ELSEVIER

PII: S0965-9978(97)00064-1

Advances in Engineering Software Vol. 29, No. 10, pp. 801–808, 1998

0965-9978/98/$19.00+0.00

Collaborative case-based estimating and design

Srinath Perera[a] **& Ian Watson**[b,*]

[a]*Department of Building Economics, University of Moratuwa, Moratuwa, Sri Lanka*
[b]*BUHU, University of Salford, Bridgewater Building, Salford M5 4WT, UK*

Information Technology (IT) solutions to problems in construction design need to consider the perspectives of all the participants in the process; only then can IT provide a platform for integration. The research described examines issues involved in the integration of construction disciplines by using Case-Based Reasoning (CBR). It describes a hierarchical case memory structure and a context-based indexing method for retrieval and reuse of previous designs and their costs. Estimating and design cases selected for reuse are adapted with the use of sub-cases and domain specific adaptation rules. A prototype system, NIRMANI, was successfully implemented to support collaborative design.

1 INTRODUCTION

Design in construction involves many collaborators (e.g. architects, quantity surveyors, structural engineers, service engineers, etc.). The product, the final design of a building, is the result of the collaborative effort of these individuals. These people have different perspectives of the building.[1] Take for example the office area in a warehouse building. To an architect, this is a functional space with aesthetic considerations, whilst for a quantity surveyor it is an element of cost that is dependent on a specification (i.e. quality and quantity). To the structural engineer, the same floor area is a structural element supported by beams, columns and foundation. Therefore, IT systems that provide solutions to design problems in construction should support such perspectives. Moreover, a design's complexity makes it difficult for the client to understand the design[2] and makes cost control difficult.[3] An additional problem is that comparative design evaluation is traditionally expensive; however, not comparing alternatives may result in sub-optimal designs and premature bias to designs.[4] Finally, it is recognised that construction problems are often caused by inadequate design.[5]

IT solutions developed in the past have often created *islands of automation*.[6,2]. For example, ELSIE,[6] and EMMY[7] automate the cost estimating process, but do not provide design information. Conversely, rule-based and case-based systems such as HI-RISE,[8] ARCHIE,[9] CADRE[10] and ADA,[11] support architectural design or structural design in isolation, without considering the cost implications of the design. We believe that IT solutions to design problems must consider all the perspectives of the design problem.

The research described in this paper supports collaborative design through a case-based estimating and design system. This paper first briefly describes the CBR process and its application to estimating and design. It then proposes a conceptual model for collaborative design. Finally, it describes the organisational structure of a case memory for case-based estimating and design and describes a retrieval method that supports collaborative design.

2 CBR AND ITS APPLICATIONS TO ESTIMATING AND DESIGN

2.1 An overview of the CBR process

CBR provides both a methodology for building systems and a cognitive model of how people solve problems.[12,13] CBR was defined as follows:[14]

> "Case-Based Reasoning is the process of solving new problems by adapting solutions that were used to solve old problems".

At the highest level of generality CBR may be described as a cycle with four processes[15] as shown in Fig. 1.[6]

A CBR system retrieves a suitable case from the case library by matching indexes established for the new case (or problem case). The information and knowledge in the retrieved case is then reused to provide an initial solution to the problem. When it does not fully satisfy the problem specification the retrieved case is revised (or adapted) using domain rules, heuristics and/or human intervention.

*Corresponding author.

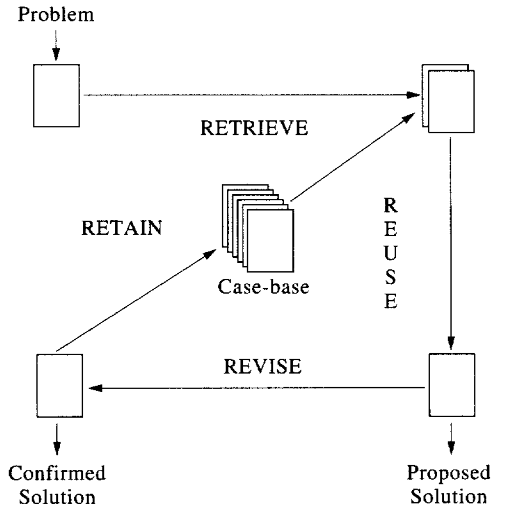

Fig. 1. The CBR cycle.

The revised case is then evaluated and criticised to assess the suitability of the solution[17] and if suitable is retained.

2.2 Case-Based Design and Case-Based Estimating

2.2.1 Case-Based Design (CBD)
Design is an ill-structured task[18] where design experience and heuristics play an important role.[19] Kolodner[13] describes design as a process of constraint satisfaction in creating an artefact that performs a certain function or fulfils a need. In most design problems these constraints may be under-specified. Thus, there are usually many solutions to any given design problem. Sometimes none of the constraints are able to be satisfied which means a compromise is required. Moreover, constraints cannot usually be considered in isolation.

During the design process architects use parts of previous architectural designs.[19–21] In case-based design, the designer is offered previous solutions to a similar problem,[21,22] indicating how a previous combination of constraints were handled.[13] This process of using previous designs in the creation of new designs is called Case-Based Design (CBD). Hence, CBD can be defined as:

> "The process of creating a new design solution by combining and/or adapting previous design solution(s)".

Maher *et al.*[23] describes CBD as a hybrid approach because it uses specific design cases in conjunction with generalised or compiled knowledge. This often involves knowledge based adaptation of the design from an existing, original design context to a new design context.

2.2.2 Case-Based Estimating (CBE)
Cost estimating is the prediction of the cost of an artefact, process or project either by using experience and/or a methodology. In the construction industry, there are several methods for estimating; for example, the unit method, storey enclosure method, or elemental estimating method.[25,26] These methods are dependant on the experience of the estimator and cost data derived from previous construction projects.

Cost planning primarily uses the elemental estimating method for cost estimation.[24] This method uses similar cost analysis of projects (Elemental Cost Analysis — ECA) as the basis for estimation. The elemental rates are adjusted or adapted for quantity, quality and price (tender, market conditions, location etc.).[25,26] Thus, this uses a previous case (previous ECA) as the basis of the new estimate. Elemental estimating is, therefore, a classic example where CBR is applied by quantity surveyors to solve an estimating problem. Hence, Case-Based Estimating (CBE) can be defined as:

> "The process of estimating the cost of a new artefact, process or project using previous estimates or cost analyses".

Surprisingly, there are very few examples where CBR has been used for estimation in any domain. FACE — Finding Analogies for Cost Estimation[27] is a system for estimating the cost of software development projects. It uses a case-base of previous estimates and a general global database for adaptation. Retrieved cases are adapted using an analogy based algorithm to provide the new estimate. The system is implemented using CBR-Express and ART IM. Estor[28,29] is another system used for software effort estimation, while CBR systems that help engineers to prepare bids in manufacturing were reported.[13,30–32]

3 NIRMANI: A CASE-BASED ESTIMATING AND DESIGN SYSTEM

3.1 The conceptual model

NIRMANI (schematically described in Fig. 2) generates a schematic design for light industrial warehouse buildings by retrieving previous design solutions that match a client's problem specification.[16,33] The retrieved design will be adapted if required for architectural, structural and services requirements. Depending on the extent of adaptation of the design, costs for the chosen design will be adapted to provide an elemental cost plan for the building. The cost plan acts as a budgetary guide for further design development. The entire case-based estimation and design process is interactive giving the design team authority to guide the design.

The stages of the case-based estimating and design process are briefly explained later. NIRMANI starts a session using an initial client brief (e.g. building function, number of occupants, budget, shape, etc.) and retrieves a set of cases ranked according to similarity. Designers and the client can visualise design cases as 2D and 3D CAD images, video images, scanned photographs, and text. The design team can evaluate and confirm the selection of the most preferred design case and the system maps all its design information to the new case creating a detailed design brief (i.e. a building specification). The system compares the developed brief and the original requirements and establishes design constraints for each decomposed design perspective.

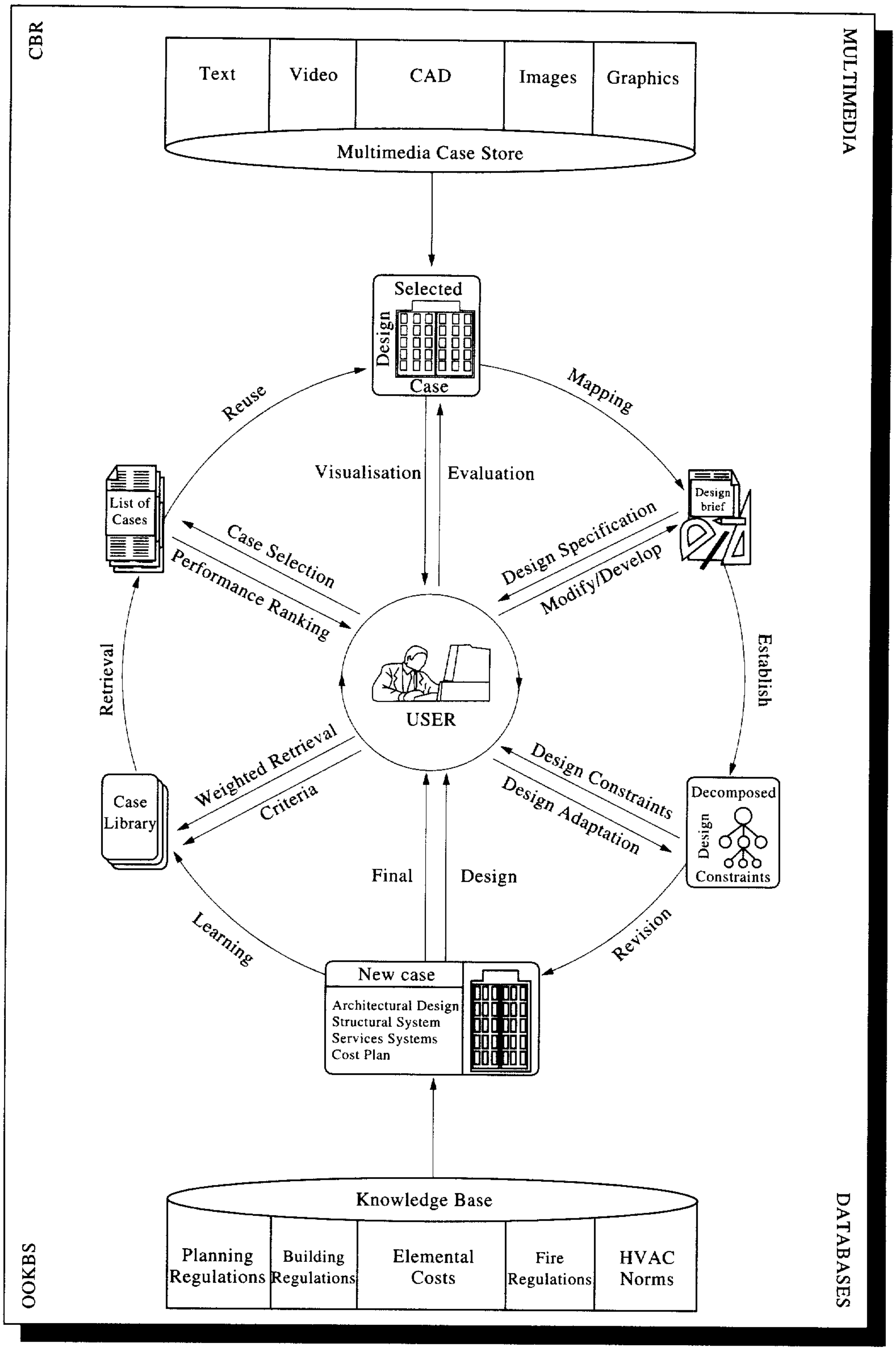

Fig. 2. NIRMANI: schematic overview.

NIRMANI uses an interactive adaptation process to satisfy design constraints and to revise the design using knowledge in a knowledge-base. Case-base support is provided where appropriate. Once constraints are satisfied (to a level acceptable to the designers) and the designers accept the new design, it is stored in the case library as a new case, thus completing the CBR cycle and learning from the new experience; i.e. the system conforms to a dynamic memory model.[34,35]

NIRMANI maintains the many perspectives of the design by using agents to represent each perspective. As pointed out by Oxmanm,[39] a multi-perspective representation is

Table 1. Comparison of NIRMANI with other CBD systems

Feature	Nirmani	Other Systems
Domain	NIRMANI operates on industrial buildings (warehouses), but could be used for other types of buildings. It deals witha comprehensive range of design aspects including Architectural design, Structural girding, Services System Selection and Cost Planning	ARCHIE,[13] ARCHIEII,[13] MEMORABILIA,[36] CADRE[10] and SEED[44] operate in the domain of Architectural Design. CADSYN,[22] CASECAD,[43] STRUPLE deal with structural analysis and design. Systems are developed for office buildings, court house, museums, apartment blocks etc. FABEL[20] supports the detailed design of industrial buildings.
Case Storage	OO feature-based storage for indexing in hierarchical case-bases, CAD file storage, graphic file store (video and scanned images), database of costs, building regulations etc. The system uses a dynamic case-base derived from existing databases.	Comparable to CASECAD but distinguished clearly from other systems such as SEED, CADRE ec. FABEL stores cases as CAD images and uses an object-oriented representation and bitmaps where cases are design segments or elements.
Case Indexing	An OO structuring of cases is used with a flexible multi-level indexing schema. It has four main perspectives of Architectural, Structural, Services and Estimating. Case indexing is user directed and guided by the context-based indexing methods.	CASECAD and SEED use an OO indexing feature hidden from the user. Comparisons could be drawn to FABEL's distance-based approach.
Case Retrieval	Provides retrieval of a set of cases ranked accoding to the degree of match to a set of weighted retrieval criteria (the design brief). Allows retrieval based on a set of cluesinput by the user. Therefore, high flexibility of retrieval. Uses a similarity metric to the analysis the degree of match.	Comparable with SEED, CASECAD and FABEL to some extent, but differentiated from CADRE which argues for a user selection of case. FABEL provides multiple retrieval methods.
Case Prevention	Multimedia case presentation enabling visualisation and understanding of the design. Cases are linked to CAD drawings, video clips, photographs, related documents with WWWURL references.	CASECAD uses multimedia case presentation but only in the form of attribute-value pairs, 2D and 3D CAD images. FABEL uses CAD for case presentation.
Case Adaptation	Uses four main adaptation processors, i.e. Architectural Adaptation, Structural Adaptation, Services Adaptation, Cost Adaptation. Rule based guidance is provided for case adaptation using a constraint satisfaction approach.	Most systems avoid adaptation and let the user adapt solutions. SEED uses an interactive adaptation (Architectural Design) while CADRE uses dimensional topological adaptation.
Implementation	Integration of CBR within an OO development environment that facilities the use of CAD and databases. Runs in the MS Windows environment (PC based).	Most systems have been implemented in an UNIX X-windows environment. Some use CBR development shells while others use common LISP or C.
Learning	NIRMANI learns by storing new designs generated by the system. It grows with the experience of the users forming a repository of co-operative experience of the design organisation. A case maintenance module allows the user to add, delete and edit cases. It keeps an account of the frequency of the usage of cases in order to prune the case-base.	Natural learning has been limited to a great extent in most CBD systems. In FABEL, cases are generated from a more generic CAD based industrialised building system (ARMILLA and MIDI).

useful for representing all possible similarities and dissimilarities that occur among different agent's perspectives. The concept of agents[36,37] enables the system to support collaborative design. Bhat[40] states that there is no commonly accepted definition of agents; whilst Woolridge[41] proposes an agent in terms of an object or system that incorporates beliefs, actions and communications. In NIRMANI an agent is defined as a set of objects representing a perspective of a generic design function.

Table 1 provides a brief comparison of NIRMANI with other notable CBD systems. A critical and more detailed comparison of NIRMANI to other CBD systems can be found in Perera et al.[38] and Watson and Perera.[33,45]

4 CASE MEMORY

A building in NIRMANI is a meta-case, consisting of a hierarchy of cases and sub-cases. At the top of the hierarchy is the Project Context case. The second level contains Architectural Context and Estimating Context cases representing the perspectives (or views) or architects and quantity surveyors. A third level decomposes the design into functional spaces and aesthetic requirements hierarchies and the estimating problem into an industry standard elemental classification hierarchy.[33,42]

Each node in the hierarchy is stored in a separate case-base. The cases are stored as records in a relational database external to the system, since this has the benefit of allowing a design organisation to obtain their case data from their existing databases. An object hierarchy within the system maps to the tables in the database and cases are presented (when required) as instances. Cases contain attribute–value pairs as case features as in Table 2.

A Project Context case describes the environment within which the project was carried out (features such as the type

Table 2. A selection of attributes from the project context case definition

	Attribute	Value(s)	Data Type
1.	Case No.	Value per project	cat-nir: capitols
2.	Number key	Unique integer value per case per case-base	cat: integer-or-nil
3.	Source cases	List of cases	default
4.	Name of Project	Text	default
5.	Site Address	Text	default
6.	Site Post Code	Text	default
7.	Client	Text	default
8.	Client Address	Text	default
9.	Client Post Code	Text	default
10.	Type of warehouse	Storage Distribution Retail	catnl:wh-type
11.	Type of occupier	Owner occupier Tenant occupier Developer	catnl:occupier
...	...	...	...
66.	Structural Engineer	Text	default
67.	Services Engineer	Text	default
68.	Other Consultants	Text	default
69.	Contractor	Text	default
70.	Contractor Address	Text	default

of building, its intended function, gross internal floor area, the site conditions, and other features common to the project context, shown schematically in Fig. 3). The second level cases (architectural and estimating) describe the context of the sub-problems. The system prefers to retrieve sub-cases with similar contexts (i.e. with similar parents in the hierarchy) in order to reduce problems of case adaptation and solution re-composition caused by contextual dissimilarity.

The interface of NIRMANI allows cases to be viewed as attribute–value pairs along with CAD drawings and other multimedia elements. It supports case comparison using a tabulated form (similar to a spreadsheet).

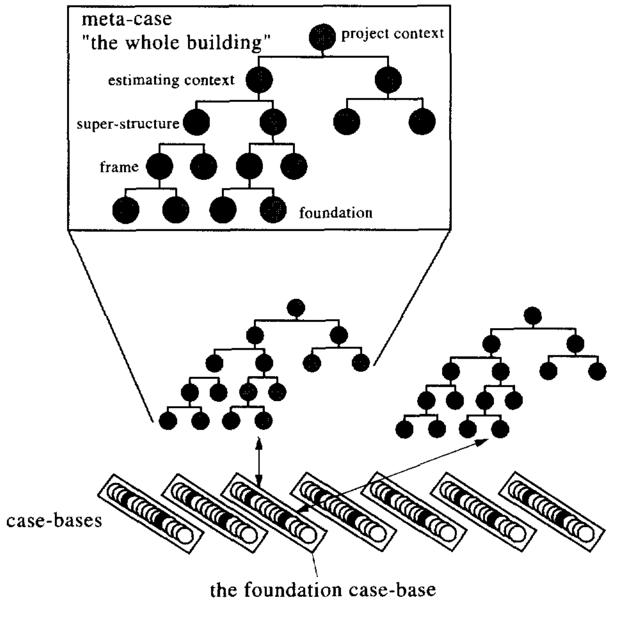

Fig. 3. Schematic of the hierarchical case representation.

5 CASE RETRIEVAL

NIRMANI provides a variety of retrieval methods, of which only two are described in this paper. Full details of these retrieval methods can be found in Perera and Watson.[38] ART*Enterprise uses a nearest neighbour algorithm with weighted features. Its programming environment gives the developer considerable control of the algorithm making it a good environment to explore different retrieval strategies. The two strategies described in this paper are described later.

5.1 Default retrieval

This is essentially standard nearest neighbour retrieval. The user is allowed to select which features are indexed. These will usually be the majority of the features in the Project Context case (except the construction cost) plus some other significant features from other aspects of the building. For example, the user may want a glazed curtain wall on the front elevation of the building but have no definite views or wishes as to the roofing type. The user may set weights on features reflecting their relative importance to them. In default retrieval an index is prepared dynamically at run-time for those case features entered by the user. Feature comparison is carried out as in normal nearest neighbour retrieval. A normalised match score for each entire meta-case is calculated and the highest ranking cases are then presented to the user. Only an entire meta-case can then be selected for adaptation.

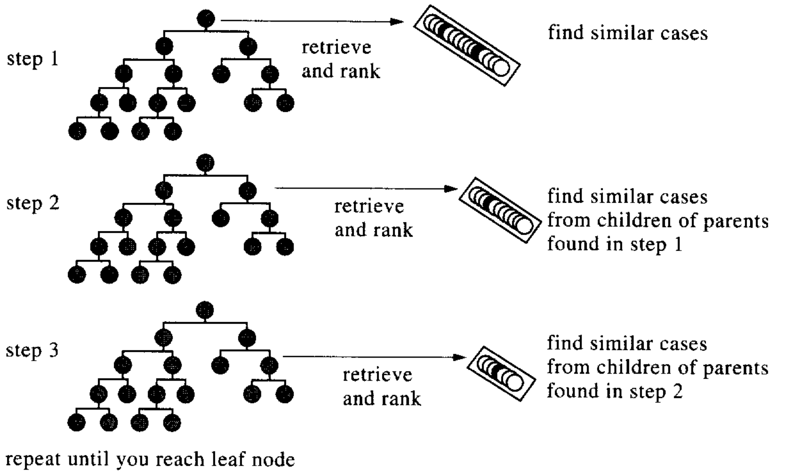

Fig. 4. Context guided retrieval.

5.2 Context guided retrieval

Context guided retrieval proceeds in series of recursive steps down the hierarchy of the case representation. In the first step, the features of the Projet Context case (at the top of the hierarchy) are used to retrieve similar Project Context cases from the Project Context case-base. This is done using ART*E's standard nearest neighbour algorithm. In the second step, retrieval of cases from the estimating or architectural case-bases (the next nodes down the hierarchy) is restricted to those cases that are the children of the cases found similar in the first retrieval step. That is, retrievals limited to those sub-cases that share similar project contexts (i.e. similar parents). This process is repeated all the way down the hierarchy. Retrieval at each level is restricted to those cases in a case-base that have similar parents as in Fig. 4.

This process reduces the search space by enforcing contextual similarity. However, if a close enough match cannot be found at any level (this is more likely to occur at leaf nodes since the number of cases included in the search may reduce at each level) then the contextual guiding can be relaxed. This relaxation is achieved by back-tracking up the hierarchy and reducing the threshold at which similarity is judged acceptable for the parent case. This will increase the number of cases allowed into the children's retrieval process. This relaxation can proceed all the way to zero, if necessary, allowing retrieval from all cases in a child's case-base, thus removing the context guidance completely.

6 CASE ADAPTATION

Retrieved cases are ranked and presented to the user. Users are allowed to select cases and case features for adaptation. Note that using the default retrieval method only sub-cases from one meta-case can be used for adaptation. Whereas, for context guided retrieval, sub-cases from different meta-cases (i.e. different buildings) with a similar context can be used. Moreover, using context guided retrieval, adaptation can occur at the elemental unit level of detail, whereas for the default retrieval, adaptation occurs at the level of the project context case (i.e. only the total estimated construction cost is adapted). A modification knowledge-base, containing a set of rules, functions and procedures provides the adaptation. In general, adaptation is in the form of parameter adjustment through extrapolation. For example, if a retrieved case has the feature floor finishes at a cost of £12,000 with a GIFA of 2000 m^2, then the adaptation function will calculate a rate for floor finishes of £6 per m^2. This rate can then be applied to a new case with a different GIFA but a similar specification for floor finishes.

7 CONCLUSIONS

It is evident that architects and quantity surveyors use previous solutions to create new design solutions and their cost estimates. Thus the paradigm of CBR can be applied to both estimating and design through the development of a case-based estimating and design system. NIRMANI integrates estimation and design in a single system. The use of multimedia enhances the usability of the system improving the understanding of complex designs by the client and the design team. It also brings the design team to a single design platform with the client thereby improving communication.

The hierarchical case memory organisation structure provides an efficient and effective case memory organisation for this complex design domain. It is efficient because retrieval does not need to search the entire case-base to find a solution. It is effective because it considers multi-perspectives of the same design space and provides adaptable sub-cases for design and cost adaptation. The system achieves data integration by directly accessing data for cases from existing legacy databases.

NIRMANI provides an elemental estimate of a building along with the design specification and an exemplar design (the source design as a CAD layout design). Design alternatives can therefore be developed cost effectively. The system grows with the experience of the organisation providing libraries of corporate experience for the design organisation. The use of previous design and construction experience helps to achieve best practice in design by avoiding repetition of previous mistakes. An interesting facet of the system is that it allows the development of the design brief from a basic set of requirements to a detailed and comprehensive brief. Moreover, the entire CBD process is interactive, giving the designers sufficient authority and control to guide the design.

NIRMANI's estimating performance was tested against a commerically available estimating package. A statistical evaluation of the performance of the system reveals a coefficient of variation of $\pm$ 2% for the cost estimation of warehouse buildings.[43]. Its design performance has also been qualitatively evaluated by several experts. All these initial tests and evaluations have produced favourable results. Further developments will involve the development of its verification and adaptation knowledge bases and the improvement of its user interface.

REFERENCES

1. Aouad, C. F., Ford, S., Kirkham, J. A., Brandon, P. S., Brown, F. E., Child, T., Cooper, G. S., Oxman, R. E. and Young, B., An approach to knowledge elicitation in the construction industry. *Construction Management and Economics*, 1994, **12(3)**, 384–397.
2. Kartam, N. A., *A Knowledge-Intensive Database System for Making Effective Use of Construction Lessons Learned.* American Society of Civil Engineers, 1994.
3. Brandon, P. S., Cost vs quality: A zero sum game? In Powell, J. A., *et al.*, ed., *Designing for building utilisation*. E and FN SPON, London, pp. 108–123, 1984.
4. Harper, D. R., *Building: The process and the product*. Construction Press, London, 1978.
5. Alshawi, M. and Underwood, J., Information analysis approach for integrating design and construction. Dept. of Surveying, University of Salford, 1994.
6. Brandon, P. S., Basden, A., Hamilton, I. and Stockley, J., *Expert Systems. The Strategic Planning of Construction Projects.* RICS, 1988.
7. Watson, I. D., Brandon, P. S. and Basden, A., EMMY: an expert system for predicting maintenance for housing assocations. In A. Bezelga and P. S. Brandon, ed., *Management Quality and Economics in Building*. E and FN SPON, 1991, pp. 1603–1611.
8. Maher, M. L., An expert system for preliminary structural design. In M. D. Rychener, ed., *Expert System for Engineering Design*, 1988, pp. 37–52.
9. Goel, A., Kolodner, J. L., Pearce, M., Billington, R. and Zimring, C., Towards a case-base tool for aiding conceptual design problem solving. *Proceedings: Workshop on Case-Based Reasoning (DARPA)*, ed. E. R. Bareiss. Morgan Kaufman, Washington, D.C. San Mateo, CA, 1991.
10. Hua, K. and Faltings, B., Exploring case-based building design-Cadre. In *AI EDAM*, Vol. 7(2). Academic Press Limited, 1993, pp. 135–143.
11. Hunt, J. and Miles, R., Towards an intelligent architectural design aid expert systems. *Expert Systems*, 1995, **12**(3), 209–218.
12. Slade, S., Case-based reasoning: A research paradigm. *AI Magazine*, 1991, 42–55.
13. Kolodner, J. L., *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc., 1993.
14. Riesbeck, C. K. and Schnak, R. S., *Inside Case-Based Reasoning*. Northvale, NJ, 1989.
15. Aamodt, A. and Plaza, E., Case-based reasoning: Foundational issues, methodological variations and system approaches. *The European Journal on Artificial Intelligence, AICOM,*, 1994, **7**(1), 39–59.
16. Perera, R. S., A case-based design approach for the integration of design and estimating. MSc Dissertation, Department of Surveying, University of Salford, Salford, UK, 1994.
17. Watson, I. D. and Marir, F., Case-based reasoning: A review. *The Knowledge Engineering Review*, 1994, **9**(4).
18. Simon, H. A., The structure of the ill structured problems. *Artificial Intelligence*, 1973, **4**, 181–201.
19. Akin, O., Expertise of the architect. In *Expert Systems for Engineering Design*, ed. M. Rychener. Academic Press, London, 1988, pp. 173–196.
20. Bartsch-Sporl, B., Towards the integration of case-based, schema-based and model-based reasoning for supporting complex design tasks. *Proceedings of the ICCBR 95*, Portugal, 1995.
21. Schmitt, G., Case-based design and creativity. *Automation in construction*, 1993, **2**, 11–19.
22. Maher, M. L. and Zhang, D. M., CADSYN: A case-based design process model. *AI EDAM*, 1993, **7**(2), 97–110.
23. Maher, M. L., Balachandran, M. B. & Zhang, D. M., *Case-Based Reasoning in Design*. Lawrence Erlbaum Associates, Publishers, USA, 1995.
24. Nisbet, J., *It's Elemental My Dear Bill Said the QS to the Builder*. London, 1970.
25. Cartlidge, D. P. and Mehrtens, I. N., *Practical Cost Planning*. Hutchinsons, London, 1982.
26. Ferry, D. J. and Brandon, P. S., *Cost Planning of Buildings*, 6th edn. BSP, 1991.
27. Bisio, R. and Malabocchia, F., Cost estimation of software projects through case-based reasoning. *Proceedings of ICCBR-95*, Portugal, 1995.
28. Mukhopadhyay, T., Vicinanza, S. S. and Prietula, M. J., Examining the feasibility of a case-based reasoning model for software effort estimation (technical). *MIS Quarterly*, 1992, **16**(2).
29. Eliot, L. B., Case-based reasoning in estimation. *AI Expert*, **8**(5). Miller Freeman, 1993.
30. Klein, G. A. and Calderwood, R., How do people use analogues to make decisions? *Proceedings of Workshop on CBR (DARPA)*, ed. J. L. Kolodner. Morgan Kaufmann, Clearwater, Florida, CA, USA, 1988.
31. Klein, G. A., Whitaker, L. A. and King, J. A., Using analogues to predict and plan. *Proceedings of Workshop on CBR (DARPA)*, ed. J. L. Kolonder. Morgan Kaufmann, Clearwater, Florida, CA, USA, 1988.
32. Strottler, R. H., Case-based reasoning for bid preparation. *AI Expert*, **7**(3). Miller Freeman, 1992.
33. Watson, I. D. and Perera, R. S., NIRMANI: A case-based expert system for integrated design and estimating. Proceedings of the Expert Systems 1995 (accepted), Cambridge, U.K., 1995.
34. Schank, R. C., *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, New York, 1982.
35. Schank, R. C. and Abelson, R. P., *Scripts, Plas, Goals and Understanding*. Erlbaum, Hillsdale, New Jersey, USA, 1977.

36. Oxman, R. and Oxman, R., PRECEDENTS: Memory structure in design case libraries. *CAAD Futures '93*, ed. U. Flemming & S. V. Wyk, pp. 273–287.
37. Sycara, K. P., Argumentation: planning other agents' plans. *Proceedings of IJCAII-89*, Michigan, 1989.
38. Perera, R. S., Watson, I. D. and Alshawi, M., Case-based design approach for integration of design and estimating. *Proceedings of the 4th International Conference on the Application of AI to Civil and Structural Engineering, Civil Comp 95*, (to be published). Cambridge, UK, 1995.
39. Oxman, R., Data, knowledge and experience in multi-user information systems. *Construction Management and Economics*, **13,** 401–409. E. and F. N. Spon, 1995.
40. Bhat, R. R., Agent systems for case adaptation. *Proceedings of the International Conference on Computer-Aided Architectural Design, Centre for Advanced Studies in Architecture*. National University of Singapore, Singapore, 1995.
41. Woolridge, M., The logical modelling of computational multi-agent systems. PhD Thesis, Department of Computation, UMIST, Manchester, UK, 1992.
42. Watson, I. D. and Perera, R. S., Case-based design: a review and analysis of building design applications. In *Artificial Intelligence for Engineering Design, Analysis and Manufacturing AIEDAM*, ed. C. L. Dym. (Accepted for publication).
43. Maher, M. L. and Blanchandran, B., Multimedia approach to case-based structural design. *The Journal of computing in civil engineering*, 1994, **8**(3), 359–376.
44. Flemming, U., Case-based design in the SEED system. In *Automation in Construction*, Vol. 3. Elsevier Science B.V., 1994, pp. 123–133.
45. Watson, I. and Perera, S., The evaluation of a hierarchical case-representation using context guided retrieval. In *Case-Based Reasoning Research and Development, Proceedings of 2nd International Conference on Case-Based Reasoning, ICCBR-97*, ed. D. Lake & E. Plaza. Springer-Verlag.