Watson, I. & Perera, R.S. (1997), The Evaluation of a Hierarchical Case Representation using Context Guided Retrieval. In Case-Based Reasoning Research & Development, Leake, D.B. & Plaza, E. (Eds.). Lecture Notes in Artificial Intelligence Vol. 1266. Springer-Verlag, Berlin.

The Evaluation of a Hierarchical Case Representation using Context Guided Retrieval

Ian Watson[†] & Srinath Perera[*]
[†]AI-CBR, Bridgewater Building, University of Salford, Salford, M5 4WT, UK
[*]Department of Building Economics, University of Moratua, Moratua, Sri Lanka
{i.d.watson | r.s.perera}@surveying.salford.ac.uk

Abstract. This paper presents the results of the comparison of the performance of a hierarchical case representation using a context guided retrieval method against that of a simpler flat file representation using standard nearest neighbour retrieval. The estimation of the construction costs of light industrial warehouse buildings is used as the test domain. Each case comprises approximately 400 features. These are structured into a hierarchical case representation that holds more general contextual features at its top and specific building elements at its leaves. A modified nearest neighbour retrieval algorithm is used that is guided by contextual similarity. Problems are decomposed into sub-problems and solutions recomposed into a final solution. The comparative results show that the context guided retrieval method using the hierarchical case representation out performs the simple flat file representation and standard nearest neighbour retrieval.

Keywords. Case-Based Reasoning, Context Guided Retrieval, Hierarchical Case-Representation

1. Introduction

Representing cases as a set of constituent pieces [Barletta & Mark, 1988, Macedo et al., 1996], *snippets* [Kolodner, 1988; Redmond, 1990; Sycara & Navinchandra, 1991] or *footprints* [Veloso, 1992; Bento et al., 1994], instead of as a single large entity, has long been proposed as a way of improving the effectiveness of a CBR system. These parts, when represented as separate structured cases, can be represented, retrieved and recomposed separately to create new solutions [Flemming, 1994; Maher & Balchandran, 1994; Bartsch-Sporl, 1995; Hunt & Miles, 1995]. Some systems, for example, CADSYN, explicitly take into account the context of a snippet or sub-problem to reduced constraint problems when recomposing solutions [Maher & Zhang, 1991].

Many successful CBR systems use relatively simple case representations of attribute-value pairs stored in flat files or record structures similar to those of a conventional database. There are good reasons for this. A primary one is, that for many commercial applications, the knowledge engineering effort required to create case-bases must be kept to a minimum. These case representations may be characterised as being *knowledge-poor*. That is they do not contain many (or any) structures that describe the relationships or constraints between case features. However, these case representations usually describe relatively simple cases with few indexed features, perhaps in the order of ten to twenty indexed features.

As the number of indexed case features increases (i.e., the number of features that are predictive of a case's solution or outcome) the utility of this knowledge-poor approach reduces. As the problem space increases, from say a 20 dimensional space to a 200 dimensional space it becomes statistically less likely that a close matching case will exist. Thus, a retrieve and propose CBR system (i.e., one without adaptation) may be proposing a relatively distant solution. If adaptation is used, the adaptation effort or distance will increase correspondingly, possibly reducing the accuracy or utility of the solution. This is illustrated in the two figures below, after Leake [p8, 1996]. Figure 1, shows, on the left, a relatively small problem space and assumes a similar sized solution space. Notice that the retrieval distance (the arrow labelled R) and the adaptation distance (the arrow labelled A) are both quite short. As the size of the problem space increases (shown on the right) the retrieval and adaptation distances may increase, as shown by the lengths of the arrows.

Moreover, as has been reported by Maher et al. [1995] there is often an inverse relationship between the number of cases in a case-base and the number of indexed features in the cases. This is because it often harder to collect a few large cases than it is to collect hundreds of small cases. Thus, case coverage is often likely to be lower in a large problem space than in a small problem space. This may cause the case-base to return a mediocre match that will require considerable adaptation, resulting in poorer solutions.

A potential solution to this problem is the *divide and conquer* approach. This suggests that, where suitable, a large problem is divided into several smaller sub-problems, each of which can be solved separately using CBR. The sub-solutions can then be combined to produce an accurate solution to the entire problem [Maher & Zhang, 1991]. A key assumption for this approach is that the sub-problems are not highly constrained one upon the other, so that they can be solved independently (i.e., that the problem can be sensibly decomposed and the solution recomposed). This approach may be visualised as in Figure 2.

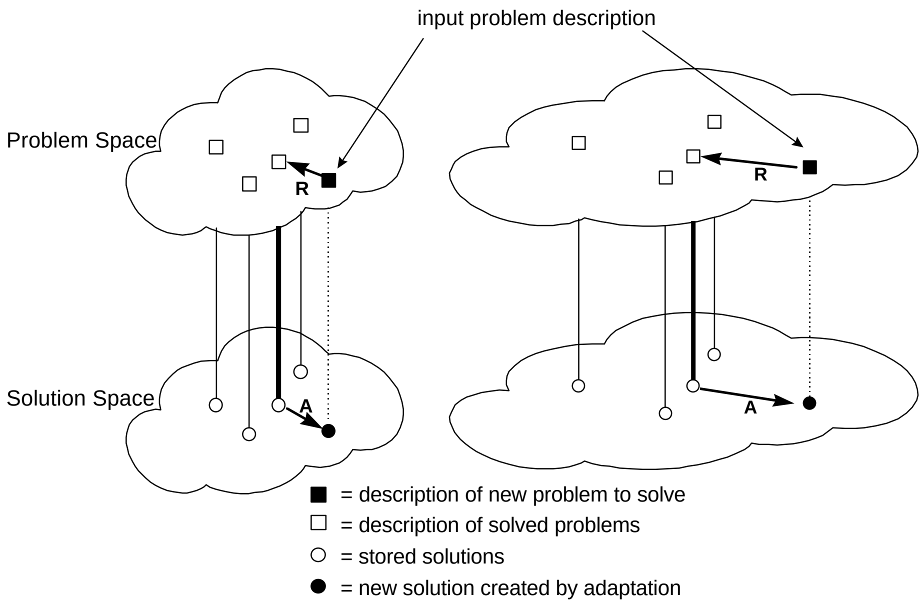

Figure 1. Small and Large Problem & Solution Spaces, after Leake [p.8, 1996]

The advantage of this approach is that each individual sub-problem is represented by a case-base that is significantly smaller (in terms of problem and solution space size) than if the whole problem were represented by a single case-base. Because each sub-problem space has fewer case features, the theory predicts, that each individual sub-case retrieval distance will be shorter than for the un-decomposed problem. Therefore, the adaptation distance will be shorter and a better sub-solution will be generated. Assuming there are no conflicting constraints, the recomposition of sub-solutions will produce a better solution than would have been obtained by using a single large case-base. One way that has been suggested to reduce constraint problems with solution recomposition is to use contextual information to guide retrieval [Hammond, 1986; Hennessy & Hinkle, 1992; Kolodner, 1993; Maher et al., 1995; Marir & Watson, 1995; Ram & Francis, 1996]. The argument being, that if cases share similar contexts, this will reduce constraint problems during solution recomposition.

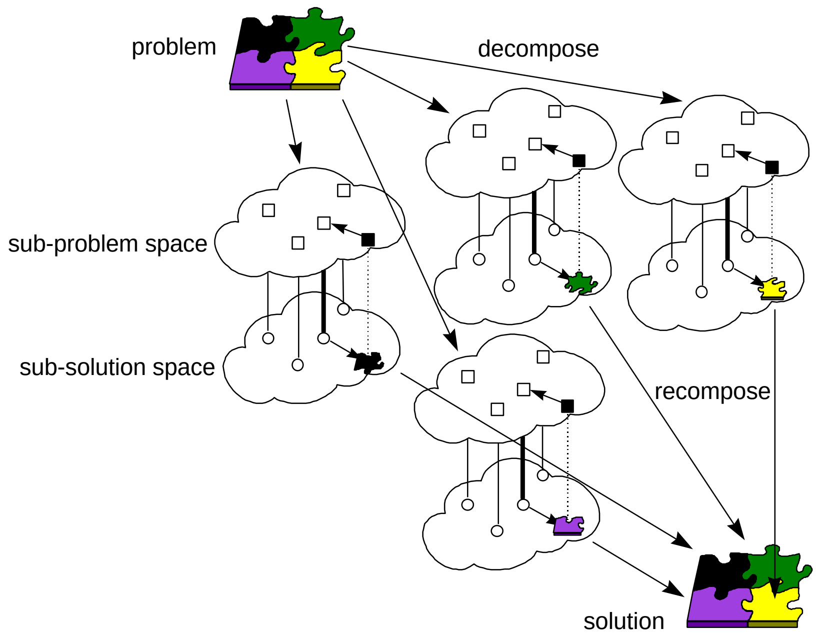

Figure 2. Problem Decomposition and Solution Recomposition

The purpose of the study presented in this paper was to quantitatively assess the accuracy of a CBR system that uses a hierarchical case representation and context-guided retrieval to decompose a complex problem and recompose a solution. The accuracy of this complex case representation and retrieval technique is compared to that of a simple flat record of attribute-value pairs using a standard nearest neighbour retrieval algorithm. The evaluation will show that the more complex representation and retrieval method out performs the simpler representation, thereby justifying the knowledge engineering and programming effort put into it.

2. The Problem Domain

For this study we selected the estimation of the construction costs of light industrial warehousing as a suitable domain. These buildings are used as storage and distribution warehouses, as low cost retail buildings, and as light industrial factory units. They were particularly suitable for this study for the following reasons:

- These warehouses are strictly functional buildings with aesthetic issues being very secondary (i.e., they rarely win design prizes). Consequently, cost is a more important issue than for most other building types.
- They are constructed using steel frames that are produced in standard sizes along with many other components (e.g., roofing sheets) that are also produced in standard sizes.
- The buildings are structurally fairly simple and consequently the constraints between different building elements are small. This therefore suggested that divided and conquer would be appropriate.
- The cost of a building is derived directly from the cost of its sub-assemblies. Thus, the problem decomposes naturally. This is supported by the way that cost estimators usually work. They calculate the cost of each sub-assembly and sum them to obtain a total cost.
- Finally, we had access to a cost estimating computer system for this building type. This has significant methodological importance and will be discussed later.

3. The Case Representation

The system, called NIRMANI, was implemented in ART*Enterprise, from BrightWare, on Windows 95 [Watson & Perera, 1995]. The environment provides an object-oriented knowledge-based development environment, that supports objects, rules (a forward chaining Rete algorithm), a procedural programming environment, case-based reasoning (nearest neighbour), a GUI builder, and an ODBC database interface [Watson, 1997].

Representing cases hierarchically is a popular approach to the use and reuse of sub-cases (e.g., Redmond, 1990; Goel, 1994; Aha & Branting, 1995). 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) of architects and cost estimators. A third level decomposes the design into functional spaces and aesthetic requirements hierarchies and the estimating problem into an industry standard elemental classification hierarchy [Perera & Watson 1996].

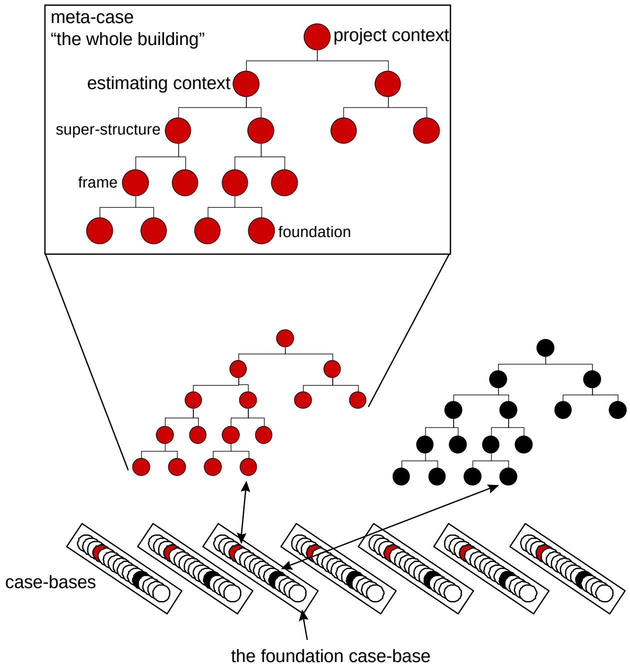

Figure 3. Schematic of the Hierarchical Case Representation

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 keep their case data in their existing databases [Brown et al., 1995]. 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.

A *Project Context* case describes the environment within which the project was carried out (features such as the type of building, its intended function, gross internal floor area (GIFA), the site conditions, and other features common to the project context). 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 recomposition due to contextual dissimilarity.

	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
12.	Use	Basic materials Completed - goods Perishable - goods Equipment Retail goods Electronic - Components Re-distribution- Goods	catnl:use
13.	Storage_Category	Flammable Non-flammable Chemical Flammable Chemical Non-flammable	catnl:st-cat
14.	Region	List of regions (BCIS)	catnl:regions
15.	Construction_period	Months	cat:float-or-nil
16.	Project_duration	Months	cat:float-or-nil
17.	Project_Cost_limit	£	cat:integer-or-nil
18.	Tender_Month	List of Months	catnl:months
19.	Tender_Year	1983-1998	cat-nir:year1983-1998
20.	Actual_project_cost	£	cat:integer-or-nil
21.	Total_variations	£	cat:integer-or-nil
22.	Completed_duration	Months	cat:float-or-nil
23.	Completed_date	Text	default
24.	Type_of_contract	List of Contract Types	catnl:contract
25.	Gross_Floor_Area	m^2	cat:integer-or-nil
26.	Gross_office_area	m^2	cat:integer-or-nil
27.	Type_of_Structure	Portal Frame Propped Portal- Frame Steel Frame & Joists Clear Span - Frame Structural Steel - Frame Timber Frame	catnl:struct-type
…	…	…	…
66.	Structural_Engineer	Text	default
67.	Services_Engineer	Text	default
68.	Other_Consultants	Text	default
69.	Contractor	Text	default
70.	Contractor_Address	Text	default

Table 1. A Selection of Attributes from the Project Context Case Definition

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).

4. Retrieval

NIRMANI provides a variety of retrieval methods, of which only two are compared in this paper. Full details of these retrieval methods can be found in Perera & Watson [1996]. 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 compared in this paper are described below.

4.1 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.

4.2 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 *Project 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, retrieval is 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.

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.

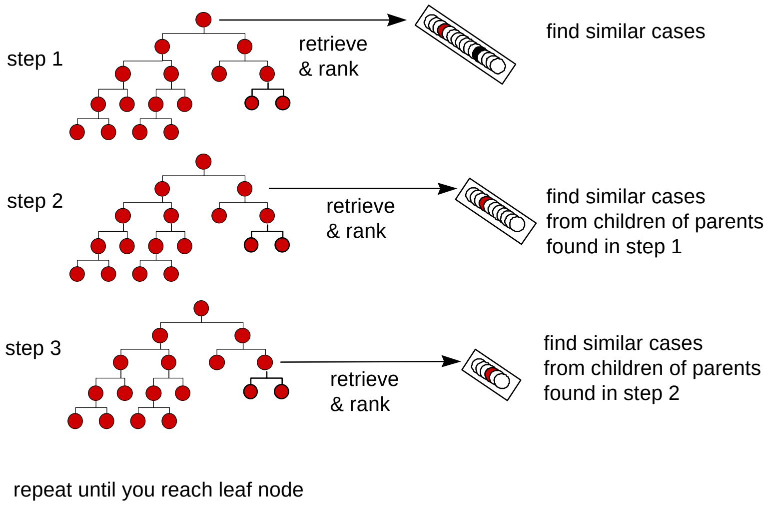

Figure 4. Context Guided Retrieval

5. Adaptation

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 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 interpolation. For example, if a retrieved case has the feature "*floor finishes*" at a cost of "*£12,000*" with a GIFA of "*2000m^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.

6. Methodology

In the 1980s and early 1990s Salford University, in collaboration with the Royal Institution of Chartered Surveyors (the RICS is the professional institution for cost estimators in the UK), developed several knowledge-based construction cost estimation systems. The first of these, a rule-based system called ELSIE, could estimate the construction costs of commercial office developments [Brandon et al., 1988]. In a subsequent development another rule-based system, called ELI, was developed for estimating the construction costs of light industrial warehouse units. These systems are sold commercially, by a joint venture company, and have sold over a thousand copies world-wide.

The RICS commissioned a study to check the accuracy of the systems [Castell et al., 1992], which found that their estimates are within plus or minus 5% of eventual construction costs. This is well within acceptable error and is a good as the most experienced cost estimators [Skitmore, 1990]. For our study we used ELI as both a case generator (i.e., to produce projects to populate our case-base) and as an evaluator (i.e., to test the accuracy of the CBR systems).

6.1 Case Acquisition

Details of thirty construction projects were obtained from the Building Construction Cost Information Service (BCIS), an information service for the UK construction industry. ELI was used to generate a further twelve hypothetical construction projects. These projects were carefully designed to fill in the gaps between the thirty real projects from the BCIS. These were then entered into a database that NIRMANI used for its case data. The projects generated by ELI were carefully designed so as to create a case-base with an even case distribution. Thus, projects were created which had a variety of functions (e.g., dry goods distribution warehouses, cold storage warehouses, flammable goods storage and distribution, retail warehouses, etc.). The projects varied in size consistently in graduations of approximately 100 m^2, from 1,500 m^2 to 3,500 m^2. In addition, a range of construction complexity with additional features, such as office space, were included. We recognise that this case-base is artificial. We felt that a well distributed case-base should be analysed before attempting a randomly distributed one.

6.2 Evaluation

Evaluation of the accuracy of NIRMANI using the two retrieval techniques described above was done in three ways.

1. Cases with a known construction cost that were in NIRMANI's case-base were removed and used as target cases (i.e., as a new problem to solve). This would remove a known case from the well-distributed case-base and force NIRMANI to solve the problem using neighbouring cases. This test was performed five times.
2. New projects (i.e., ones that NIRMANI had never seen) were developed by ELI and hence we new ELI's estimation of their construction cost. These were then presented to NIRMANI as new problems for it to estimate. This test was performed ten times.
3. Finally, as a test of both ELI's and NIRMANI's accuracy, real projects (with known costs) were obtained from the Building Cost Information Service and given to ELI and NIRMANI to solve. This acted as an independent check on the accuracy of both systems.

These evaluation methods are shown schematically in Figure 5.

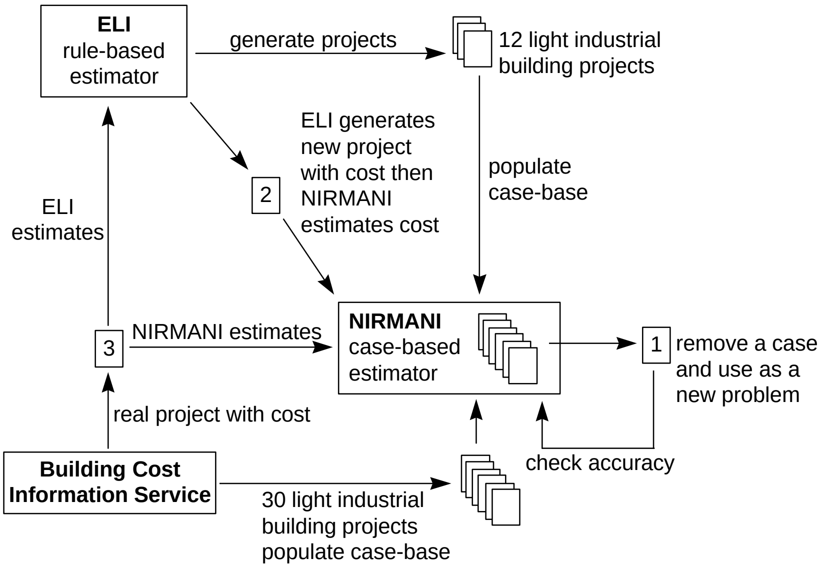

Figure 5. The Case Generation & Evaluation Methodology

The results from the evaluation tests were statistically analysed using the coefficient of variation method. This technique is widely used as the most common criteria for the determination of the accuracy of an estimating method or model [McCaffer, 1975]. CV is defined as:

$$CV = \frac{\text{Standard Deviation of Residuals } (S_r)}{\text{Mean Cost of All Schemes - Actual } (M_a)}$$

Thus, *CV* can be termed as the estimating error where:

accuracy = 1 - % estimating error

and therefore:

accuracy = 100 - CV

7. Results

A summary of the tests is given below and shown in Figure 6. Exactly the same feature weightings were used for both the NN retrieval and the context guided retrieval.

7.1 Test 1

For test 1 a case was removed from NIRMANI's case-base and used a target case. The results of the five tests are summarised in Table 2.

	Data				**NN Retrieval**		**Context Guided NN**	
Test No	GIFA m^2	Office Area m^2	Building Use	Actual Cost £	Estimate £	% Diff.	Estimate £	% Diff
T1	2325	111	Storage	500,562	525,314	4.94	499,539	-0.20
T2	2138	244	Retail	648,500	468,750	-27.72	603,825	-6.89
T3	2000	100	Storage	660,100	678,628	2.81	663,129	0.46
T4	2590	250	Storage	593,697	657,029	10.67	592,075	-0.27
T5	1500	0	Storage	399,506	294,636	-26.25	421,566	5.52

Table 2. Results of Test 1

Two major studies on the accuracy of estimation in the construction industry revealed that an accuracy ranging from ± 15% to ± 20% [Ashworth & Skitmore, 1983] and ± 8% to ± 15% [Skitmore et. al., 1990] are acceptable for early stage estimating of construction costs. Therefore, all the estimates using context guided retrieval were well within acceptable error. However, the flat representation using standard nearest neighbour failed in tests T2 and T5 (with context guided retrieval the percentage difference was also considerably greater for these two). This was because the cases in these two tests do not have close nearest neighbours within NIRMANI's case-base. The accuracy of the context guided retrieval is increased because it can find nearest neighbours for individual elements of buildings, whereas the other technique cannot find a whole building that matches well enough. A detailed examination of these two tests revealed that the poor estimate was caused by a poor match for the substructure, for test T2, and for external works for test T5.

	Nearest Neighbour		**Context Guided NN**	
Test No	% Diff.	Contributor	% Diff.	Contributing Cases
T1	4.94	WHS_A3	-0.20	*WHS_A2*3*, WHS_A3*2
T2	-27.72	WHS_A3	-6.89	*WHS_T1*, WHD_GG1, WHS_A4, WHS_A1*2, WHS_A2
T3	2.81	WHS_T2	0.46	*WHS_T2*5*
T4	10.67	WHS_C2	-0.27	*WHS_C2*3*, WHS_C3*2
T5	-26.25	WHS_A4	5.52	*WHS_E1*, WHS_A4, WHS_A2, WHS_T2, WHS_G1, WHD_K1,

Table 3. Cases Contributing to a Solution for Test 1

For the standard nearest neighbour retrieval only one entire meta-case can contribute to the solution. For context guided retrieval parts of different meta-cases can contribute. In Table 3, the case reference number that is underlined and in italics contributed most to the solution.

7.2 Test 2

For test 2, ELI was used to generate ten new projects and to estimate their construction costs. NIRMANI was then given the same projects to estimate.

	Data			**ELI**	**Nearest Neighbour**		**Context Guided NN**	
Test No.	GIFA m^2	Office Area m^2	Building Use	Estimate £s	Estimate £s	% Diff.	Estimate £s	% Diff.
CS1	1,500	75	Storage	329,600	320,773	-2.68	322,177	-2.25
CS2	1,750	100	Storage	388,500	430,783	10.88	391,598	0.80
CS3	2,000	125	Storage	486,600	477,114	-1.95	488,709	0.43
CS4	2,000	200	Retail	575,600	614,457	6.75	581,639	1.05
CS5	2,250	175	Storage	607,400	474,812	-21.83	606,749	-0.11
CS6	2,500	200	Storage	663,200	602,832	-9.10	661,294	-0.29
CS7	2,750	200	Storage	1,221,100	809,090	-33.74	1,233,125	0.98
CS8	3,000	250	Retail	825,100	903,321	9.48	809,475	-1.89
CS9	3,250	300	Retail	898,400	970,686	8.05	910,048	1.30
CS10	1,250	50	Distribution	363,700	326,330	-10.27	369,652	1.64

Table 4 Results of Test 2

This test gave consistently similar estimates with a maximum percentage difference of -2.25% for context guided retrieval. However, the standard nearest neighbour retrieval was more inconsistent, ranging from 10.88% to -33.74%.

	Nearest Neighbour		**Context Guided NN**	
Test No.	% Diff.	Contributor	% Diff.	Contributors
CS1	-2.68	WHS_A4	-2.25	*WHS_A4*5*, WHS_F1, WHS_W1
CS2	10.88	WHS_T2	0.80	*WHS_A4*4*, WHS_T2*3, WHS_D1*2
CS3	-1.95	WHS_T1	0.43	*WHS_T1*3*, WHS_T2, WHR_B1, WHS_A3*2, WHR_BB1*2
CS4	6.75	WHR_BB1	1.05	*WHR_BB1*4*, WHR_M1, WHS_A4, WHS_A3
CS5	-21.83	WHS_A3	-0.11	*WHS_X1*3*, WHS_A1, WHS_A3, WHS_T3
CS6	-9.10	WHS_T4	-0.29	*WHS_T4*3*, WHS_C1, WHS_C5, WHS_D1
CS7	-33.74	WHS_C2	0.98	*WHS_AA1*2*, WHS_C2*4, WHR_BB1, WHS_W1

	Nearest Neighbour		Context Guided NN	
Test No.	% Diff.	Contributor	% Diff.	Contributors
CS8	9.48	WHR_N1	-1.89	*WHR_N1*3*, WHR_Q1*2, WHR_FF1, WHR_V1
CS9	8.05	WHR_N1	1.30	*WHR_N1*2*, WHR_Z1, WHR_M1, WHS_AA1, WHS_W1
CS10	-10.27	WHD_HH1	1.64	*WHD_HH1*3*, WHR_V1, WHS_W1*2, WHS_T2*2

Table 5. Cases Contributing to a Solution For Test 2

7.3 Statistical Analysis

The results from test 1 and 2 were combined (i.e., n = 15)and are summarised in Table 6.

Statistic	BCIS or ELI	Nearest Neighbour			Context Guided NN		
	Cost/m^2	Cost/m^2	Diff Absolute	% Diff. Absolute	Cost/m^2	Diff. Absolute	% Diff. Absolute
Mean	275.92	256.51	37.58	12.47	276.02	4.53	1.61
Standard Deviation (Population)	55.09806	40.93379	37.559007	10.008	56.080815	7.192149	1.993
Coefficient of Variation (CV)			14.090187			2.095961	
T Test		0.282962			0.9958138		
Co-relation coefficient		0.502196			0.9917861		
Estimating Accuracy			85.909813			97.904034	

Table 6. Summary of Results from Tests 1 & 2

Since the sample size was less than 30 (i.e., n = 15), the Students' *t* Test was used for statistical analysis. Three tests were carried out. All were carried out initially for 95% confidence limits, which is accepted as providing statistically significant results.

1. *Hypothesis Test 1 (HT 1)*

   The context guided retrieval achieves a mean accuracy of 98% (i.e. 2% error in estimating). In statistical terms this means hypothesising a population mean of 2% (μ = 2 null hypothesis. The statistical aim of the test is to prove that $\mu = 2$ is not possible. H_O has to be accepted, because the hypothesis $\mu = 2$, or estimating accuracy $EA_N = 98\%$, cannot be disapproved at a 95% level of confidence.

2. *Hypothesis Test 2 (HT 2)*

   The same test as HT 1 was carried out to check if the standard nearest neighbour retrieval could achieve a mean accuracy of 86% (i.e. 14% error in estimating). The test hypothesis was, H_O: $\mu = 14$ Null hypothesis. H_O has to be accepted, because the hypothesis $\mu = 14$ or estimating accuracy EA_F = 86% cannot be disapproved at a 95% level of confidence.

3. *Hypothesis Test 3 (HT 3)*

   The aim of this test is to determine whether the results obtained for the standard nearest neighbour and context guided retrieval represent significantly different approaches. In statistical terms this involve testing whether the test samples could be from the same population. In order to achieve these results a "Paired Sample Student' *t* Test" was carried out. The test hypothesis was as follows: H_O: μ = 0 (The mean of the difference between the two techniques is zero). The test was repeated for the differences in estimated values (absolute) obtained from Table 6. T-Tests were carried out as for HT 1 for a 95% level of confidence. This found that H_O could not be rejected at 95% confidence levels. However, at 90% confidence levels H_O could be rejected.

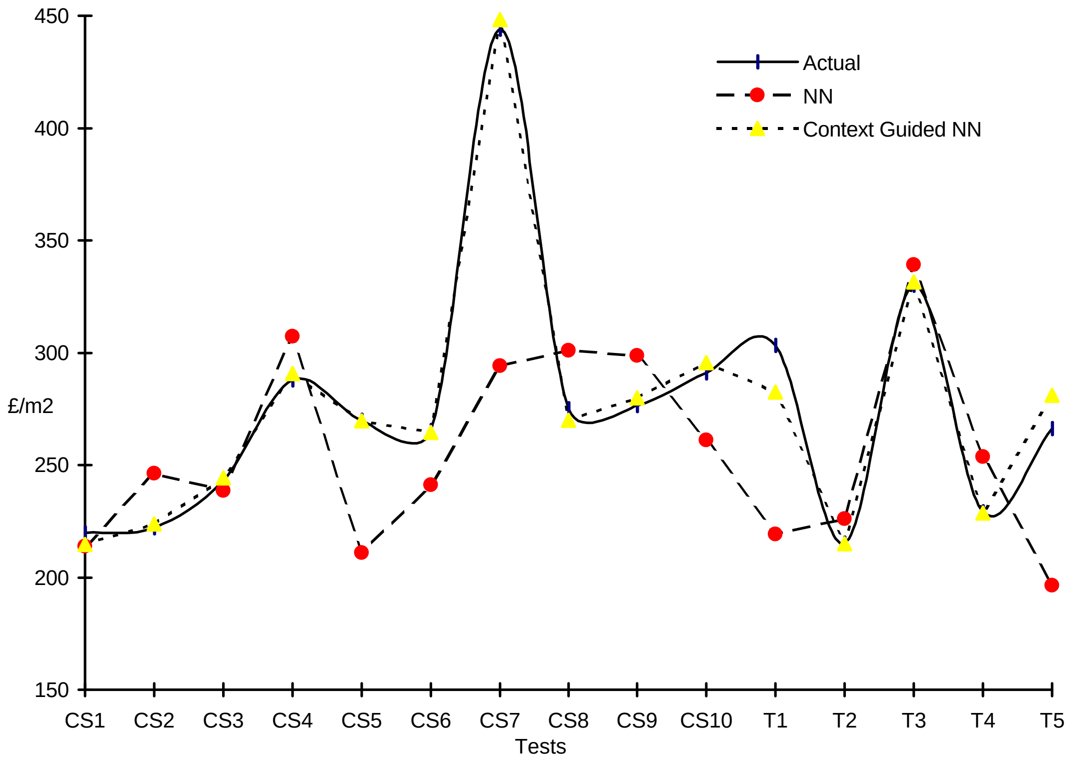

Figure 6. Summary of Test Results

8. Conclusion

The systematic evaluation of a CBR system is very difficult because such systems are typically very complex with many interacting components [Santamaria & Ram, 1996]. Consequently, this study has simplified the performance of our system down to a single quantifiable measure - estimating accuracy. We accept that this measure is a simplification of the performance of our system. Nonetheless, the evaluation demonstrates that the context guided retrieval method out performs that of the simpler flat-file nearest neighbour method. The only times that the simpler technique performed acceptably were when a problem happened to find a close near neighbour within the case-base. When the simpler technique performed badly it was because it was unable to find a complete matching case and was forced to use the closest case that matched on a subset of features. Conversely, when the context guided retrieval method significantly out performs the simpler technique it is because it has composed a solution from many cases. Thus, when a close near neighbour cannot be found the divide and conquer approach, using context guided retrieval, performs better as the theory predicts. It is interesting to note that that the simpler technique usually recognises which case can contribute most to solution, but, by being unable to use snippets from other cases as well, its accuracy is reduced.

We recognise that this has been a fairly limited study, with a small sample size. We have shown that for our tests the context guided retrieval (HT 1) was accurate. However, there was only a 90% confidence that this technique was statistically different from standard nearest neighbour retrieval (HT 3). Because of the size of each meta-case (i.e., approx. 400 case features) each single evaluation test took one day to perform. Consequently, the number of tests was limited and therefore it would be unwise to rely too heavily on the simple statistical analysis performed here. However, the results are indicative and support the view that divide and conquer, through problem decomposition and solution recomposition, is an effective method of solving problems with large complex cases. The context guided retrieval method evaluated here may also be a useful way of reducing the problems of conflicting constraints between parts of the solution.

The fact that the case-base was populated with an evenly distributed set of cases may have skewed our results. Although from the results it would appear that this should skew the results in favour of the simpler method. Since it performs better when a close good match can be found, one would expect it to perform more erratically with a more unevenly distributed case-base.

Finally, it was interesting to see that the case-based estimator performed as well as the rule-based estimation system, with a mean error of 2%. The rule-based estimator took over three person years to implement, whilst the case-based estimator took less than half that time. This further supports the many findings that show that CBR systems can be implemented quicker than their rule-based counterparts [Simoudis & Miller, 1991; Mark et al., 1996].

9. References

Aha, D., & Branting, K. (1995). Stratified Case-Based Reasoning: Reusing Hierarchical Problem Solving Episodes. In, *Proceedings of the 13th International Joint Conference on Artificial Intelligence, IJCAI-95*, pp.384-390. IJCAI, Menlo Park, Calif., US.

Ashworth, A., & Skitmore, R.M. (1982), Accuracy in Estimating. *The Chartered Institute of Building*, Occasional Paper No. 27.

Barletta, R., & Mark, W. (1988). Breaking Cases into Pieces. In, *Proceedings of the DARPA Case-Based Reasoning Workshop,* Kolodner, J.L., (Ed.), Morgan Kaufmann, Calif., US.

Bartsch-Sporl, B. (1995), Towards the Integration of Case-Based, Schema-Based and Model-Based Reasoning for Supporting Complex Design Tasks. In, *Case-Based Reasoning Research & Development - ICCBR '95*, Veloso, M., & Aamodt, A. (Eds.), pp. 145 - 156, Springer-Verlag, Berlin.

Bento, C., Macedo, L., & Costa, E. (1994). RECIDE - Reasoning with Cases Imperfectly Described and Explained. In, *Proceeding of the 2nd European Workshop on Case-Based Reasoning*.

Brandon, P., Basden, A., Hamilton, I., & Stockley, J. (1988). *The Strategic Planning of Construction Projects*, RICS Publications, London, UK.

Brown, M., Watson, I., & Filer, N. (1995). Separating the cases from the data: Towards more flexible case-based reasoning. In, *Case-Based Reasoning Research and Development*, Veloso, M., & Aamodt, A. (Eds.), Lecture Notes in Artificial Intelligence 1010, Springer-Verlag, Berlin.

Castell, A.M., Basden, A., Erdos, G., Barrows, P., & Brandon, P.S. (1992). Knowledge Based Systems In Use: A Case Study. In, *Proceedings of the 12th. Annual Conference of the British Computer Society Specialist Group on Expert Systems*. Cambridge University Press, Cambridge, UK.

Flemming, U., (1994). Case-based design in the SEED system,. *Automation in Construction,* 3: pp.123-133, Elsevier Science B.V.

Goel, A., Ali, K., Donnellan, M., Garza, A., & Callantine, T. (1994). Multistrategy Adaptive Navigational Path Planning. *IEEE Expert*, 9(6): pp.57-65

Hammond, K.J. (1986), CHEF: A model of case-based planning. In, *Proc. American Association for Artificial Intelligence, AAAI-86*, Philadelphia, PA, US.

Hennessy, D., & Hinkle, D. (1992), Applying Case-Based Reasoning to Autoclave Loading, *IEEE Exper,t* 7(5): pp.21-26.

Hunt, J., & Miles, R. (1995), Towards an Intelligent Architectural Design Aid. *Expert Systems,* 12(3): pp.209-218,

Kolodner, J.L. (1988). Retrieving events from a case memory: A parallel implementation. In, *Proceedings of the DARPA Case-Based Reasoning Workshop*, Kolodner, J.L. (Ed.), Morgan Kaufmann, Calif., US.

Kolodner, J.L. (1993), *Case-Based Reasoning*. Morgan Kaufmann Publishers Inc., CA, US.

Leake, D.B., (1996). CBR in Context: The Present and Future. In, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, Leake, D.B. (Ed.). AAAI Press / The MIT Press, Menlo Park, Calif., US.

Macedo, L., Pereira, F.C., Grilo, C., & Cardoso, A. (1996). Plans as Structured Networks of Hierarchically and Temporally Related Case Pieces. In, *Advances in Case-Based Reasoning,* Smith, I., & Faltings, B. (Eds.), Lecture Notes in Artificial Intelligence 1168, Springer-Verlag, Berlin.

Maher, M.L., & Balachandran, B. (1994). Flexible Retrieval Strategies for Case-Based Design. In, *Artificial Inteliigence in Design '94,* Gero, J.S., & Sudweeks, F. (Eds.), pp.163-180, Kluwer Academic Publishers, Netherlands.

Maher, M.L., Balachandran, M.B., & Zhang, D.M. (1995). *Case-Based Reasoning in Design.* Lawrence Erlbaum Associates, US.

Maher, M.L. & Zhang, D.M. (1991). CADSYN: using case and decomposition knowledge for design synthesis. In, *Artificial Intelligence in Design,* Gero, J.S. (Ed.), Butterworth-Heinmann. Oxford. UK.

Marir, F., & Watson, I. (1995), Representing and Indexing Building Refurbishment Cases for Multiple Retrieval of Adaptable Pieces of Cases. In, *Case-Based Reasoning Research & Development - ICCBR '95*, Veloso, M., and Aamodt, A. (Eds.), pp.55 - 66, Springer-Verlag, Berlin.

Mark, W., Simoudis, E., & Hinkle, D. (1996). Case-Based Reasoning: Expectations and Results. In, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, Leake, D.B. (Ed.). AAAI Press / The MIT Press, Menlo Park, Calif., US.

Perera, R.S., and Watson, I.D. (1996). Multi-Agent Collaborative Case-Based Estimating and Design in NIRMANI: Organising a Multi-Perspective Case Memory. In, *Information Processing in Civil and Structural Engineering Design*, pp.53-64, Kumar, B. (Ed.), Civil-Comp Press, Scotland.

Ram, A., & Francis, A.G. (1996). Multi-plan Retrieval and Adaptation in an Experience-Based Agent. In, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, Leake, D.B. (Ed.). AAAI Press / The MIT Press, Menlo Park, Calif., US.

Redmond, M. (1990). Distributed Cases for Case-Based Reasoning: Facilitating Use of Multiple Cases. In, *Proceedings of AAAI*.

Santamaria, J.C., & Ram, A. (1996). Systematic Evaluation of Design Decisions in Case-Based Reasoning Systems. In, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, Leake, D.B. (Ed.). AAAI Press / The MIT Press, Menlo Park, Calif., US.

Simoudis, E. & Miller, J.S. (1991). The Application of CBR to Help Desk Applications. In, *Proceedings of the DARPA Workshop on Case-Based Reasoning*, Bareiss R. (Ed.). Morgan Kaufmann, San Francisco, Calif., US.

Sycara, K.P. & Navinchandra D. (1991). Influences: A Thematic Abstraction for Creative Use of Multiple Cases . In, *Proceedings of the DARPA Case-Based Reasoning Workshop*, Bareiss, E.R., Morgan Kaufmann, Calif., US.

Veloso, M.M. (1992). *Learning by analogical reasoning in general problem solving*. Carnegie Mellon University, School of Computer Science Technical Report no. CMU-CS-92-174.

Watson, I. (1997). *Applying Case-Based Reasoning*. Morgan Kaufmann, CA, USA (forthcoming).

Watson, I., and Perera, R.S. (1995). NIRMANI: A Case-Based Expert System for Integrated Design & Estimating. In, *Applications and Innovations in Expert Systems III*, pp.335-348, Macintosh, A. and Cooper, C. (Eds.). SGES Publications, UK.

10. Acknowledgements

The authors would like to acknowledge the Association of Commonwealth Universities, the University of Moratuwa, Sri Lanka, and EPSRC, whose grants: GR/J42496, GR/J43660 and GR/L16330, helped support this work.