Case-Based Reasoning in an Integrated Design and Construction System

G N SCHMITT*

ABSTRACT

In representing and manipulating complex objects, such as buildings, or complex processes, such as design and construction, we propose to employ a new approach. *Case-based reasoning* (CBR) is a promising method for solving design and construction problems with minimal search. CBR maintains the high quality of an original case by adapting known and proven solutions from a case base to a new situation. *Integrated design and construction systems* are a key to making practical use of computers in design, building, and management. They guarantee the consistency of database information from phase to phase and prevent the unnecessary and error prone re-generation of information. Applying CBR to an integrated design and construction process helps to protect positive aspects of past solutions and allows for stepwise improvements. The paper presents application examples for case-based design, case-based design repair and management of the design process for a large project.

Key Words

case-based reasoning; integrated design systems; design process; construction process; project information management

INTRODUCTION

Rather than re-generating every new design from scratch, we propose using the knowledge embodied in existing cases for the design, construction and management processes. Cases represent knowledge in uncompiled form and are therefore quite different from prototypes which contain knowledge in compiled form. A case can represent both a physical object, such as a building, or a process, such as the construction or facility management process. These processes are parts of the overall building description because they contain essential procedural knowledge of how the building was designed, constructed and managed. Appropriate representations of the embodied knowledge exist for each of the individual processes. A case contains good solutions to specific problems, but it also includes unavoidable trade-offs. The case-based design systems we developed are able to adapt a selected case semi-automatically in order to solve a new design problem. Case-based design systems have the advantage of achieving a new, complex and complete product

* *Professor, Architecture and CAAD, Swiss Federal Institute of Technology, ETH Zürich, CH-8093 Zürich, Switzerland.*

description with a minimal search effort, while maintaining the quality of the existing case. With a case-based system, we can gather and manage important experiences for future project information systems (PIS).

This paper first provides necessary definitions and then discusses the use of cases to provide intelligent support for the following phases in the life cycle of a building: program development, design, construction, and facilities management. Following two examples, demonstrating the use of cases for design adaptation and repair, we propose an integrated design system. It is based on our experience in the development and testing of the Integrated Building Design Environment (IBDE) at Carnegie Mellon University (Fenves, 1990), and on extensive research work on case-based reasoning applied to design (Hua, 1992). The proposed system is under development for one of the largest current construction projects in Switzerland which is presently in its early design stages.

DEFINITIONS

Design reasoning with and based on cases has a long history in engineering and architecture. While experienced designers use their own case base accumulated over the years, inexperienced designers rely more on generative methods and on external case bases. Most architecture students learn to design using cases. Cases help us to understand a building or a structure as a complete entity including all of its inherent trade-offs. A case is the final, complex result of a successful design process. Therefore, cases are a very interesting topic for exploration in design computing. To limit the scope of the paper and to clarify the discussion, a definition of the main expressions is necessary.

Case-Based Reasoning and Case-Based Design

Case-based Reasoning (CBR) is an effective knowledge acquisition and representation scheme for producing complete problem solutions with minimal search. A significant advantage of case-based reasoning, over design using parameterised objects or prototypes, is that cases need not be parameterised before use. The parameterisation can occur during the reasoning process and need only to concern with the parameters of interest for a specific problem. Solutions are found quickly if a new problem has a close match in an existing case base or library. The disadvantages are similar to those of prototypes, in that limitations exist for the generation of truly new solutions.

Case-based Design (CBD) is the application of CBR techniques to design. Because most designs consist of complex objects, design cases can reach considerable size. Design cases may contain, among other things, abstractions and representations of geometry, function, performance, circulation, and the structural system. Adaptation of complex cases is the major challenge in architectural CBD. Starting from known cases is a common strategy for solving complex problems. CBD is different from the traditional rule-based approach for knowledge engineering, in that it does not attempt to generalise and compile the knowledge before the problem is specified. Applying this concept to computational processes, means using a given case as the initial state of the search process and searching through only the variations with significant similarity to the initial case. The process is divided into case selection and case adaptation, each of which can be solved independently. If the initial case is properly selected, the search for a solution can be carried out much more efficiently. The separation of case selection and case adaptation allows the introduction of manual guidance in the case selection which in turn reduces the difficulty of the machine computations. The implementation challenges of this approach are the capability of recognising potential solutions, if cases are to be selected automatically, and the capability to analyse the case and to derive case specific knowledge. For example, it is desirable but difficult to implement a CBD system that is able to recognise the positive aspects of a case that should be preserved during the adaptation, or to detect the negative aspects of the case that should be changed. The final challenge is to efficiently generate variations to solve the new problem and to modify the case accordingly while not disturbing desirable features.

CBD can integrate concerns related to the different representations and to apply adaptation mechanisms to the entire case, thus avoiding problems of combinatorial explosion or endless evaluation loops. Disadvantages are the difficulty of expressing design semantics and the problems of CBD in finding truly new solutions.

Integrated Design and Construction Systems

Integrated design and construction systems (IDCS) provide a computer-based environment for dealing with a building throughout its entire existence, from the early conceptual phase through its use and eventual demolition. As a simplification, we consider four phases: program development, design development, construction, and facility management. An IDCS must guarantee that relevant information passes from phase to phase and that no information is destroyed in the process. An IDCS normally consists of several compatible modules which allow the smooth exchange of information. The IBDE (Integrated Building Design Environment) project, initiated in 1986 at the Engineering Design Research Centre at Carnegie Mellon University, has achieved a prototype IDCS (Fenves, 1990). In IBDE, seven knowledge-based processes create information and display the results on message and status blackboards, which are accessible from all processes. A controller and a data manager handle the flow of data between the project data store and the other components of the system. A common display interface connects all modules (see Figure 1). The advantages of IDCS ' s over existing programs are obvious. Traditional CAD and analysis programs only deal with isolated aspects of the design process. In developing an IDCS, two extreme options arise: either the entire process from program development through facility management is seen as one object that can be manipulated and developed as a single entity; or, the process is seen as a sequence of interlocking modules. The output of each module must then be suitable input for the next module. While the first option is probably closer to the human memory representation, the second option is preferable for practical reasons.

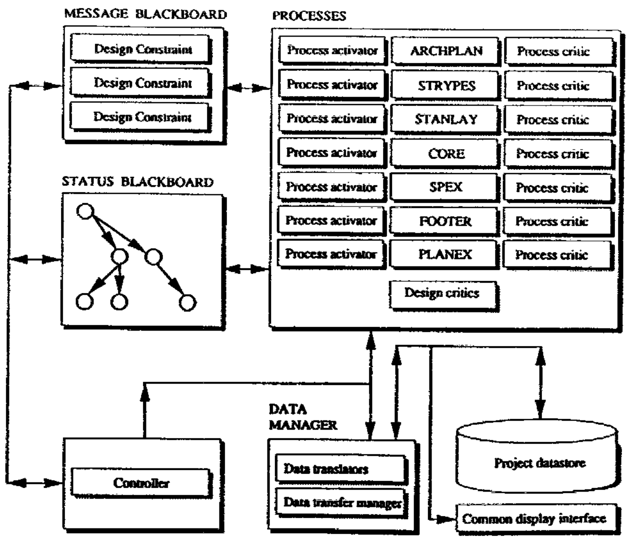

Figure 1. Architecture of the IBDE system. Processes, blackboards, data manager and project data store are linked in a distributed computing environment.

THE USE OF CASES

The case, as a collection of specific knowledge, is very different from the prototype as a form of generalised knowledge (Rosenman, 1992). We have found that in case-based design it is more important to have a few, high quality cases rather than many, lower quality cases. We measure the task-specific quality of a case based on two criteria: the performance of a case and the closeness or applicability of the selected case to the design problem at hand. The first criterion is self-evident, only cases which perform well in a multi-criteria evaluation should be included in a case base. The second criterion is problem specific: although a given case in the case base might have excellent performance ratings, it may not be a valid starting point for the solution of the current problem. This measure of case quality is a common concern for all CBD applications. Another general problem in CBD is the trade-off that occurs between case complexity and process complexity. On one hand, even complex design tasks require a minimum of changes in the form of case adaptation if a suitable case can be found; on the other, almost any design problem can be solved by combining small fragments of cases, but this involves a complex process of combination and adaptation.

Using Cases for Design

Design case bases contain successful, high quality architectural examples. In order to be included in a case base, buildings need to fulfil a number of requirements. Some of these requirements are:

- Architectural quality. The building must earn the respect of the professional community as well as the acceptance of its users. Although this is a very subjective criterion, it is important.
- Timelessness. The design should be a product of its time, but not be merely fashionable. Some of the Alexander's criteria may be used here (Alexander, 1979).
- Environmental responsiveness. The building must offer an appropriate answer to the environmental conditions of the site.
- Contextual responsiveness. The building must make a clear statement regarding its position within a context. Extreme examples adapt to the context or propose a bold new beginning.
- Functional quality. The building must fulfil all functional requirements and in addition offer possibilities for future adaptation.
- Structural stability. The design must offer structural safety and also comply with special local ordinances regarding conditions, such as earthquakes or tornadoes.

If all of these criteria are strictly applied, very few buildings will be included in a case base. The fact is, that even the best buildings contain a number of trade-offs which balance positive and negative properties, overall quality of the case is positive. The case base for construction will contain a collection of successful construction techniques connected to certain construction types. A facilities management (FM) case base contains histories of buildings in which FM is crucial issue, such as for institutional and hospital buildings.

EXAMPLES

All reasoning which involves physical or functional objects requires efficient adaptation techniques. Adaptation is the key to efficient change.

Adaptation helps to avoid excessive re-generation and the subsequent loss of important qualities of an object. As design, and especially architecture, have been dominated by geometric concerns for centuries, sophisticated methods of geometric and dimensional adaptation have developed. Geometric constraints and rules are sufficient for covering adaptation of proportional and size properties. They fail when the distance between the selected case and the design goal is too large. For such cases, topological adaptation must be available. Topological adaptation allows changes to be made at a more abstract level than that which is required for solving dimensional problems. In practice, this involves switching, adding or subtracting entire spaces or functions. Once a topological adaptation has been successfully completed, the changes must be studied again at the dimensional level.

Case-Based Housing Design Adaptation

Since 1989, we have developed several case-based procedures for engineering and architectural design. After testing them with practical applications, we believe that we have found a valid approach for case adaptation which can be applied in design and related fields. The first design tasks for our CBD system were to fit an existing design into a new site and to adapt a floor plan to a new context. The entire process consists of the following steps:

- Case selection for the case base. The first buildings we selected for the case base - according to the criteria listed above - were the Felder and Maggi houses by Campi and Pessina and a low cost Vienna housing project by Schweighofer. For test reasons, we often include more simple cases, such as a residence in Massachusetts by Hugh Stubbins.
- Case and site description. In this necessary first step, the case and the environment are described carefully. For this purpose, we developed the pre-processor Mod4 to model the case by inserting walls, doors, windows, rooms and contextual elements such as parcel lines, roads, neighbouring buildings, lakes and parks. Mod4 converts the input into AutoCAD format.
- Deriving knowledge from the case. The case input process produces additional knowledge which is not obvious in the graphical representation but which is present in the model. The system automatically identifies topological relationships among spaces by analysing the model and its labels. Labels of wall, door, and window positions are used to deduce required spatial adjacencies. Thus we build different abstractions of a building into a case description which will be later used for reasoning.
- Input of new design requirements. If a case must be adapted to programmatic changes as well as to a new context, the following steps are necessary: Specify the maximum and minimum sizes, areas and proportions of spaces (dimensional adaptation). If new spaces need to be added, specify

any required adjacencies to other spaces. Indicate the search depth by defining how many rooms are to be re-allocated (topological changes).

- Case insertion. The existing case is inserted into a new environment, usually a new site. The insertion into a new site takes place according to a knowledge base that defines the positive or negative weights of direct links between spaces and contextual elements. For example, a living room facing south, a park or a yard carry positive weighting factors, whereas a bedroom facing a road or other buildings carries a negative weighting factor. The case is checked against its original site first. If a positive link is found, the weight is doubled, if a negative link is found, the negative weight is considered not that significant and thus is divided by two. The placement is completed after the sum of all links between the new site and the inserted case has been maximised by rotating and mirroring the case.
- Dimensional adaptation. If the insertion satisfies the internal and external requirements, but not the dimensions of the new site, the first step is to dimensionally adapt the existing case. For this purpose, an evaluation detects discrepancies between the inserted case and the new site and converts these conflicts into parameters. The number of parameters can be large in the beginning. The process of dimensionality reduction (Hua, 1992) helps to reduce the parameters to a manageable number and also allows the integration of other design considerations, such as those related to the structural system. If the dimensional adaptation is unsuccessful or produces visually unsatisfactory results, the topological adaptation process can be triggered.
- Topological adaptation. We have tested different approaches for solving this problem. The first attempt was the use of case-specific rules which fired whenever a conflict between the existing situation and the target situation was detected. As could be expected, the number of rules that were needed grew rapidly and degraded the overall performance of the system. A second attempt was the use of shape grammars and the application of neural networks to recognise changes in the grammar. The third and most promising approach employs the wall representation algorithms developed by Flemming. A derivation of Flemming's wall representation method is applied to search for topological variations (Flemming, 1988 and Coyne, 1991). Linear programming is used to check the dimensional feasibility of each configuration. The selected case is used as the starting configuration. The process first removes the number of spaces that are to be reallocated. This already creates some alternatives. Based on these alternatives, spaces are re-inserted one by one. If a configuration is found to be unacceptable, the entire search branch is discarded. If more than one positive solution is found, the user is prompted to graphically select a solution from the proposed set for insertion into the new site.

- Evaluation and visual inspection. After a successful dimensional or topological adaptation, the geometric results are displayed and evaluated. A visual evaluation of the geometric model is the last step. If the visual evaluation proves unsatisfactory, the process of topological adaptation is restarted.

Using this approach to adaptation, we are able to avoid some of the problems inherent in specialised layout programs, such as the limitation to about ten configurable objects in a fixed layout. Rather than starting with a set of unrelated objects, we use the existing case as a starting point for the re-configuration and can thus handle a larger number of objects. The first justification for this is, that the case was selected because of its good layout and that the adaptation should maintain as much of the original layout as possible. The second reason is computational efficiency. For example, the time required to solve a 30 space layout problem resulting from a topological change was only about five minutes of CPU time on a Sparc Station 10.

Case-Based Office Building Modification and Design Repair

The second example describes the proposed modification and design repair of a research office building at the Swiss Federal Institute in Lausanne. Case modification, rather than case election or case adaptation is the main concern. The starting situation was the completed office building, based on a regular grid of concrete columns. Late changes in the design phase positioned a large lecture hall at the end of the building on the second floor. The change was not compatible with the structural system. A structural column was located in the centre of the lecture hall. As this is a rather undesirable situation, we attempted to resolve the conflict with the case-based reasoning system that was developed for adaptation purposes. The existing building, consisting of three floors and a basement was input as a case. The design goal was to accommodate the same space program, including a similar sized lecture hall, but undisturbed by a column. In other words, the existing design case had to be repaired as follows:

- Definition of all known constraints. The size of the lecture hall was defined as a constraint and the column was labelled as an unwanted interruption of the space. Size and proportion of the other rooms, as well as location and spacing of the structural columns were defined as parameters with constraints.
- Layout modification to satisfy the new constraints. The system attempted to re-size or re-arrange all spaces on the same floor without violating any of the constraints. Because of the size and proportional requirements of the lecture hall, this attempt failed.
- Topological adaptation. After the dimensional adaptation failed because of the size of the lecture hall, topological change, *ie*, re-arrangement of the

spaces was attempted. We again used an adaptation of Flemming's wall representation algorithm (Flemming, 1988). This attempt also failed.

- Integration of spatial and structural concerns. It became clear at this point that the structural grid had to be changed. Made possible by a common representation in the form of constraints, dimensionality reduction was applicable to the process of adapting the structural definition as well as the architectural definition of the case. By changing the number of bays and by proposing a new structural grid, all constraints on the floor in question could be satisfied.
- Verification of the structural change in the other floors of the building. In all but the ground floor, the new positioning of columns did not cause any spatial conflicts. The layout problem on the ground floor could not be solved completely by re-arranging the spaces (topological adaptation), but the result was considered acceptable (see Figure 2).

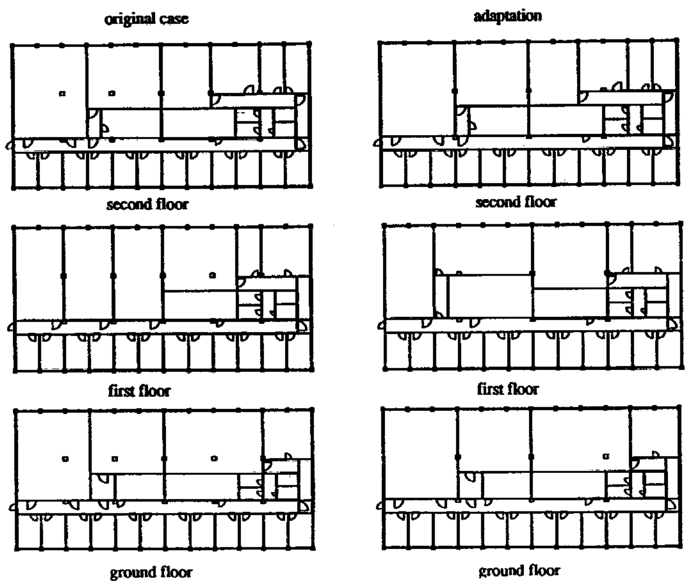

Figure 2. Modification and repair of a case. Left: Case before adaptation. Right: Case after architectural and structural adaptation. Note the automatic changes to the space layout and to the structural grid.

- Graphical selection of the final solution. At the end of the process, the designer could graphically select alternatives from the screen. A selection was necessary because the spatial layout algorithm produced several alternatives which are close to the original case. The best way to judge them is by visual inspection.

The use of constraints and the integration of structural and architectural concerns characterise this example. If structural and architectural constraints can be solved by dimensional adaptation, endless feedback cycles are avoided. This is not true for topological adaptation, where the integration of architectural and other concerns has not been achieved yet. The entire adaptation process took about 10 minutes of CPU time on a Sparc Station 10. A common user interface allows the control of the entire program from within AutoCAD.

Case-Based Reasoning in a Large Integrated Design Project

In 1993, a new centre for integrated planning will be founded at ETH Zürich. The leadership consists of faculty from architecture and civil engineering who specialise in planning (IBETH), building realisation (HBT) and CAAD. The purpose of the centre is twofold: to demonstrate the applicability of computer-based planning and facility management programs for large projects and the development of a new management approach, that accompanies a building from the early programming phase, through its life time, to its final demolition. The first goal will be realised by accompanying one of the largest construction projects in Switzerland, an office and retail complex from its conception onward. The second goal requires the formation of a potent research group to develop new representations for the processes and products of integrated design and construction. For practical implementation of the integrated planning, design and construction project, IBDE could serve as a case. The case in this situation is not a building or a physical object, but rather an organised collection of interrelated processes. IBDE could be adapted to learn from the past and to correct problems encountered during the course of using the system. The case-based reasoning process must adapt the following system components (compare also Figures 1 and 3):

- Processes. This is an adaptation, in that the knowledge-based processes in the IBDE project are replaced by human interaction (see Figures 1 and 3). All project partners begin their work at the same time, therefore they must be able to share the growing amount of project information in the form of drawings, models, photographs, memos, and specifications.

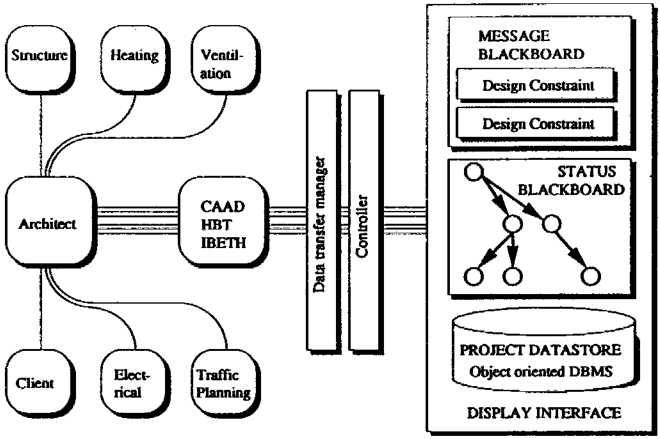

Figure 3. Case-based reasoning in an integrated design and construction process. Adaptation and repair of the IBDE process (see also Figure 1). Note the replacement of the seven expert systems by human experts.

- Data manager. The data manager is a program that provides the necessary interfaces between applications (processes), the project database and the controller. Information exchange and storage are controlled by an intelligent file management system.
- Controller. This program decides on the content of the information to be displayed on the message and status blackboard.
- Message blackboard. The message blackboard displays constraints and changes made by any of the project partners.
- Status blackboard. The status blackboard displays the status of each project partner's work as well as the status of the overall project.
- Project data store. An object oriented database stores all project related information. It changes in content and purpose during the course of the project.
- Common display interface. All project partners work on UNIX workstations. The interface will be a motif application.
- Communication. The physical medium is an ISDN network that connects all partners. There is an additional Ethernet connection between IBETH and CAAD.

Following these adaptations, work on the office design project will begin in the summer of 1993 and is scheduled to accompany the entire design and construction process of the selected office building. For the second and more research oriented phase of the project, we can draw on human experience in the form of professional interaction case studies. Although professionals can describe rather well which information needs to be present and how it must be manipulated, no existing computer tool supports this process. Instead, existing computer programs disrupt and split a project into several insular solutions, the consolidation of which regularly causes loss of crucial information. Human problem solving techniques and processes, extracted from the project partners during various sessions, form the second case, which will be adapted and implemented in the next phase of the project. The advantage of this approach is that information flow and the data model can be implemented in software. The enormous amount of information and secondary data - which often irritates and paralyses human interaction - can be handled by the file management system and the object oriented database. The main advantages for the client will be a reduction of redundant information, minimisation of mistakes and better communication between project partners.

CONCLUSIONS

By implementing and testing examples, we have discovered interesting properties of cases and case-based reasoning. First, case adaptation is an attractive method for design development which, when based on selected examples in a case base, offers complete solutions to routine design problems. Second, case repair is an excellent method, using adaptation, for solving serious design flaws in otherwise acceptable buildings. Third, a computer-based, integrated design process can serve as a case and provide the model for a real integrated design and construction project. As demonstrated in this project, case adaptation can be successfully applied to an early example like the IBDE project.

There are major differences between these two recent, but complementary approaches towards design: prototypes and cases. Prototypes make use of generalised design knowledge, whereas cases draw from specific examples. The two methods contribute to different stages of the design process or to the management of buildings. Methods, which have proven successful over time can be formalised and stored in the form of prototypical solutions. New and special projects are supported by similar experiences or cases from the past. Once a case has been adapted several times to solve similar problems, causal relations between requirements and final shape may emerge. At this point, these experiences can be structured and eventually converted into prototype knowledge. The major finding of the project is that case-based reasoning can not only be applied to the design of individual buildings, but also to integrated design and construction processes. This new method can improve both the process and the results.

ACKNOWLEDGEMENTS

Research in case-based reasoning is supported by the Swiss Nationalfonds under NFP23 "Artificial Intelligence and Robotics". I want to thank Boi Faltings, Ian Smith, Kefeng Hua, Simon Bailey, and Shen-Guan Shih for their contributions.

References

Alexander, C (1979), *The timeless way of building.* Oxford University Press, New York, NY.

Fenves, S J, M L Maher, U Flemming, C Hendrickson and G Schmitt (1990), Integrated Software Environment for Building Design and Construction. *CAD*, Vol 22, No 1, pp 27-36.

Coyne, R (1991), *ABLOOS: An Evolving Hierarchical Design Framework*, Ph.D. thesis, EDRC 02-15-91, Carnegie Mellon University, Pittsburgh, PA.

Flemming, U, R F Coyne, T J Glavin and M D Rychener (1988), *A Generative Expert System for the Design of Building Layouts,* Technical Report EDRC-48-08-88, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

Hua, K, I Smith, B Faltings, S Shih and G Schmitt (1992), Adaptation of Spatial Design Cases. In *Artificial Intelligence in Design*, (ed J Gero). Kluwer Academic Publishers, The Netherlands, pp 559-575.

Rosenman, M A, J S Gero and R E Oxman (1992), What's in a Case: The Use of Case Bases, Knowledge Bases, and Databases in Design. In *CAAD futures '91* (ed G Schmitt). Vieweg Verlag, Wiesbaden, Germany, pp 285-300.

Schmitt, G and C C Cheng (1991), Classes of Design - Classes of Methods - Classes of Tools. *Design Studies*, Butterworth-Heinemann, The Netherlands, March, pp 246-252.

Shih, S G (1992), Case-Based Representation and Adaptation in Design. In *CAAD futures '91* (ed G Schmitt). Vieweg, Wiesbaden, Germany, pp 301-312.

---

c. 1993, Management of Information Technology for Construction, K. Mathur et al (Eds), World Scientific Publishing Co., Singapore.