Criteria of Good Project Network Generator and its Fulfillment using a Dynamic CBR Approach

[1] Hyun Woo Kim and [2] Kyoung Jun Lee*

[1] Graduate School of Management, KAIST
Cheongryangri-dong, Dongdaemun-Ku, Seoul, Korea
Kimhyunu@kgsm.kaist.ac.kr

[2] School of Business, Kyung Hee University
Hoegi-Dong, Dongdaemun-Ku, Seoul, Korea
klee@khu.ac.kr

Abstract. Most project-based industries such as construction, shipbuilding, and software development etc. should generate and manage project network for successful project planning. We suggest a set of criteria of good project network generator such as network generation efficiency, quality of network, and economics of system development. For the efficiency of the planning, the first criterion, we decided to take a CBR approach. However, using only previous cases is insufficient to generate a proper network for a new project. By embedding rules and constraints in the case-based system, we could improve the quality of the project network: the second criterion. The integration of CBR approach and the knowledge-based approach makes feasible the development of the project network generator and improves the quality of the network by mutual enhancement through crosschecking the knowledge and cases in the development and maintenance stages. For some complex project network planning, a single-case assumed project network generation methodology is refined into Dynamic Leveled Multiple Case approach. The methodology contributes again the efficiency and effectiveness of project network generation and reduces the efforts of the system development.

1. Introduction

Generation, verification, and modification of construction project schedule networks in the PERT-CPM chart are the essential tasks for successful project planning and management in the construction industry. Because a project network consists of hundreds of activities and precedence relationships, project planning is a time-consuming and knowledge-intensive task. To compete with other companies for a contract, it is critical for a construction company to quickly generate a good and consistent project plan.

To generate a project network, much domain knowledge, experience and the control knowledge are needed. The domain knowledge describes the domain world

* Corresponding Author: Kyoung Jun Lee (klee@khu.ac.kr)

and the available actions to the planner. The control knowledge indicates how the planner will achieve its goals; it controls the planner's search for a plan. In other words, control knowledge is prescriptive, whereas domain knowledge is descriptive. So the project manager responsible for generating project network should have much domain knowledge, control knowledge and time. Especially in the construction domain, the generation of project network is a time consuming task. It takes a couple of days for generating the simplest construction project network such as project network for APT (apartment) construction although the 5 years experienced construction manager worked. We need the project network generation system to respond to management's requirements quickly and flexibly, to reduce time to generate network, and to improve quality of network. The quality of network can be measured by technical soundness, satisfaction of due date, and resource efficiency. To achieve these objectives, system contains much knowledge.

In the area of project planning, there had been a lot of research and developments on project scheduling methods and management techniques assuming that a project network is given to the project manager. However, since the earliest research prototype CONSTRUCTION PLANEX [1], there has only been a limited amount of research to automate or support the project network generation using knowledge-based techniques, such as GHOST [5], SIPE-2 [2], and HISCHED [6]. To generate a plan by these systems, the users have to input a lot of activities' information since these systems are not designed to utilize past cases.

Most of the previous systems were not designed to use past cases; so, their users had the burden of inputting vast amounts of information, or their developers had to provide this knowledge for the systems. OARPLAN [7], a model-based planning system, does use past cases, but the user has to input the precedence relationships between activities. In contrast to these systems, the system that we developed for this project doesn't require the users to input any precedence relationships because the system uses past cases containing these precedence constraints and adaptation knowledge. Zhang and Maher [8] used a CBR method for the structural design of buildings. They claimed that CBR as a design model is intuitively appealing because much of the design knowledge comes through the experience of multiple, individual design situations. The same holds true in the construction planning situation.

To select an approach among these alternatives, we need a set of criteria to evaluate a project network generation system. Therefore, in this paper, we first suggest the set of criteria of good project network generator.

2. Criteria of Good Project Network Generator

In this section, we discuss about the criteria of good project network generator. Some of the criteria are related to the system itself and the other criteria are related to the project network. The system relating criteria are classified into the efficiency feature and the implementation feature. The efficiency feature can be evaluated by time for network generation. The implementation feature can be evaluated by the cost. The project network relating criteria can be evaluated by the quality of project network. Though there is much variation in the characteristics of project network depending on the specific features of the domains such as construction, shipbuilding, and SW development etc., we can generalize and define a project network as follows.

A project networks is a set of activities and their interrelated precedence relationships to start and finish a project. Generating a project network, whether it is done by human or computer, is a function of project requirement, project-related domain knowledge, and domain-independent commonsense knowledge. The project requirement in other word project specification consists of the function requirement given by client and the engineering requirement. The project-related domain knowledge includes the work breakdown structure, the activity selection knowledge and the precedence knowledge. The domain independent commonsense knowledge means network principles kept to compose a sound network structure of the project. This knowledge is necessary in project network evaluation as well as generation.

2.1. Efficiency of project network generation

The efficiency can be evaluated from two viewpoints. The one is the total generation time for project network that can be used in real field. Generally, the project manager can understand required activities and their precedence relations with only the project network for the project management. So, even if the generated project network has no error, the project network generator's operator has to customize the system generating project network in the cognitive view point. Efficiency should therefore be evaluated by the total time of the system generation time and the customizing time.
Another evaluation method is comparing the pure system time of alternatives those generate same quality of project network if they have same input.

2.2. Effectiveness of the project network generator

Is it the well-formed project network?
This question is concerned with the structure of project network. There is the principle for project network structure. We call the project network as a well-formed project network if the project network satisfies the principle.

One is there should be no isolated part in the project network. Every part in project network is necessary to achieve the goal of project. In this viewpoint, it is not difficult to conclude that they should be tightly coupled by relationship. If there is any part should be isolated, it means the project network contains another project. And it should be divided into another project network.

Another requirement is that there should be no cycled relationship. An example of a cycled relationship is that 'B' activity has to start after 'A' activity, 'C' activity has to start after 'B' activity, but 'A' activity has to start after 'C 'activity. Even though it is a logic error, we can find it without checking and fix it up temporary by removing a relationship with heuristic algorithm.

The other is the optional principle. There should be only one start and end node. It is helpful when we analyze project network with CPM method and interpret the result. Especially in the construction domain, this principle is a strong custom.

Is it the valid network?
This criterion is interested in the conformity of activity. In other word, this criterion evaluate that the project network generator decides correctly about what activities are required according to the project specification. If the project network includes only the necessary activities, then we call it as a valid project network. The evaluation result of this criterion is largely dependent on the degree of detail in the input project specification. The issue in this research is how to maximize the conformity of activity without detail project specification such as CAD data.

Is it the verified network?
This criterion is concerned with the relationship between activities. If there is no violation of critical precedence constraints then we call it the verified project network.

2.3. Economics of development of project network generator

The system development cost is very important evaluation criterion for the project network generation system. As we have reviewed, the project network generator is a typical knowledge based system generating the proper project network for a new project using domain knowledge and the commonsense knowledge. Therefore we can evaluate which alternative is better by comparing the development cost in the knowledge based system development perspective. The cost consists of the knowledge acquisition cost and the inference engine development cost.

3. Case and Knowledge Based Project Network Generator

Network generation by case and knowledge is inspired by an expert's generation method and adapted for efficiency and effectiveness. The reasons why we adapt case approach is currently the expert start with the most similar past network to get a new project network quickly and to use the implicit knowledge in the past project network. We have understood that if we would use a similar past project network, we could generate a new initial project network without much knowledge and time. In other words, the case based approach could be very helpful to generating a new project network efficiently if we could modify the retrieved past project network by system.

This is the reason why we have integrated the knowledge approach. Currently most of experts finish generating a new project network by modification of the selected past project network with their domain knowledge gained from experience. Through this process, a new project network is being generated. The project network can be represented formally. So we had represented the project network using frames for machine understanding and we had identified the required adaptation knowledge. And, we have acquired and represented the knowledge by constraints and rules. We have comprehended that case based approach helps to generate project network efficiently and the knowledge based modification make the quality of project network higher.

We could get initial project network by past case retrieval. And we use rules to analysis the difference between the new project and the selected case. The project network generator modifies the initial project network with the set of operators for activity modification. Next it uses a set of knowledge that is represented by constraint to satisfy the precedence constraints between activities. We call the constraint type knowledge as precedence constraint. We have implemented several construction project network generators using case and knowledge based methodology.

3.1 Procedure of case and knowledge based project network generation

The first step is project specification analysis. In this step, we can reduce the search space by the case base filtering with the result of analysis. It needs almost new generating effort if we would modify a project network for concrete framed building to a project network for still beam framed building. For this reason, we have introduced the project specification analysis process with case base filtering function.

The second step is project network retrieval under the least modification principle.. The output of the project network generator is not the project itself but the project network for the project. By this viewpoint, we use the amount of modification effort as a similarity measure. It means that a project network needs the least modification efforts is the most similar project network. An obvious question that arises here is how to calculate the amount of modification effort before the adaptation is performed.

The third step is the project network modification by addition or deletion of activities. The most of initial project network have discrepancies with the new project. We fill the gap.

The fourth step is activity's features modification such as activities' duration and assigned resource.

The fifth step is network analysis and constraint satisfaction. In this step, we have the modified project network for the new project. We analyze the project network with PERT/CPM. By this task, we can get the earliest start time (ES), the earliest finish time (EF), the latest start time (LS) and the latest finish time (LF). We check the set of precedence constraints. If there are any violated precedence constraints, then we fix up the project network.

Finally, the network will be executed and customized by the project manager. The executed project network will be restored to case base.

3.2 Case Representation

A case consists of a design specification and project network. The project network consists of hundreds of activities and relationships. We use a frame-based representation scheme for representing design specification and project network. For this representation, we used the expert system tool UNIK-FRAME [3], which was developed by KAIST. A project frame has the slots such as the name, address, start date, due date, ground type, topography, and construction area. A project can have more than one building.

3.3 Project network retrieval under the least modification principle

We retrieve the most similar case under the least modification principle. The amount of modifications required can be calculated before the adaptation process is performed by the following process. If it is different the new projects specification with a past case's, then this discrepancy is added to the discrepancy list. After finished the comparison, the system run the forward rule inference to identify what the modifications are required to fill gap in between the past project network and the new project network. The result of rule inference is represented by the set of primitive operators for modification such as add, delete, replication, add-replication and reduce-replication. We can count how many modifying actions will be required for each operator. The distance between the new project and the past project is the sum of modifying actions counting in the previous process. The past project that has the least distance is the most similar case for the new project.

Distance = $\Sigma \text{dist}(X_{Ni}, X_{Pi})$

X_{Ni}: ith specification for the new project

X_{Pi}: ith specification for the past project in case base

$\text{dist}(X_{Ni}, X_{Pi})$: Distance function

If Xi is the interior finish work and the new project(X_{Ni}) is stone finish and the past project(X_{Pi}) is no interior, then the distance function calculates the distance value and returns through the following step:

Step1: Identify the set of required primitive operators.

In this case, the operator set is as following. Operator set = {add interior frame, add attach stone}

Step2: Calculate the number of modifying actions.

For executing the [add interior frame] operator, it should add the interior framing activity. To add an activity to project network, it has to create a proper activity and give several relationships to the new activity. The relationships can be generated by converting the related precedence constraints. If we would count the number of action for [add interior frame], it needs one of modifying action for activity creation, and add the number of the related precedence constraints to the total number of modifying actions. We could formalize the number of modification action for each network modification operators as follows.

ADD

Activity node addition: $\text{count}(A_N^*)$

New Relationship addition: $\text{count}(\text{Con}(A_i:A_N^*)) + \text{count}(\text{Con}(A_N^*: A_j))$ actions

A_N^*: set of newly added activities to the project network

A_i: set of activities which are predecessor to A_N^*

A_j: set of activities which are successor to A_N^*

$\text{Con}(A_1, A_2)$: set of precedence constraints which the predecessor is in set A_1 and the successor is in set A_2

count(S): function for counting the number of set S

DELETE
Activity node deletion: count(A_S^*)
Connected Relationship deletion: count(Con(A_i: A_S^*)) + count(Con(A_S^*:A_j)) actions
New Relationship addition: count(Con(A_i: A_i^C)) + count(Con(A_j^C:A_j)) actions
As*: set of deleting activities from the project network
A_i: set of activities which are predecessor to A_S^*
A_j: set of activities which are successor to A_S^*
A^C: complementary set of A
Con(A_1, A_2): set of precedence constraints which the predecessor is in set A_1 and the successor is in set A_2
count(S): function for counting the number of set S

REPLACE
Activity node addition (1 action) and activity node deletion (1 action)

ADD_REPLICATION
Activity node addition (1 action) and relationship addition (1 action)

REDUCE_REPLICATION
Activity node deletion (1 action) and relationship deletion (1 action)

Step3: Repeat Step 2 for other operators and summation the results.

3.4 Knowledge based project network modification

We could get the list of discrepancies between the new project network and the past project network when we calculate the modification effort for the project network retrieval. And we could get the set of primitive operators for filling gap in between the past project network and the new project network by rule based inference. We design five primitive operators to modify project network. The first operator is the ADD operator. If the retrieved project network is without interior but the new project is stone interior then we have to add necessary activities into the retrieved project network. The procedure for executing the ADD operator is described as below Figure 1(a).

Notation for Procedure of Operators
act_i – activities included in selected project network
A_i , A_j: set of activities
$R(A_i : A_j)$ – set of relationships between the predecessor is in the set A_i and the successor is in the set A_j
$W(act_i)$ - act_i's work breakdown structure considered as a class of activities
F_{Ni} - the i th information of new project
F_{Si} - the i th information of selected project
W_{Fi} – set of work breakdown structure codes related with F_{Ni} and F_{Si}

W_{FSi} – set of work breakdown structure codes related with F_{Si}
W_{FNi} – set of work breakdown structure codes related with F_{Ni}
A^* - set of activities with W_{Fi} WBS ; $\{act_i \mid W(act_i) \in W_{Fi}\}$
A_S^* - set of activities with W_{FSi} WBS ; $\{act_i \mid W(act_i) \in W_{FSi}\}$
A_N^* - set of activities with W_{FNi} WBS ; $\{act_i \mid W(act_i) \in W_{FNi}\}$
act^*_{FSi} - last activity in A^*
$Con(A_1, A_2)$: set of precedence constraints which the predecessor is in set A_1 and the successor is in set A_2

The second operator is DELETE operator. The DELETE operator is needed at the situation to the contrary of the ADD's. The procedure for executing DELETE operator is described in Figure 1(b).

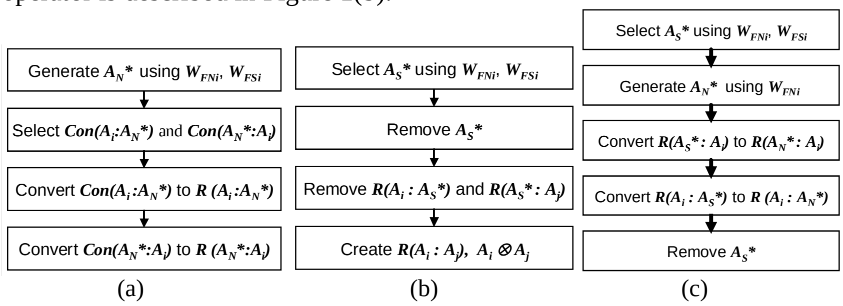

Figure 1. ADD (a), DELETE (b), and REPLACE (c) operator

The third operator is REPLACE operator. If the retrieved project network with aluminum interior but the new project is stone interior then we have to delete unnecessary activities and add necessary activities into the retrieved project network. But some of these modifications can be finished by replacement. In this case, we don't have to consider relationship modification. So we can reduce the modification efforts. The procedure for executing REPLACE operator is described in Figure 1(c).

The fourth operator is ADD-REPLICATION operator. If the retrieved project network is 18th floor but the new project is 20th floor then we have to add 19th and 20th activities into the retrieved project network as in Figure 2.

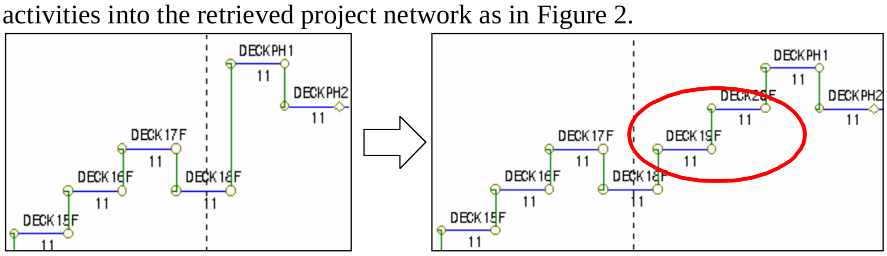

Figure 2. ADD-REPLICATION operator

The last operator is REDUCE-REPLICATION operator. The REDUCE-REPLICATION operator is needed at the situation to the contrary of the ADD-REPLICATION's. The procedure for executing ADD-REPLICATION and REDUCE-REPLICATION operator is described in Figure 3.

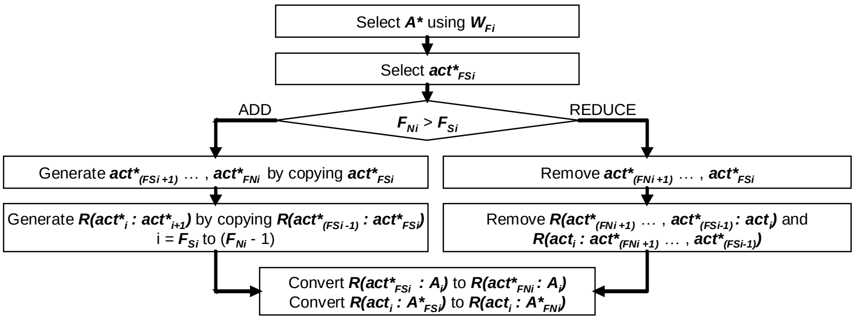

Figure 3. ADD-REPLICATION and REDUCE-REPLICATION operator

3.5 Implementation of FASTRAK-APT

We have developed the apartment building project network generator FASTRAK-APT with case and knowledge approach [4] in 1996. First, the system user inputs the project specification. Second, the FASTRAK-APT generates a new project network and the operator reflects his or her intention by interaction with the FASTRAK-APT. Finally, the operator modifies the activities' row, font, or color with project network viewers such as PERTware or Primavera.

FASTRAK-APT has been proved to reduce the effort required for generating an initial project plan from seven person-days to one person-day. The cost of updating a plan, which occurs every three months on a project, has been also reduced from 2 person-days to half a person-day. The running time of the system is about 5 minutes for a building from case loading to adaptation (a 20-floor building has about 500 activities and 700 precedence relationships).

The plans generated by FASTRAK-APT have been proved by human experts to be sound technically and have even satisfied more constraints than the cases prepared by domain experts. Therefore, the generated and executed project networks were used for enhancing the case base, and the refined case base helped improve the quality of generated plans.

The case approach great helps to reduce the development cost. The FASTRAK-APT had 430 precedence constraints and 50 cases when it had started. If we did not use a CBR approach, we would have to gather more precedence constraints, up to 13,203 ($_{163}C_2$) for FASTRAK-APT theoretically. If we didn't adopt knowledge based adaptation, actually the project network auto generation was impossible.

4. Dynamic Leveled Multiple Case Approach

Although the methodology used for FASTRAK-APT satisfies the criteria of good project network generator, it has problems to be generalized into other domains. For example, in the office building construction, the structures of project networks are very much different according to the construction method of each project while in the apartment building construction the structures of project networks are similar to each other. _If we use the adaptation of a single case, we need much knowledge for network adaptation or large number of cases for various project network structures, which leads to low efficiency and development economy.

The analysis of the past cases of office building project networks found that the project networks can be divided into several subnetworks and each subnetwork also can be divided into several lower subnetworks. We could represent a project network by a subnetwork hierarchical structure. We also found that if we divide the project networks with a proper level then the subnetworks are expected to be modified with a small amount of adaptation knowledge. By adding only a new type of knowledge for connecting the subnetworks, we can improve the efficiency and economy of the case-based project network generation. In this section, we explain such a methodology which divides a project network into multiple subnetworks, modifies and integrates them into a new project network.

4.1 Dynamic leveled multiple cases approach

For most of large project networks in the domains such as construction and software development, a project network is composed of subnetworks that are also composed of subnetworks. In such domain, we may have three kinds of strategy for utilizing project networks: maximization, minimization, and middle-level strategy. Maximization strategy (Figure 4) uses a past case as a whole project and corresponds to the single case approach as in FASTRAK-APT. This strategy is inefficient and not economical when the shape of the project network structure is various.

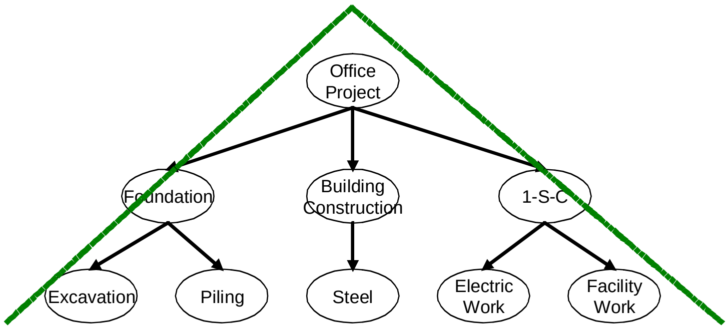

Figure 4. Maximization strategy

Minimization strategy (Figure 5) creates a new project networking by retrieving subnetworks at most primitive level and synthesizing them into a whole network and can be called a static leveled multiple cases approach. Using the primitive level subnetworks contribute to the efficiency and economics of project network generator by reducing the variety of project network structures. However, in this approach, synthesizing many primitive level subnetworks depends on the network synthesis knowledge therefore the quality of the project network depends on this knowledge. Furthermore, when we create a project network which is almost the same as the executed and validated one in the case base, we should retrieve the primitive level cases and synthesize them into one network. This approach can be used in generating a bridge construction project composed of simple relationships among subnetworks.

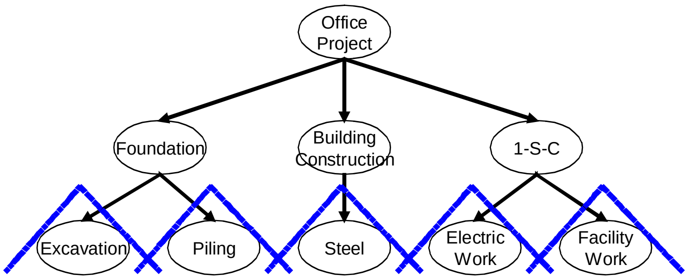

Figure 5. Minimization strategy

Middle level strategy (Figure 6) is acquiring the subnetworks in case base at a proper level for minimizing the network synthesis and modification efforts and described as a dynamic leveled multiple case approach. Using this strategy solves the problem of efficiency and economics resulting from the variety of project networks. By varying the subnetwork level from the high level to the low level for minimizing modification knowledge the strategy removes the flaws of static leveled multiple cases approach and reduces the risk that determines the effectiveness of project network generation.

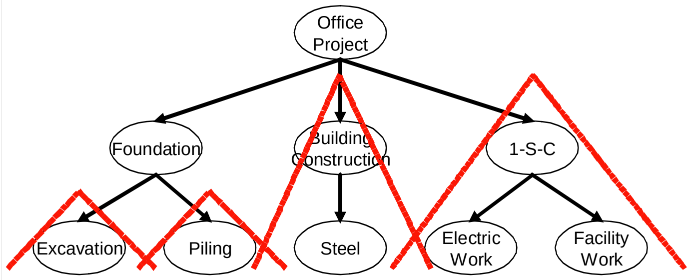

Figure 6. Middle level strategy

Network Breakdown Structure

To implement Dynamic Leveled Multiple Case (DLMC) approach we need information on the subnetworks that a specific case has and a Network Breakdown Structure for fast retrieval of subnetworks as in Figure 7. Network Breakdown Structure is stored in the case base with past project information and project networks and provides subnetwork information when a project network generator demands subnetwork information in case base.

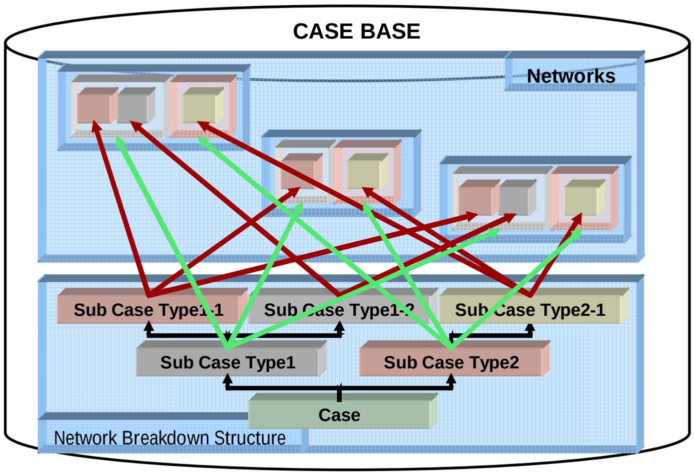

Figure 7. Network Breakdown Structure

4.2 Procedure for dynamic leveled multiple cases approach

The first step: The project specification analysis
In this step, we not only reduce the search space by filtering but also generate NBS for the new project with meta NBS. Meta NBS describes the relationship among subnetworks in a whole project network and consists of AND, OR, XOR (exclusive OR), and replication nodes. As in Figure 8, a project must include a Foundation, a Building Construction, and a Finalizing. Here Building Construction subnetworks can be included multiple times and should select either concrete structure or steel structure. Finalizing can include Electric Work or Facility Work or both.

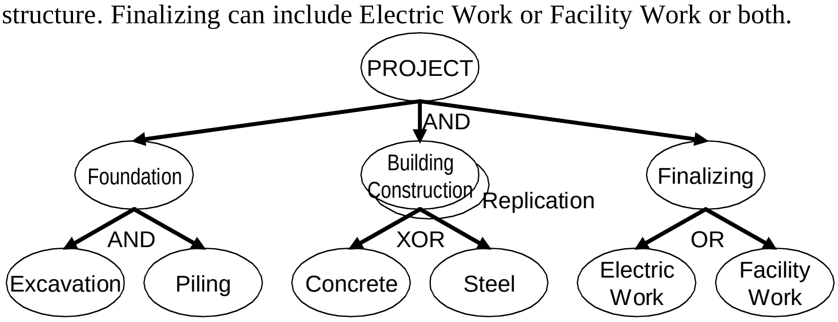

Figure 8. Meta NBS

Analyzing a project specification with rule-based inference produces a network breakdown structure necessary for a new project network. The following rule states that if a construction method is “top-down-concrete” or “bottom-up-concrete” then we need to select “concrete” rather than “steel”.

(fwd-rule NBS-gen-001 (office ^construction-method << "'top-down-concrete" '"bottom-up-concrete" >>)

```
-->
        (new-value      'concrete       'instance-num    1)
        (new-value      'steel          'instance-num    0))
```

Using the rules in the above, when a project network has both a "concrete"-type tower and a "steel"-type tower, we can produce a Network Breakdown Structure which is proper for a new project (Figure 9).

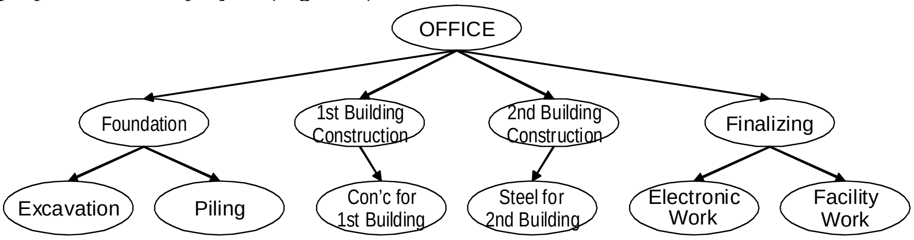

Figure 9. Generated Network Breakdown Structure

The second step: The dynamic leveled multiple sub network retrieval

The criteria for selecting subnetworks depend on the network modification efforts and the network synthesis efforts. Calculating modification effort is described in a previous section. Synthesis effort is calculated in such a way similar to that of adding a new activity. Therefore, the calculation of synthesis efforts is as follows: Synthesis effort: count(Con(A_{NS}^C : A_{NS}) $\cup$ Con(A_{NS} : A_{NS}^C)) where A_{NS} is the set of activities included in newly selected subnetwork. The level of subnetworks retrieved is determined so that the calculated efforts for modification and synthesis can be minimized.

The third step: Subnetwork modifications by addition or deletion of activities

This step is the same as that explained in section 3.

The fourth step: Initial project network generation by subnetworks synthesis

Synthesizing new subnetworks into a new project network is implemented by converting the precedence relationship knowledge into precedence relationship between activities in the subnetwork and other selected activities (Figure 10). To combine 'Piling' sub-network to the project network, add sub-network {A3, A4, R3} and generate relationships {R13, R14, R16, R17} by converting the related precedence constraints.

The next steps are the same as those explained in section 3.

The fifth step: Activity features (duration and assigned resource) modification

The sixth step: Network analysis and constraint satisfaction

The final step: Network executed, customized, and restored into case base

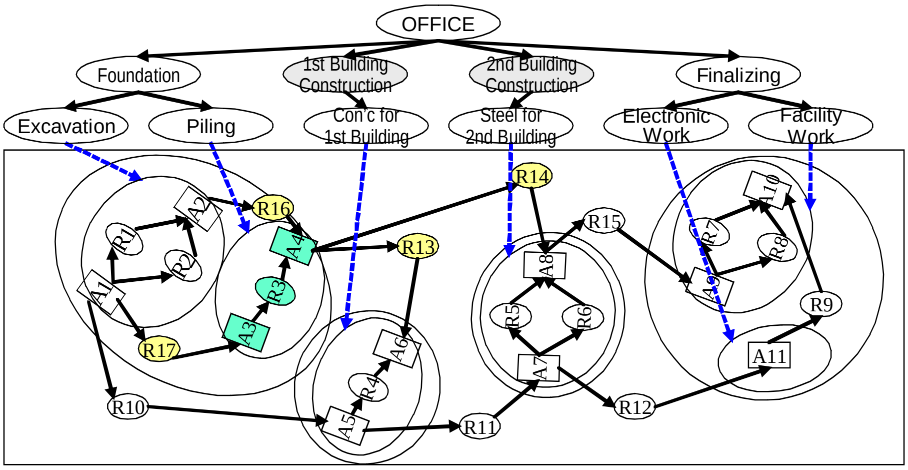

Figure 10. Subnetwork synthesis

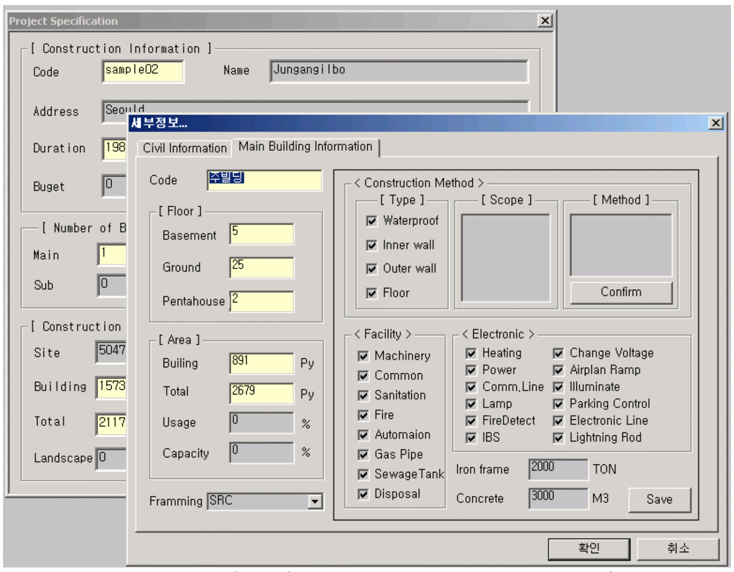

Figure 11. User interface for inputting the project specifications

4.3 Implementation of FASTRAK-OFFICE with DLMC

We have developed the office building project network generator (FASTRAK-OFFICE, Figure 11) using the dynamic leveled multiple cases approach. The FASTRAK-OFFICE had 236 precedence constraints, 25 analysis rules for generating NBS and with only 5 full past project network when it had started even the office project network is more complex then the apartment project network. If we had developed it with the single case approach then we had to gather more precedence constraints in the worst case 12,090 constraints would be needed ($_{156}C_2$) with only 5 full past project network, and gather more past project networks up to 3,360 networks with only 236 precedence constraints.

5. Conclusion

Most of project-based industries such as construction, shipbuilding, and software development etc. should generate and manage project network for successful project planning. In this paper, we suggested a set of criteria of good project network generator such as network generation efficiency, the quality of the network, and economics of system development. For the efficiency of the planning, the first criterion, we decided to take a case-based approach. However, using only previous cases is insufficient to generate a proper network for a new project. By embedding rules and constraints in the case-based system, we could improve the quality of the project network: the second criterion. Interestingly, we found that the integration of CBR approach and the knowledge-based approach makes feasible the development of the project network generator and improves the quality of the network by mutual enhancement through crosschecking the knowledge and cases in then development and maintenance stages. However, for some complex project network planning, we had to refine a single-case assumed project network generation methodology into Dynamic Leveled Multiple Case (DLMC) approach. The DLMC methodology contributes again to the efficiency and effectiveness of project network generation and reduces the system development effort.

References

[1] Hendrickson, C.; Zozaya-Gorostiza, C.; Rehak, D.; Baracco-Miller, E.; and Lim, P. 1987. An Expert System for Construction Planning. Journal of Computing in Civil Engineering 1(4): 253–269.

[2] Kartam, N.; Levitt, R.; and Wilkins, D. 1991. Extending Artificial Intelligence Techniques for Hierarchical Planning. Journal of Computing in Civil Engineering 5(4): 464–477.

[3] Lee, J. K. 1994. UNIK User's Manual (in Korean), Intelligent Information Systems Laboratory, Korea Advanced Institute of Science and Technology, Seoul, Korea.

[4] Lee, K., Kim, H., Lee, J., and Kim, T., "Case- and Constraint-Based Project Planning for Apartment Construction", AI Magazine, vol.19, no.1, pp.13-24, Spring, 1998.

[5] Navinchandra, D.; Sriram, D.; and Logcher, R. 1988. GHOST: Project Network Generator. Journal of Computing in Civil Engineering 2(3): 239–254.

[6] Ory, S., and Abraham, W. 1995. Knowledge-Based System for Construction Planning of High-Rise Buildings. Journal of Construction Engineering and Management 121(2): 172–182.

[7] Winstanley, G.; Chacon, M.; and Levitt, R. 1993. Model-Based Planning: Scaled-Up Construction Application. Journal of Computing in Civil Engineering 7(2): 199–217.

[8] Zhang, D., and Maher, M. 1995. Case-Based Reasoning for the Structural Design of Buildings. Paper presented at the Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE) 95, 6–8 June, Melbourne, Australia.