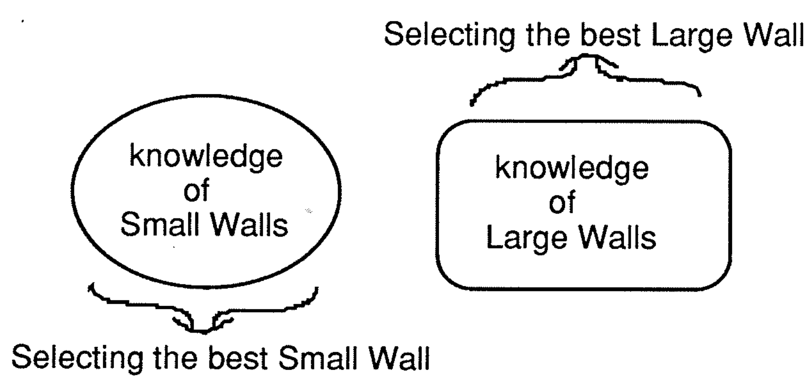


Figure 3-5 Selection by distinct knowledge groups

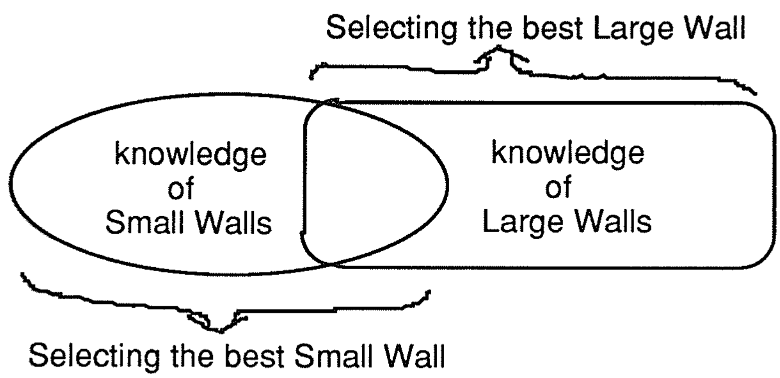


Figure 3-6 Selection by both knowledge groups

As described in Figure 3-7 and Figure 3-8, wall depths, concrete ranges, (i.e., 0-10 meters, 10-30 meters, etc.) are set. In real situations, the ranges are fluid, flexible, continuous, and ambiguous.

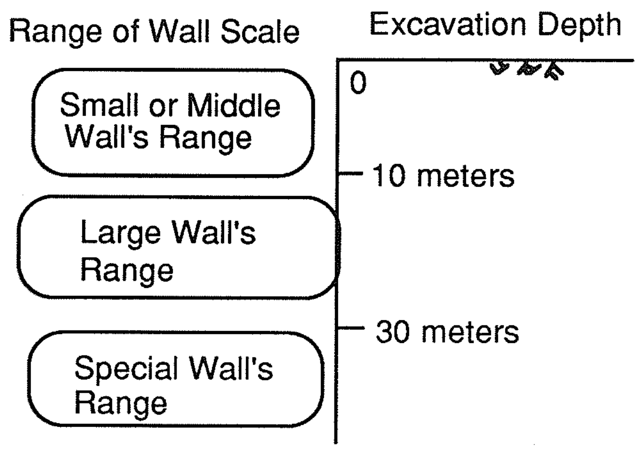


Figure 3-7 Distinct Ranges of Wall Scale

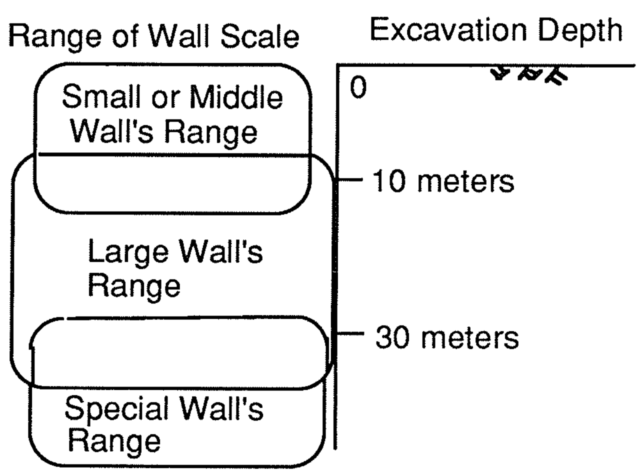


Figure 3-8 Real Ranges of Wall Scale

To solve these problems, Fuzzy Theory is applied as shown in Figure 3-9A and Figure 3-9B. To apply this Fuzzy Theory, Mr. Koichi Yufu[2] assisted the author considerably.

---

2 Koichi Yufu: Information Systems Office, Kumagai Gumi Co.,Ltd, Tokyo, Japan

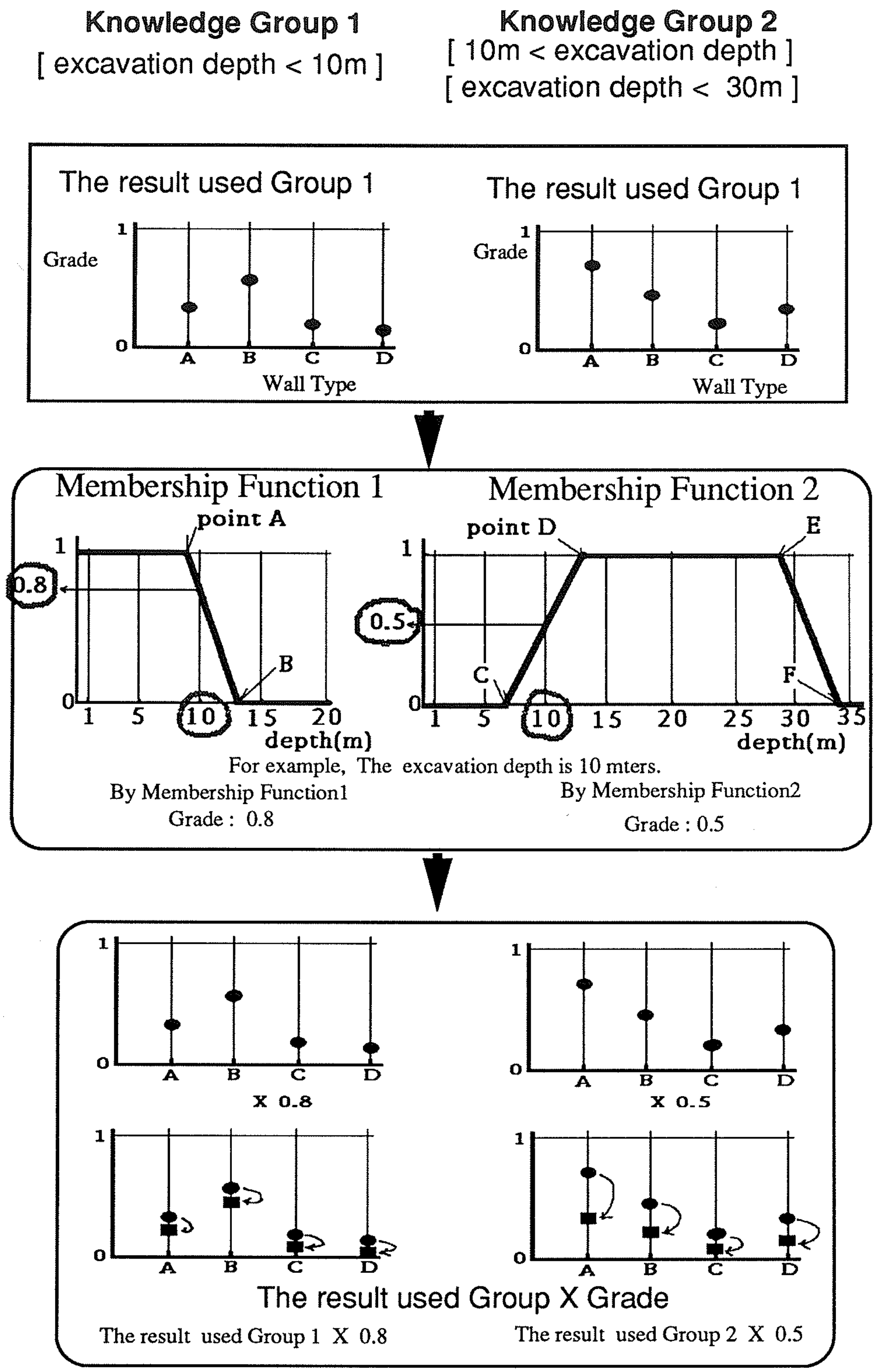


Figure 3-9A Applied Fuzzy Theory to the System

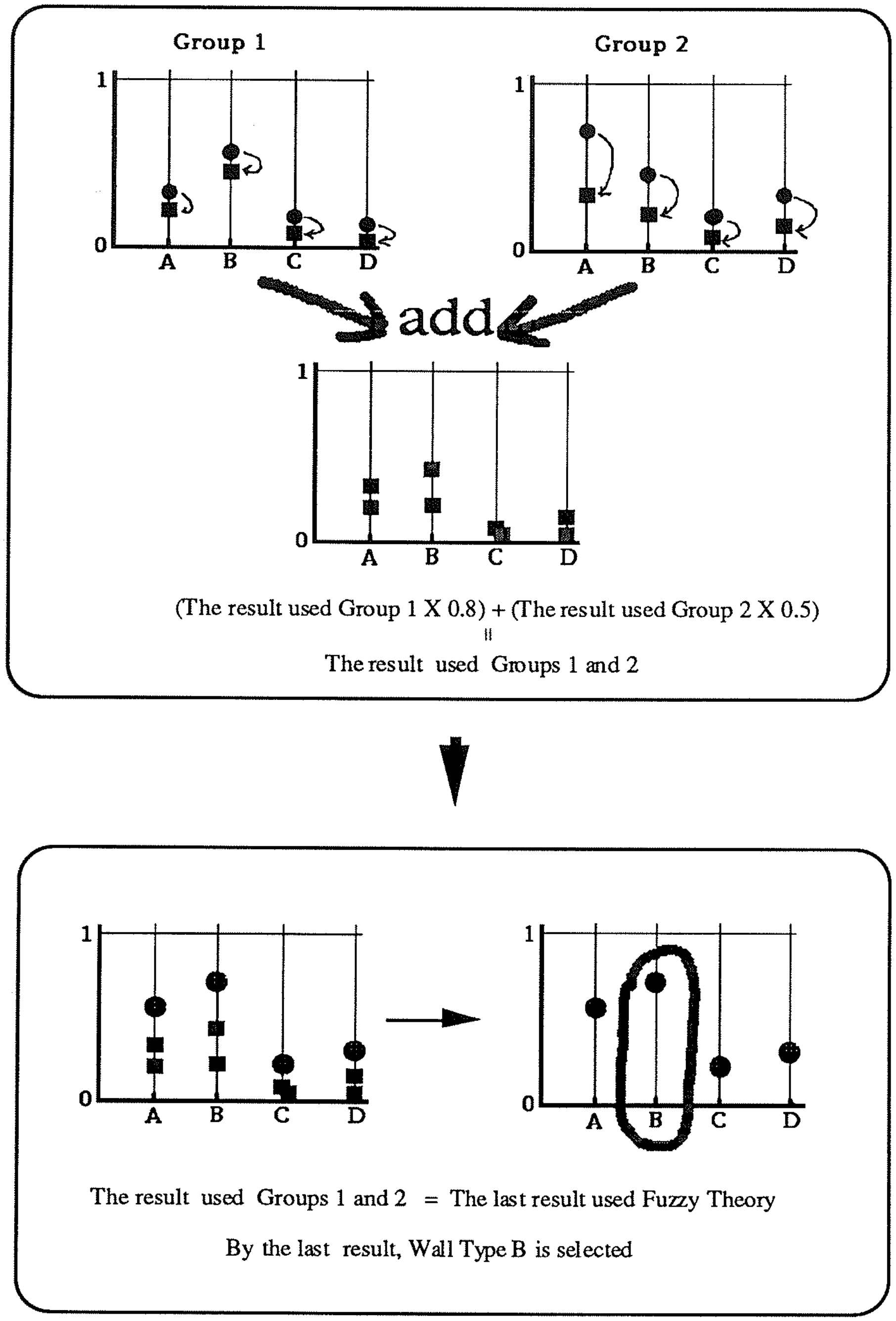


Figure 3-9B Applied Fuzzy Theory to the System

3.3.4 Link of Advantage Table Method and Fuzzy System

In the prototype model, HyperCard is used to determine Membership Function [Index] and to assign grades to each group in the Fuzzy System. HyperCard is very user friendly. HyperTalk is very easy to program, even for a beginner. In the system, experts (specialists) decide the Membership Functions based on databases or their experience. By using the user interface of Hypercard, it is fairly easy for them to recognize Membership Functions' tendencies as shown in Figure, 3-10A and Figure 3-10B.

How to use this user interface is descried below. On figure 3-10A, an expert can select point A and B to determine a Membership Function for Knowledge Group 1, also can select point C and D to determine a Membership Function for Knowledge Group 2. To click the button: "Draw Membership Function", the Membership Function will appear. The Membership Function can be recognized as the dark line shown in Figure 3-10B. After a excavation depth is input, the grades of Knowledge Group 1 and Knowledge Group 2 are calculated as shown in Figure 3-10B.

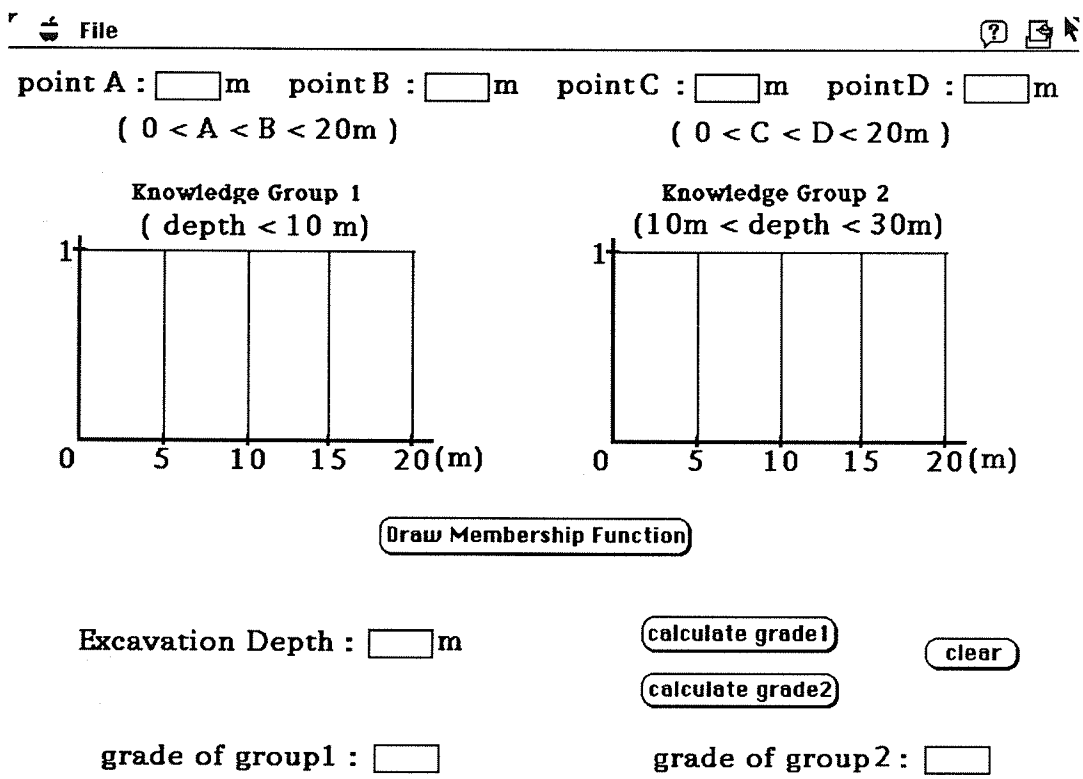


Figure 3-10A User Interface for the Fuzzy System

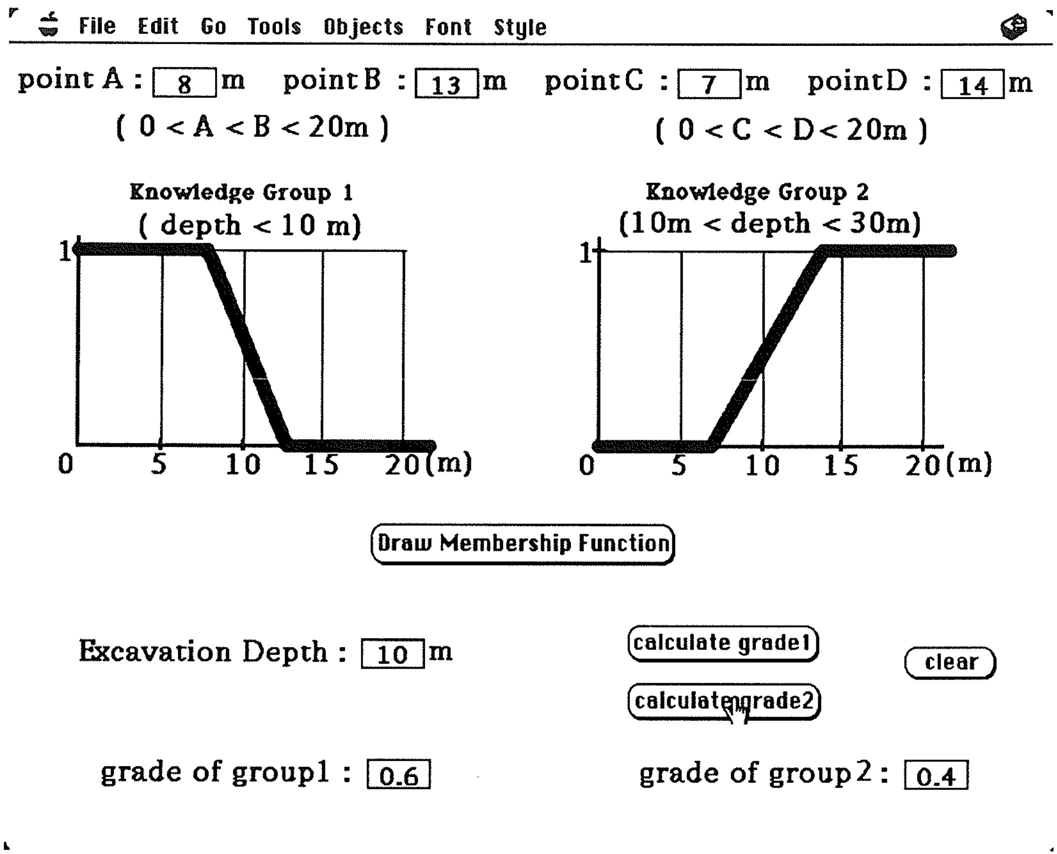


Figure 3-10-B User Interface for the Fuzzy System

For the Advantage Table, a spreadsheet is used. It is Microsoft Excel V.3.0. It is shown in Table 3-3. Evaluation characteristics are selected as shown in Table 3-3 for the prototype model. For future studies, these evaluation items should be modified in accordance with additional experience. In this case, a spreadsheet-based model is very easy to manipulate.

A concrete flow chart is shown in Figure 3-11.

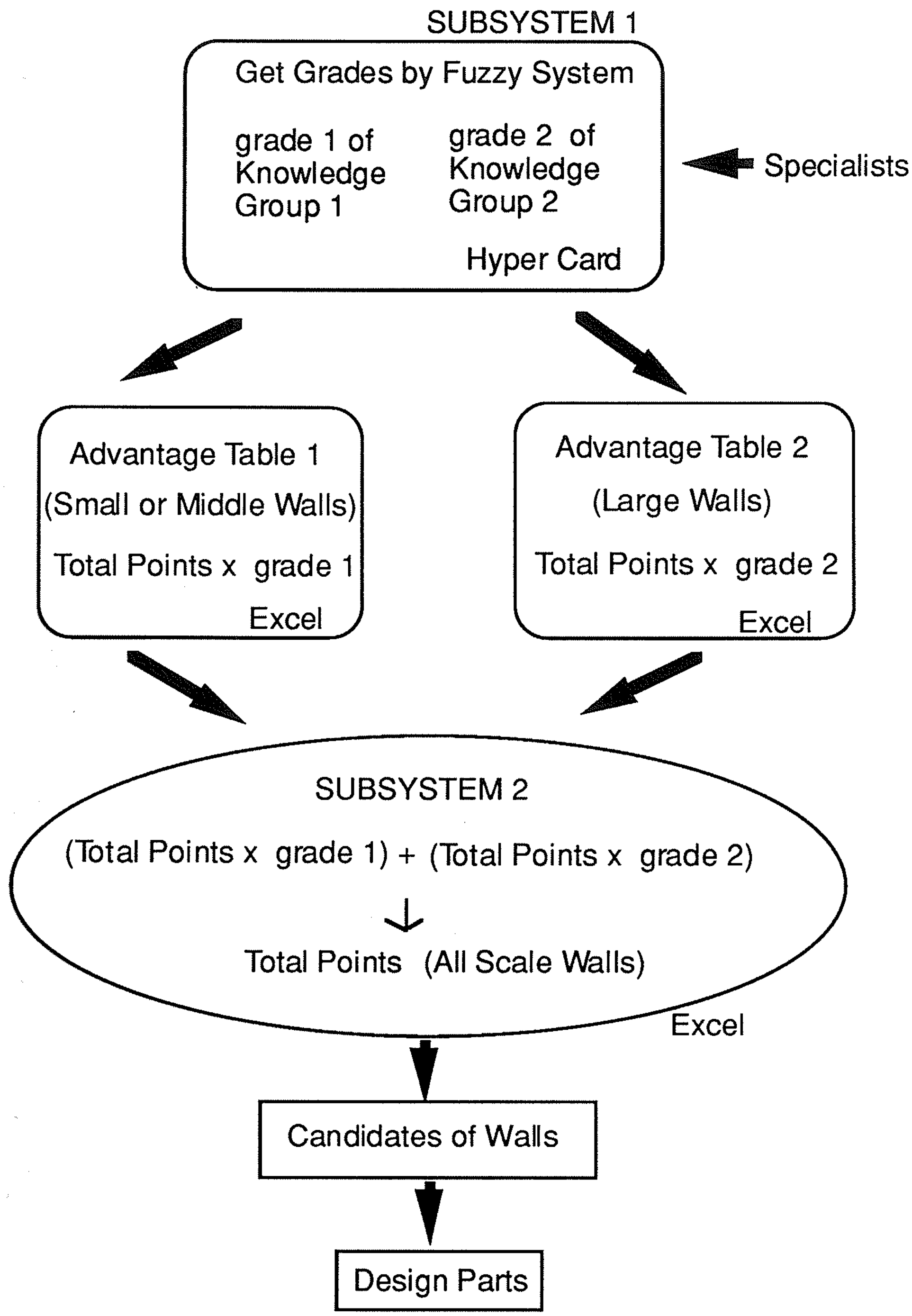


Figure 3-11 Calculation Procedure of Subsystems and Advantage Table

The outline of this flow is explained as follows:

(1) Determine Fuzzy System grades referred to as "Subsystem 1" (HyperCard) on the flow chart.

(2) Total points are calculated on Advantage Tables (Excel):

- Advantage Table 1 from Knowledge Group 1 (small or middle scale of walls), and
- Advantage Table 2 from Knowledge Group 2 (large scale of walls).

(3) By using these grades from Subsystem 1 and total points from Advantage Tables 1 and 2, the final total points based on both groups are calculated by "Subsystem 2" (Excel).

(4) Considering the final total points, wall-type candidates for the next design steps are determined.

4. Test Cases

Wall types are selected by using Advantage Tables and Fuzzy System on the expert system. The test cases are shown for this part of the system. Six cases have been selected. The site conditions for the test cases are shown on Table 4-1. Cases 1-4 are on the same site conditions. The only difference between these cases is excavation depth. Excavation depth is one of the most important factors in wall types' selections. Cases 5-6 have the same excavation depth (11 meters) as Case 3. The only difference between Case 5-6 is the condition of the ground. The condition of the ground is also an important factor in wall types' selections.

Table 4-1 Site Conditions of Test Cases

Site Conditions of Test Cases

Case Number	excavation depth (meters)	condition of ground	ground water
1	5	sandy soil and cohesive soil (both layers)	especially much ground water
2	9		
3	11		
4	15		
5	11	all sandy soil	no ground water
6	11	poor soil	ground water
7	11	same as Case 3	ground water
8	11		no ground water

Comments and results for each test case are described below: (The results of the test cases are shown on Figures 4-1 to 4-12 and Tables 4-1 to 4-9 after each comment.)

Case 1 excavation depth is 5 meters. In this case, selection is done by using Knowledge Group 1 (small or middle walls) in accordance with Membership Function shown in Figure 4-1. The result of both Fuzzy and No Fuzzy must be the same as shown in Figure 4-2.

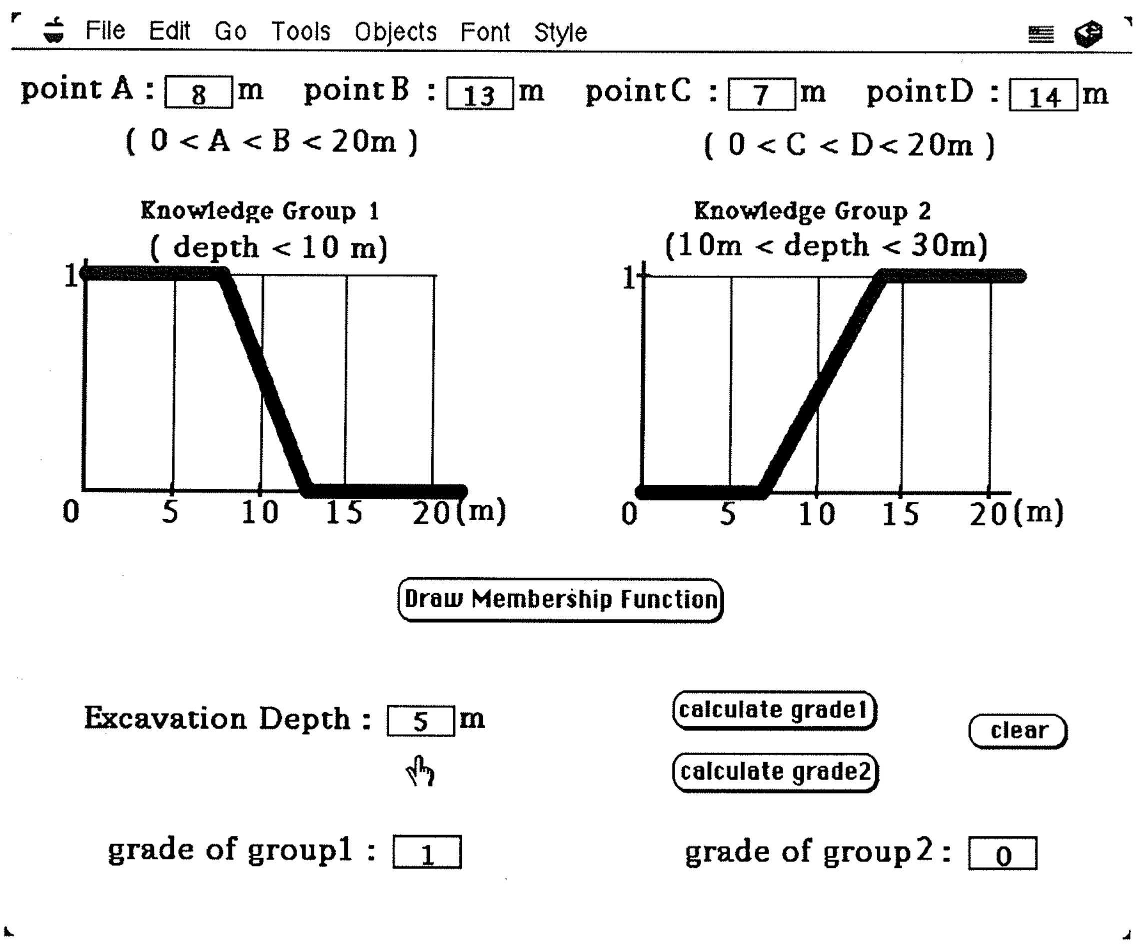


Figure 4-1 Case 1

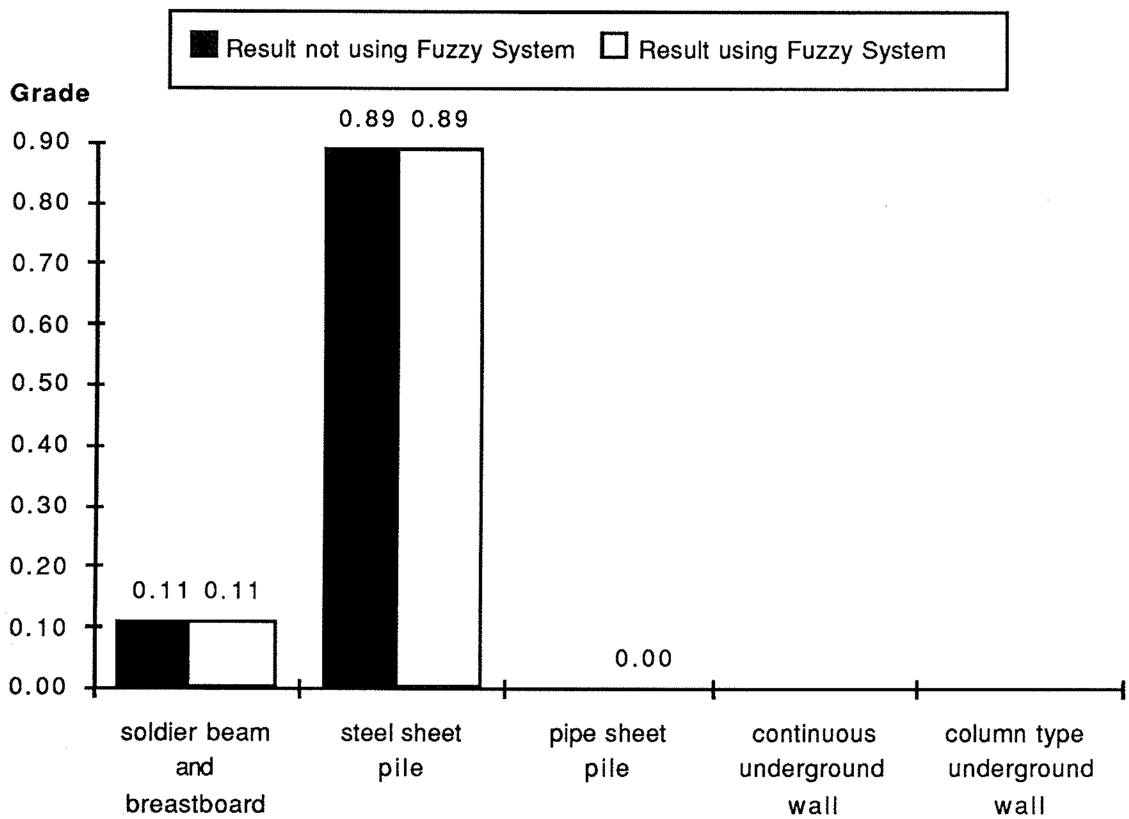


Figure 4-2 Case 1

Table 4-2 Case 1 (Advantage Table)

	Advantage Table2 for large walls						
		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	10
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	0
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	14	44	28	36	16	
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE				
BEST	○○	high mark	3				
BETTER	○	middle mark	2				
Possible but not preferred	△	low mark	1				
Disadvantage	×	negative mark	0				
Grade from MemberShip Function2	0						
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Results by Table 2	Converted TOTAL	14	33.64705882	33.0909091	42.54545455	23.11111111	
	GRADE	0.095631986	0.229838219	0.22603924	0.29062188	0.157868675	
	Coverted GRADE	0	0	0	0	0	
Excavation Depth > 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Tables 2	Result not using Fuzzy System	0.10	0.23	0.23	0.29	0.16	
Grade by Tables 1&2	Result using Fuzzy System	0.11	0.89	0.00	0.00	0.00	
Excavation Depth < 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Table 1	Result not using Fuzzy System	0.11	0.89				
Grade by Tables 1&2	Result using Fuzzy System	0.11	0.89	0.00	0.00	0.00	

	Advantage Table1 for Small or middle walls			
		WALL TYPE		Weight Coefficient
		horizontal sheeting	steel sheet pile	
EVALUATION ITEM				
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	10
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	10
SCALE of EXCAVATION	depth(deep)	△	○	0
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	3	50	
	Converted TOTAL	3	25	
	GRADE	0.107142857	0.892857143	
	Coverted GRADE	0.107142857	0.892857143	
Grade from MemberShip Function1	1			
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE	
BEST	○○	high mark	3	
BETTER	○	middle mark	2	
Possible but not preferred	△	low mark	1	
Disadvantage	×	negative mark	0	

Case 2 excavation depth is 9 meters. If selection procedures shown in Figure 3-3 are followed with no changes, Knowledge Group 2 (large walls) is not applied because the excavation depth is less than 10 meters. Walls are selected by using only Knowledge Group 1 (small or middle walls). In this system, Walls are selected by using both Knowledge Group 1 and 2. In case 2, steel sheet pile is the best selection for walls. By using Fuzzy System, the grade of the best selection becomes smaller and possibilities for other walls can be determined.

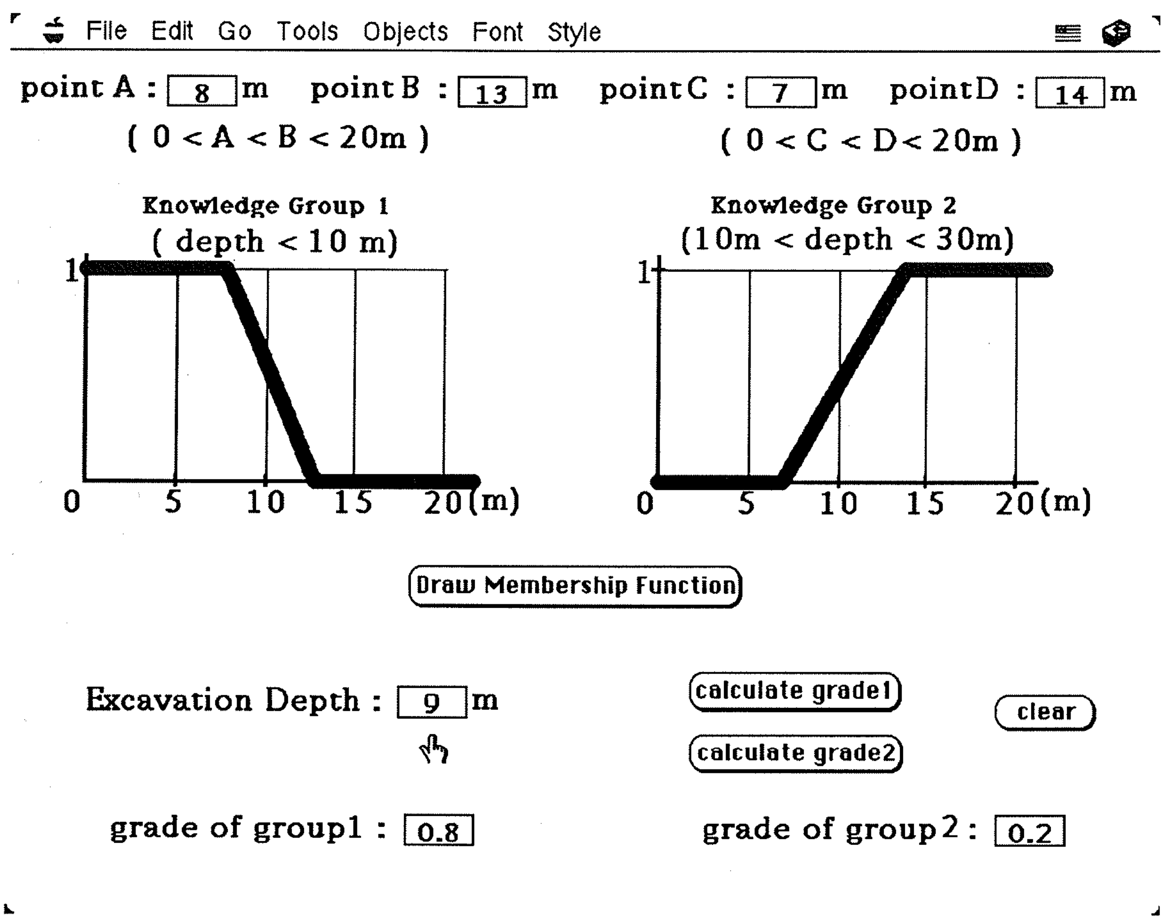


Figure 4-3 Case 2

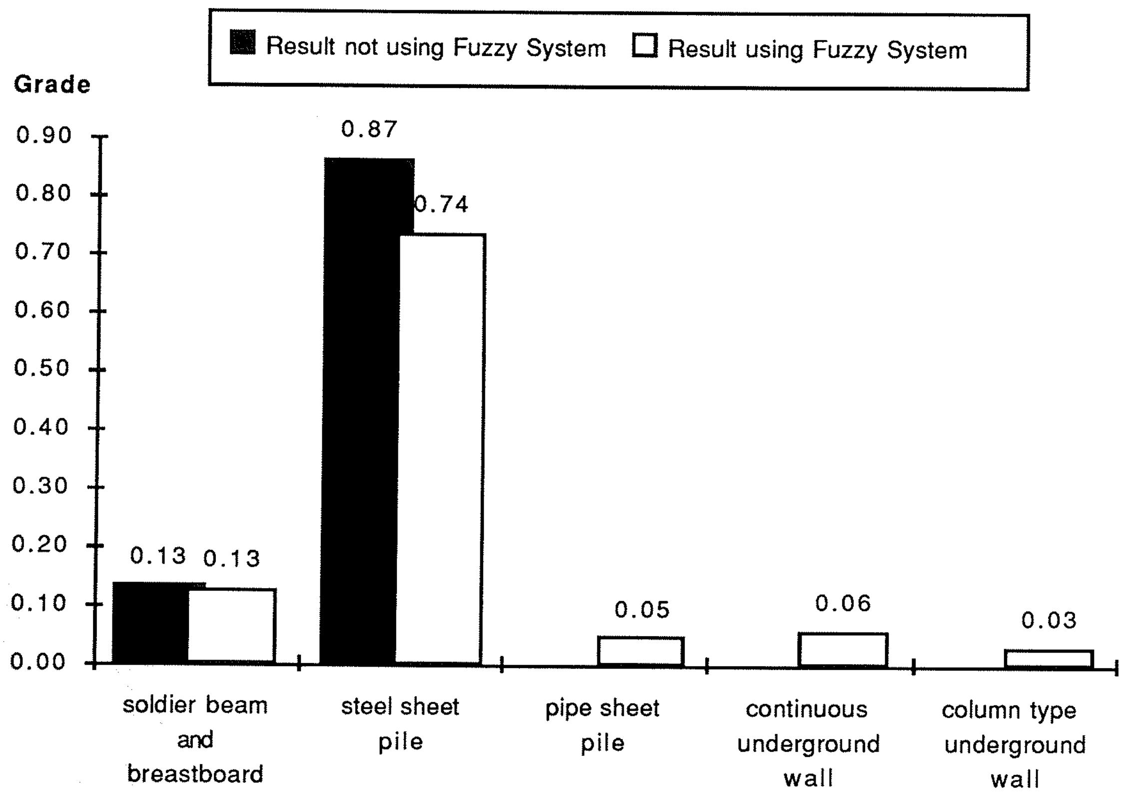


Figure 4-4 Case 2

Table 4-3 Case 2 (Advantage Table)

	Advantage Table2 for large walls						
		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	10
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	46	31	39	17	
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE				
BEST	○○	high mark	3				
BETTER	○	middle mark	2				
Possible but not preferred	△	low mark	1				
Disadvantage	×	negative mark	0				
Grade from MemberShip Function2	0.2						
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Results by Table 2	Converted TOTAL	15	35.17647059	36.6363636	46.09090909	24.55555556	
	GRADE	0.095262713	0.223400402	0.23267196	0.292716336	0.155948589	
	Coverted GRADE	0.019052543	0.04468008	0.04653439	0.058543267	0.031189718	
Excavation Depth > 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Tables 2	Result not using Fuzzy System	0.10	0.22	0.23	0.29	0.16	
Grade by Tables 1&2	Result using Fuzzy System	0.13	0.74	0.05	0.06	0.03	
Excavation Depth < 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Table 1	Result not using Fuzzy System	0.13	0.87				
Grade by Tables 1&2	Result using Fuzzy System	0.13	0.74	0.05	0.06	0.03	

	Advantage Table1 for Small or middle walls			
		WALL TYPE		Weight Coefficient
		horizontal sheeting	steel sheet pile	
EVALUATION ITEM				
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	10
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	10
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	52	
	Converted TOTAL	4	26	
	GRADE	0.133333333	0.866666667	
	Coverted GRADE	0.106666667	0.693333333	
Grade from MemberShip Function1	0.8			
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE	
BEST	○○	high mark	3	
BETTER	○	middle mark	2	
Possible but not preferred	△	low mark	1	
Disadvantage	×	negative mark	0	

Case 3, excavation depth is 11 meters. If selection procedures shown in Figure 3-3 are followed with no changes, Knowledge Group 1 (small or middle walls) is not applied because the excavation depth is greater than 10 meters. Walls are selected by using only Knowledge Group 2 (large walls). In this system, walls are selected by using both Knowledge Group 1 and 2. Steel sheet pile is the best selection for walls. A continuous underground wall is the best selection without Fuzzy system. For large walls, a continuous underground wall is a good type. A continuous underground wall has great stiffness and little displacement even for large excavation depth as described in Table 3-2 (Chapter 3). Great stiffness and little displacement are merits of this wall. For demerits of this wall, long duration is necessary and the cost is expensive. Steel sheet pile has a great displacement for a very large excavation depth, but it can be used economically. In Case 3, the excavation depth is only 11 meters. 11 meters actually belongs to Knowledge Group 2 (large walls). 11 meters is very close to 10 meters which belongs to Knowledge Group 1 (small or middle walls). There is a great possibility of steel sheet pile being the best selection. In this case, steel sheet pile must be selected by Experts. TRW Expert System selected this wall by using Fuzzy system.

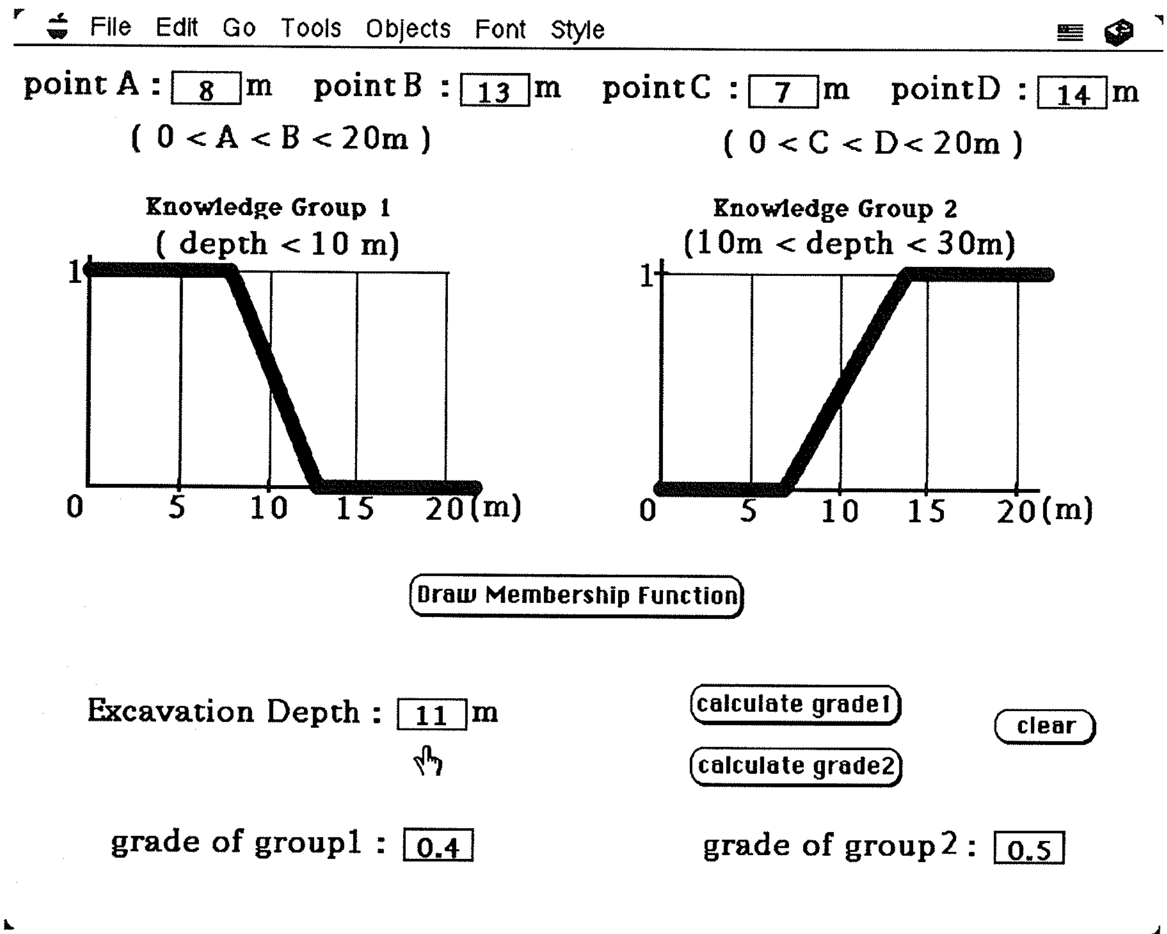


Figure 4-5 Case 3

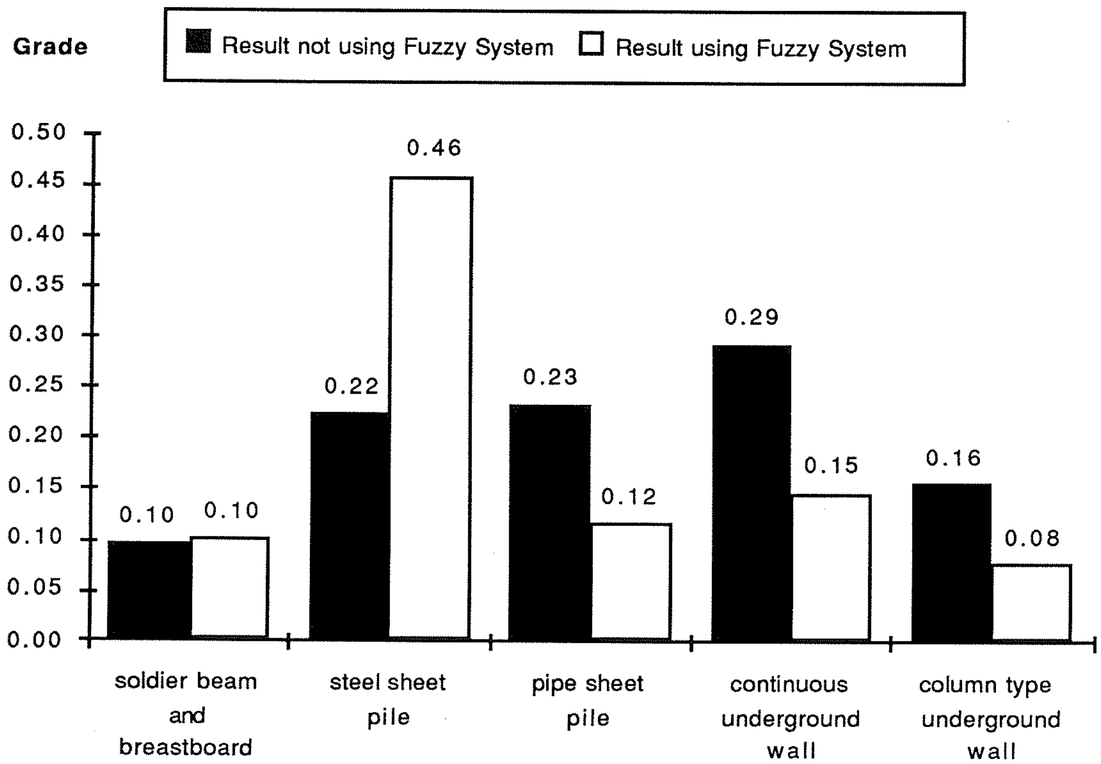


Figure 4-6 Case 3

Table 4-4 Case 3 (Advantage Table)

Advantage Table2 for large walls

		WALL TYPE					Weight Coefficient
EVALUATION ITEM		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	10
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	46	31	39	17	

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Grade from MemberShip Function2: 0.5

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Results by Table 2	Converted TOTAL	15	35.17647059	36.6363636	46.09090909	24.55555556
	GRADE	0.095262713	0.223400402	0.23267196	0.292716336	0.155948589
	Coverted GRADE	0.047631357	0.111700201	0.11633598	0.146358168	0.077974295

Excavation Depth > 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Tables 2	Result not using Fuzzy System	0.10	0.22	0.23	0.29	0.16
Grade by Tables 1&2	Result using Fuzzy System	0.10	0.46	0.12	0.15	0.08

Excavation Depth < 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Table 1	Result not using Fuzzy System	0.13	0.87			
Grade by Tables 1&2	Result using Fuzzy System	0.10	0.46	0.12	0.15	0.08

Advantage Table1 for Small or middle walls

		WALL TYPE		Weight Coefficient
EVALUATION ITEM		horizontal sheeting	steel sheet pile	
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	10
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	10
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	52	
	Converted TOTAL	4	26	
	GRADE	0.133333333	0.866666667	
	Coverted GRADE	0.053333333	0.346666667	

Grade from MemberShip Function1: 0.4

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Case 4 excavation depth is 15 meters. In Case 4, walls are selected by only Knowledge Group 2 (large walls) in accordance with Membership Function shown in Figure 4-1. The result of both Fuzzy and No Fuzzy must be the same as shown in Figure 4-8.

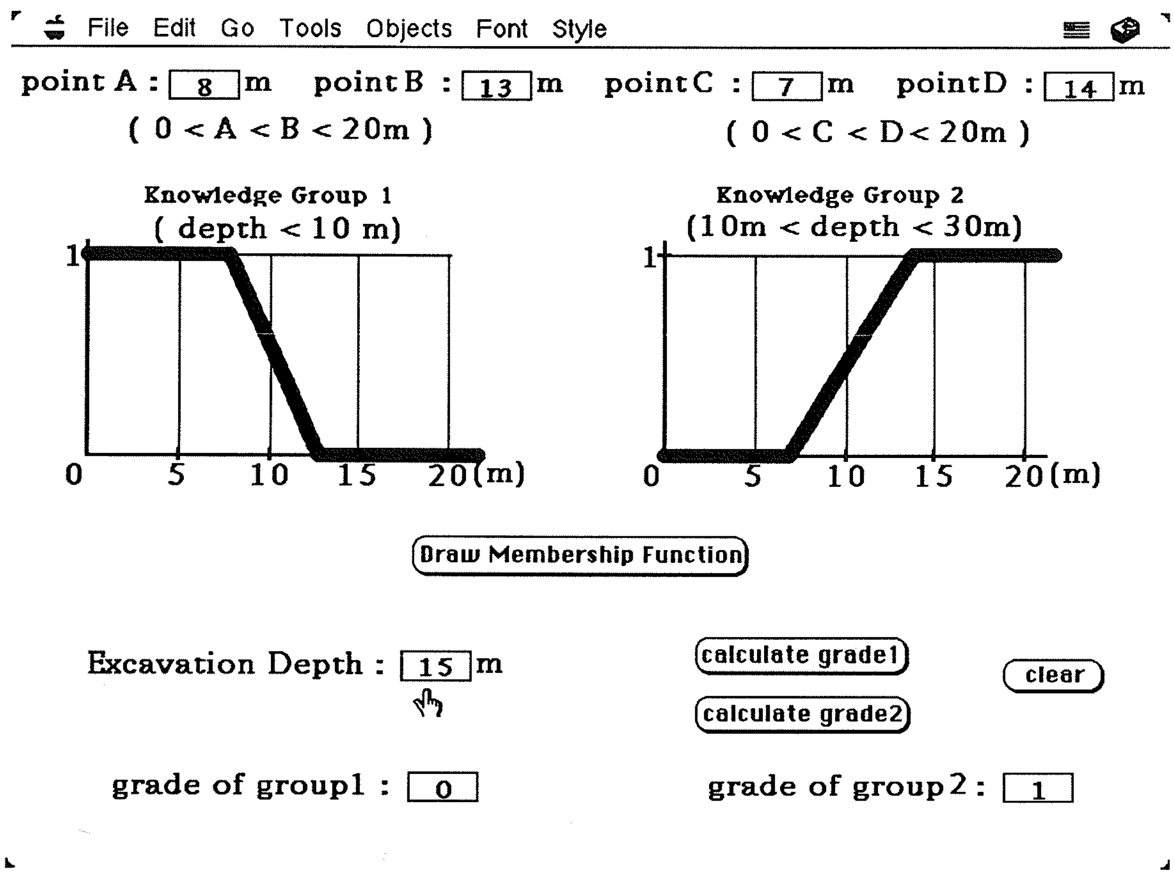


Figure 4-7 Case 4

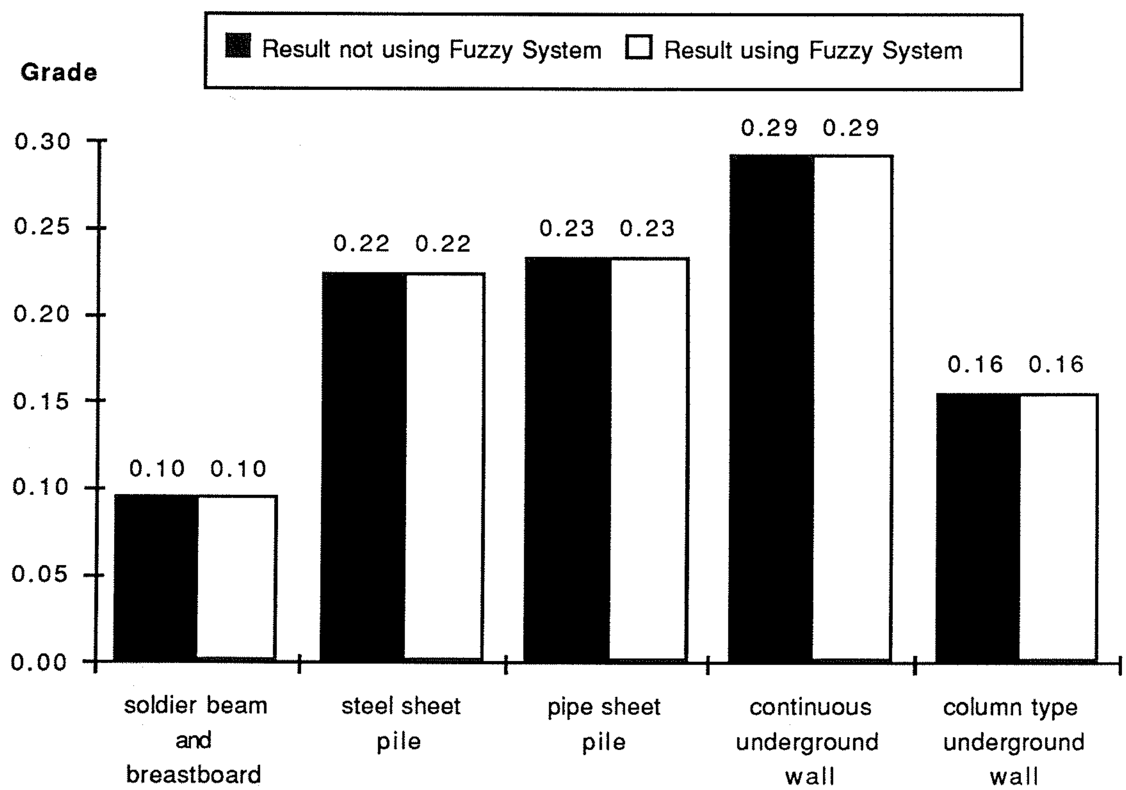


Figure 4-8 Case 4

Table 4-5 Case 4 (Advantage Table)

	Advantage Table2 for large walls						
		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	10
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
COMBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	46	31	39	17	
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE				
BEST	○○	high mark	3				
BETTER	○	middle mark	2				
Possible but not preferred	△	low mark	1				
Disadvantage	×	negative mark	0				
Grade from MemberShip Function2	1						
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Results by Table 2	Converted TOTAL	15	35.17647059	36.6363636	46.09090909	24.55555556	
	GRADE	0.095262713	0.223400402	0.23267196	0.292716336	0.155948589	
	Coverted GRADE	0.095262713	0.223400402	0.23267196	0.292716336	0.155948589	
Excavation Depth > 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Tables 2	Result not using Fuzzy System	0.10	0.22	0.23	0.29	0.16	
Grade by Tables 1&2	Result using Fuzzy System	0.10	0.22	0.23	0.29	0.16	
Excavation Depth < 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Table 1	Result not using Fuzzy System	0.13	0.87				
Grade by Tables 1&2	Result using Fuzzy System	0.10	0.22	0.23	0.29	0.16	

	Advantage Table1 for Small or middle walls			
		WALL TYPE		Weight Coefficient
		horizontal sheeting	steel sheet pile	
EVALUATION ITEM				
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	10
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	10
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	52	
	Converted TOTAL	4	26	
	GRADE	0.133333333	0.866666667	
	Coverted GRADE	0	0	
Grade from MemberShip Function1	0			
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE	
BEST	○○	high mark	3	
BETTER	○	middle mark	2	
Possible but not preferred	△	low mark	1	
Disadvantage	×	negative mark	0	

Case 5 excavation depth is 11 meters as is Case 3. The condition of ground is sandy soil without ground water. In Case 3, steel sheet pile is selected because there is much ground water. In Case 5, soldier beam and breastboard is selected as being the best wall. 11 meters (excavation depth) belongs to Knowledge Group 2 (large walls). 11 meters is very close to 10 meters which belongs to Knowledge Group 1 (small or middle walls). The grade of soldier beam and breastboard becomes greater by using Fuzzy system.

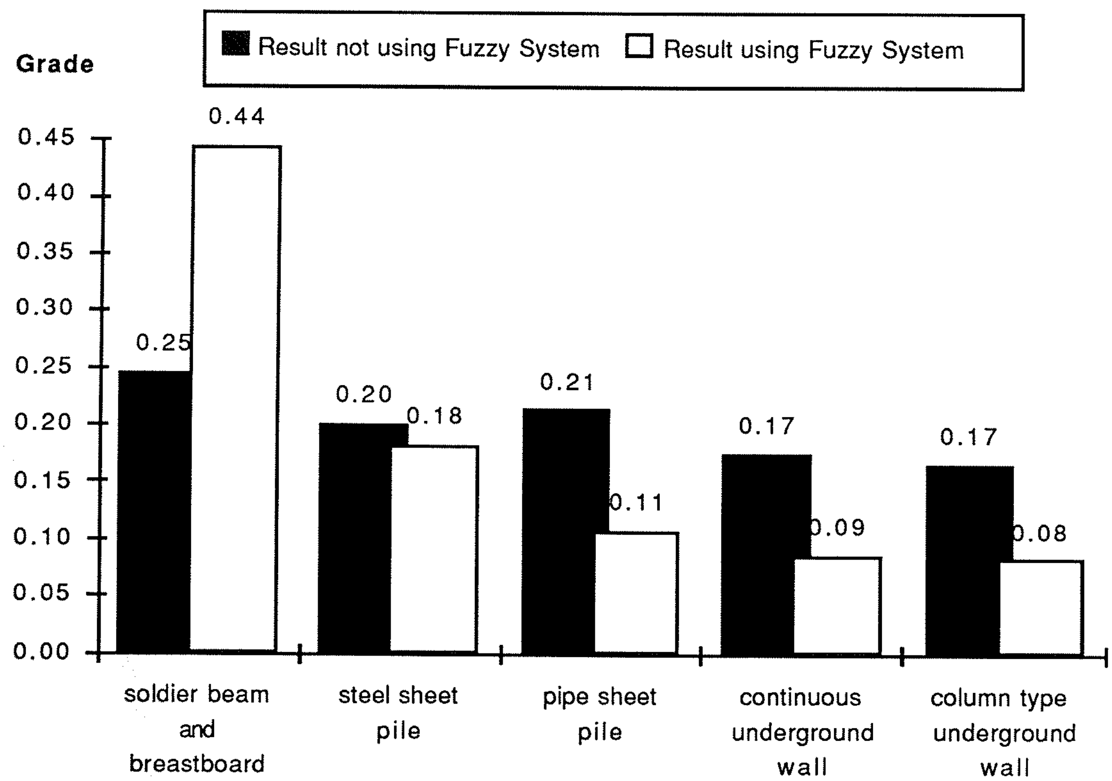


Figure 4-9 Case 5

Table 4-6 Case 5 (Advantage Table)

Advantage Table2 for large walls

EVALUATION ITEM		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	Weight Coefficient
		WALL TYPE					
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	0
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
COMBINATION of TIMBERING	(brace,tie back…)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	16	11	9	7	

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Grade from MemberShip Function2	0.5

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Results by Table 2	Converted TOTAL	15	12.23529412	13	10.63636364	10.11111111
	GRADE	0.245971121	0.200635267	0.21317497	0.174415886	0.165802755
	Coverted GRADE	0.12298556	0.100317634	0.10658749	0.087207943	0.082901378

Excavation Depth > 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Tables 2	Result not using Fuzzy System	0.25	0.20	0.21	0.17	0.17
Grade by Tables 1&2	Result using Fuzzy System	0.44	0.18	0.11	0.09	0.08

Excavation Depth < 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Table 1	Result not using Fuzzy System	0.80	0.20			
Grade by Tables 1&2	Result using Fuzzy System	0.44	0.18	0.11	0.09	0.08

Advantage Table1 for Small or middle walls

EVALUATION ITEM		horizontal sheeting	steel sheet pile	Weight Coefficient
		WALL TYPE		
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	0
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	0
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	2	
	Converted TOTAL	4	1	
	GRADE	0.8	0.2	
	Coverted GRADE	0.32	0.08	
Grade from MemberShip Function1	0.4			

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Case 6 excavation depth is 11 meters as is Case 3. The condition of ground is poor ground with ground water. Steel sheet pile is selected. A continuous underground wall is the best selection without Fuzzy system. For large walls, a continuous underground wall is a good type. In Case 6, the excavation depth is 11 meters which belongs to Knowledge Group 2 (large walls). 11 meters is very close to 10 meters which belongs to Knowledge Group 1 (small or middle walls). There is a great possibility of steel sheet pile being the best selection. TRW Expert System selected this wall by using Fuzzy system.

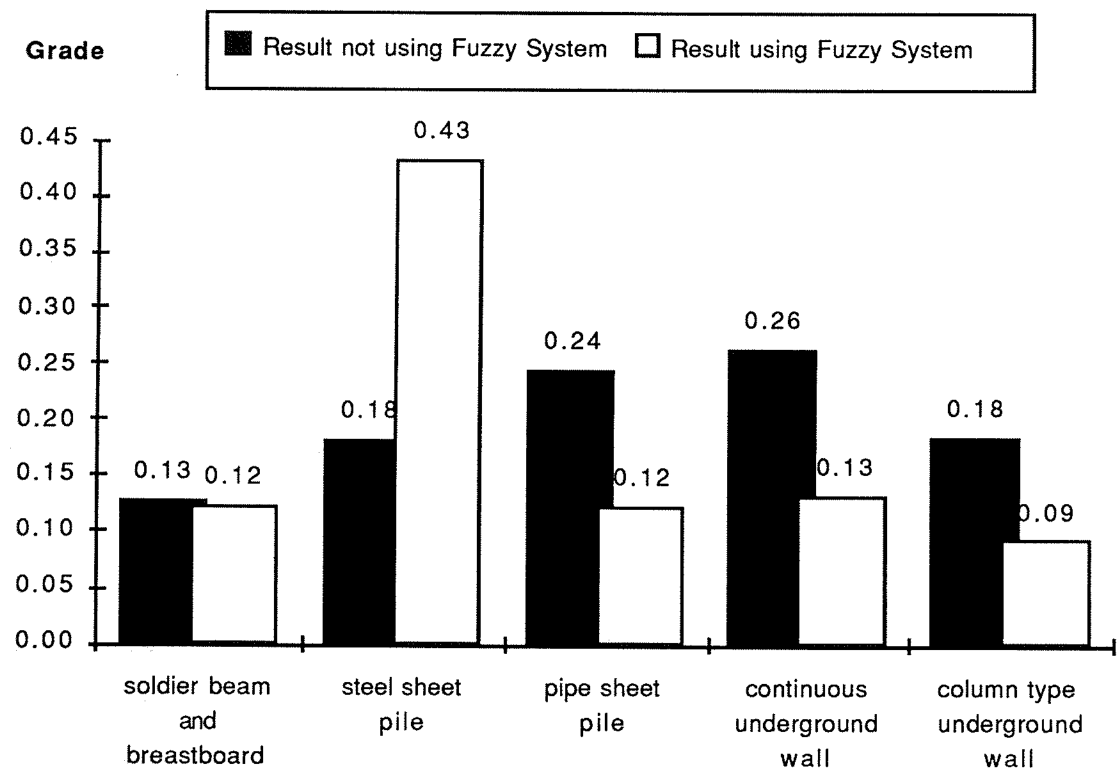


Figure 4-10 Case 6

Table 4-7 Case 6 (Advantage Table)

Advantage Table2 for large walls

EVALUATION ITEM		WALL TYPE: soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	Weight Coefficient
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	1
	cohesive soil	○○	○○	○	○	○	0
	sandy soil	○○	○○	○	△	△	0
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	1
	placing	○○	○○	○	△	△	0
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	1
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	8	15	13	14	8	

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Grade from MemberShip Function2: 0.5

Results by Table 2	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
	Converted TOTAL	8	11.47058824	15.3636364	16.54545455	11.55555556
	GRADE	0.127114804	0.182260196	0.2441182	0.262896526	0.183610272
	Coverted GRADE	0.063557402	0.091130098	0.1220591	0.131448263	0.091805136

Excavation Depth > 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Tables 2	Result not using Fuzzy System	0.13	0.18	0.24	0.26	0.18
Grade by Tables 1&2	Result using Fuzzy System	0.12	0.43	0.12	0.13	0.09

Excavation Depth < 10m

	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall
Grade by Table 1	Result not using Fuzzy System	0.14	0.86			
Grade by Tables 1&2	Result using Fuzzy System	0.12	0.43	0.12	0.13	0.09

Advantage Table1 for Small or middle walls

EVALUATION ITEM		WALL TYPE: horizontal sheeting	steel sheet pile	Weight Coefficient
CONDITION of GROUND	poor ground	×	○○	1
	gravel soil	○○	×	0
	much groundwater	×	○○	1
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	1
	cut off water	×	○	1
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	1	12	
	Converted TOTAL	1	6	
	GRADE	0.142857143	0.857142857	
	Coverted GRADE	0.057142857	0.342857143	

Grade from MemberShip Function1: 0.4

EVALUATION RANK	SIGN	SCORE	YOUR CHOICE
BEST	○○	high mark	3
BETTER	○	middle mark	2
Possible but not preferred	△	low mark	1
Disadvantage	×	negative mark	0

Case 7 excavation depth is 11 meters as is Case 3. The condition of the ground is also the same as Case 3. In Case 3, there is much ground water and the weight coefficient is 10. In Case 7, there is some ground water and the weight of coefficient is 1. How to determine the weight coefficient is important. In order to fix this value, many test cases must be made. The results of this (TRW) system must be compared to actual projects in the future.

When some factors are very important like ground water as in Case 3, these factors can be expressed by using the weight coefficient. In Case 3, steel sheet pile is selected because of a great amount of ground water. In Case 7, there is a small amount of ground water. Soldier beam and breastboard construction is the first selection by using Fuzzy System. Steel sheet pile is the second selection as second by using Fuzzy System. When Fuzzy System is not used, the grades of all walls are almost the same because only Knowledge Group 2 (Large scale walls) is used.

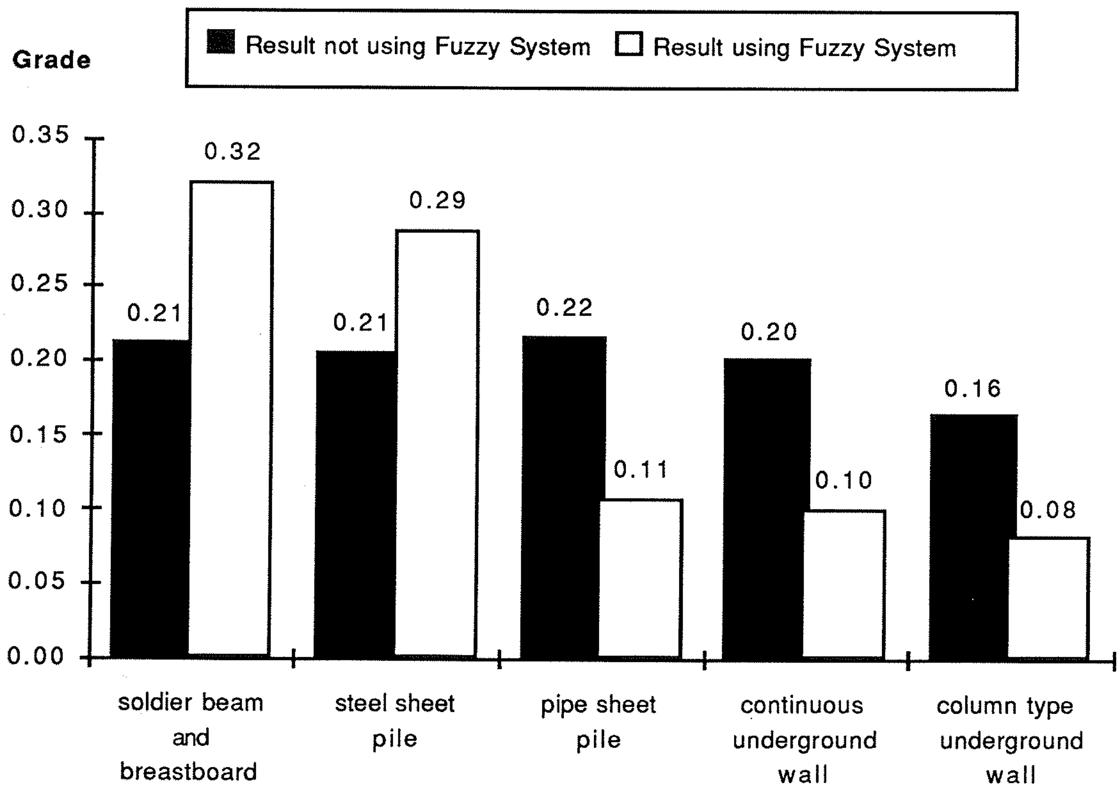


Figure 4-11 Case 7

Table 4-8 Case 7 (Advantage Table)

Advantage Table2 for large walls

		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	1
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	19	13	12	8	
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE				
BEST	○○	high mark	3				
BETTER	○	middle mark	2				
Possible but not preferred	△	low mark	1				
Disadvantage	×	negative mark	0				
Grade from MemberShip Function2	0.5						
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Results by Table 2	Converted TOTAL	15	14.52941176	15.3636364	14.18181818	11.55555556	
	GRADE	0.212373077	0.205710392	0.21752151	0.200789091	0.163605926	
	Coverted GRADE	0.106186538	0.102855196	0.10876076	0.100394545	0.081802963	
Excavation Depth > 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Tables 2	Result not using Fuzzy System	0.21	0.21	0.22	0.20	0.16	
Grade by Tables 1&2	Result using Fuzzy System	0.32	0.29	0.11	0.10	0.08	
Excavation Depth < 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Table 1	Result not using Fuzzy System	0.53	0.47				
Grade by Tables 1&2	Result using Fuzzy System	0.32	0.29	0.11	0.10	0.08	

Advantage Table1 for Small or middle walls

		WALL TYPE		Weight Coefficient
		horizontal sheeting	steel sheet pile	
EVALUATION ITEM				
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	1
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	1
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	7	
	Converted TOTAL	4	3.5	
	GRADE	0.533333333	0.466666667	
	Coverted GRADE	0.213333333	0.186666667	
Grade from MemberShip Function1	0.4			
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE	
BEST	○○	high mark	3	
BETTER	○	middle mark	2	
Possible but not preferred	△	low mark	1	
Disadvantage	×	negative mark	0	

Case 8 excavation depth is 11 meters as is Case 3. The condition of the ground is also the same as Case 3. In Case 3, there is much ground water and the weight coefficient is 10. In Case 7, there is a small amount of ground water and the weight coefficient is 1. In Case 8, there is no ground water so the weight coefficient is 0.

When Fuzzy System is not used, the grades of all walls are almost the same because only Knowledge Group 2 (Large scale walls) is used. Soldier beam and breastboard construction is selected by using Fuzzy system.

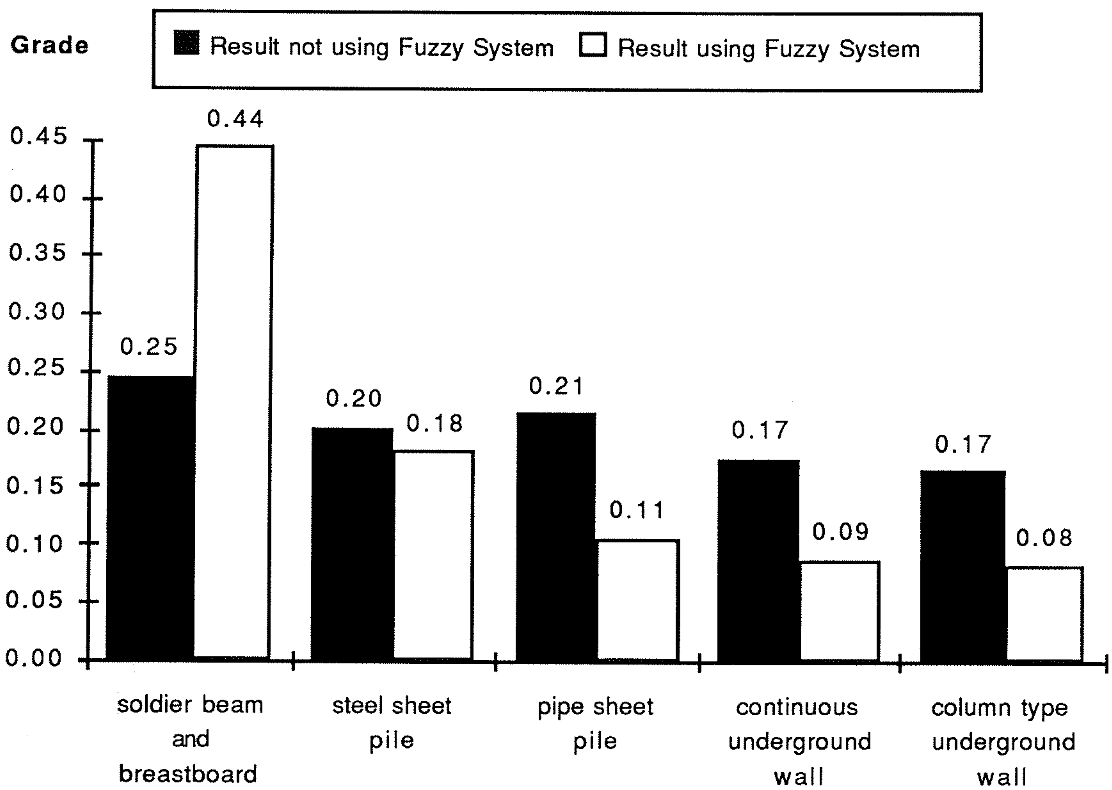


Figure 4-12 Case 8

Table 4-9 Case 8 (Advantage Table)

Advantage Table2 for large walls

		WALL TYPE					Weight Coefficient
		soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
EVALUATION ITEM							
CONDITION of GROUND	poor ground	△	○○	○○	○○	○	0
	cohesive soil	○○	○○	○	○	○	1
	sandy soil	○○	○○	○	△	△	1
CONDITION of EXECUTION	much groundwater	×	○○	○	○○	△	0
	placing	○○	○○	○	△	△	1
	noise, vibration	△	△	△	○	○	0
	settlement	△	○	○○	○○	○	0
SCALE of EXCAVATION	depth(deep)	△	○	○○	○○	△	1
	width (large area)	○○	○○	○	○	○	1
CONBINATION of TIMBERING	(brace,tie back--)	○○	○○	○	○	○	0
REUSE	REUSE	○	○○	×	×	×	0
PERIOD	PERIOD	○○	○○	×	×	○	0
COST	COST	○	○	×	×	×	1
	neutral TOTAL	26	34	22	22	18	
	TOTAL POINT	15	16	11	9	7	
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE				
BEST	○○	high mark	3				
BETTER	○	middle mark	2				
Possible but not preferred	△	low mark	1				
Disadvantage	×	negative mark	0				
Grade from MemberShip Function2	0.5						
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Results by Table 2	Converted TOTAL	15	12.23529412	13	10.63636364	10.11111111	
	GRADE	0.245971121	0.200635267	0.21317497	0.174415886	0.165802755	
	Coverted GRADE	0.12298556	0.100317634	0.10658749	0.087207943	0.082901378	
Excavation Depth > 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Tables 2	Result not using Fuzzy System	0.25	0.20	0.21	0.17	0.17	
Grade by Tables 1&2	Result using Fuzzy System	0.44	0.18	0.11	0.09	0.08	
Excavation Depth < 10m							
	Wall TYPE	soldier beam and breastboard	steel sheet pile	pipe sheet pile	continuous underground wall	column type underground wall	
Grade by Table 1	Result not using Fuzzy System	0.80	0.20				
Grade by Tables 1&2	Result using Fuzzy System	0.44	0.18	0.11	0.09	0.08	

Advantage Table1 for Small or middle walls

		WALL TYPE		Weight Coefficient
		horizontal sheeting	steel sheet pile	
EVALUATION ITEM				
CONDITION of GROUND	poor ground	×	○○	0
	gravel soil	○○	×	1
	much groundwater	×	○○	0
CONDITION of EXECUTION	noise, vibration	○	×	0
	settlement	×	○	0
	cut off water	×	○	0
SCALE of EXCAVATION	depth(deep)	△	○	1
Results by Table 1	neutral TOTAL	6	12	
	TOTAL POINT	4	2	
	Converted TOTAL	4	1	
	GRADE	0.8	0.2	
	Coverted GRADE	0.32	0.08	
Grade from MemberShip Function1	0.4			
EVALUATION RANK	SIGN	SCORE	YOUR CHOICE	
BEST	○○	high mark	3	
BETTER	○	middle mark	2	
Possible but not preferred	△	low mark	1	
Disadvantage	×	negative mark	0	

5. Conclusions

The basis of the whole system should consist of the following four parts:

(1) necessity checks of walls for the site,

(2) selection of wall types,

(3) design of walls, and

(4) check of safety, cost, and duration.

Design functions are comparatively easy to program because the method to calculate design is known. For the other functions, especially selection of wall types, expert systems are desirable. There are few established rules to select walls. Expert systems can help engineers to select walls. New ideas (Advantage Table Method, Fuzzy System, and so on) to select walls have been proposed in this paper. An Advantage Table is formed based on design experience that every specialist has acquired. It is very easy to modify these tables to reflect new experience and knowledge. These ideas will improve progress in future research. Even if the scale of a wall can not be decided easily (i.e., small scale walls or large scale walls), the best wall is selected easily by using Advantage Table and Fuzzy System.

The critical problem to make a TRW Expert System is a shortage of specialists' knowledge. There are no formalized knowledge for this. To obtain this information could help to produce an excellent TRW Expert System.

6. Future Work

In order to accomplish the integration of the TRW Design Expert System, many future works are mentioned as follows:

(1) Concept of the whole system

As mentioned in Chapter 2, in order to realize the concept of whole TRW Design Expert System, the integration of every part of the system is necessary. Generally speaking for a build up of expert systems, some expert systems, especially structural design parts of walls, are made as a continuous process style. (KUMAGAI 1). Some expert systems, especially selection parts of wall types and construction methods, make little progress. For selection parts, even the idea to realize expert systems is not clear for many cases. For the future, many ideas for this part should be proposed and realized to build up expert systems, then the integration of every part of the system will succeed.

In order to find out more practical concepts of the wall system, it is necessary to interview more experts and to coordinate their knowledge.

(2) Selection areas of walls

About the Advantage Table Method, every expert should have his own knowledge like an Advantage Table. This table will be able to be modified for real usage according to coordinate knowledge from other experts. How to decide weight coefficients and scores of rankings as mentioned in Chapter 3.3.2 is a very important issue. Some rules based on knowledge of experts are necessary to decide weight coefficients and scores of rankings.

About the Fuzzy system, the way to determine Membership Functions should become clearer to realize this system for real usage. In this stage, experts can decide this freely, but even experts of TRW can not fix this so simply. There are possibilities to be able to set some basic patterns of Membership Functions to compare real site examples with the results of many test cases from the model.

In a case of over 30 meter's excavation depths, this system does not deal with that now. For such a special huge wall, there are few examples and few experts have experience to design and build such walls. In this case, it is very hard to formalize the steps of selecting and designing walls. Considering the existing state of things, it is safer that this case be handled directly by specialists now. For the future, some expert systems based on new ideas will be able to help us to design such huge walls.

(3) Recheck part of safety, cost, and duration

There are often special conditions unique to specific sites. There is a possibility of neglecting to consider such conditions in other parts of this system. Such conditions should be checked according to safety, cost, and duration. An expert system is needed for this part. To produce such a part of an expert systems should be part of a future study.

It is not possible to find an expert system for selection of wall types of TRW in the U.S. There seems to be very few of such kinds of systems in the U.S. The only system that could be found is the Retaining Wall Rehabilitation Design (ADAMS 1). In Japan, there are also very few systems for real usage. Only one excellent expert system could be found. It is created by KAJIMA CORPORATION (KAJIMA 1). The production of such kinds of expert systems has only just recently started. There is a great need of such expert systems for TRW for construction sites. More research needs to focus on this problem. Therefore this research provides a new chart for future research.

7. References

[SEKOU 1] Sogo Doboku Kenkyusho, "The Foundation Engineering & Equipment", Monthly. Vol. 15, No. 3. Mar. 1987.

[SAWADA 1] Ichiro Sawada, "The Method Selection of Retaining walls on Construction Sites". The Foundation Engineering & Equipment, Monthly. Vol. 15, No. 3. pages 29-36. Mar. 1987.

[MEPC 1] Metropolitan Expressway Public Corporation, "Design Standards for Temporary Structures". Aug. 1990.

[FUZZY 1] Fuzzy Study Group of KUMAGAI GUMI Co.,Ltd. "Applications of Fuzzy Theory in Construction Industry". Apr. 1992.

[FUZZY 2] Toshirou Terano, Kiyoji Asai, Michio Sugeno, "Fuzzy Systems Theory and Its Applications." Dec. 1989.

[FUZZY 3] Kiyoji Asai, Constantin Virgil Negoita, "Introduction to Fuzzy Systems Theory." Dec. 1987

[FUZZY 4] Toshirou Terano, Kiyoji Asai, Michio Sugeno, "Applied Fuzzy Systems." May. 1989.

[KUMAGAI 1] KUMAGAI GUMI Co.,Ltd, "CADDET (Computer Aided Design, Drawing, and Estimating System for Temporary Structures)." 1989.

[ADAMS 1] Adams, T., Hendrickson, C., and Christiano, P. "An Expert System Architecture for Retaining Wall Design." Transportation Research Record 1187, Transportation Research Board, National Research Council, pages 9-20. 1988.

[ADAMS 1] Adams, T., Christiano, P., and Hendrickson, C. "Some Expert System Applications in Geotechnical Engineering." In proceedings, ASCE Foundation Engineering Congress, Northwestern University, Evanston, IL, June. 1988.

[ADAMS 1] Adams, T., Christiano, P., and Hendrickson, C. "A Knowledge Base for Retaining Wall Rehabilitation Design." In Lambe, P. and Hansen, L., editors Design and Performance of Earth Retaining Structures, Geotechnical Special Publication No. 25, pages 125-138, ASCE. 1990.

[ADAMS 1] Adams, T., Hendrickson, C., and Christiano, P. "Computer Aided Rehabilitation Design." (AIEDAM), 5(2). to appear. 1991.

[ADAMS 1] Adams, T. "RETAIN: "An Integrated Knowledge Based System for Retaining Wall Rehabilitation Design." Tech. Report R-89-181, Dept. of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA, Aug. 1989.

[ADAMS 1] Ciarico, A., Adams, T., and Hendrickson, C. "A Cost Estimating Module to Aid Integrated Knowledge-Based Preliminary Design." In Barnwell, Jr., T. O., editor, Computing in Civil Engineering: Computers in Engineering Practice, pages 52-59, ASCE, Atlanta, GA, Sept. Proceedings of the Sixth Conference.

[JOSEPH 1] Joseph E. Bowles. "Foundation Analysis and Design." Third Edition, McGraw-Hill Book Company, 1982.

[RANDALL 1] Randall McMullan. "Dictionary of Building." Nichols Publishing 1988.

Appendix I

A. Examples of the programming by using several tools

A. 1 HyperCard and Excel on a Macintosh

Considering the members, there should be both domain engineers (specialists of TRW) and system engineers for a project to produce an Expert System for real usage. For a domain engineer, it seems to be very important to explain ideas and procedures that are being done daily. HyperCard is a great tool to propose ideas and to practice simple tests. There is sometimes a big gap in understanding between domain engineers and system engineers. By using HyperCard, even a domain engineer can propose ideas based on models to system engineers. Such a presentation can help a domain engineer to explain images effectively and certainly to system engineers. To bridge the gap is one of the most important factors to produce an excellent Expert system.

HyperCard and Excel were used to formulate ideas and to practice simple tests. The system that is mentioned in Chapter 3 is produced by HyperCard and Excel on a Macintosh. The graphics (for input and output) are shown in Figure 3-10, Figure A1-1 to A1-3, and Table 3-3. The lists of programming (written by HyperTalk) are in Appendix II.

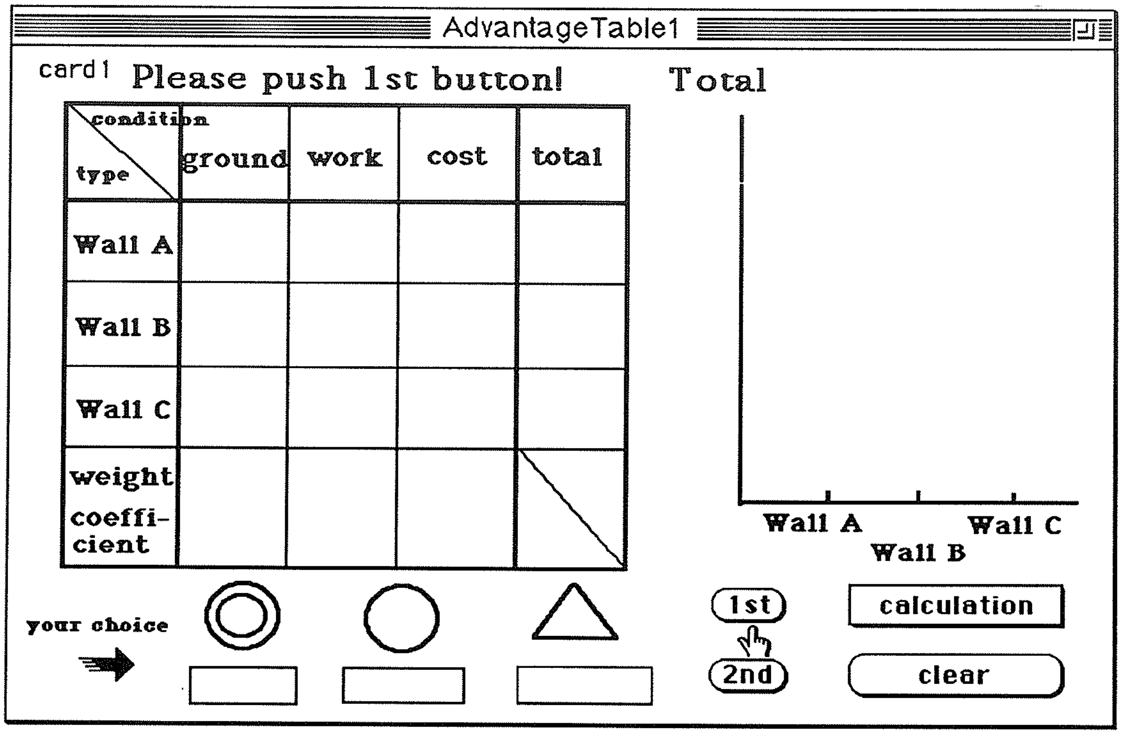


Figure A1-1 User Interface by HyperCard (Step1)

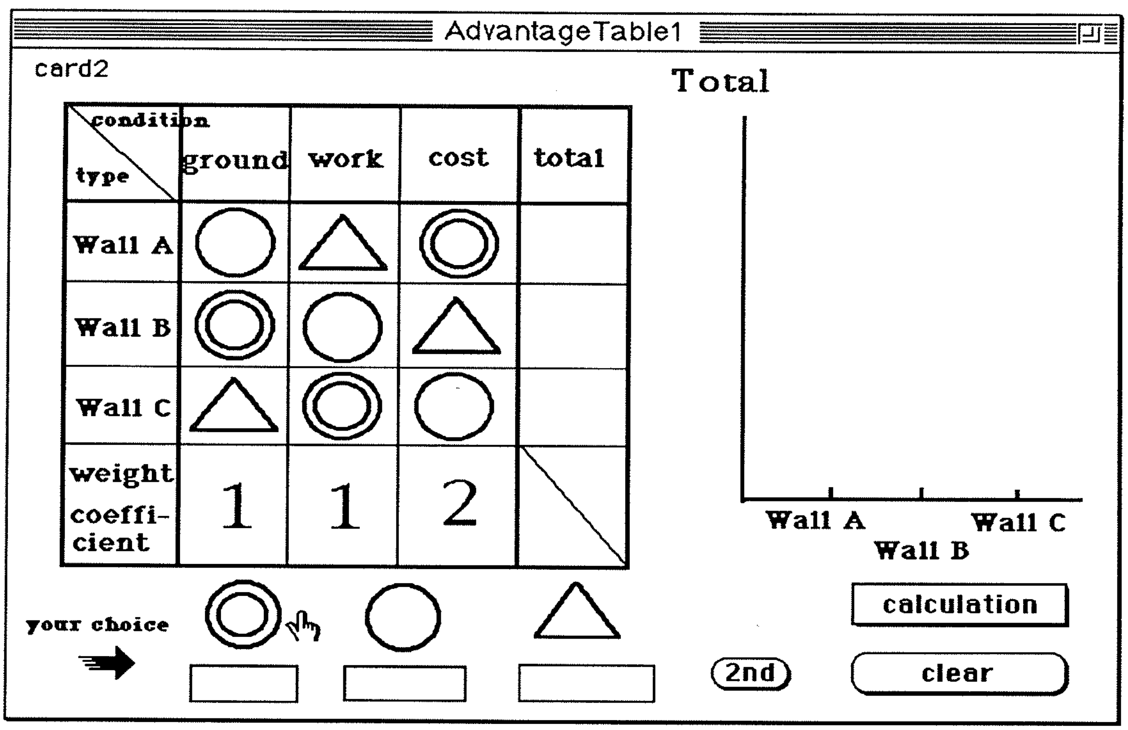


Figure A1-2 User Interface by HyperCard (Step2)

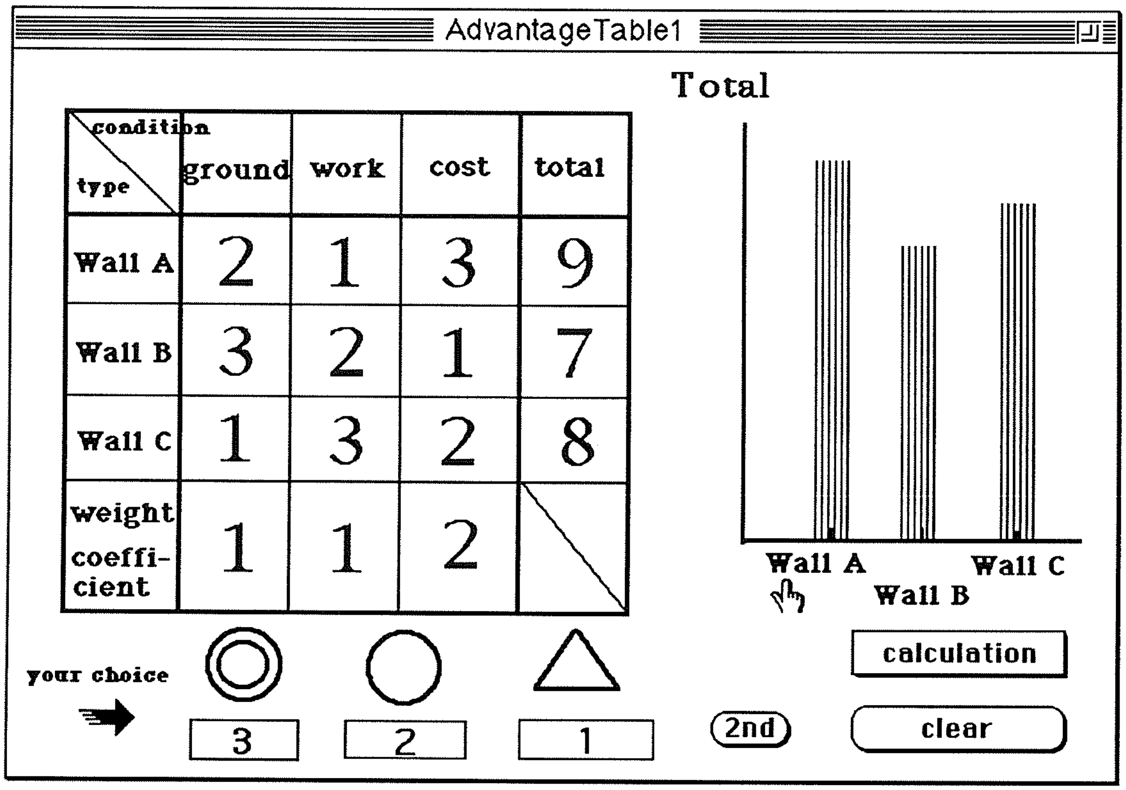


Figure A1-3 User Interface by HyperCard (Step3)

A. 2 Prokappa on an engineering work station

Prokappa is a tool for expert system development based on Object Oriented Programming. Prokappa is also useful for this system. For the first step, make an initial prototype model in Prokappa on Sun work station. The prototype model was built to analyze structure of Retaining Walls because of familiarity with structural designs Most of the existing programs for structural designs of TRW are not written using Object Oriented methods. By using this method, modifications to the model can easily change the attributes of each object.

The model is shown in Figure A2-1 to A2-4. User Interface is also shown in Figure A2-5 to A2-9. The lists of programming are in Appendix II. It is written by Protalk (which is a Prokappa's particular language based on C).

The model is simplified as follows:

(1) The kind of walls -- only steel sheet pile (Type I, III, IV, V);

(2) The soil -- no ground water, one layer, all sand; and

(3) Others -- no struts, no load on the ground.

The steps of calculation are as follows;

(1) Input-- ø: angle of internal friction r: unit weight, depth of excavation;

(2) Calculate -- active(and passive) earth pressure;

(3) Calculate -- active(and passive) side stability moment;

(4) Iterate-- select the length of the wall(penetration depth);

(5) Calculate -- active side bending moment;

(6) Calculate -- stress of the wall;

(7) Find -- section modulus which is required;

(8) Select -- best wall class (which has reasonable section modulus);

(9) Calculate -- weight of the wall; and

(10) Output -- the best class of wall, the length and weight of the wall.

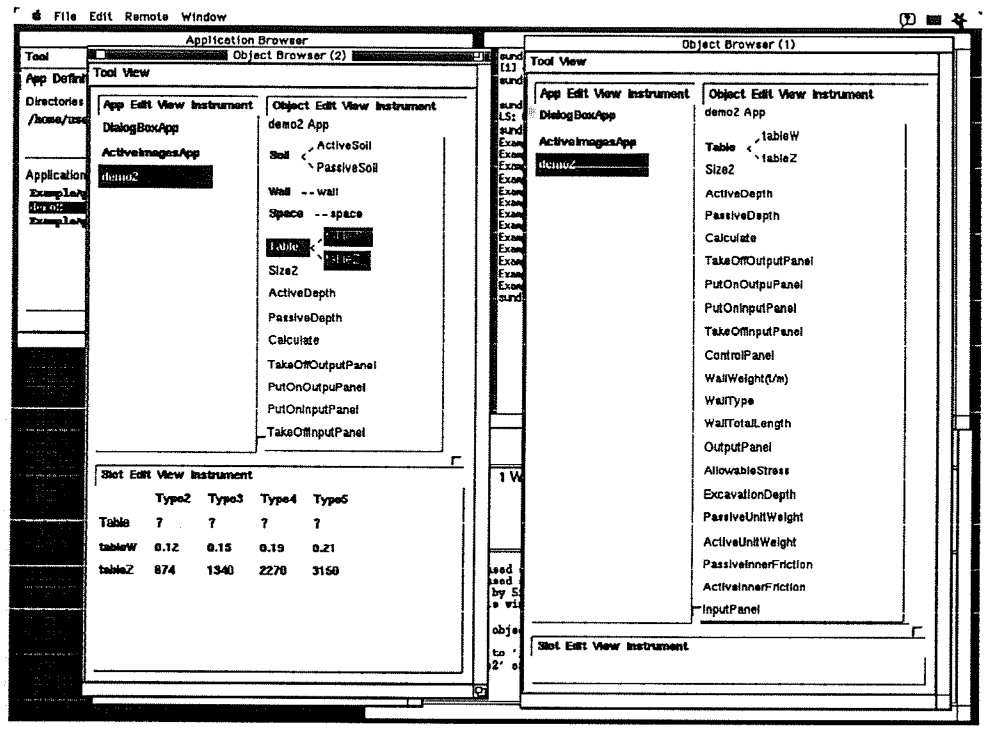


Figure A2-1 The model by Prokappa [Object Browser (1) and (2)]

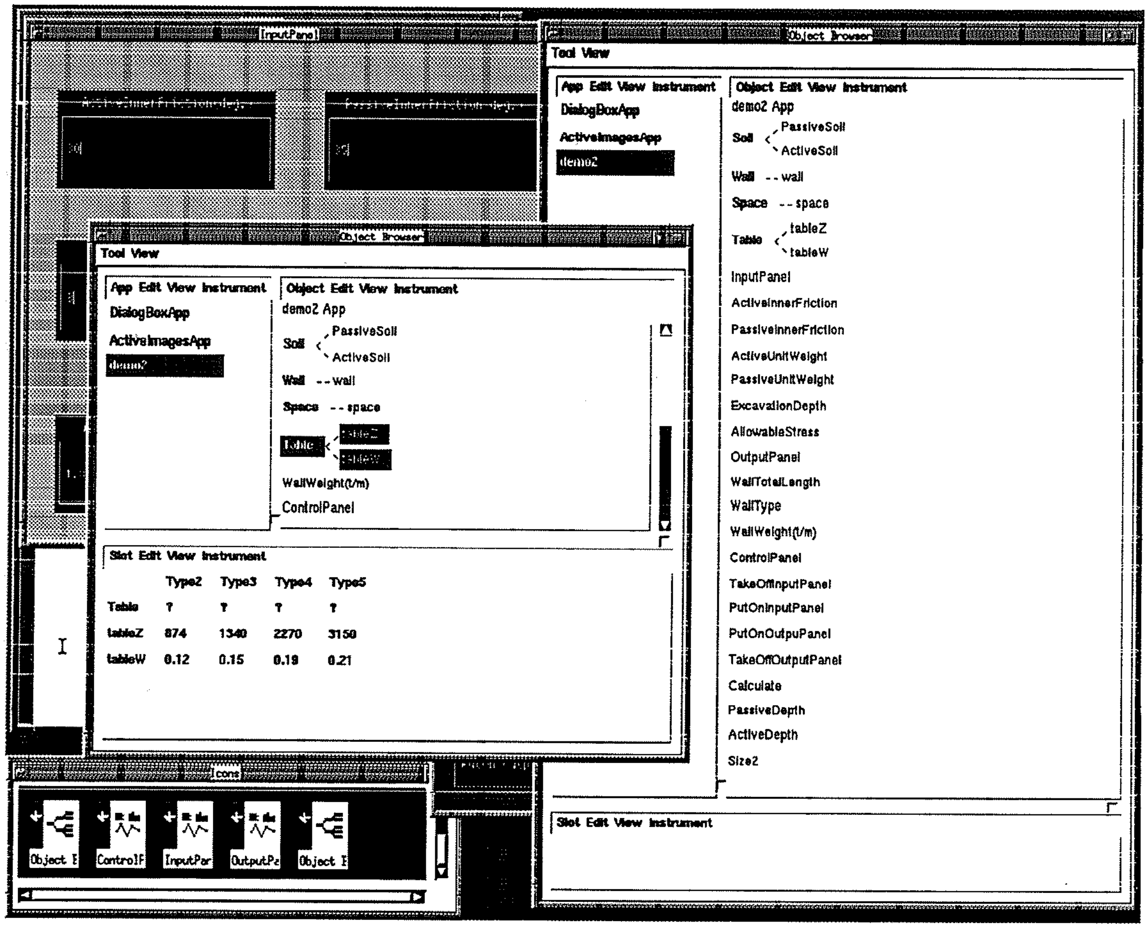


Figure A2-2 The model by Prokappa [Object Browser (1) and (2)]

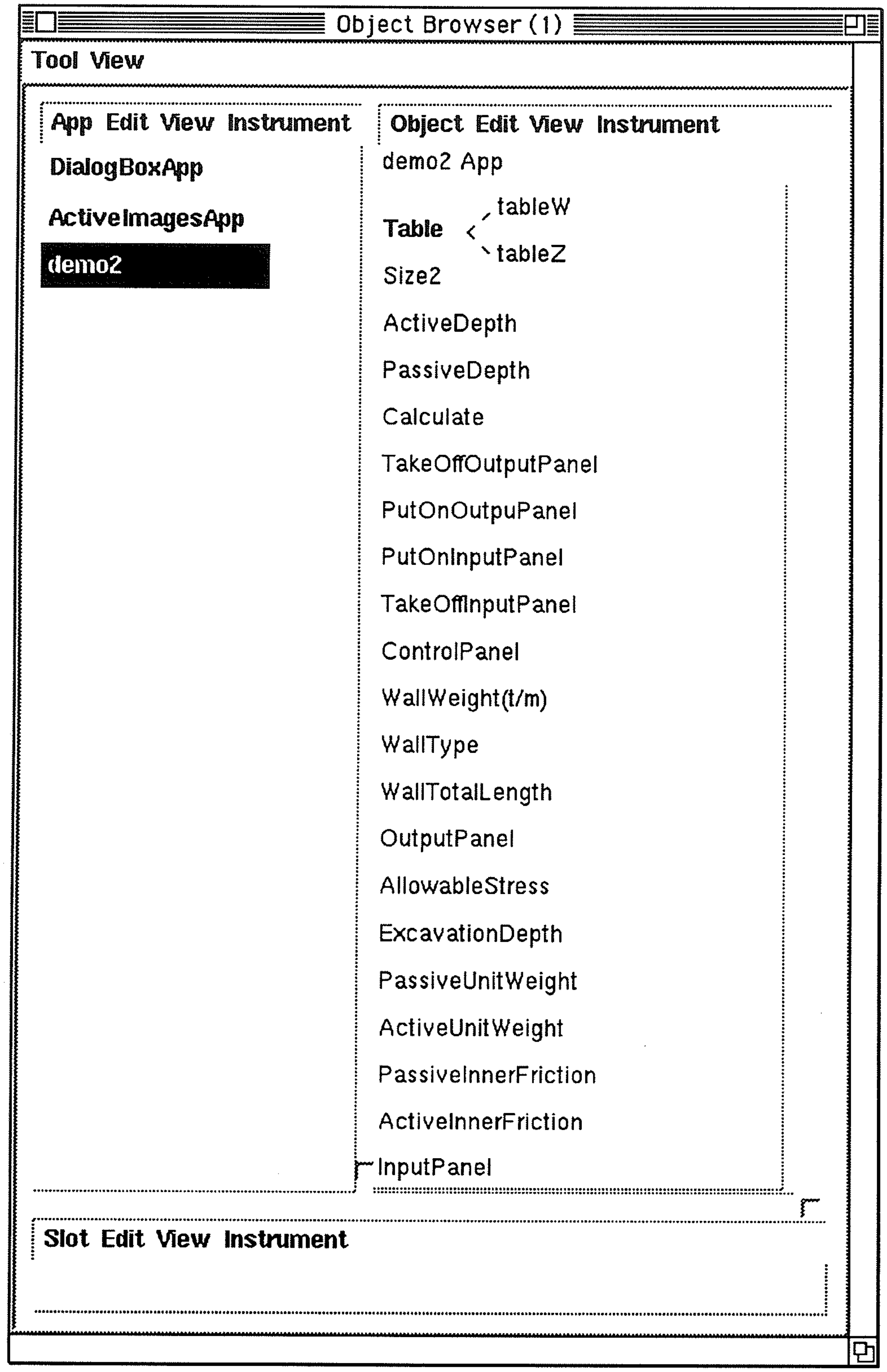


Figure A2-3 The model by Prokappa [Object Browser (1)]

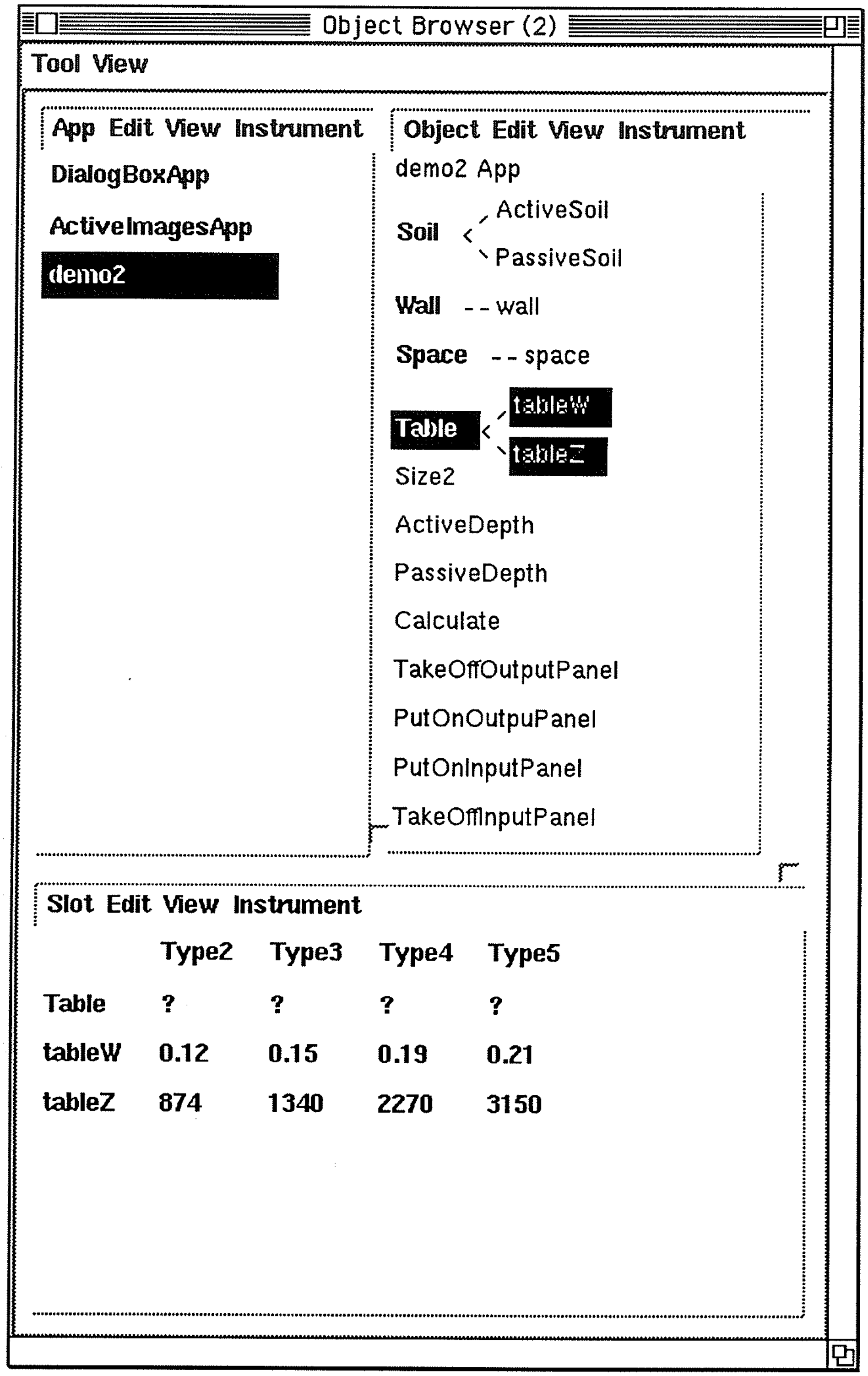


Figure A2-4 The model by Prokappa [Object Browser (2)]

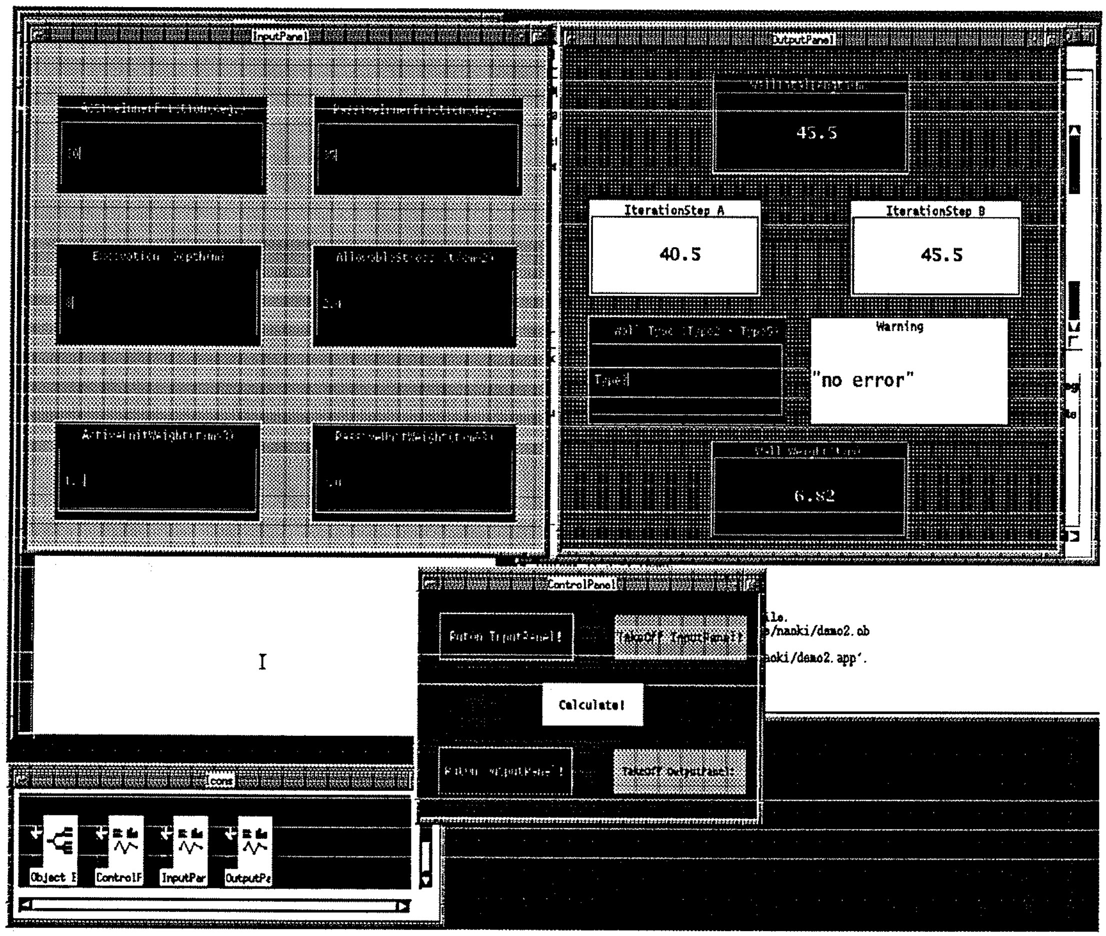


Figure A2-5 User Interface by Prokappa [all]

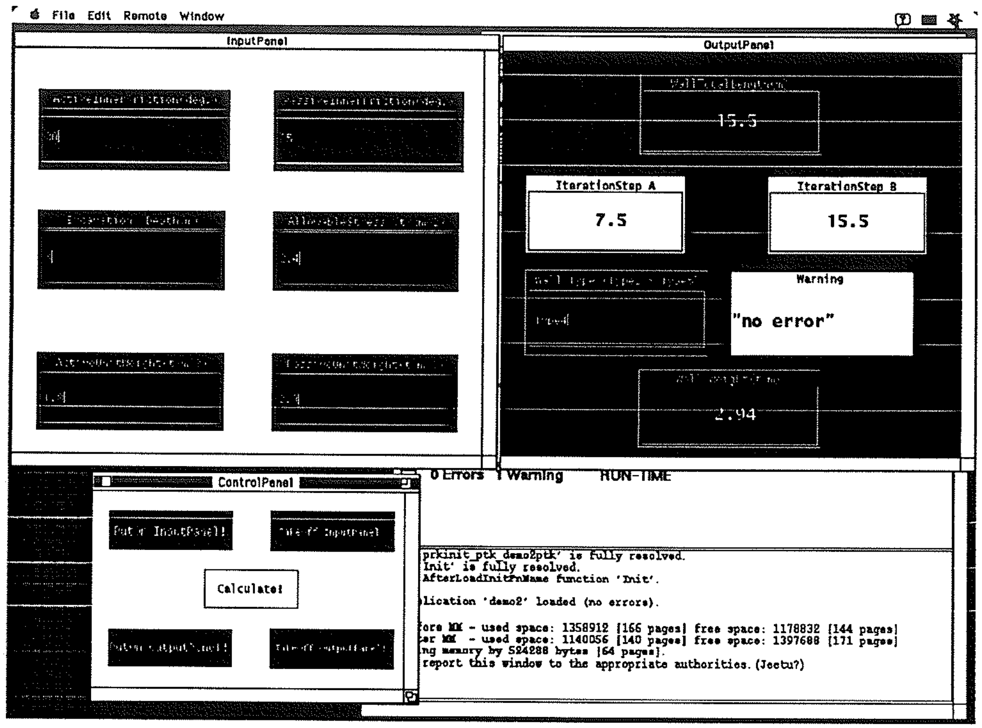


Figure A2-6 User Interface by Prokappa [all]

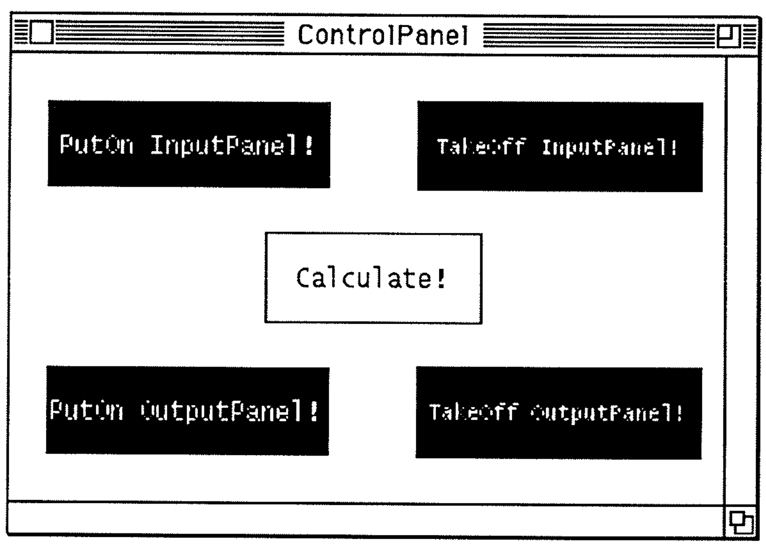


Figure A2-7 User Interface by Prokappa [Control Panel]

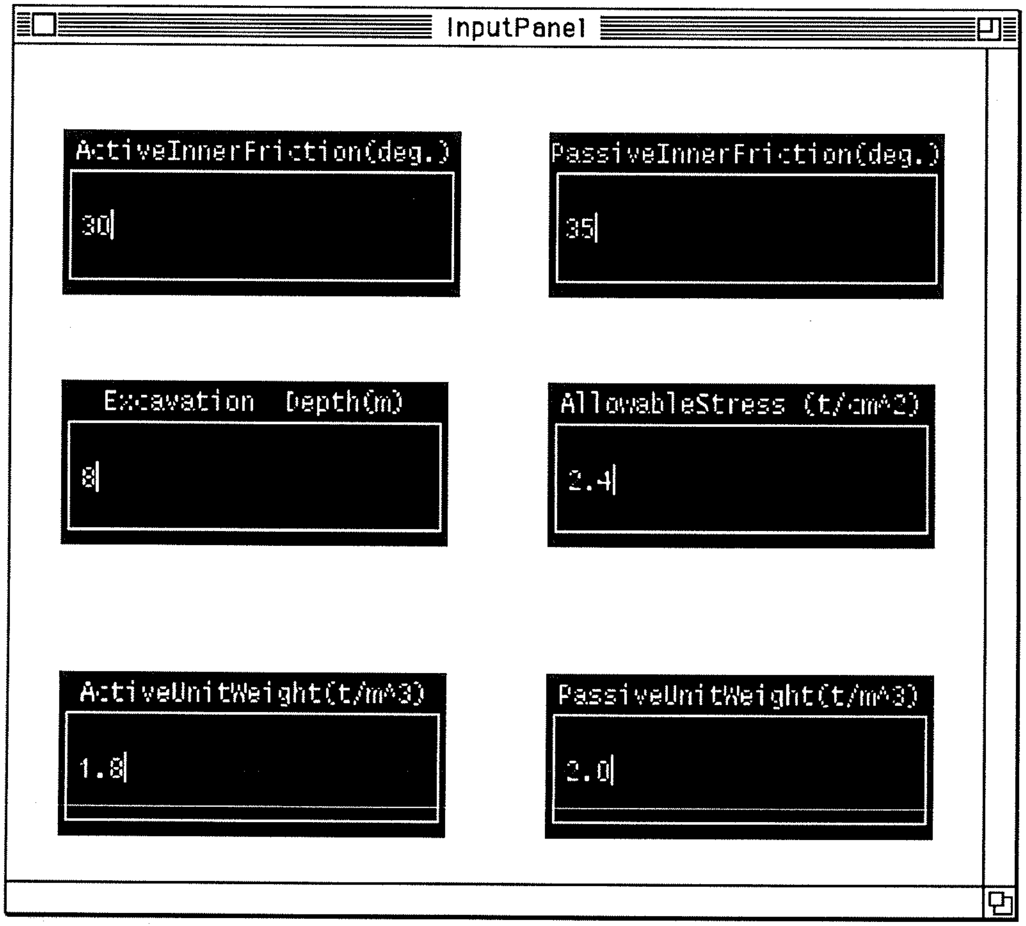


Figure A2-8 User Interface by Prokappa [Input Panel]

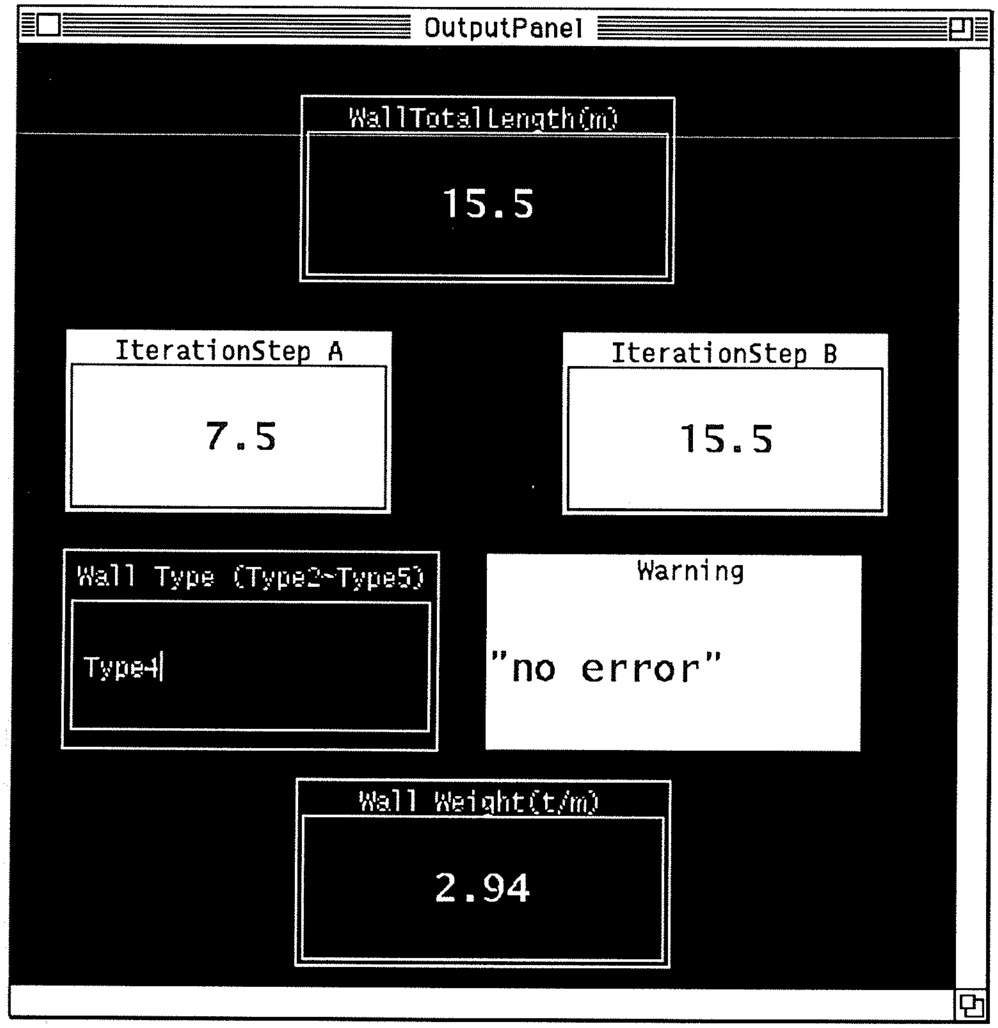


Figure A2-9 User Interface by Prokappa [Output Panel]

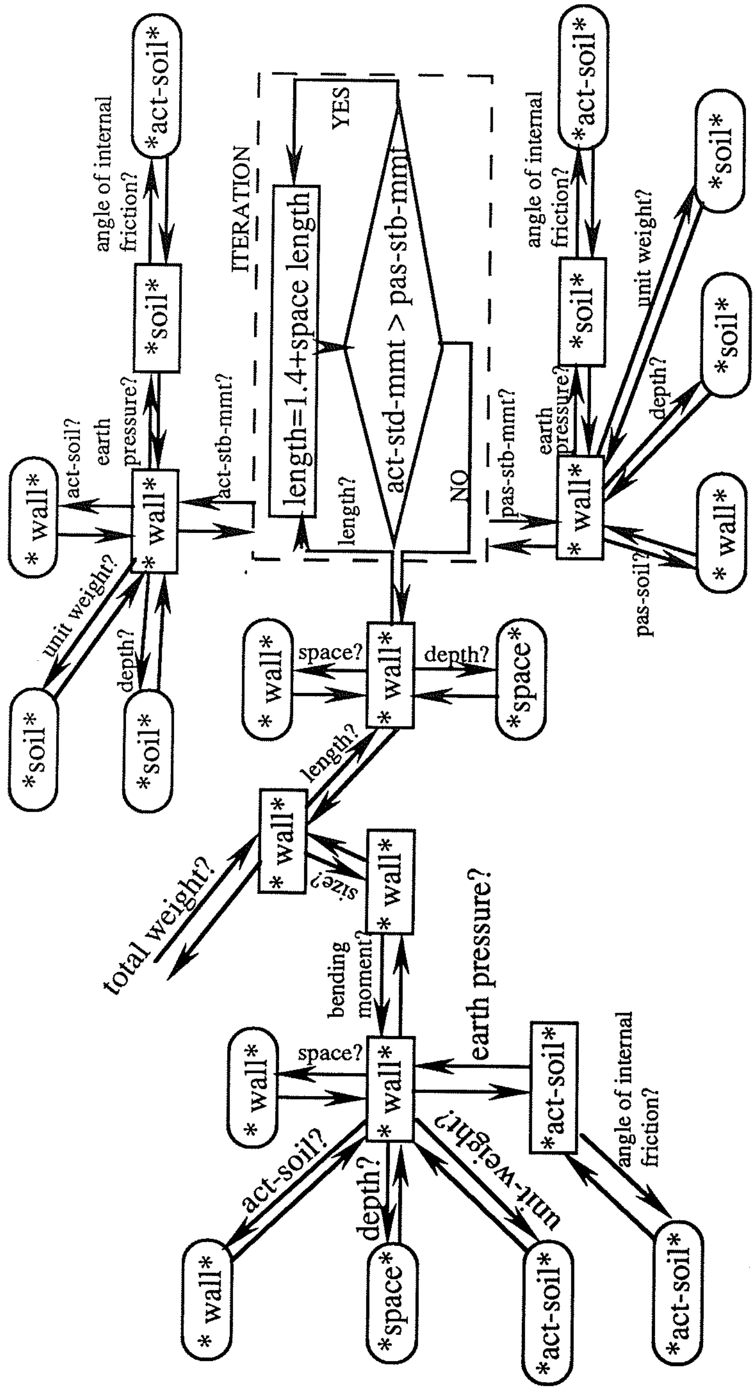
A. 3 Common Lisp on a Macintosh

This programming is implemented by Mr. Hiroyuki Fuyama[3]. This was by using Common Lisp on the Macintosh. The model is shown in Figure A3-1. The User Interface is shown in Figure A3-3 to A3-5. Search method is applied to find penetration depths of walls as Figure A3-5. The model is also very easy to modify by changing attributes of any object. The remarkable point is the User Interface. As the wall itself is one of the objects, the object has a value of the length of wall as its attribute during the steps of calculation of penetration depths. It is fairly easy to show the length of walls during the calculation on the User Interface like Figure A3-3. The lists of programming are in Appendix II.

A. 4 Others

The modeling by HyperCard and Excel on Macintosh is fairly easy and effective even for beginners. By using a tool for expert systems based on Object Oriented Programming like Prokappa, expert systems can be produced for TRW, but these are a little difficult for beginners. Considering the existing state of things, system engineers (not domain engineers) might be able to deal with these tools effectively for real projects to produce Expert Systems for real usage. An important thing that must be recognized is that the Object Oriented Programming is fairly useful for the Temporary Retaining Wall Expert System.

3 Hiroyuki Fuyama, Ph. D. candidate, Stanford University, CA.



: Objects that have an answer to the question

: Objects that do not have an answer to the question

Figure A3 -1 Message Exchanges between Objects

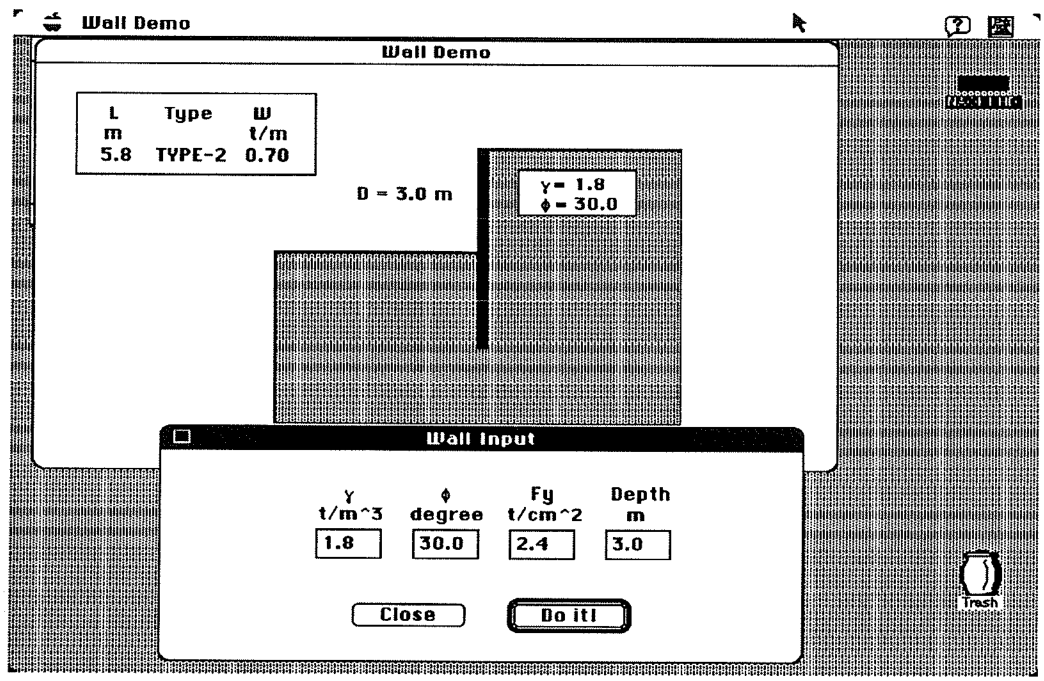


Figure A3-2 User Interface on a Macintosh [all]

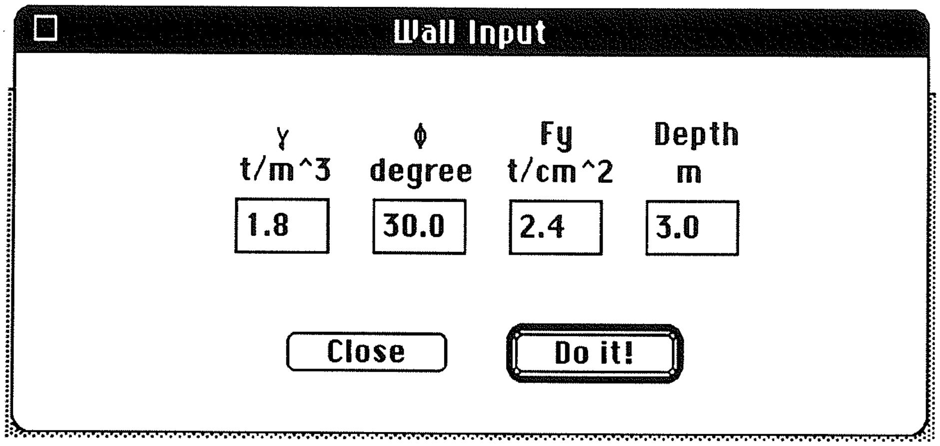


Figure A3-3 User Interface on a Macintosh [In Put]

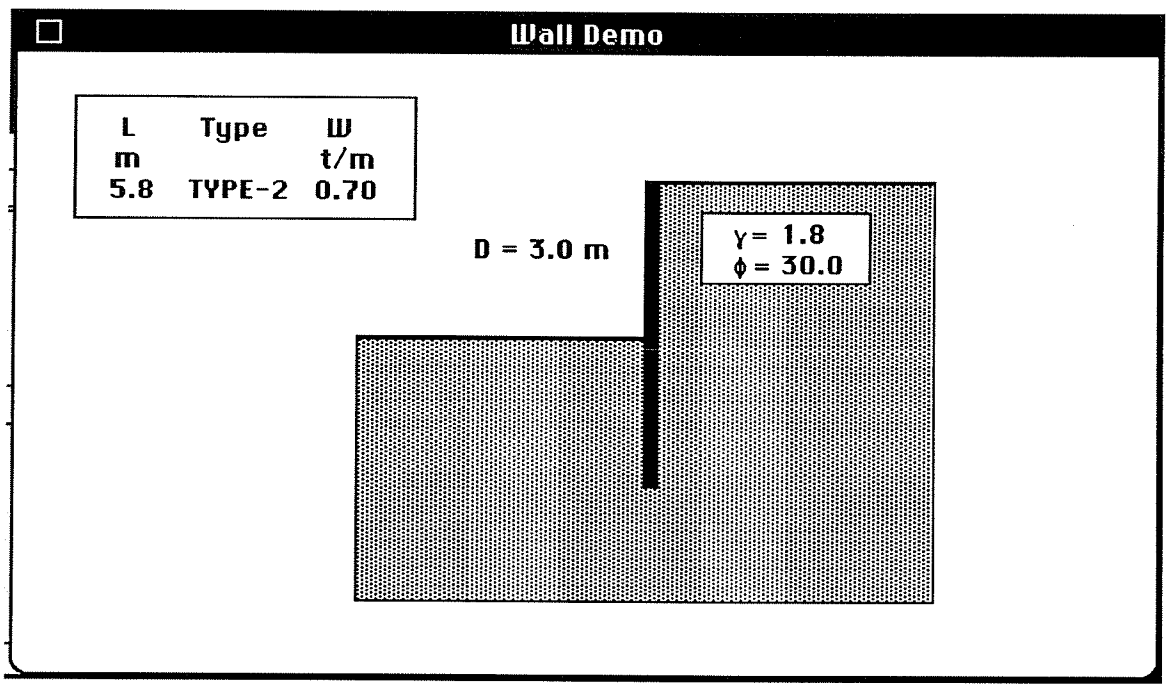


Figure A3-4 User Interface on a Macintosh [Out Put]

Appendix II

B. Lists of Programming

B. 1 HyperTalk by HyperCard on Macintosh

B. 1. 1 Fuzzy System (Fuzzy1)

--- Fuzzy1 bkgnd button id1 = "Draw Membership Function" ---

```
on mouseUp

  put card field "pointA" into pa
  put card field "pointB" into pb
  put card field "pointC" into pc
  put card field "pointD" into pd

  -- seisuuka of zahyouten --

  put round(43 + 9.8 * pa) into paa
  put round(43 + 9.8 * pb) into pbb
  put round(361 + 9.8 * pc) into pcc
  put round(361 + 9.8 * pd) into pdd

  -- draw graphic --

  choose brush tool
  set the brush to 7

  drag from 43,137 to  paa,137
  drag from  paa,137 to  pbb,249
  drag from  pbb,249 to 258,249

  drag from 361,249 to pcc,249
  drag from  pcc,249 to  pdd,137
  drag from  pdd,137 to 576,137

  choose browse tool


end mouseUp
```

--- Fuzzy1 bkgnd button id2 = "calculate grade1" ---

```
on mouseUp

  --  naming of hensuu   --

  put card field "pointA" into pa
  put card field "pointB" into pb
  --put card field "pointC" into pc
  --put card field "pointD" into pd

  put card field "excavationDepth" into ed

  -- excavation depth < pointA --

  if ed <= pa  then
    put 1 into card field "grade1"
    exit mouseUp

    -- pointA < excavation depth < pointB

  else
    if (ed > pa) and (ed < pb)  then

      --  cal. of katamuki --

      put (43 + 9.8 * pa) into a1
      put (43 + 9.8 * pb) into a2
      put 112/(a2-a1) into k

      -- cal. of grade --

      put (43 + 9.8 * ed) into x1
      put (k * (x1 - a1) + 137) into y1
      put (249 - y1)/112 into y2
      put y2 into card field "grade1"
      exit mouseUp

      -- pointB < excavation depth

    else  put 0 into card field "grade1"

  end if


end mouseUp
```

--- Fuzzy1 bkgnd button id 9 = "calculate grade2" ---

```
on mouseUp

  --  naming of hensuu   --

  --put card field "pointA" into pa
  --put card field "pointB" into pb
  put card field "pointC" into pc
  put card field "pointD" into pd

  put card field "excavationDepth" into ed

  -- excavation depth < pointC --

  if ed <= pc  then
    put 0 into card field "grade2"
    exit mouseUp

    -- pointC < excavation depth < pointD --

  else
    if (ed > pc) and (ed < pd)  then

      --  cal. of katamuki --

      put (361 + 9.8 * pc) into a11
      put (361 + 9.8 * pd) into a22
      put -112/(a22-a11) into k1

      -- cal. of grade --

      put (361 + 9.8 * ed) into x11
      put (k1 * (x11 - a11) + 249) into y11
      put (249 - y11)/112 into y22
      put y22 into card field "grade2"
      exit mouseUp

      -- pointD < excavation depth

    else  put 1 into card field "grade2"

  end if

end mouseUp
```

--- Fuzzy1 bkgnd button id 3 = "clear" ---

```
on mouseUp

  -- clear text --

  delete first line of card field "pointA"
  delete first line of card field "pointB"
  delete first line of card field "pointC"
  delete first line of card field "pointD"
  delete first line of card field "excavationDepth"
  delete first line of card field "grade1"
  delete first line of card field "grade2"

  -- clear graphic --

  choose eraser tool

  repeat with y1 = 0 to 12
    drag from 43,(137 + 10 * y1) to 579,(137 + 10 * y1)
  end repeat

  choose browse tool

end mouseUp
```

B. 1. 2 Advantage Table (Advantage Table1)

```
--- Advantage Table1 bkgnd button id 7 = "clear" ---

-- Advantage Table1 --

on mouseUp

  -- clear total A, B, C --

  delete first char of bkgnd field "A1"
  delete first char of bkgnd field "A2"
  delete first char of bkgnd field "A3"
  delete first char of bkgnd field "B1"
  delete first char of bkgnd field "B2"
  delete first char of bkgnd field "B3"
  delete first char of bkgnd field "C1"
  delete first char of bkgnd field "C2"
  delete first char of bkgnd field "C3"

  -- clear total A, B, C --

  delete first char of bkgnd field "weight of best"
  delete first char of bkgnd field "weight of maru"
  delete first char of bkgnd field "weight of sankaku"

  delete first char of card field "totalA"
  delete first char of card field "totalB"
  delete first char of card field "totalC"

  delete first char of bkgnd field "weight1"
  delete first char of bkgnd field "weight2"
  delete first char of bkgnd field "weight3"

  -- clear graphic --

  choose eraser tool
  drag from 380,0 to 380,231
  drag from 420,0 to 420,231
  drag from 466,0 to 466,231
  choose browse tool


  go to card id 3616
  -- to card2
  delete first char of bkgnd field "weight of best"
  delete first char of bkgnd field "weight of maru"
  delete first char of bkgnd field "weight of sankaku"

  go to card id 2934
  -- to card1
  delete first char of bkgnd field "weight of best"
  delete first char of bkgnd field "weight of maru"
```

```
    delete first char of bkgnd field "weight of sankaku"

  end mouseUp
```

--- Advantage Table1 bkgnd button id 6 = "calculation" ---

```
on mouseUp

  -- the result of cal(total) to field of  card3 --

  put bkgnd field "weight1" into dw1
  put bkgnd field "weight2" into dw2
  put bkgnd field "weight3" into dw3

  put bkgnd field "A1" into da1
  put bkgnd field "A2" into da2
  put bkgnd field "A3" into da3
  put bkgnd field "B1" into db1
  put bkgnd field "B2" into db2
  put bkgnd field "B3" into db3
  put bkgnd field "C1" into dc1
  put bkgnd field "C2" into dc2
  put bkgnd field "C3" into dc3

  put da1 * dw1 + da2 * dw2 + da3 * dw3 into card field
"totalA"
  put db1 * dw1 + db2 * dw2 + db3 * dw3 into card field
"totalB"
  put dc1 * dw1 + dc2 * dw2 + dc3 * dw3 into card field
"totalC"

  -- apper grafh --

  put card field "totalA" into ya
  put card field "totalB" into yb
  put card field "totalC" into yc

  choose brush tool
  set the brush to 30
  drag from 380,230 to 380,(231 - 20 * ya)
  drag from 420,230 to 420,(231 - 20 * yb)
  drag from 466,230 to 466,(231 - 20 * yc)
  choose browse tool

end mouseUp
```

--- Advantage Table1 card id 4372 = "card3" ---

```
on openCard

  -- global hensuu no hikitori from card2 --

  global
da1,da2,da3,db1,db2,db3,dc1,dc2,dc3,dw1,dw2,dw3,b1,m1,s1

  put da1 into bkgnd field "A1"
  put da2 into bkgnd field "A2"
  put da3 into bkgnd field "A3"
  put db1 into bkgnd field "B1"
  put db2 into bkgnd field "B2"
  put db3 into bkgnd field "B3"
  put dc1 into bkgnd field "C1"
  put dc2 into bkgnd field "C2"
  put dc3 into bkgnd field "C3"

  put dw1 into bkgnd field "weight1"
  put dw2 into bkgnd field "weight2"
  put dw3 into bkgnd field "weight3"

  put b1 into bkgnd field "weight of best"
  put m1 into bkgnd field "weight of maru"
  put s1 into bkgnd field "weight of sankaku"

end openCard
```

--- Advantage Table1 bkgnd button id 27 = "2nd" ---

```
on mouseUp

  --naru sankaku ireta suuji no hyoji on table--

  global
da1,da2,da3,db1,db2,db3,dc1,dc2,dc3,dw1,dw2,dw3,b1,m1,s1

  put bkgnd field "weight of best" into da3
  put bkgnd field "weight of best" into db1
  put bkgnd field "weight of best" into dc2
  put bkgnd field "weight of maru" into da1
  put bkgnd field "weight of maru" into db2
  put bkgnd field "weight of maru" into dc3
  put bkgnd field "weight of sankaku" into da2
  put bkgnd field "weight of sankaku" into db3
```

```
    put bkgnd field "weight of sankaku" into dc1

    put bkgnd field "weight of best" into b1
    put bkgnd field "weight of maru" into m1
    put bkgnd field "weight of sankaku" into s1

    put bkgnd field "weight1" into dw1
    put bkgnd field "weight2" into dw2
    put bkgnd field "weight3" into dw3

    --move card2 to card3 maru disapper--

    visual effect dissolve slow
    go to card id 4372

end mouseUp
```

B. 2 Protalk and C on Prokappa

B. 2. 1 Demo2.app

```
#PrkDefn ProKappa : $Revision:       3.132 $
#
# Definition for: demo2
#
   Application: demo2

   CFiles                     = :demo2.pkc
   ProTalkFiles               = :demo2.ptk
   ProTalkCompileFlags   =
   LoadFlags                  =
   ObjectBase                 = :demo2.ob
   UserModules                =
   RequiredModules            = ActiveImagesApp
   AboutAppFile               =
   AfterLoadInitFnName   = Init RunFnName  =
```

B. 2. 2 Demo2.pkc

```
/*
 *
 * C method source file
 *
 */


#include <prk/cmethods.h>


/*-------------------------------------------------------------------------
--
 *
 * C method  --  Init
 *
 *       This is the default method, a simple tracer.
 *       It prints the name of the object and method slot.
 *       Note: if you add any arguments, remember that they (and the
return
 *             value) must be of type PrkType.
 *
```

```
 */

PrkType Init ()
/* PrkType Init (PrkObject self, PrkSymbol slot_name) */
{
/* Uncomment one of these if you don't use an argument.  PrkIgnore will
* prevent not-used complaints from the compiler or interpreter. */

     /* PrkIgnore (self); */
     /* PrkIgnore (slot_name); */
/*
     printf ("\nMethod:");
     printf ("\n  object  %s", (char *) self);
     printf  ("\n   slot     %s",  (char *)
     slot_name); printf ("\n");
*/


     PrkSendMsg(ControlPanel@,`PutOnScreen!
);

     return PrkNull;
}
```

B. 2. 3 Demo2.ptc

```
/*
 *
 * ProTalk method source file
 *
 */


#include <prk/lib.pth>
#include <prk/math.pth>


/*-----------------------------------------------------------------------
--
 *
 * ProTalk method  --  Soil.Calculate!
 *
 *        This is the default method, a simple tracer.
```

```
 *        It prints the name of the object and method slot.
 *
 */

method Soil.Calculate! ()
{
     /* Methods must always have all their inputs bound: */
     bound inputs;/*
     Print ("\nMethod:",
              "\n  object ", ?self,
              "\n  slot   ", ?slot,
              "\n");
*/
          if {ActiveSoil.InnerFriction == Null;}  /* if ? or
          not  */  {Print("Please  input  InnerFriction  of
          ActiveSoil.\n");
                    return Null;}

          if {PassiveSoil.InnerFriction == Null;} /* if ? or
          not  */  {Print("Please  input  InnerFriction  of
          PassiveSoil.\n");
                    return Null;}

                              wall.Length = 0.0;  wall.Weight
                    = 0.0;
          wall.Size = none;  wall.Size2 ="no error";

?pi = Acos(-1);
/*          Step  1    Calculation  of  ka  :  earth  pressure  coefficient
ActiveSide */
?AngleA = (?pi / 4) - ActiveSoil.InnerFriction * ?pi / 180 / 2;
?tanA = Tan(?AngleA);
?ka = ?tanA*?tanA;
ActiveSoil.k = ?ka;

/*          Step  2    Calculation  of  kp  :  earth  pressure  coefficient
PassiveSide */
?AngleP = (?pi / 4) + PassiveSoil.InnerFriction * ?pi / 180 / 2;
?tanP = Tan(?AngleP);
?kp = ?tanP*?tanP;
PassiveSoil.k = ?kp;

/*         Step  3          Calculation  of  BendingMoment  ;Mb  Wall
*/

?Mb = (ActiveSoil.k * ActiveSoil.UnitWeight * space.depth * space.depth
          * space.depth) / 6;
wall.BendingMoment = ?Mb;

/*        Step  4     Calculation  of  PenetrationDepth     iteration
*/
/*        Step 4-1  ActiveStabilityMoment  Wall  */
/*        Step  4-2   PassiveStabilityMoment
Wall  */
/*     Step 4-3   Iteration   */
```

```
PassiveSoil.Depth  =  PassiveSoil.InitDepth;
ActiveSoil.Depth  =  PassiveSoil.Depth  +
space.depth; SendMsg(Soil,CalcMoment!);

/*      Step  5   Calculation  of  RequestedZ
*/

?Zr  =  wall.BendingMoment  *  100  /  wall.Fy;
wall.RequestedZ = ?Zr;
if ?Zr > Max(tableZ.Type2,tableZ.Type3,tableZ.Type4,tableZ.Type5);
{
          Print("You need over Type5 of SheetPile.\n");
          wall.Size = OUT;
          wall.Size2 = "OUT!(over Type5)";
}
else {
          if ?Zr <= tableZ.Type5;
          wall.Size = Type5;
          if ?Zr <= tableZ.Type4;
          wall.Size = Type4;
          if ?Zr <= tableZ.Type3;
          wall.Size = Type3;
          if ?Zr <= tableZ.Type2;
          wall.Size = Type2;
}

/*    Step 6  Calculation of WallTotalLength    */

?Lt = space.depth + PassiveSoil.Depth;
wall.Length = ?Lt;

/*    Step 7  Calculation of WallWeight     */

if wall.Size != OUT;
{
          if wall.Size == Type2;
          ?w = tableW.Type2;
          if wall.Size == Type3;
          ?w = tableW.Type3;
          if wall.Size == Type4;
          ?w = tableW.Type4;
          if wall.Size == Type5;
          ?w = tableW.Type5;
          ?W = ?w * wall.Length;
          wall.Weight = ?W;
}

     return Null;

}


/*-----------------------------------------------------------------
--
 *
```

```
 * ProTalk method  --  Soil.CalcMoment!
 *
 *        This is the default method, a simple tracer.
 *        It prints the name of the object and method slot.
 *
 */

method Soil.CalcMoment! ()
{
     /* Methods must always have all their inputs bound: */
     bound inputs;
/*
     Print ("\nMethod:",
              "\n    object   ",
?self,
              "\n    slot     ",
              ?slot, "\n");
*/
?Ma1 = ActiveSoil.k * ActiveSoil.UnitWeight;
          ?Ma2 = ActiveSoil.Depth;
          ?Mas = ?Ma1 * ?Ma2 * ?Ma2 * ?Ma2 / 6.0;
?Mp1 = PassiveSoil.k * PassiveSoil.UnitWeight;
          ?Mp2 = PassiveSoil.Depth;
          ?Mps = ?Mp1 * ?Mp2 * ?Mp2 * ?Mp2 / 6.0;
          if ?Mas > ?Mps;
          {         PassiveSoil.Depth = PassiveSoil.Depth +
Soil.d;
                    ActiveSoil.Depth  = ActiveSoil.Depth +
                    Soil.d; SendMsg(Soil,CalcMoment!);
          }

     return Null;
}
```

B. 2 Common Lisp on a Macintosh

[Macintosh Common Lisp, Version 2.0, beta 1, patch 3, (MCL2.0b1p3)]

B. 2. 1 Calculations part (wall : source : wall)

```
;;*********************************************************************
;;                                 soil
;;*********************************************************************


;;            wall : wall.
;;            spce : space.
;;
;;            act-p : whether or not active soil.
;;
;;            unit-wght : unit weight (ton/m^3).
;;            innr-frct : inner friction (degree).
;;
;;            k : earth pressure coefficient.
;;
;;            dpth : depth (m).

(defclass soil ()
;;
  ((wall :accessor wall :initform nil)
   (spce :accessor spce :initform nil)
;;
   (act-p :accessor act-p :initarg :act-p)
;;
   (unit-wght :accessor unit-wght :initform 1.8 )
   (innr-frct :accessor innr-frct :initform 30.0)
;;
   (k :accessor k :initform nil)
;;
   (dpth :accessor dpth :initform nil)
;;
   ))


;;                              (k soil) :before
;;
;; *  fun  * stores soil earth pressure coefficient.

(defmethod k :before ((soil soil))
  (let ((val (slot-value soil 'k)))
    (unless val
      (setf (slot-value soil 'k)
            (get-soil-k soil))
      )))
```

```
;;                              (dpth soil) :before
;;
;; *  fun  * stores soil depth (m).

(defmethod dpth :before ((soil soil))
  (let ((val (slot-value soil 'dpth)))
    (unless val
      (setf (slot-value soil 'dpth)
            (get-soil-dpth soil))
      )))


;;=======================================================================


;;                              (get-soil-k soil)
;;
;; *  fun  * obtains soil earth pressure coefficient.
;; *  in   * soil : soil.

(defun get-soil-k (soil)
  (if (act-p soil)
    (expt (tan (- (/ pi 4) (* (innr-frct soil) 1/360 pi))) 2)
    (expt (tan (+ (/ pi 4) (* (innr-frct soil) 1/360 pi))) 2)))


;;                              (get-soil-dpth soil)
;;
;; *  fun  * obtains soil depth (m).
;; *  in   * soil : soil.

(defun get-soil-dpth (soil)
  (let ((wall (wall soil))
        (spce (spce soil)))
    (if (act-p soil)
      (lgth wall)
      (- (lgth wall) (dpth spce)))))


;;*********************************************************************
;;                                 wall
;;*********************************************************************


;;           act-soil : active  side soil.
;;           pas-soil : passive side soil.
;;               spce : excavated space.
;;
;;           bnd-mmt  : bending moment (t*m/m).
;;           req-z    : required section mudulus (cm^3/m).
;;           size     : size.
;;
;;           lgth : length (m).
```

```
;;            wght : weight (ton).
;;
;;            act-stb-mmt : active  side stability moment (t*m/m).
;;            pas-stb-mmt : passive side stability moment (t*m/m).

(defclass wall ()
;;
  ((act-soil :accessor act-soil :initform nil)
   (pas-soil :accessor pas-soil :initform nil)
   (    spce :accessor     spce :initform nil)
;;
   (fy      :accessor fy      :initform 2.4)
   (bnd-mmt :accessor bnd-mmt :initform nil)
   (req-z   :accessor req-z   :initform nil)
   (size    :accessor size    :initform nil)
;;
   (lgth :accessor lgth :initform nil)
   (wght :accessor wght :initform nil)
;;
   (act-stb-mmt :accessor act-stb-mmt :initform nil)
   (pas-stb-mmt :accessor pas-stb-mmt :initform nil)
;;
   ))


;;                                  (bnd-mmt wall) :before
;;
;; *  fun  * stores wall bending moment (t*m/m).

(defmethod bnd-mmt :before ((wall wall))
  (let ((val (slot-value wall 'bnd-mmt)))
    (unless val
      (setf (slot-value wall 'bnd-mmt)
            (get-wall-bnd-mmt wall))
      )))


;;                                  (req-z wall) :before
;;
;; *  fun  * stores wall required section modulust (cm^3/m).

(defmethod req-z :before ((wall wall))
  (let ((val (slot-value wall 'req-z)))
    (unless val
      (setf (slot-value wall 'req-z)
            (get-wall-req-z wall))
      )))


;;                                  (size wall) :before
;;
;; *  fun  * stores wall size (type w z).

(defmethod size :before ((wall wall))
  (let ((val (slot-value wall 'size)))
```

```
    (unless val
      (setf (slot-value wall 'size)
            (get-wall-size wall))
      )))


;;                              (lgth wall) :before
;;
;; *  fun  * obtains wall length (m) by iteration.

(defmethod lgth :before ((wall wall))
  (let ((val (slot-value wall 'lgth)))
    (unless val
      (get-wall-lgth wall)
      )))


;;                              (wght wall) :before
;;
;; *  fun  * stores wall weight (ton).

(defmethod wght :before ((wall wall))
  (let ((val (slot-value wall 'wght)))
    (unless val
      (setf (slot-value wall 'wght)
            (get-wall-wght wall))
      )))


;;                          (act-stb-mmt wall) :before
;;
;; *  fun  * stores wall active side stability moment (t*m/m).

(defmethod act-stb-mmt :before ((wall wall))
  (let ((val (slot-value wall 'act-stb-mmt)))
    (unless val
      (setf (slot-value wall 'act-stb-mmt)
            (get-wall-stb-mmt wall 'act))
      )))


;;                          (pas-stb-mmt wall) :before
;;
;; *  fun  * stores wall passive side stability moment (t*m/m).

(defmethod pas-stb-mmt :before ((wall wall))
  (let ((val (slot-value wall 'pas-stb-mmt)))
    (unless val
      (setf (slot-value wall 'pas-stb-mmt)
            (get-wall-stb-mmt wall 'pas))
      )))
```

```
;;                          (setf (lgth wall) lgth) :before
;;
;; *  fun  * clears depthes of active & passive side soils and active &
passive
;;            side stability moment.

(defmethod (setf lgth) :before (lgth (wall wall))
  (declare (ignore lgth))
  (let ((act-soil (act-soil wall))
        (pas-soil (pas-soil wall)))
    (setf (dpth act-soil) nil)
    (setf (dpth pas-soil) nil)
    (setf (wght wall) nil)
    (setf (act-stb-mmt wall) nil)
    (setf (pas-stb-mmt wall) nil)))



;;                          (setf (lgth wall) lgth) :after
;;
;; *  fun  * draws wall section.

(defmethod (setf lgth) :after (lgth (wall wall))
  (when lgth
    (let ((wnd *wall-wnd*)
          (act-soil (act-soil wall))
          (pas-soil (pas-soil wall))
          (spce     (spce     wall)))
      (draw-wall wnd wall)
      (draw-soil wnd act-soil)
      (draw-soil wnd pas-soil)
      (draw-lower-soil wnd  wall)
      (write-wall-rslt wnd wall)
      (write-spce-inp wnd spce)
      (dotimes (i 50000))
      (erase-rect wnd  40 60  70 70)
      (erase-rect wnd 130 60 160 70)
      )))


;;==================================================================


;;                          (get-wall-bnd-mmt wall)
;;
;; *  fun  * obtains wall bending momnet (t*m/m).
;; *  in   * wall : wall.

(defun get-wall-bnd-mmt (wall)
  (let* ((    spce (    spce wall))
         (act-soil (act-soil wall))
         (dpth (dpth spce))
         (unit-wght (unit-wght act-soil))
         (k         (k         act-soil)))
    (* 1/6 k unit-wght (expt dpth 3))))
```

```
;;                         (get-wall-stb-mmt wall soil-type)
;;
;; *  fun   * obtains wall stability moment (t*m/m).
;; *  in    * wall : wall.
;;            soil-type : ('act,'pas).

(defun get-wall-stb-mmt (wall soil-type)
  (let* ((soil (case soil-type
                 (act (act-soil wall))
                 (pas (pas-soil wall))))
         (dpth      (      dpth soil))
         (unit-wght (unit-wght soil))
         (k         (k          soil)))
    (* 1/6 k unit-wght (expt dpth 3))))


;;                               (get-wall-req-z wall)
;;
;; *  fun   * obtains wall required section modulus (cm^3/m).
;; *  in    * wall : wall.

(defun get-wall-req-z (wall)
  (let ((bnd-mmt (bnd-mmt wall))
        (fy (fy wall)))
    (/ (* bnd-mmt 100) fy)))


;;                               (get-wall-size wall)
;;
;; *  fun   * obtains wall size (type w z).
;; *  in    * wall : wall.

(defun get-wall-size (wall)
  (let ((req-z (req-z wall)))
    (dolist (wall *wall-list*)
      (let ((z (nth 2 wall)))
        (when (< req-z z)
          (return wall))))))


;;                               (get-wall-lgth wall)
;;
;; *  fun   * obtains wall length (m).
;; *  in    * wall : wall.

(defun get-wall-lgth (wall)
  (let* ((spce (spce wall))
         (spce-dpth (dpth spce))
         (lgth      (+ 1.4 spce-dpth))
         (un-stb-p t))
    (loop while un-stb-p do
          (setf lgth (+ lgth 0.1))
          (setf (lgth wall) lgth)
```

```
          (setf un-stb-p (< (pas-stb-mmt wall) (act-stb-mmt wall)))
          )))


;;                              (get-wall-wght wall)
;;
;; *  fun  * obtains wall weight (ton/m).
;; *  in   * wall : wall.

(defun get-wall-wght (wall)
  (let* ((size (size wall))
         (unit-wght (nth 1 size))
         (lgth (lgth wall)))
    (* unit-wght lgth)))


;;*******************************************************************
;;                                space
;;*******************************************************************


;;            dpth : depth (m).

(defclass spce ()
;;
  ((dpth :accessor dpth :initform 3.0)
;;*******************************************************************
;;                              soil
;;*******************************************************************


;;            wall : wall.
;;            spce : space.
;;
;;            act-p : whether or not active soil.
;;
;;            unit-wght : unit weight (ton/m^3).
;;            innr-frct : inner friction (degree).
;;
;;            k : earth pressure coefficient.
;;
;;            dpth : depth (m).

(defclass soil ()
;;
  ((wall :accessor wall :initform nil)
   (spce :accessor spce :initform nil)
;;
   (act-p :accessor act-p :initarg :act-p)
;;
   (unit-wght :accessor unit-wght :initform 1.8 )
   (innr-frct :accessor innr-frct :initform 30.0)
;;
   (k :accessor k :initform nil)
```

```
    (k :accessor k :initform nil)
;;
    (dpth :accessor dpth :initform nil)
;;
   ))


;;                              (k soil) :before
;;
;; *  fun  * stores soil earth pressure coefficient.

(defmethod k :before ((soil soil))
  (let ((val (slot-value soil 'k)))
    (unless val
      (setf (slot-value soil 'k)
            (get-soil-k soil))
      )))


;;                              (dpth soil) :before
;;
;; *  fun  * stores soil depth (m).

(defmethod dpth :before ((soil soil))
  (let ((val (slot-value soil 'dpth)))
    (unless val
      (setf (slot-value soil 'dpth)
            (get-soil-dpth soil))
      )))


;;======================================================================


;;                              (get-soil-k soil)
;;
;; *  fun  * obtains soil earth pressure coefficient.
;; *  in   * soil : soil.

(defun get-soil-k (soil)
  (if (act-p soil)
    (expt (tan (- (/ pi 4) (* (innr-frct soil) 1/360 pi))) 2)
    (expt (tan (+ (/ pi 4) (* (innr-frct soil) 1/360 pi))) 2)))


;;                              (get-soil-dpth soil)
;;
;; *  fun  * obtains soil depth (m).
;; *  in   * soil : soil.

(defun get-soil-dpth (soil)
  (let ((wall (wall soil))
        (spce (spce soil)))
```

```
     (if (act-p soil)
       (lgth wall)
       (- (lgth wall) (dpth spce)))))


;;**********************************************************************
;;                              wall
;;**********************************************************************


;;            act-soil : active  side soil.
;;            pas-soil : passive side soil.
;;                spce : excavated space.
;;
;;            bnd-mmt  : bending moment (t*m/m).
;;            req-z    : required section mudulus (cm^3/m).
;;            size     : size.
;;
;;            lgth : length (m).
;;            wght : weight (ton).
;;
;;            act-stb-mmt : active  side stability moment (t*m/m).
;;            pas-stb-mmt : passive side stability moment (t*m/m).

(defclass wall ()
;;
  ((act-soil :accessor act-soil :initform nil)
   (pas-soil :accessor pas-soil :initform nil)
   (    spce :accessor     spce :initform nil)
;;
   (fy      :accessor fy      :initform 2.4)
   (bnd-mmt :accessor bnd-mmt :initform nil)
   (req-z   :accessor req-z   :initform nil)
   (size    :accessor size    :initform nil)
;;
   (lgth :accessor lgth :initform nil)
   (wght :accessor wght :initform nil)
;;
   (act-stb-mmt :accessor act-stb-mmt :initform nil)
   (pas-stb-mmt :accessor pas-stb-mmt :initform nil)
;;
  ))


;;                              (bnd-mmt wall) :before
;;
;; *  fun  * stores wall bending moment (t*m/m).

(defmethod bnd-mmt :before ((wall wall))
  (let ((val (slot-value wall 'bnd-mmt)))
    (unless val
      (setf (slot-value wall 'bnd-mmt)
            (get-wall-bnd-mmt wall))
      )))
```

```
;;                              (req-z wall) :before
;;
;; *  fun  * stores wall required section modulust (cm^3/m).

(defmethod req-z :before ((wall wall))
  (let ((val (slot-value wall 'req-z)))
    (unless val
      (setf (slot-value wall 'req-z)
            (get-wall-req-z wall))
      )))



;;                              (size wall) :before
;;
;; *  fun  * stores wall size (type w z).

(defmethod size :before ((wall wall))
  (let ((val (slot-value wall 'size)))
    (unless val
      (setf (slot-value wall 'size)
            (get-wall-size wall))
      )))



;;                              (lgth wall) :before
;;
;; *  fun  * obtains wall length (m) by iteration.

(defmethod lgth :before ((wall wall))
  (let ((val (slot-value wall 'lgth)))
    (unless val
      (get-wall-lgth wall)
      )))



;;                              (wght wall) :before
;;
;; *  fun  * stores wall weight (ton).

(defmethod wght :before ((wall wall))
  (let ((val (slot-value wall 'wght)))
    (unless val
      (setf (slot-value wall 'wght)
            (get-wall-wght wall))
      )))



;;                         (act-stb-mmt wall) :before
;;
;; *  fun  * stores wall active side stability moment (t*m/m).

(defmethod act-stb-mmt :before ((wall wall))
```

```
  (let ((val (slot-value wall 'act-stb-mmt)))
    (unless val
      (setf (slot-value wall 'act-stb-mmt)
            (get-wall-stb-mmt wall 'act))
      )))



;;                          (pas-stb-mmt wall) :before
;;
;; *  fun  * stores wall passive side stability moment (t*m/m).

(defmethod pas-stb-mmt :before ((wall wall))
  (let ((val (slot-value wall 'pas-stb-mmt)))
    (unless val
      (setf (slot-value wall 'pas-stb-mmt)
            (get-wall-stb-mmt wall 'pas))
      )))



;;                        (setf (lgth wall) lgth) :before
;;
;; *  fun  * clears depthes of active & passive side soils and active &
passive
;;           side stability moment.

(defmethod (setf lgth) :before (lgth (wall wall))
  (declare (ignore lgth))
  (let ((act-soil (act-soil wall))
        (pas-soil (pas-soil wall)))
    (setf (dpth act-soil) nil)
    (setf (dpth pas-soil) nil)
    (setf (wght wall) nil)
    (setf (act-stb-mmt wall) nil)
    (setf (pas-stb-mmt wall) nil)))



;;                        (setf (lgth wall) lgth) :after
;;
;; *  fun  * draws wall section.

(defmethod (setf lgth) :after (lgth (wall wall))
  (when lgth
    (let ((wnd *wall-wnd*)
          (act-soil (act-soil wall))
          (pas-soil (pas-soil wall))
          (spce     (spce     wall)))
      (draw-wall wnd wall)
      (draw-soil wnd act-soil)
      (draw-soil wnd pas-soil)
      (draw-lower-soil wnd  wall)
      (write-wall-rslt wnd wall)
      (write-spce-inp wnd spce)
      (dotimes (i 50000))
      (erase-rect wnd  40 60  70 70)
      (erase-rect wnd 130 60 160 70)
```

```
      )))


;;=====================================================================


;;                              (get-wall-bnd-mmt wall)
;;
;; *  fun   * obtains wall bending momnet (t*m/m).
;; *  in    * wall : wall.

(defun get-wall-bnd-mmt (wall)
  (let* ((    spce (    spce wall))
         (act-soil (act-soil wall))
         (dpth (dpth spce))
         (unit-wght (unit-wght act-soil))
         (k         (k         act-soil)))
    (* 1/6 k unit-wght (expt dpth 3))))


;;                         (get-wall-stb-mmt wall soil-type)
;;
;; *  fun   * obtains wall stability moment (t*m/m).
;; *  in    * wall : wall.
;;            soil-type : ('act,'pas).

(defun get-wall-stb-mmt (wall soil-type)
  (let* ((soil (case soil-type
                 (act (act-soil wall))
                 (pas (pas-soil wall))))
         (dpth      (     dpth soil))
         (unit-wght (unit-wght soil))
         (k         (k         soil)))
    (* 1/6 k unit-wght (expt dpth 3))))


;;                              (get-wall-req-z wall)
;;
;; *  fun   * obtains wall required section modulus (cm^3/m).
;; *  in    * wall : wall.

(defun get-wall-req-z (wall)
  (let ((bnd-mmt (bnd-mmt wall))
        (fy (fy wall)))
    (/ (* bnd-mmt 100) fy)))


;;                              (get-wall-size wall)
;;
;; *  fun   * obtains wall size (type w z).
;; *  in    * wall : wall.

(defun get-wall-size (wall)
```

```
  (let ((req-z (req-z wall)))
    (dolist (wall *wall-list*)
      (let ((z (nth 2 wall)))
        (when (< req-z z)
          (return wall))))))


;;                            (get-wall-lgth wall)
;;
;; *  fun   * obtains wall length (m).
;; *  in    * wall : wall.

(defun get-wall-lgth (wall)
  (let* ((spce (spce wall))
         (spce-dpth (dpth spce))
         (lgth      (+ 1.4 spce-dpth))
         (un-stb-p t))
    (loop while un-stb-p do
          (setf lgth (+ lgth 0.1))
          (setf (lgth wall) lgth)
          (setf un-stb-p (< (pas-stb-mmt wall) (act-stb-mmt wall)))
          )))


;;                            (get-wall-wght wall)
;;
;; *  fun   * obtains wall weight (ton/m).
;; *  in    * wall : wall.

(defun get-wall-wght (wall)
  (let* ((size (size wall))
         (unit-wght (nth 1 size))
         (lgth (lgth wall)))
    (* unit-wght lgth)))


;;*******************************************************************
;;                                 space
;;*******************************************************************


;;           dpth : depth (m).

(defclass spce ()
;;
  ((dpth :accessor dpth :initform 3.0)
;;
  ))


;;*******************************************************************
;;                                   instances
;;*******************************************************************
```

```
(defvar *act-soil* (make-instance 'soil :act-p t  ))
(defvar *pas-soil* (make-instance 'soil :act-p nil))


(defvar *wall* (make-instance 'wall))
(defvar *spce* (make-instance 'spce))


(setf (wall *act-soil*) *wall*)
(setf (spce *act-soil*) *spce*)

(setf (wall *pas-soil*) *wall*)
(setf (spce *pas-soil*) *spce*)


(setf (act-soil *wall*) *act-soil*)
(setf (pas-soil *wall*) *pas-soil*)
(setf (    spce *wall*)     *spce*)

;;
  ))

;;*******************************************************************
;;                                 instances
;;*******************************************************************


(defvar *act-soil* (make-instance 'soil :act-p t  ))
(defvar *pas-soil* (make-instance 'soil :act-p nil))


(defvar *wall* (make-instance 'wall))
(defvar *spce* (make-instance 'spce))


(setf (wall *act-soil*) *wall*)
(setf (spce *act-soil*) *spce*)

(setf (wall *pas-soil*) *wall*)
(setf (spce *pas-soil*) *spce*)


(setf (act-soil *wall*) *act-soil*)
(setf (pas-soil *wall*) *pas-soil*)
(setf (    spce *wall*)     *spce*)
```

B. 2. 2 Graphics part (wall : source : graphics)

```
;;*******************************************************************
;;                                  graphics
;;*******************************************************************



;;                               (show-wall-wnd)
;;
;; *  fun  * shows wall window.

(defun show-wall-wnd ()
  (let ((wnd *wall-wnd*))
    (if (and wnd (wptr wnd))
      (window-select wnd)
      (open-wall-wnd))))



;;                               (open-wall-wnd)
;;
;; *  fun  * opens wall window.

(defun open-wall-wnd ()
  (open-grh-wnd *wall-wnd* "Wall Demo")
  (let ((wnd *wall-wnd*))
    (draw-section wnd)
    (window-show wnd)))



;;                               (draw-section wnd)
;;
;; *  fun  * draws section.
;; *  in   * wnd : drawing widnow.

(defun draw-section (wnd)
  (bgn-pict wnd)
  (draw-soil wnd *act-soil*)
  (draw-soil wnd *pas-soil*)
  (write-soil-inp wnd *pas-soil*)
  (draw-lower-soil wnd *wall*)
  (draw-wall       wnd *wall*)
  (write-wall-rslt wnd *wall*)
  (write-spce-inp  wnd *spce*)
  (end-pict wnd))



;;                               (redraw-section wnd)
;;
;; *  fun  * redraws section.
;; *  in   * wnd : drawing window.

(defun redraw-section (wnd)
  (draw-wall       wnd *wall*)
  (del-draw-inf wnd)
```

```
  (bgn-pict wnd)
  (draw-soil wnd *act-soil*)
  (draw-soil wnd *pas-soil*)
  (write-soil-inp wnd *pas-soil*)
  (draw-lower-soil wnd *wall*)
  (draw-wall       wnd *wall*)
  (write-wall-rslt wnd *wall*)
  (write-spce-inp  wnd *spce*)
  (end-pict wnd))


;;                              (del-section wnd)
;;
;; *  fun  * deletes section.
;; *  in   * wnd : drawing window.

(defun del-section (wnd)
  (erase-rect wnd 0 0 500 300))


;;                              (draw-wall wnd wall)
;;
;; *  fun  * draws wall.
;; *  in    * wnd : drawing window.
;;            wall : wall.

(defun draw-wall (wnd wall)
  (let* ((unit *unit*)
         (bgn-pnt *bgn-pnt*)
         (lgth (lgth wall))
         (x  (+ (nth 0 bgn-pnt) (* 5 unit)))
         (bgn-y (nth 1 bgn-pnt))
         (end-y (+ bgn-y (round (* lgth unit))))
         (bgn-crd `(,x ,bgn-y))
         (end-crd `(,x ,end-y)))
    (set-pen-size wnd #@(7 1))
    (draw-line wnd bgn-crd end-crd)
    (set-pen-size wnd #@(1 1))
    ))


;;                              (draw-soil wnd soil)
;;
;; *  fun  * draws soil.
;; *  in   * wnd : drawing window.
;;           soil : soil.

(defun draw-soil (wnd soil)
  (let* ((unit *unit*)
         (bgn-pnt *bgn-pnt*)
         (spce (spce soil))
         (spce-dpth (dpth spce))
         (soil-dpth (dpth soil))
         (act-p (act-p soil))
         (bgn-x (if act-p
```

```
                  (+ (nth 0 bgn-pnt) (* 5 unit) 7)
                  (nth 0 bgn-pnt)))
         (bgn-y (if act-p
                  (nth 1 bgn-pnt)
                  (+ (nth 1 bgn-pnt) (round (* spce-dpth unit)))))
         (end-x (if act-p
                  (+ bgn-x (* 5 unit) -7)
                  (+ bgn-x (* 5 unit) -2)))
         (end-y (+ bgn-y (round (* soil-dpth unit)))))
    (set-pen-pattern wnd *light-gray-pattern*)
    (paint-rect wnd bgn-x (+ bgn-y 2) (+ end-x 2) end-y)
    (set-pen-pattern wnd *black-pattern*)
    (set-pen-size wnd #@(2 2))
    (draw-line wnd `(,bgn-x ,bgn-y) `(,end-x ,bgn-y))
    (set-pen-size wnd #@(1 1))
    (if act-p
      (draw-line wnd `(,(+ end-x 2) ,bgn-y) `(,(+ end-x 2) ,end-y))
      (draw-line wnd `(,(- bgn-x 1) ,bgn-y) `(,(- bgn-x 1) ,end-y)))
    ))


;;                          (draw-lower-soil wnd wall)
;;
;; *  fun  * draws lower soil.
;; *  in   * wnd : drawing window.
;;           wall : wall.

(defun draw-lower-soil (wnd wall)
  (let* ((unit *unit*)
         (bgn-pnt *bgn-pnt*)
         (lgth (lgth wall))
         (bgn-x (nth 0 bgn-pnt))
         (bgn-y (+ (nth 1 bgn-pnt) (round (* lgth unit))))
         (end-x (+ bgn-x (* 10 unit) 2))
         (end-y (+ (nth 1 bgn-pnt) (* 8 unit))))
    (set-pen-pattern wnd *light-gray-pattern*)
    (paint-rect wnd bgn-x bgn-y end-x end-y)
    (set-pen-pattern wnd *black-pattern*)
    (draw-line wnd `(,(1- bgn-x) ,bgn-y) `(,(1- bgn-x) ,end-y))
    (draw-line wnd `(,(1- bgn-x) ,end-y) `(,end-x  ,end-y))
    (draw-line wnd `(    ,end-x  ,end-y) `(    ,end-x  ,bgn-y))
    ))


;;                          (write-wall-rslt wnd wall)
;;
;; *  fun  * writes result.
;; *  in   * wnd : drawing window.
;;           wall : wall.

(defun write-wall-rslt (wnd wall)
  (let* ((lgth (lgth wall))
         (size (size wall))
         (type (nth 0 size))
         (wght (wght wall)))
    (write-text wnd '( 45 40) (format nil "L"))
```

```
    (write-text wnd '( 80 40) (format nil "Type"))
    (write-text wnd '(135 40) (format nil "W"))
;;
    (write-text wnd '( 42 55) (format nil "m"))
    (write-text wnd '(132 55) (format nil "t/m"))
;;
    (write-text wnd '( 40 70) (format nil "~3,1f" lgth))
    (write-text wnd '( 75 70) (format nil "~a" type))
    (write-text wnd '(130 70) (format nil "~4,2f" wght))
;;
    (frame-rect wnd 25 20 175 80)
    ))


;;                          (write-soil-inp wnd soil)
;;
;; *  fun  * writes input.
(defun write-soil-inp (wnd soil)
  (let* ((unit-wght (unit-wght soil))
         (innr-frct (innr-frct soil)))
    (erase-rect wnd 300 75 375 110)
    (set-window-font wnd '("symbol" 12))
    (write-text wnd '(315  90) "g =")
    (write-text wnd '(315 105) "f =")
    (set-window-font wnd '("chicago" 12))
    (write-text wnd '(335  90) (format nil "~3,1f" unit-wght))
    (write-text wnd '(335 105) (format nil "~3,1f" innr-frct))
;;
    (frame-rect wnd 300 75 375 110)
    ))


;;                (write-spce-inp wnd spce)
;;
;; *  fun  * writes space input.
;; *  in   * wnd : drawing window.
;;           spce : space.

(defun write-spce-inp (wnd spce)
  (let* ((unit *unit*)
         (bgn-pnt *bgn-pnt*)
         (bgn-x (nth 0 bgn-pnt))
         (bgn-y (nth 1 bgn-pnt))
         (dpth (dpth spce))
         (x (+ bgn-x (round (* 2.0 unit))))
         (y (+ bgn-y (round (* 1/2 dpth unit)))))
    (write-text wnd `(,x ,y) (format nil "D = ~3,1f m" dpth))))
```

B. 2. 3 Window and menu part <1> (wall : source : basics)

```
;;********************************************************************
;;                                menu basics
;;********************************************************************



;;                             (dfn-menu menu title)
;;
;; *   fun   * define menu.
;; *   in    * menu : menu name.
;;             title : menu title.

(defmacro dfn-menu (menu title)
  `(setf ,menu (make-instance 'menu :menu-title ,title)))



;;                      (put-menu-item menu name title fnct)
;;
;; *   fun   * puts menu item.
;; *   in    * menu : menu name.
;;             title : item title.
;;             fnct : item function.

(defmacro put-menu-item (menu title fnct)
  `(add-menu-items ,menu
                   (make-instance 'menu-item
                                  :menu-item-title ,title

,fnct))))



;;********************************************************************
;;                             graphics window
;;********************************************************************



;; inherits characters of both *dialog* and *window*.

(defclass grh-wnd (window)
  ((saved-pict :accessor saved-pict :initform nil)))



;; redraws picture, if *grh-wnd* has 'saved-pict.
;; saved-pict : drawing procedure of present picture.

(defmethod view-draw-contents ((grh-wnd grh-wnd))
  (call-next-method)
  (when (saved-pict grh-wnd)
    (draw-picture grh-wnd (saved-pict grh-wnd))))
```

```
;; unbounds 'saved-pict information

(defmethod window-close ((grh-wnd grh-wnd))
  (call-next-method)
  (let ((saved-pict (saved-pict grh-wnd)))
    (when saved-pict (kill-picture saved-pict)))
  (setf (saved-pict grh-wnd) nil))



;;                         (open-grh-wnd grh-wnd title size)
;;
;; *  fun  * opens graphics window.

(defmacro open-grh-wnd (wnd title &optional (size 'big))
  (let* ((wnd-x (case size (big 500) (sml 400)))
         (wnd-y (case size (big 295) (sml 150)))
         (wnd-size (make-point wnd-x wnd-y))
         (wnd-pst    (case size (big  42) (sml 100))))
    `(setf ,wnd (make-instance 'grh-wnd
                        :window-type     :tool
                        :window-title    ,title
                        :view-size      ,wnd-size
                        :view-font       '("Chicago" 12 :plain)
                        :window-show     nil
                        :view-position '(:top ,wnd-pst)))))



;;                                   (del-draw-inf wnd)
;;
;; * fun  * deletes drawing information.

(defun del-draw-inf (wnd)
  (let ((saved-pict (saved-pict wnd)))
    (when saved-pict (kill-picture saved-pict)))
  (setf (saved-pict wnd) nil))



;;                                    (bgn-pict wnd)
;;
;; *  fun  * declares beginning of picture.
;; *  in   * wnd : drawing window.

(defun bgn-pict (wnd)
  (start-picture wnd))



;;                                    (end-pict wnd)
;;
;; *  fun * declares end picture and stores picture information.
;; *  in  * wnd : drawing window.

(defun end-pict (wnd)
  (setf (saved-pict wnd) (get-picture wnd))
  (view-draw-contents wnd))
```

```
;;                              (draw-line pnt pnt-1 pnt-2)
;;
;; *  fun   * draws line between two points on window.
;; *  in    * wnd : drawing window.
;;            pnt-1 : coordinate list of point-1.
;;            pnt-2 : coordinate list of point-2.

(defun draw-line (wnd crd-1 crd-2)
  (let ((x1 (first  crd-1))
        (y1 (second crd-1))
        (x2 (first  crd-2))
        (y2 (second crd-2)))
    (move-to wnd x1 y1)
    (line-to wnd x2 y2)))


;;                              (write-text wnd pnt text)
;;
;; *  fun   * writes text at pnt on window.
;; *  in    * wnd : drawing window.
;;            pnt : coordinate list of pnt.
;;            text : text.

(defun write-text (wnd crd text)
  (let ((x (first  crd))
        (y (second crd)))
    (move-to wnd x y)
    (princ text wnd)))


;;*********************************************************************
;;                                   dialog window
;;*********************************************************************


;;                         (put-text-box wnd name pnt text)
;;
;; *  fun   * puts editable text dialog item.
;; *  in    * wnd : dialog window.
;;            name : item name.
;;            pnt  : corrdinate list of point.
;;            text : text.
;; *  var   * pnt-x : x-coordinate of point.
;;            pnt-y : y-coordinate of point.
;;            pstn : window position of point.

(defun put-text-box (wnd name pnt text &optional (size 'sml))
  (let* ((pnt-x (first pnt))
         (pnt-y (second pnt))
         (pstn (make-point pnt-x pnt-y))
         (box (if (eq size 'sml) #@(35 16) #@(40 16))))
    (add-subviews wnd
```

```
                    (set name
                         (make-instance 'editable-text-dialog-item
                                    :dialog-item-text     text
                                    :view-size     box
                                    :view-position pstn)))))



;;                  (put-radio-bttn wnd name pnt text clst pshd-p)
;;
;; *  fun   * puts radio button dialog item.
;; *  in    * wnd : dialog window.
;;            name : item name.
;;            pnt : coordinate list of point.
;;            text : item text.
;;            clst : item cluster.
;;            pshd-p : t    --- radio button is     pushed.
;;                     nil --- radio button is not pushed.
;; *  var   * pnt-x : x-coordinate of point.
;;            pnt-y : y-coordinate of point.
;;            pstn : window position of point.

(defun put-rdo-bttn (wnd name pnt text clst pshd-p)
  (let* ((pnt-x (first pnt))
         (pnt-y (second pnt))
         (pstn (make-point pnt-x pnt-y)))
    (add-subviews wnd
                    (set name
                         (make-instance 'radio-button-dialog-item
                                        :dialog-item-text     text
                                        :radio-button-cluster clst
                                        :radio-button-pushed-p pshd-p
                                        :view-position pstn)))))



;;                        (put-bttn wnd pnt text dflt-p fnct)
;;
;; *  fun   * puts button.
;; *  in    * wnd : dialog & graphic window.
;;            pnt : coordinate list of point.
;;            text : button text.
;;            dflt-p : t    --- default button.
;;                     nil --- no default button.
;;            fnct : function.
;; *  var   * pnt-x : x-coordinate of point.
;;            pnt-y : y-coordinate of point.
;;            pstn  : window position.

(defmacro put-bttn (wnd pnt text dflt-p fnct)
  (let* ((pnt-x  (first  pnt))
         (pnt-y  (second pnt))
         (pstn   (make-point pnt-x pnt-y)))
    `(add-subviews ,wnd
                    (make-instance 'button-dialog-item
                                   :dialog-item-text     ,text
                                   :view-size     #@(70 16)
                                   :view-position ,pstn
```

```
                                  :default-button       ,dflt-p
                                  :dialog-item-action
                                  #'(lambda (item) item ,fnct)))))


;;                            (get-val-frm-box name)
;;
;; *  fun   * obtains value from box dialog item.
;; *  in    * name : box name.

(defun get-val-frm-box (name)
  (read-from-string (dialog-item-text name)))


;;               (get-val-frm-2-rdo wnd cluster name-1 name-2)
;;
;; *  fun   * obtains value from two radio buttons.
;; *  in    * wnd : dialog window.
;;            cluster : radio button cluster.
;;            name-1 : button name -1.
;;            name-2 : button name -2.

(defun get-val-frm-2-rdo (wnd cluster name-1 name-2)
  (let ((slct-id (pushed-radio-button wnd cluster)))
    (or (when (eq name-1 slct-id)
          (read-from-string (dialog-item-text name-1)))
        (when (eq name-2 slct-id)
          (read-from-string (dialog-item-text name-2))))))


;;**********************************************************************
;;                               wall menu
;;**********************************************************************


(dfn-menu *wall-menu* "Wall Demo")


(set-menubar `(,*wall-menu*))


(let* ((menu *wall-menu*)
       (file-menu (second *default-menubar*))
       (file-menu-items (menu-items file-menu))
       (quit-item       (first (last file-menu-items))))
  (put-menu-item menu "Open Wall" (show-wall-wnd))
  (put-menu-item menu "Input"       (open-inp-dlg))
  (put-menu-item menu  "-" ())
  (add-menu-items menu quit-item)
  (set-command-key quit-item #\Q))
```

```
(let* ((apple-menu (first  *default-menubar*))
       (menu-items (menu-items apple-menu))
       (about-item (first menu-items))
       (line       (second menu-items)))
  (remove-menu-items apple-menu about-item)
  (remove-menu-items apple-menu line)
  (put-menu-item apple-menu "About Wall Demo..." (open-what-wnd))
  (put-menu-item apple-menu "-" ()))
```

B. 2. 4 Window and menu part <2> (wall : source : wnd)

```
;;*******************************************************************
;;                                open-what-wnd
;;*******************************************************************


;;                               (open-what-wnd)
;;
;; *  fun  * opens *what-wnd*.

(defun open-what-wnd ()
  (open-grh-wnd *what-wnd* "Welcome to Wall Demo!!" sml)
  (let ((wnd *what-wnd*))
    (put-bttn wnd (160 110) "Return" t (window-close wnd))
    (bgn-pict wnd)
    (write-text wnd '(80 30)
                "This is just concept demonstration.")
    (write-text wnd '(110 50)
                "Symplified by Naoki Ikoma.")
    (write-text wnd '(85 70)
                "Programmed by Hiroyuki Fuyama.")
    (write-text wnd '(140 90) "October 3, 1991")
    (end-pict wnd)
    (window-show wnd)
    ))


;;*******************************************************************
;;                                open-inp-dlg
;;*******************************************************************


;;                               (open-inp-dlg)
;;
;; *  fun  * opens input dialog.

(defun open-inp-dlg ()
  (open-grh-wnd *inp-dlg* "Wall Input" sml)
  (let ((wnd *inp-dlg*)
        (unit-wght (write-to-string (unit-wght *act-soil*)))
        (innr-frct (write-to-string (innr-frct *act-soil*)))
        (fy    (write-to-string (fy   *wall*)))
        (dpth  (write-to-string (dpth *spce*))))
;;
    (put-text-box wnd 'unit-wght '(100 60) unit-wght)
    (put-text-box wnd 'innr-frct '(160 60) innr-frct)
    (put-text-box wnd 'fy        '(220 60) fy  )
    (put-text-box wnd 'dpth      '(280 60) dpth)
;;
    (put-bttn wnd (120 110) "Close"  nil (window-close wnd))
    (put-bttn wnd (220 110) "Do it!" t   (inp-dlg-actn))
;;
    (bgn-pict wnd)
    (set-window-font wnd '("symbol" 12))
```

```
    (write-text wnd '(115 35) "g")
    (write-text wnd '(175 35) "f")
    (set-window-font wnd '("chicago" 12))
    (write-text wnd '(230 35) "Fy")
    (write-text wnd '(280 35) "Depth")
;;
    (write-text wnd '(98 50) "t/m^3")
    (write-text wnd '(155 50) "degree")
    (write-text wnd '(215 50) "t/cm^2")
    (write-text wnd '(290 50) "m")
;;
    (end-pict wnd)
    (window-show wnd)
    ))


;;                                      (inp-dlg-actn)
;;
;; *  fun  * replaces match indices.

(defun inp-dlg-actn ()
  (declare (special dpth unit-wght innr-frct fy))
  (let ((dpth (get-val-frm-box dpth))
        (unit-wght (get-val-frm-box unit-wght))
        (innr-frct (get-val-frm-box innr-frct))
        (fy (get-val-frm-box fy)))
    (del-section *wall-wnd*)
;;
    (setf (dpth *spce*) dpth)
    (setf (unit-wght *act-soil*) unit-wght)
    (setf (unit-wght *pas-soil*) unit-wght)
    (setf (innr-frct *act-soil*) innr-frct)
    (setf (innr-frct *pas-soil*) innr-frct)
    (setf (fy *wall*) fy)
;;
    (setf (k *act-soil*) nil)
    (setf (k *pas-soil*) nil)
    (setf (lgth     *wall*) nil)
    (setf (bnd-mmt  *wall*) nil)
    (setf (req-z    *wall*) nil)
    (setf (size     *wall*) nil)
    (setf (wght     *wall*) nil)
;;
    (redraw-section *wall-wnd*)
    ))
```

B. 2. 5 Definition of global variable (wall : source : glb var)

```
(require 'quickdraw)



(defvar *wall-menu*)          ;; wall menu.
(defvar *wall-wnd* nil)       ;; wall window.
(defvar *inp-dlg*)
(defvar *what-wnd*)            ;; explanation window.



(defvar *unit* 25)
(defvar *bgn-pnt* '(150 60))


(setf *lisp-startup-functions* '(show-wall-wnd))




(defvar *wall-list*
;;           w     z
;;          t/m  cm^3/m
  '((type-2 0.12  874)
    (type-3 0.15 1340)
    (type-4 0.19 2270)
    (type-5 0.21 3150)))
```

Index

Technical Terms

Temporary Retaining Wall: Retaining Wall is a free-standing wall designed to hold back earth or other solid material that rises higher on one side of the wall. Temporary Retaining Wall is used only for the duration of the excavation work. It is not permanent but temporary. Sometimes , it must be removed when the work is finished.

GEO-FRONT: A GEO-FRONT is a development projects of a ground in Japan in order to extending humans' living space. This is a project digging into the ground to make large underground towns.

MINATO-MIRAI 21: A MINATO-MIRAI 21 is a huge development project of port areas at Yokohama in Japan.

cofferdam: A cofferdam is a retaining wall which can cut off water.

sheet pile: A sheet pile is a row of piles that are set close together in the ground as a continuous barrier. They may be used around the edge of an excavation to keep out earth or water.

tie back (tieback anchor): A tie back is an anchor supporting the retaining wall. This is preferred in most temporary construction where it is possible to encroach on the adjacent ground to install the anchor. The principal advantages are allowing an unobstructed area in front of the wall. The major disadvantage is in encountering underground utilities.

boiling (boil blow): A boiling is a flow of fine soil, such as sand or silt, into the bottom of an excavation that is under pressure of water or air.

heaving (ground heave): A heaving is a rise in level of a soil surface, or foundation, cause by expansion of a soil such as CLAY.

cohesion: A The general process of attraction between molecules that holds a soil or liquid together.

Fuzzy Theory: A Fuzzy Theory is the method which can express numerically something ambiguous that is felt by a human's sense.

grade: A grade is a technical term used in Fuzzy Theory. It means a degree or an extent of the result.

Membership Function: A Membership Function is a function which can determine grades.