Case- and Constraint-Based Project Planning for Apartment Construction

Kyoung Jun Lee, Hyun Woo Kim, Jae Kyu Lee, and Tae Hwan Kim

■ To effectively generate a fast and consistent apartment construction project network, Hyundai Engineering and Construction and Korea Advanced Institute of Science and Technology developed a case- and constraint-based project-planning expert system for an apartment domain. The system, FASTRAK-APT, is inspired by the use of previous cases by a human expert project planner for planning a new project and the modification of these cases by the project planner using his/her knowledge of domain constraints. This large-scale, case-based, and mixed-initiative planning system, integrated with intensive constraint-based adaptation, utilizes semantic-level metaconstraints and human decisions for compensating incomplete cases imbedding specific planning knowledge. The case- and constraint-based architecture inherently supports cross-checking cases with constraints during system development and maintenance. This system has drastically reduced the time and effort required for initial project planning, improved the quality and completeness of the generated plans, and is expected to give the company the competitive advantage in contract bids for new contracts.

Generation, verification, and modification of construction project schedule networks in the PERT-CPM (project evaluation and review technique–critical path method) chart are the essential tasks for successful project planning and management in the construction industry. Figure 1 is a partial screen of a project network for apartment building construction. The horizontal solid lines denote the various construction activities, and the thick lines indicate the activities on the critical path.

Because a project network consists of hundreds of activities and precedence relationships, project planning is a time-consuming and knowledge-intensive task. For example, even for a senior engineer with 10 years of experience in construction planning, it takes a couple of days to make a project network of apartment building construction. To compete with other companies for a contract, it is critical for a construction company to quickly generate a good and consistent project plan.

Hyundai Engineering and Construction (HDEC) has won a worldwide reputation among its clients in South East Asia, Middle East, and North America, as well as in domestic markets, during the past 50 years. The company's key construction activities consist of building multistory housing complexes, hospitals, hotels, airports, offices, and multipurpose commercial buildings. Hyundai supplied about 17,000 units of housing in 1995 especially for the housing project. The total revenue of this company for 1995 was about five billion dollars.

HDEC has been dedicated to the development of a series of expert systems for the automatic generation, verification, and modification of construction project networks. This five-year project has been performed in cooperation with the Intelligent Information Systems Laboratory of the Korea Advanced Institute of Science and Technology (KAIST) and Hyundai Information Technology (HIT) Co.

The first domain we attacked was the apartment project because this domain is relatively structured, and we already had much experience and data. Since 1996, we have been developing the system for bridge and power transmission tower construction planning. The core methodology of this project has been the case- and constraint-based approach.

Rationale for Using the Case- and Constraint-Based Approach

The reason we chose the case- and constraint-based approach is intuitive. We observed how the domain experts at Hyundai made project

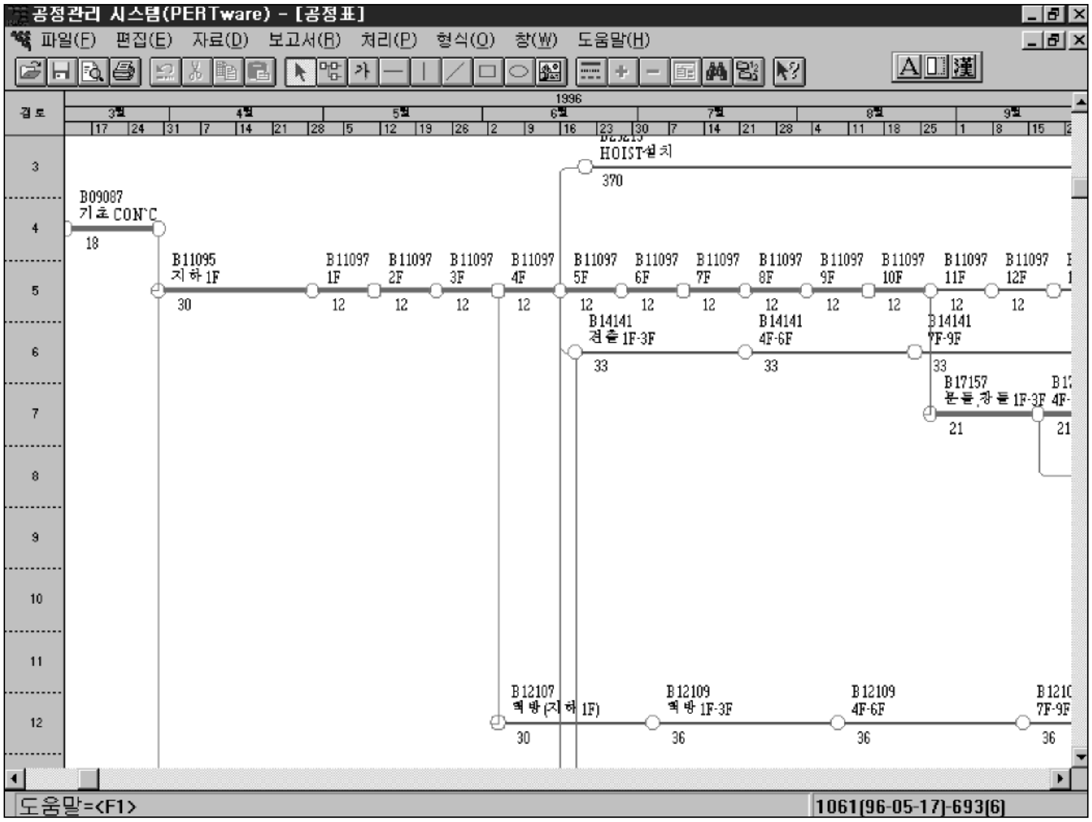


Figure 1. A Construction Project Network.

networks. From this observation, we could see that the experts refer to past cases and start their work based on the most similar case. We also found that they modify the old case to create a new case using their own constraint-type knowledge.

It is difficult for a human project planner to newly generate a project network without referring to any previous cases because the amount of required knowledge is too large and complex. Therefore, human experts usually look for a similar previous project network for reference and get previous human expert knowledge by referencing previous similar networks.

To simulate the expert's use of a past similar case, the case-based reasoning (CBR) approach was adopted as the fundamental AI technique in this project. To date, most studies in CBR were developed for toy problems (see, however, Aamodt and Plaza [1994]). Our project is a full-scale study applicable to a real-world situation and deals with very large cases, each of which typically consists of hundreds of activities and hundreds of precedence relationships. In addition, as Shrobe (1996) pointed out in the excellent review of past Innovative Applications of Artificial Intelligence conferences, there has been a significant gap between practical uses of CBR, which are limited almost totally to case retrieval, and the full paradigm as it emerged in the research community, which involves not just retrieval but also case adaptation and debugging. Our system, FASTRAK-APT, deals intensively with case adaptation and maintenance as well as case retrieval in a large-scale, real-world context.

A new case usually has some discrepancies with the past cases; so, a retrieved case has to be modified to fill the gaps. In the construction domain, the modification implies the addition and deletion of some activities and their associated precedence relationships. For example, suppose the past case involves a 20-floor building, but the new one is for an 18-floor building. We have to delete the activities for the nineteenth and twentieth floors. After deletion, the succeeding activities should be pulled forward, unless a constraint is violated. To effectively modify the previous network to make it suitable for the new construction project, we adopt a constraint-based case-adaptation approach.

Integration of the case-based and constraint-based approaches has the following two advantages: First, construction domain constraints acquired from field experts compensate for incomplete cases that imbed specific planning knowledge and consequently improves the system's performance. Second, during system development, cross-checking of cases with constraints improves the quality of both of them. Through the cross-checking process, the system developers can refine the previous cases to the high-quality referential cases and simultaneously validate and verify the domain constraints.

Previous Approaches

In the area of project management, there has been a lot of research and development of network-based project-planning methods and management techniques, assuming that a project network is given to the project manager (Bent and Thumann 1994). However, as Levitt, Kartam, and Kunz (1988) pointed out, the traditional network-based planning tools are knowledge-poor analysis tools that depend on knowledgeable managers, who must analyze the project to provide meaningful data and interpret the significance of the output data. In addition, they have no capability for generating project plans.

To overcome the limitations, there has been some research to automate or support the generation of progress networks using AI or knowledge-based techniques, such as CONSTRUCTION PLANEX (Hendrickson et al. 1987), GHOST (Navinchandra, Sriram, and Logcher 1988), SIPE-2 (Kartam, Levitt, and Wilkins 1991), and HISCHED (Ory and Abraham 1995).

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 (Winstaniey, Chacon, and Levitt 1993), 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 constraints because the system uses past cases containing these precedence constraints.

Zhang and Maher (1995) 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. For many domains where construction planning knowledge is difficult to acquire and might not be applicable objectively, the case-based paradigm can provide a model for the acquisition and reuse of specific planning knowledge because previous cases contain much valuable knowledge in themselves.

This project is the first time that Hyundai has adopted an AI technique for project planning. Previously, human planners created the project plan without any automated aid. The company has also never had a structured case base accumulating past project cases. Some other companies have tried a database-oriented approach, but it is not effective because it requires much knowledge and effort to modify a project network. For these reasons, Hyundai and KAIST considered using AI techniques: the case-based technique for reusing and accumulating good cases and the constraint-based technique for adapting a past case according to the construction knowledge.

Application Description

Figure 2 shows the architecture of FASTRAK-APT. The central box is the kernel system with six modules for case retrieval and adaptation. On the right, we have a couple of constraint bases containing knowledge about activity existence, precedence relationships between activities, and subnetwork connection as well as a database for work-breakdown structure (WBS) and resources. On the top is a project case base and a case-management knowledge base. When the user inputs a design specification for a new project, the system retrieves the most similar case from the case base, adapts it, and shows it to the user. The user can then express his/her intention or management strategy, and the system can respond to the new requirement. Sometimes, the system can ask the user about an important decision such as the relaxation of constraints. In this case, the user can choose some constraints among the candidates.

As such, the system supports a *mixed-initiative planning procedure* (Veloso 1996): A user can interact with the system by inputting the design specification, accumulating good cases into the case base, selecting constraints among the relaxation candidates, and informing the system of his/her intentions or management strategies based on the project situation.

To simulate the expert's use of a past similar case, the case-based reasoning (CBR) approach was adopted as the fundamental AI technique in this project.

Knowledge Representation

We use a frame-based representation scheme for representing design specification, cases, and constraints. For this representation, we used the expert system tool UNIK-FRAME (Lee 1994), which was developed by KAIST. The

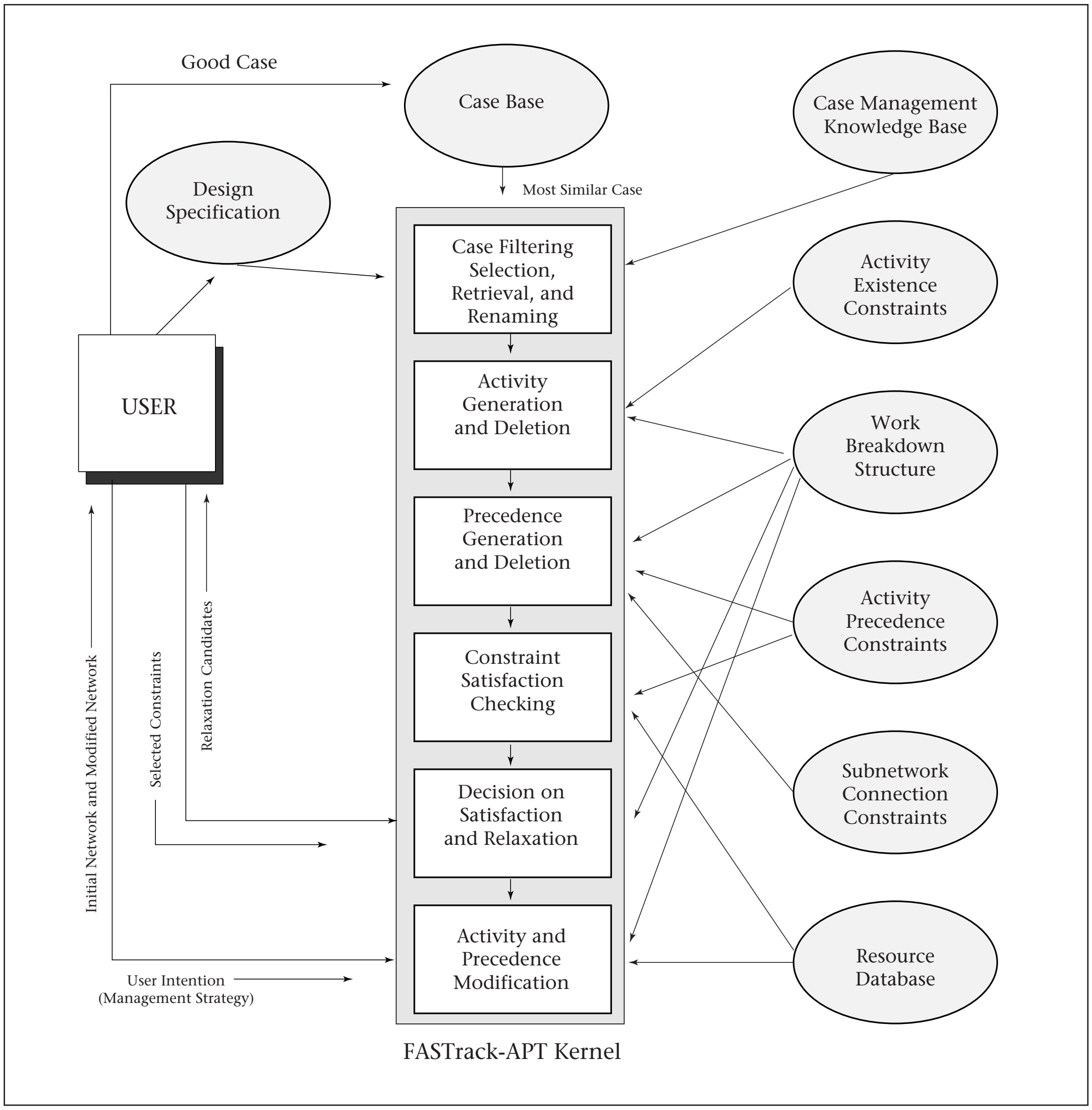


Figure 2. Architecture of FASTRAK-APT.

design specification of an apartment is illustrated as follows:

```
{{Hukseok_Hyundai_Apt_Project
    is-a                :Project
    name                : Hukseok Hyundai Apt
    address             : Hukseok, Seoul
    start-date          : 1995/05/01
    due-date            : 1996/06/21
    ground-type         : Flatland
    topography          : (clay 40%)(fragile-rock 60%)
                          (soft-rock 0%)(solid-rock 0%)
    number-of-building: 1
    area                : 5707
    construction-area   : 571
    household           : (36 68)
}}
```

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. The following example shows a frame representation of an apartment building for the previous project.

```
{{Hukseok_Hyundai_Apt_1
    is-a            : APT
    part-of         : Hukseok_Hyundai_Apt_ Project
    apt-type        : Corridor
    base-last       : 1
    ground-last     : 10
    pent-last       : 1
    phases          : 2
    start-date      : 1995/05/01
    finish-date     : 1996/06/21
    household       : (36 34)
    construction-method: PC-framing, …
}}
```

Thus, the apartment building has a 10-floor building with a 1-floor basement and a 1-floor penthouse. Users input these basic characteristics of the project into the system. Figure 3 is

Figure 3. Input Screen for Project Specification.

the input screen for the project specification. The three windows are for inputting the overall specification, the building specification, and the construction methods.

With this information from the user, the system carries out its procedures, such as case retrieval, activity generation and deletion, precedence relationship generation and deletion, constraint-satisfaction checking, decision on satisfaction and relaxation, and activity and precedence modification. We explain the main procedures of the kernel system one by one.

Case Filtering and Retrieval

When the user inputs the design specification of the current project, as in figure 3, FASTRAK-APT selects the most similar case using a two-step procedure: The first step is *case filtering*. Because not every case in the case base can always be adapted for the current project, we should filter out cases that, if adopted, could lead to adaptation failure.

The case-filtering knowledge is in the case-management knowledge base. The filtering knowledge has been accumulated through the experience of adaptation failure during system development and testing. For example, the variable *phases* is one of the variables used for the filtering process. The filtering module filters out the cases that have a different *phases* value than that of the current case because the variable determines the grouping of concurrent activities and, therefore, significantly affects the topological shape of project networks. After the filtering process comes step 2: We select the most similar case among the remaining candidates using the similarity measure. For the effective retrieval of suitable cases, we extracted six important properties from domain experts, as follows, and used the weighted sum of the distance functions of the variables:

Distance = $\Sigma_i W_i X_i$
W_i = weight of each variable
X_1 = difference in number of floors
X_2 = difference in construction method
X_3 = difference in number of households
X_4 = difference in building space
X_5 = difference in ground type
X_6 = difference in topography of buildings.

The upper window of figure 4 shows the candidate cases with each similarity value. The user can check the specification of each case in the lower window. After that, the user can finally decide which case should be used as a base plan.

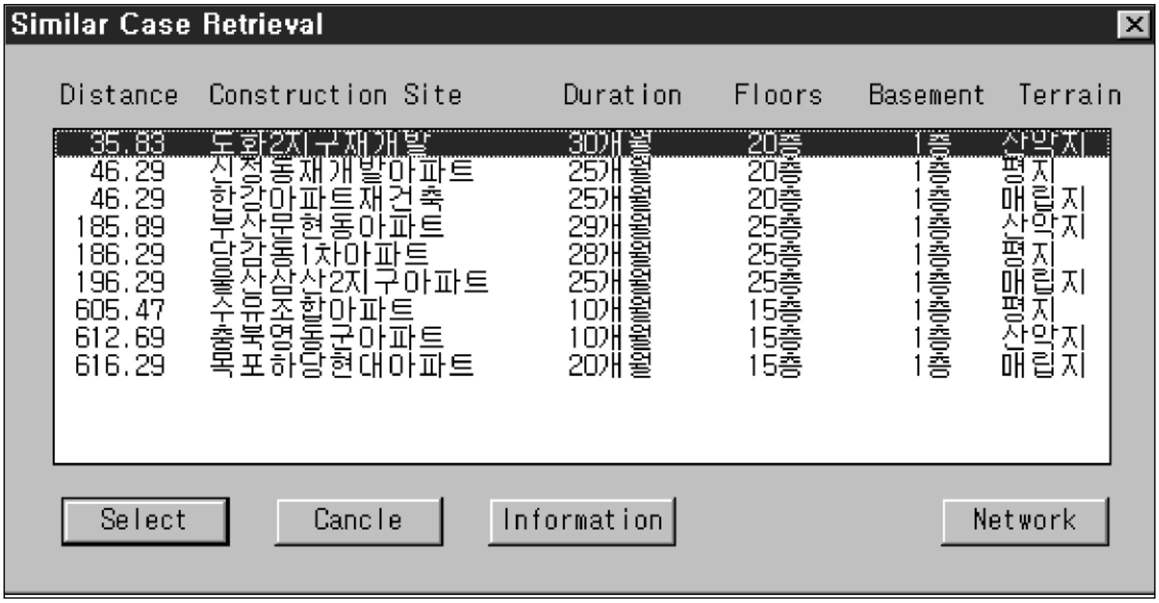


Figure 4. Screen for Case Retrieval.

After the selection and retrieval, the system renames the activities and their precedence relations of the retrieved case using a naming rule that gives the unique codes to entities.

Activity Generation and Deletion

The retrieved case usually needs some activities to be added or has some activities to be deleted; so, the next step is activity generation and deletion. There are two types of activity generation and/or deletion in our system: The first is caused by the difference of the number of floors (ground or basement level) between the selected case and a new project. For example, if the selected case is a 20-floor apartment, but the new project is an 18-floor building, then the activities for the nineteenth and twentieth floors should be deleted.

The second cause of activity generation and/or deletion is the difference between employed construction methods. In the apartment domain, there are about 20 variables for the choice of construction method. We generate and delete activities using the *activity existence constraint*, which defines the relationship between the construction method and activities. For example, if the selected case used a reinforced piling method, but the new project will use a preenforced concrete piling method, RC (reinforced concrete) piling activities (b09073, b09077, b09079) should be deleted, and PC (preenforced concrete) framing activities (b09075) should be generated. The following object represents this constraint, and figure 5 shows the screen for browsing through such existence constraints:

```
{{Construction-Method-Constraint-1
    is-a            : construction-method-
                      constraint
    method-type     : piling
    value           : PC-piling
    use             : b09075
    delete          : b09073 b09077 b09079
}}
```

Precedence Generation and Deletion

When activities are generated or deleted, the precedence relations associated with them should be created or removed. We integrate the two methods for generating and deleting the precedence relations: principle-based approach and constraint-based approach.

The *principle-based method* uses general network principles for maintaining project networks. Bell's (1989) work can be classified here. We maintain the soundness of a project network by keeping the following basic principles: (1) there should be no cycle in the network, (2) there should be no isolated activity, and (3) there should be only one start node and only one end node.

For example, in the case of the 18-floor building, if an eighteenth-floor activity succeeded by an activity is deleted, the seventeenth-floor activity of the same kind as the eighteenth should be succeeded by the activity. As such, this approach is not based on knowledge but on logical rationale.

However, it is not sufficient to use only the principle-based approach for maintaining a construction project network. To use the useful knowledge of the construction experts, we should also use the domain constraint-based method. The *constraint-based method* uses precedence constraints acquired from domain experts. If the newly generated activity has an associated precedence constraint, it can be converted to a new precedence relationship satisfying itself. In addition, when planning multiapartment building construction, we need to connect the network of each building. Thus, we use subnetwork connection constraints that define the interbuilding activity precedence relationship.

Constraint-Satisfaction Checking

Constraints are used to verify the current project network as well as to generate relevant activities and relationships. The first step in constraint-satisfaction checking is calculating the earliest start time, the earliest finish time, the latest start time, and the latest finish time of all activities using the critical path method (CPM). With these values, we check the satisfaction of the constraints. We have acquired approximately 430 domain constraints about activity existence, activity precedence relationships, and the subnetwork connection. For the convenient representation and maintenance of the large number of constraints, we should represent them on a semantic level. However, for the reasoning efficiency of the system, we should tightly couple the constraints into the activities. To satisfy the two criteria, that is, user convenience and reasoning efficiency, we developed a *metaconstraint representation method* where a semantic-level constraint for users is converted to a couple of instantiated constraints.

In addition, most constraints should be able to have different parameter values contingent to the situation. For example, when we construct a 10-floor apartment, the first floor's plastering activity starts after starting the third floor's framing activity. However, in the case of a 15-floor apartment, the first floor's plastering activity should start after starting the fourth floor's framing activity. To support the contingency, we connect forward-chaining rules with the metaconstraint. The forward-chaining inference is supported by the tool UNIK-FWD (Lee 1994) developed by KAIST. The following frame shows an illustrative constraint for the earlier example:

```
{{Precedence-Constraint-3-b
   is-a                    : precedence-constraint
   relationship-type       : FS ; Finish-to-Start
   relationship-operator   : >=
   relationship-value      : 0
   value-type              : day
   predecessor-wbs         : framing
   predecessor-floor       : <contingent-value-1>
   rule-groups             : (<contingent-value-1>
      Precedence-Constraint-3-b-rule-group)
   successor-wbs           : plastering
   successor-floor         : <base-first>
   importance              : 0.7
}}
```

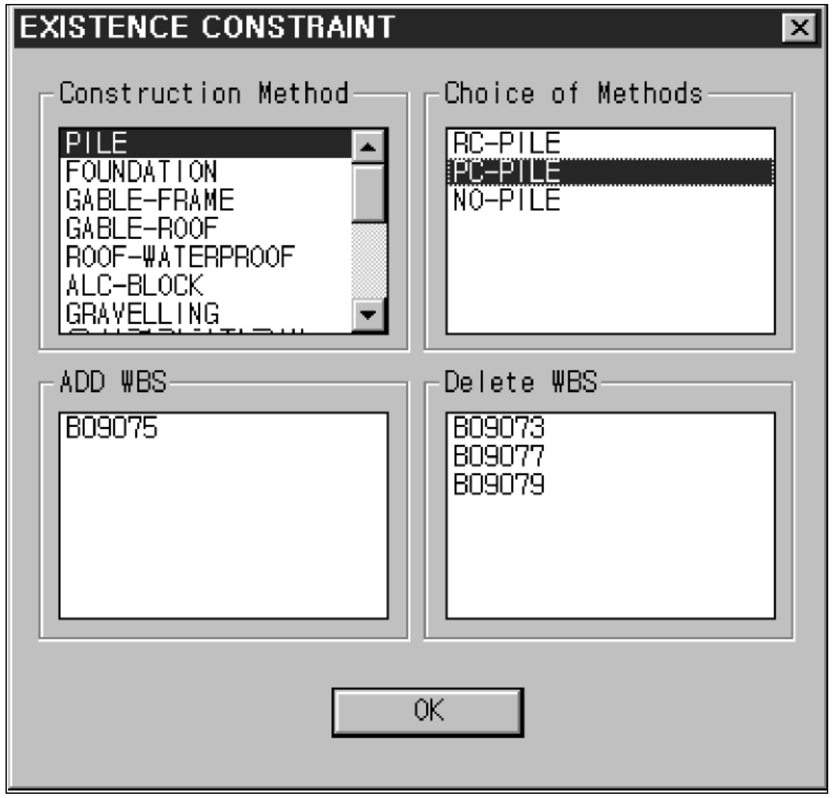


Figure 5. Browsing Existence Constraints.

This constraint means that the basement's first-floor plastering activity can start after some floor's framing activity finishes. With only this constraint, we cannot know which floor of the framing work should be finished for the plastering activity because it depends on the number of floors in the building. For this information, the system refers to the slot *rule-groups,* and then it can find the following associate rules:

```
(Fwd-Rule Precedence-Constraint-3-b-1
[Rule-Group Precedence-3-b-rule-group]
(precedence-control ^current-apt <apt>)
(apt ^frame-name (= <> <apt>)
     ^ground-last (<= <> 12 ))
→ (new-value     'precedence-control
                 'contingent-value-1 3))

(Fwd-Rule Precedence-Constraint-3-b-2
[Rule-Group Precedence-3-b-rule-group]
(precedence-control ^current-apt <apt>)
(apt ^frame-name (= <> <apt>)
     ^ground-last (>= <> 13 ))
→ (new-value     'precedence-control
                 'contingent-value-1 4))
```

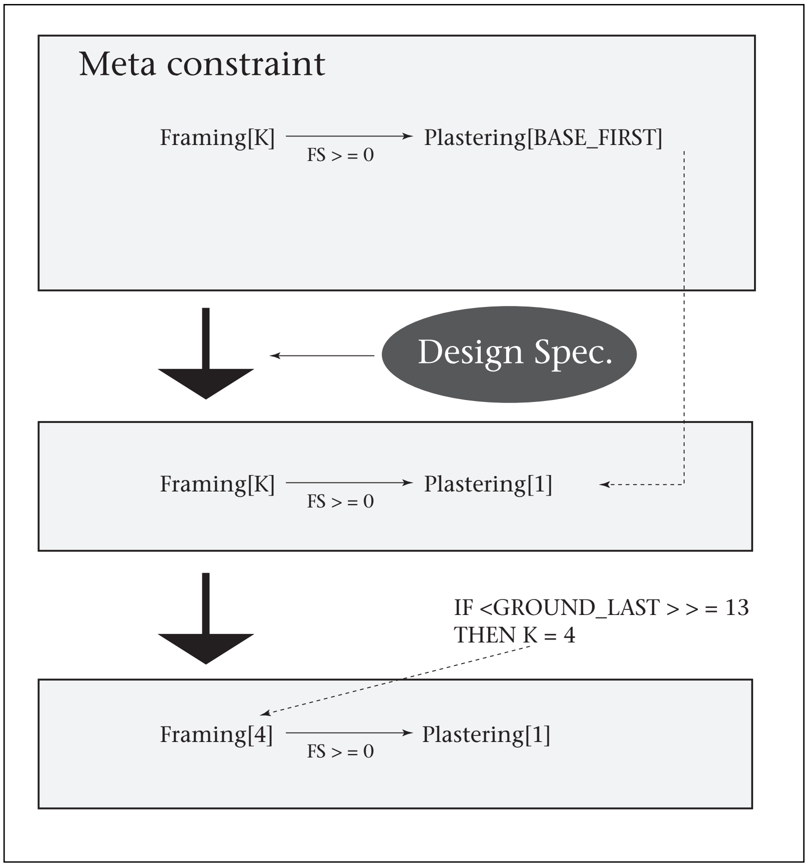


Figure 6. Instantiation of a Metaconstraint.

These two rules belong to the referred rule group of the metaconstraint *Precedence-Constraint-3-b*. The first rule means that the contingent value should be 3 when the building has less than or equal to 12 floors. The second rule implies that the contingent value should be 4 when the building has more than or equal to 13 floors.

Figure 6 shows the instantiation process using forward-chaining inference. In the first box, we can see the metaconstraint. Using the information from the design specification, we assign the value of 1 to the BASE_FIRST variable. If we assume that the building has more than 13 floors, the contingent value has the value of 4 by the rule. As a result, we can now see the instance of the metaconstraint, as follows:

```
{{PC3-B-P2B1-0
    is-a                   : precedence-constraint-instance
    value-type             : day
    relationship-value     : 0
    relationship-operator  : >=
    relationship-type      : FS
    predecessor            : Hukseok_Apt_1-Framing-04
    successor              : Hukseok_Apt_1-Plastering-01
    importance             : 0.7
}}
```

The constraint instance now has the specific predecessor and successor. This constraint says that as soon as the fourth-floor framing activity finishes, the first-floor plastering activity can start.

Decision on Satisfaction and Relaxation

Using the constraint instances made from the metaconstraints, we can check how many constraints that the current project network violates. If a violated constraint is found, we should determine whether it should be satisfied or relaxed. There are two ways that the decision can be made: (1) automatically and (2) manually. For automatic selection of the constraints to be relaxed, each constraint has an *importance* value. Constraints with the higher importance value should be satisfied preferentially. The other way is *user selection*. If the system shows the violated constraints, then the user can select some of them to be relaxed. In figure 7, the upper window shows all the violated constraints. If the user selects some constraints and clicks the Add button, then the system moves the constraints to the lower window, which means they should be satisfied. The Add All button can be used when a user wants to satisfy all the violated constraints. If the user wants to relax some constraints, he/she clicks on the Delete button after selecting the constraints in the lower box.

Activity and Precedence Modification

The modification occurs by satisfying a violated constraint or satisfying a user's intention. Because every constraint instance has a specific predecessor and successor, the simple way to satisfy a violated constraint is converting it to a new precedence relationship. However, adding a precedence relationship into the network can increase the project "makespan"; therefore, it is important to stabilize the *makespan* (that is, the total duration required to finish the project) and simultaneously satisfy important precedence constraints. We have three methods for reducing the makespan: First, we can delete relatively unimportant precedence relationships. The precedence relationships not defined in the precedence constraint base are good candidates to be deleted. Of course, the user should confirm the deletion of a precedence relationship. Second, we can reduce the value of the precedence relationship as far as satisfying its associated precedence constraint. Finally, we can reduce the duration of activities in the critical path. To prevent an unreasonable reduction in the duration, the system uses the WBS

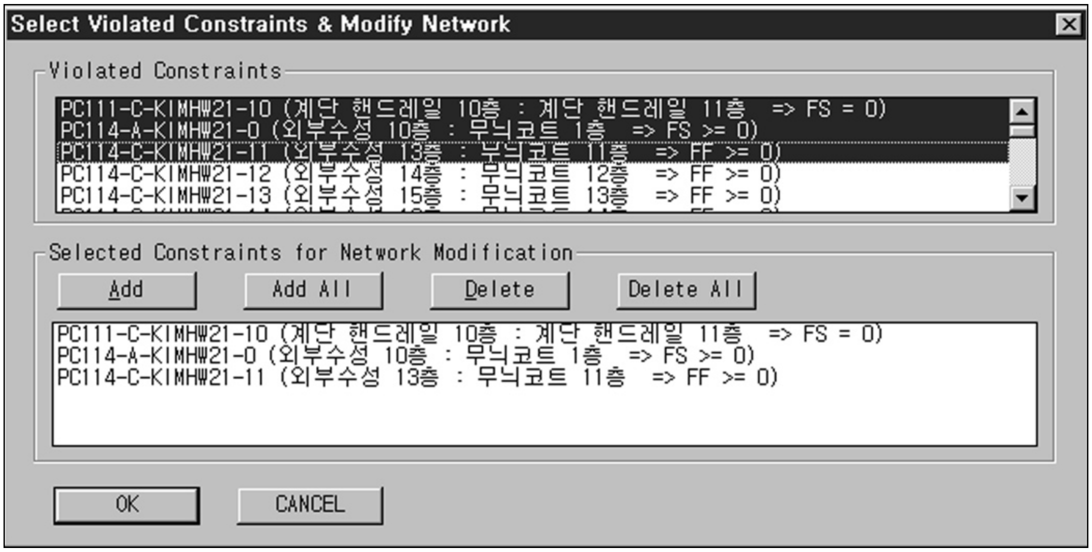


Figure 7. Selecting Violated Constraints.

database and considers the status of the resource utilization.

Evaluating and Enhancing Solutions

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 results were used for enhancing the case base, and the refined case base helped improve the quality of generated plans. Cross-checking and mutual enhancement are important benefits from the integration of the case-based and the constraint-based approaches. The running time of the system is about 5 minutes a building from case loading to adaptation (a 20-floor building has about 500 activities and 700 precedence relationships).

Knowledge Acquisition and Maintenance

Before entering the knowledge-acquisition phase, we standardized and constructed a three-level WBS database, which contains information such as average duration, hierarchical structure, location, and seasonal factor of each activity. Figure 8 shows the WBS information editor window of our system, which contains the specific information of the ground-rc framing activity, such as its average duration, resource unit, and quantity. Ten domain experts participated in the knowledge-acquisition task. We then made a cross-table of all WBS activities and let an expert with nine years' experience mark in the cell if the two activities had any technical precedence relationships. The results were checked and refined by five other experts. The collected technical precedence constraints could not be completed because there can be some managerial and conventional precedence relationships. However, we could see that the lack of information was compensated for by the case base.

Our system also provides a constraint editor for easily maintaining the expert knowledge, as in figure 9. It shows the knowledge implying that the *n*+1th activity of WBS B11095 can start immediately after finishing the *n*th activity of the same WBS. Users can correct and update the knowledge by changing the index ranges, relation types, floor numbers, and so on.

Implementation and Maintenance

FASTRAK-APT has been implemented on WINDOWS 95 using VISUAL C++ and UNIK (unified knowledge) (Lee 1994), an expert system tool developed by KAIST. Because KAIST has the right to use and enhance the source codes of UNIK, it is a good choice for developing flexible and expandable systems. Currently, FASTRAK-APT has about 430 metaconstraints that can create thousands of instantiated constraints (for example, approximately 2000 for a 20-floor building). To date, we have accumulated 50 high-quality cases prepared and verified by human experts. The project team consists of one project manager, two research program-

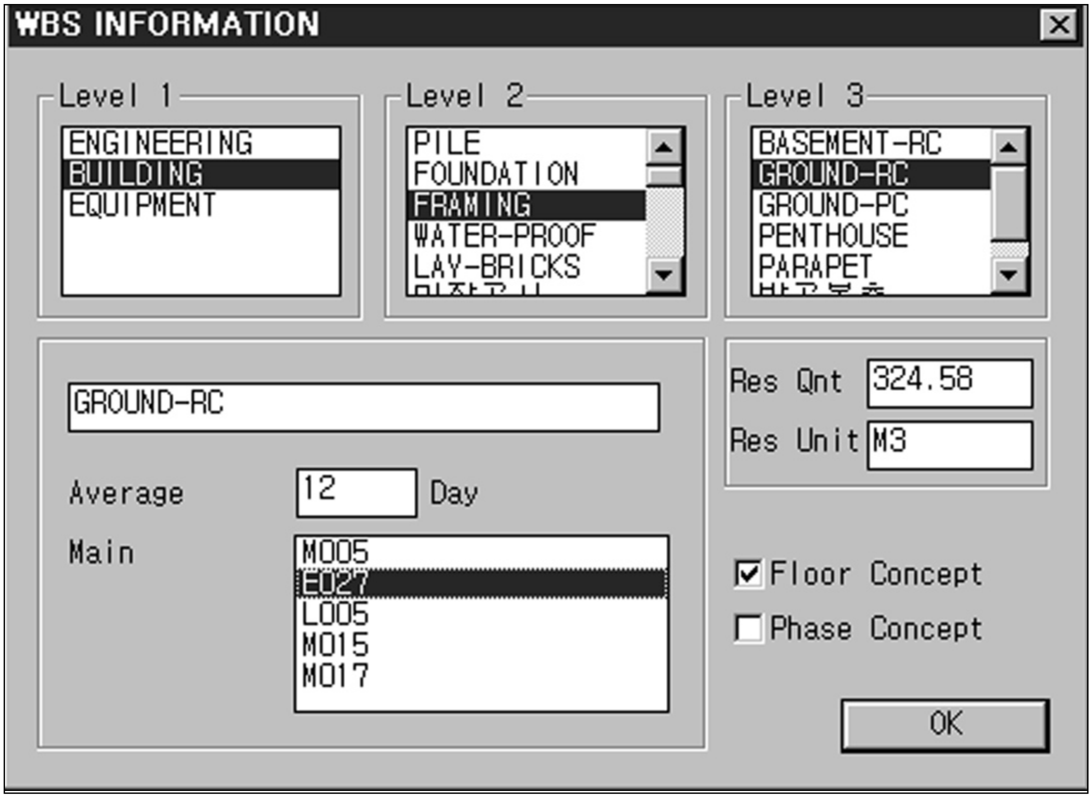


Figure 8. Work-Breakdown Structure Editor.

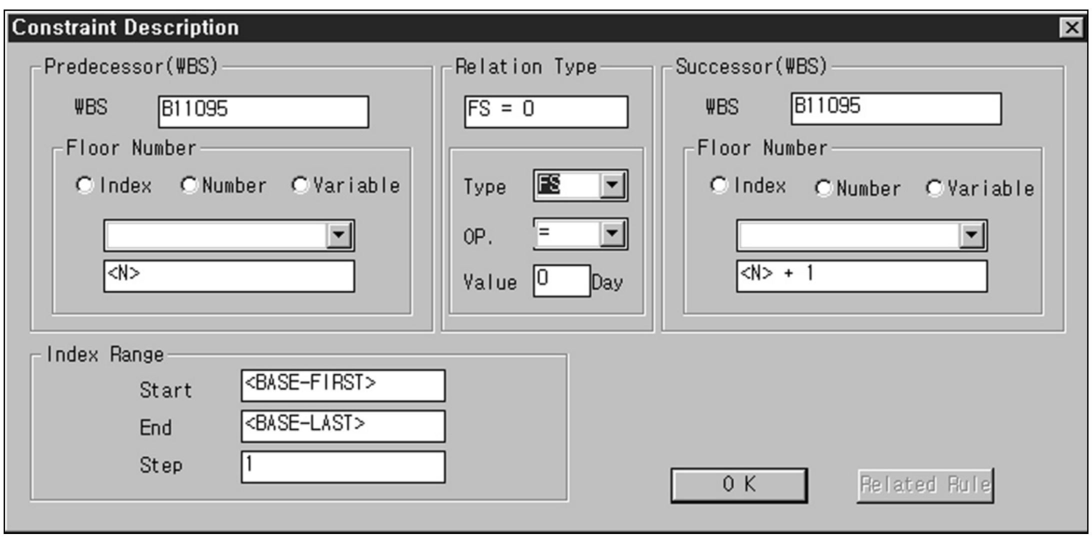


Figure 9. Precedence Constraint Editor.

mers, two application programmers, and three domain experts. The roles of each group are described in table 1. Hyundai and KAIST have been extending this system since September 1996 for bridge construction and power transmission tower construction planning. Therefore, application programmers from industry have been changing their roles to research programmers for the new systems.

Project manager	1	Methodology consulting Overall management and control
Research programmers (KAIST/HIT)	2	Knowledge engineering Case base design Kernel system development Research and development on bridge case
Application programmers (HIT/HDEC)	2	Graphical user interface development Interface to project management software System verification and maintenance Research and development on power transmission tower
Domain experts (HDEC)	3	Requirement and specification Knowledge transfer and verification Case base construction System validation Knowledge maintenance

Table 1. Roles of Members in Each Group.

Development Cost

The development costs of FASTRAK-APT for the two years were calculated at approximately $42,000 for the hardware, $162,000 for the outsourcing, and $417,000 for internal personnel. The total cost was about $621,000.

Application Use and Estimate of Payoff

FASTRAK-APT has been used by the Construction Management Department since September 1996 and is now ready for use by the construction sites. 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 0.5 person-day. The company expects to be able to reduce the effort to complete a bid document from 10 person-days to 1 person-day if the system is enhanced to support aggregated resource use and the activity cost calculation. The expected annual benefit using these parameters is about $616,000. Using FASTRAK-APT relieved the company of its problems from a deficiency in project management experts. In addition, the faster simulation and feasibility analysis gives the company a competitive advantage in bids for new contracts. The accumulated good cases and the digitalized and refined knowledge became invaluable assets for the company. In addition, the company now uses the system to train employees for construction management.

Conclusions

We think that FASTRAK-APT is a successful, large-scale, real-world application of the case-adaptation approach to construction. Our approach can be regarded as a good strategic example of work on a large-scale planning problem. The experience of cross-checking and mutually enhancing cases and constraints was interesting, and the robust solution from the process is one of most important benefits of integrating CBR and constraint-based adaptation. We expect that our methodology can be generalized to other planning problems based on network representation.

Acknowledgments

This research is funded by Hyundai Engineering and Construction Co., Ltd., and Hyundai Information Technology Co., Ltd., in Korea. We are particularly grateful to Executive Directors Jae Ho Yoon and Seok Boo Choi for their strong support and helpful advice. We would also like to thank members of this project at Hyundai: Chang Gon Kim, Myoung Kyun Yoon, Eui Jun Hwang, and Hyun Jeong Park.

References

Aamodt, A., and Plaza, E. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications* 7(1): 39–59.

Bell, C. 1989. Maintaining Project Networks in Automated Artificial Intelligence Planning. *Management Science* 35(1): 1192–1214.

Bent, J., and Thumann, A. 1994. *Project Management for Engineering and Construction*. 2d ed. Buffalo, N.Y.: Fairmount.

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.

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.

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

Levitt, R.; Kartam, N.; and Kunz, J. 1988. Artificial Intelligence Techniques for Generating Construction Project Plans. *Journal of Construction Engineering and Management* 114(3): 329–343.

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

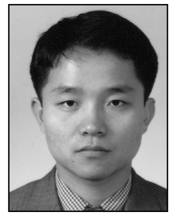
Ory, S., and Abraham, W. 1995. Knowledge-Based System for Construction Planning of High-Rise Buildings. *Journal of Construction Engineering and Management* 121(2): 172–182.

Shrobe, H. 1996. The Innovative Applications of Artificial Intelligence Conference: Past and Future. *AI Magazine* 17(3): 15–20.

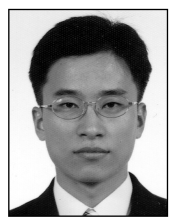
Veloso, M. 1996. Toward Mixed-Initiative Rationale-Supported Planning. In *Advanced Planning Technology,* ed. A. Tate, 277–282. Menlo Park, Calif.: AAAI Press.

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

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.



Kyoung Jun Lee is a senior researcher of the Graduate School of Management at the Korea Advanced Institute of Science and Technology (KAIST) and is also working as a chief researcher at the International Center for Electronic Commerce. He received his B.S. (1990), M.S. (1992), and Ph.D. (1995) in management science from KAIST. From 1996 to 1997, he worked as a visiting scientist in the Robotics Institute at Carnegie Mellon University. He won the Innovative Applications of Artificial Intelligence (IAAI) Award in 1995 and 1997 and has published papers in *AI Magazine, Expert Systems with Applications,* and *European Journal of Operational Researches*. His current research interests are in developing intelligent information systems for electronic commerce, supply chain management, and planning-scheduling problems.



Hyun Woo Kim is a Ph.D. candidate in management engineering at the Korea Advanced Institute of Science and Technology (KAIST) and a senior system engineer at Hyundai Information Technology Co., Ltd. He received a B.E. in industrial engineering and an M.E. in management information systems from KAIST. His research interests include planning expert systems, case-based reasoning, neural networks, and AI for electronic commerce.

Jae Kyu Lee is a professor in the Graduate School of Management at the Korea Advanced Institute of Science and Technology (KAIST) and the director of the International Center for Electronic Commerce. He received a B.A. (1973) from Seoul National University, an M.S. (1976) from KAIST, and a Ph.D. (1985) in decision sciences from the Wharton School at the University of Pennsylvania. He was the first president of the Korea Expert System Society and the chair of the Third World Congress on Expert Systems in 1996. His research interests include internet commerce, applications of AI, decision support systems, work flow, and business innovation.

Tae Hwan Kim has worked with Hyundai Engineering & Construction Co. in the field of project planning and control. Currently, he is an M.S. candidate in construction management at the University of Reading, United Kingdom. He received a B.S. in architectural engineering from Dong-A University. His research interests include the application of AI technology to construction management.