VICTORIA UNIVERSITY OF WELLINGTON

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

Iterative Design of Case Retrieval Systems

Eric K. Jones
Aaron Roydhouse

Technical Report CS-TR-94/6
May, 1994

Abstract

We argue that the process of designing effective systems for case retrieval should not be divorced from the process of evaluation. Developing a suitable representation vocabulary for indices, and suitable mechanisms for case retrieval should proceed in an iterative fashion, guided by the joint application of sound AI principles, domain expertise, and empirical evaluation.

We lay out the key components of a case retrieval system and suggest procedures for evaluating each component. We illustrate our claims using the MetVUW Workbench, a system for intelligent retrieval and display of historical meteorological data.

Publishing Information

A shorter version of this paper is to appear in *Proceedings of the AAAI'94 Workshop on Case Based Reasoning*, Seattle, Washington, August 1994.

Author Information

Eric Jones is on the academic staff of the Department of Computer Science at Victoria University of Wellington and a member of the department's artificial intelligence group.

Aaron Roydhouse is a Masters student with the Department of Computer Science at Victoria University of Wellington.

1 Introduction

Case retrieval is an important component of case-based reasoning. Design of a case retriever requires addressing the *indexing problem* (Domeshek, 1991; Schank, 1982; Schank and Fano, 1992): the problem of accessing appropriate cases using partial descriptions of their content. In this paper, we argue that solutions to the indexing problem should be developed by an iterative process of design and evaluation, and that this process is to a significant extent an experimental and empirical enterprise. Evaluation should not be carried out only after completion of a system, but should serve as an integral part of the development process.

We draw on our experience with the design and implementation of MetVUW Workbench, a system for intelligent retrieval and display of historical meteorological data (Roydhouse et al., 1993; Jones and Roydhouse, 1994). The system is intended to serve as a "memory amplifier" for meteorologists that allows them to rapidly locate historical situations of interest. Two groups of end users are anticipated:

1. *Weather forecasters* should be able to quickly retrieve past cases that are similar to the current situation in meteorologically significant respects, providing them with an additional source of information to supplement the output of numerical models.

2. *Research scientists* should be able to perform empirical analyses of meteorological phenomena of interest, using MetVUW Workbench to rapidly identify large numbers of examples of the phenomena.

In the first instance, MetVUW Workbench is intended to serve as an intelligent assistant to a human forecaster. For example, forecasters at the Meteorological Service of New Zealand currently engage in a "map session" every day at 10am, at which they compare competing prognoses to arrive at a consensus forecast. MetVUW Workbench is designed to provide valuable input to discussions such as these. It is envisaged that forecasters will follow three steps when using the system:

1. Determine the meteorological processes and systems that appear to be driving the current weather situation.

2. Formulate a query to MetVUW Workbench in terms of these processes and systems.

3. Analyse and summarise the cases that the system returns, or display any of the data associated with particular retrieved cases. Analyses and summaries can be used to provide evidence for or against the prognoses of numerical models, to help adjudicate between alternative forecasting hypotheses, and to draw attention to possible outcomes that forecasters may not have considered. Detailed examination of particular past cases can also usefully guide analysis of the current situation (Hunter, 1989).

In the context of meteorological research, MetVUW Workbench is expected to open up a wealth of possibilities for empirical investigation. In the past, large-scale investigations have been impeded by the difficulty of rapidly identifying relevant data. MetVUW Workbench solves this problem by providing a convenient summary view of a much larger archive; this summary view is small enough to be stored on hard disk at comparatively modest cost. If more detailed analyses are required, larger amounts of data relating to a particular situation can be retrieved using slower media such as CD-ROM or magnetic tape.

Each case in MetVUW Workbench is a slice of time for which meteorological data is available. The data available to us includes satellite imagery stored both in digital form and on laser disc, a document archive, and numeric fields from the European Center for Medium-range Weather Forecasting (ECMWF). Examples of numeric fields include pressure, temperature, relative humidity, wind speed, and relative vorticity, all of which are available at 14 different levels of the atmosphere. We are working with a portion of this numeric data that covers the Australasian region.

Cases are indexed and retrieved in terms of high-level descriptions of the content of their numeric fields. Numeric fields are available at 12-hour intervals. We currently possess 3.5 years (1.6Gb) of data, which allows us to build a case base containing some 2500 cases. The experiments described in this paper, however, were carried out on a 550-case subset of the data. In the near future, we plan to obtain an additional 10 years of data, allowing us to expand the case base to 10 000 cases (6Gb).

Case retrieval in MetVUW Workbench is a process of comparing queries (sometimes called *index probes* (Domeshek, 1991)) against explicitly represented *index labels*. Queries identify particular features of the current situation that appear to be meteorologically significant. Index labels are representations of high-level features of the weather situation in the case. As far as possible, index labels are extracted automatically or semi-automatically from the raw data. We are currently focusing on features such as local minima and maxima that are easy to derive automatically from ECMWF fields. In the medium term, we plan to address the much harder problem of extracting high-level features from satellite imagery.

MetVUW Workbench is a case retrieval system as opposed to a full case-based reasoner, so the only high-level features of meteorological situations that are represented in its knowledge base are the index labels for cases. The rest of each case consists of raw satellite imagery, ECMWF data, and text. It is important to emphasise, however, that index labels typically provide much richer descriptions of meteorological situations than the features of any particular query. Once a case is retrieved, its entire index label is available and can provide useful summary descriptions of the case, or can contribute to a summary description of a set of retrieved cases.

For example, a forecaster might use MetVUW Workbench to obtain information on how rapidly a particular frontal system is likely to approach New Zealand. To accomplish this, the forecaster would first issue a query describing both the current position of the relevant front, together with key features of the current situation that seem likely to affect the front's movement. The forecaster would then request a frequency distribution that summarises the rate of eastward progress of the corresponding front in each of the retrieved cases.

The remainder of this paper is structured as follows. Section 2 describes the architecture of MetVUW Workbench and identifies four kinds of evaluation for MetVUW Workbench and other case retrieval systems with this architecture. Section 3 sketches MetVUW Workbench's representations for index labels. Section 4 describes similarity assessment. Section 5 introduces important issues in case selection in MetVUW Workbench. Section 6 reports some preliminary evaluations of algorithms for case selection, and explains how they have contributed to a process of iterative design and evaluation. Section 7 concludes with a brief discussion of future work.

2 System Architecture and Evaluation

The appropriate role of evaluation in the design of a case retrieval system is in part a function of the architecture of the retrieval engine. In this section, we describe the architecture of MetVUW Workbench, and then use this description to identify four kinds of evaluation that are appropriate for this architecture.

2.1 The architecture of MetVUW Workbench

Figure 1 depicts the architecture of MetVUW Workbench. The user interacts with the system via a *query constructor*, which allows meteorologists to formulate queries in terms of high-level descriptions of the meteorological systems and processes that appear to be relevant in the current situation. Users can describe high-level features of past weather situations. These features include low and high pressure systems, ridges, troughs, and jet streams. Eventually, it should also be possible to represent properties of these objects as they develop over time, such as the track of the pressure minimum of a low-pressure system.

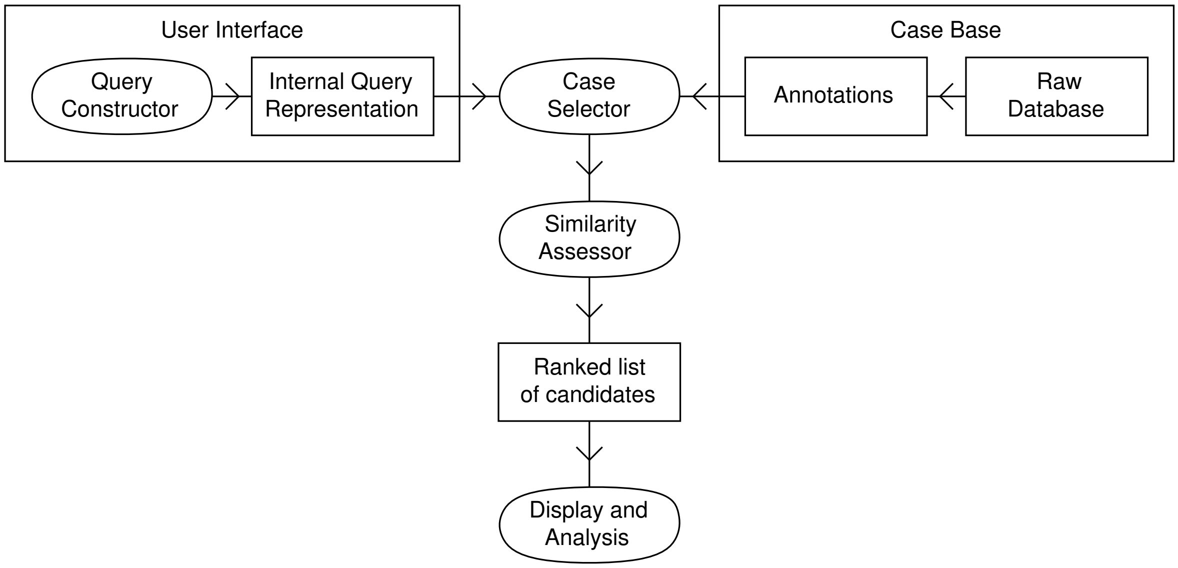


Figure 1: Architecture for case retrieval

Users construct queries by drawing representations of meteorological phenomena on a rectangular canvas that depicts a region of the globe. These phenomena are represented by points, lines, and regions, using a notation similar to an abstract weather map. This kind of interface was chosen so that the components of queries can be entered quickly and easily using a notation that is familiar to forecasters.

A sample query is shown in figure 2. The query requests a low-pressure system with an upper-level trough situated to the northwest at a distance of 400–600km. The "-30mb" marking indicates that there is a 30mb pressure difference between the minima so labeled and the perimeter of the outermost contour of the enclosing low pressure system.

After a query is constructed, a symbolic representation of it is passed to the *case selector*, which uses key features of the query as indices for retrieving past situations or cases. The purpose of case selection is to allow the time it takes to retrieve a given number of cases to scale sub-linearly with the size of the case base. In particular, case selection cannot simply scan the entire case base.[1]

Case selection presupposes a suitable memory organization for index labels, and a retrieval mechanism that exploits this memory organization to efficiently access cases whose labels match a query. As we discuss in more detail below, MetVUW Workbench uses an advanced relational database called Postgres to implement case selection (Stonebraker and Rowe, 1986). This "heavy duty" machinery was chosen so that the system can readily scale up to a very large case base. Case selection is efficient

[1] We assume serial computer hardware.

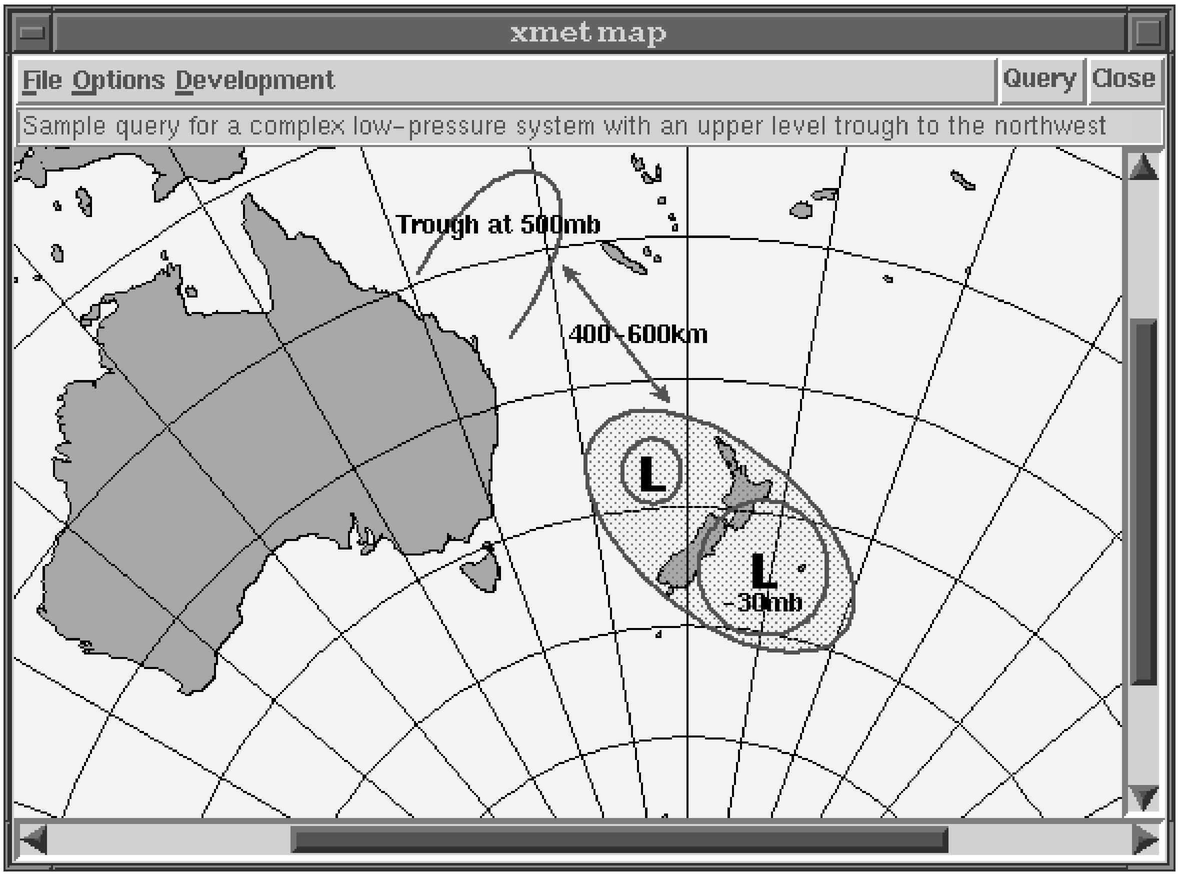


Figure 2: A query

because it employs a number of sophisticated indexing strategies provided by Postgres, including R-tree indexing of spatial data (Guttman, 1984).

Once a collection of past cases has been retrieved, they are passed to the *similarity assessor*, which uses a knowledge-intensive partial matching process to rank them according to how well they match the query. Those cases whose match quality falls below some threshold are discarded.

Case retrieval thus proceeds in two stages. First, case selection uses efficient retrieval machinery—in MetVUW Workbench, a sophisticated relational database—to quickly identify a manageable number of candidate cases that need to be explicitly considered. Second, these cases are further filtered and ranked using a more costly but more accurate process of similarity assessment. This combination of case selection and similarity assessment facilitates fast, high quality retrieval.

It is important to distinguish two notions of retrieval that arise in this architecture: case retrieval itself, and what we are calling case selection. Each has its own representation vocabulary for index labels. Case selection employs simple labels that meet the often strict constraints of its efficient retrieval mechanisms. In contrast, case retrieval employs a richer representation vocabulary that also supports similarity assessment.

For example, Postgres' R-tree indexing mechanism allows efficient retrieval of rectangular regions oriented along lines of latitude and longitude that are spatially related in some way to a similarly oriented rectangular region in a query. Spatial relations include contains, contained-in, and overlaps. Because R-tree indexing is available, queries in MetVUW Workbench employ rectangular bounding boxes of low and high-pressure systems as index labels for case selection. Moreover, queries request cases containing systems that bear one of Postgres' built-in spatial relations to the corresponding system in the query. In contrast, similarity assessment uses arbitrary polygons to represent low and high pressure systems, and computes a more complex (and more informative) relation of graded match between corresponding systems in the query and the case.

2.2 Four kinds of evaluation

The architecture of MetVUW Workbench suggests evaluating the system along two dimensions. Along the first dimension, it is possible to separately evaluate either case selection or similarity assessment components. Along the second dimension, one can evaluate either the effectiveness of each component's representations for index labels or the utility of its algorithms. We now examine each of the four kinds of evaluation that these possibilities give rise to.

Similarity assessment must be evaluated with respect to the requirements of end users of the system. Recall that similarity assessment produces a ranked list of cases for further processing. Evaluation of similarity assessment focuses on the following questions:

1. *Representation vocabulary.* Do the index labels encode representational distinctions that matter to the user?

2. *Algorithms.* Do computations of graded match between queries and cases produce an appropriate ranking of the cases?

Answering these questions typically requires detailed consultations with users and domain experts. We do not consider evaluation of similarity assessment further in this paper.

Case selection, in contrast, can often be evaluated without consulting users or experts, assuming that appropriate procedures for similarity assessment are already in place. Case selection exists for efficiency reasons only: if similarity assessment were adequately efficient, or scaling considerations unimportant, then case retrieval could be implemented simply by applying similarity assessment to the entire case base and returning those cases that well match the query. Let us call the list of cases produced in this way the *benchmark list* (for a given query).

Benchmark lists of cases produced by similarity assessment provide a convenient standard for evaluating case selection. A procedure for case selection cannot be faulted if when using this procedure, case retrieval successfully returns exactly the benchmark list.

Like similarity assessment, case selection can be evaluated with respect to both index representations and algorithms:

3. *Representation vocabulary.* Are the index labels sufficiently expressive? Are they liberal enough to permit retrieval of most cases judged to be a good match by similarity assessment? Are they conservative enough to filter out most of the irrelevant cases?

4. *Algorithms.* Does the time complexity of case selection scale well with the size of the database?

In contrast to similarity assessment, however, questions of efficiency are more central to case selection. This is not to imply that efficiency is not also important for similarity assessment. However, the quality of case selection directly impacts the efficiency of the entire retrieval system: if case selection retrieves many irrelevant cases, then similarity assessment will take much longer, because it must scan all of the cases that case selection returns.

Two standard evaluation metrics from the information retrieval literature are useful in evaluation of case selection (Jacobs, 1992):

- The *precision* of an information retrieval system is the percentage of items retrieved in response to a query that are good answers to the query.
- The *recall* of an information retrieval system is the proportion of good answers to a query that the system retrieved, expressed as a percentage of the total number of good answers that exist in the entire database.

In the context of case selection, precision measures the percentage of candidates returned by case selection that are contained in a query's benchmark list. In contrast, recall measures the proportion of candidates returned by case selection that are in the benchmark list.

For a wide variety of retrieval problems, there is a tradeoff between recall and precision. A given algorithm for case selection may be able to achieve a high level of recall, but in general this requires sacrificing precision. The extreme case is to retrieve the entire case base, which achieves 100% recall at the price of very low precision.

It follows that a good measure of the overall quality of a case selection algorithm is the precision it achieves for a given (acceptable) level of recall. This measure is especially useful for our purposes because it correlates closely with the time efficiency of the overall case retrieval system.

In the next section, we outline some of MetVUW Workbench's representations for index labels and algorithms for case selection and similarity assessment. We then report on some initial experiments concerning the evaluation of case selection in MetVUW Workbench.

3 Representing Queries and Index Labels

Queries are constructed using graphical objects such as points, vectors, and regions that denote high-level meteorological features. There are three kinds of features: static, dynamic, and relational. Static features describe meteorological phenomena at particular points in time. Examples include the centre and extent of a low pressure system, the orientation of a ridge or trough, and wind direction. Dynamic features encode properties of these phenomena as they vary over time, such as the path followed by a low pressure system or whether a trough is intensifying. Relational features encode spatial constraints between features.

Corresponding to each graphical object in a query there is an underlying symbolic representation that is used in case retrieval. Index labels are also constructed using the same representation vocabulary. This paper focuses on static features and relational features of high and low pressure systems.

In designing a suitable representation for high and low pressure systems, it is important to identify those aspects of their shape and relational structure that experts consider to have important meteorological implications. Only if these aspects are explicit in the representation can case selection and similarity assessment produce useful results. We now discuss each of these aspects.

3.1 Shape of high and low pressure regions

Regions of high pressure have more interesting shape properties than regions of low pressure. In particular, forecasters consider the orientation of the ridges of high pressure systems to be important, because the presence or absence of a ridge tends to influence the weather at that location. Figure 3 shows two high pressure systems with much the same location and extent but with differently oriented

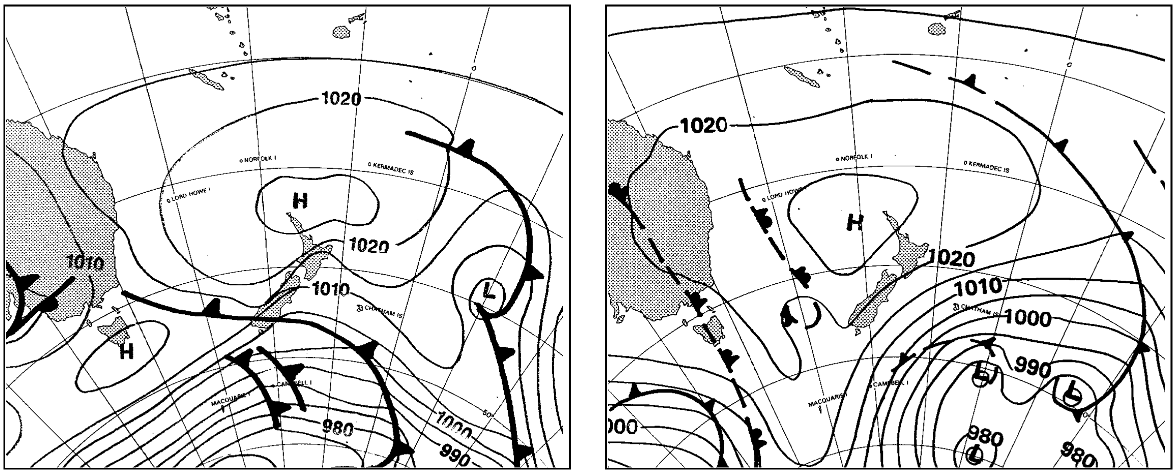


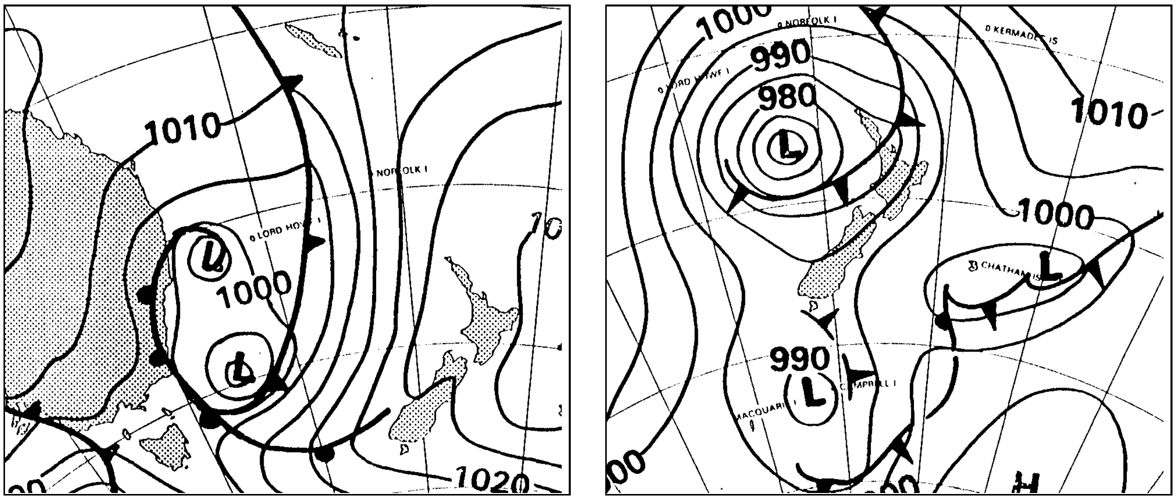
Figure 3: Two high pressure systems with different shapes

ridges. To capture this shape information, high pressure systems are encoded using a pinwheel of convergent axes that represents their ridges. Each axis is labelled by its orientation and a measure of its "curvature" or how well-defined the ridge is.

Individual regions of low pressure have fewer interesting shape properties, and are generally close to circular or elliptical. Only the geographical extent of these regions and the location of their pressure minima are considered meteorologically significant by forecasters. In contrast to high-pressure regions, however, low pressure regions are often clustered together into groups of two or more that overlap, as we now discuss.

3.2 The structure of complex low-pressure systems

Figure 4 illustrates two complex low-pressure systems. The entire region covered by such a system is likely to experience adverse weather, especially near centres of low pressure. It is therefore important to encode the shape of these systems in a way that makes clear the location of regions of low pressure.



Midnight, 4 August, 1990 NZST | Midnight, 17 August, 1990 NZST

Figure 4: Two complex low pressure systems

Complex low-pressure systems are represented as *clusters* of simple low-pressure regions. A cluster is a tree whose leaves are representations of simple low pressure systems—systems with a single low-pressure minimum—and whose internal nodes are low-pressure regions representing smaller clusters. A cluster is an ancestor of another cluster in the tree if and only if there is a pressure contour associated with the ancestor that completely encloses the contours of its descendant. This tree structure is used during similarity assessment to compute a measure of the degree of structural match between a query and a case.

3.3 Representing other spatial relations

Clusters encode one kind of meteorologically significant spatial relation between weather systems. However, further kinds of spatial relations are also important. In particular, the position of two weather systems relative to one another is often very significant, especially when one of them exerts a causal influence over the other. For example, a so-called *blocking high* can impede the eastward progress of a low pressure system lying to the west of it; similarly, an upper level pressure trough can act to intensify a low-pressure system to its southeast.

It should therefore be possible for queries to specify relevant spatial constraints between the objects in a query; moreover, the system should supply sensible defaults. Defaults can be supplied by providing a set of simple "conceptual models" or schemas that encode causal relations between systems in a certain spatial configuration. One conceptual model, for example, encodes the idea of "blocking high."

While this idea has not yet been implemented, it is proposed that recognition of such schemas be triggered from the symbolic query representation in a forward chaining manner. Schema recognition would cause the system to automatically suggest augmenting a query with one or more new spatial relations. Spatial relations can then guide both case selection and similarity assessment.

3.4 Encoding spatial representations in Postgres

All representations of index labels in MetVUW Workbench are automatically extracted from the ECMWF data and stored in the Postgres relational database. The relation scheme depicted in figure 5 is employed. Each case is allocated a unique identifier, as is each region of low pressure. Clusters are built up out of individual regions of low pressure; the tree structure of a cluster is encoded using the "Parent Id" field of the "Region" relation. Regions of high pressure are also stored using the "Region" relation, except that their axes are stored in a separate relation called "Axis." Other high-level features of cases such as troughs are encoded using further relations similar to "Region" and "Axis."

Regions of low pressure are used to represent both the simple low pressure systems at cluster leaves and the enclosing contours that make up the internal nodes of the tree. These contours are chosen to be as large as possible without sacrificing cluster structure. For each region, the location and mean sea-level pressure of the pressure minimum are stored, together with a high-resolution polygon that represents its outer boundary. Case selection works with a rectangular bounding box that approximates this polygon, but similarity assessment can access the full polygon to compute such quantities as degree of overlap with a region in the query. The mean sea-level pressure at the polygon boundary is also stored. The representation of a typical case is depicted in figure 6. (For brevity, the Perimeter field is omitted from the Region relation, although it is displayed graphically.)

Case
Case Id
Time and Date
Extent
Description

Region
Region Id
Case Id
Parent Id
Classification
Bounding Box
Perimeter
Perimeter Pressure
Pressure Centre
Minimum Pressure

Axis
Axis Id
Region Id
Case Id
Orientation
Curvature

Figure 5: Relation scheme for low-pressure systems

4 Similarity Assessment

The similarity assessor is handed a query and a set of cases retrieved by the case selector, and sorts the cases by their degree of partial match to the query. Cases that match too poorly are discarded.

Developing appropriate representations and algorithms for similarity assessment requires exploring a space of design decisions. It is useful to divide similarity assessment into two subtasks: assessing similarity between individual low pressure regions in a query and a case, and using cluster structure to combine those assessments into an overall assessment of goodness of match. We now consider each of these subtasks in turn.

4.1 Assessing similarity of individual low-pressure regions

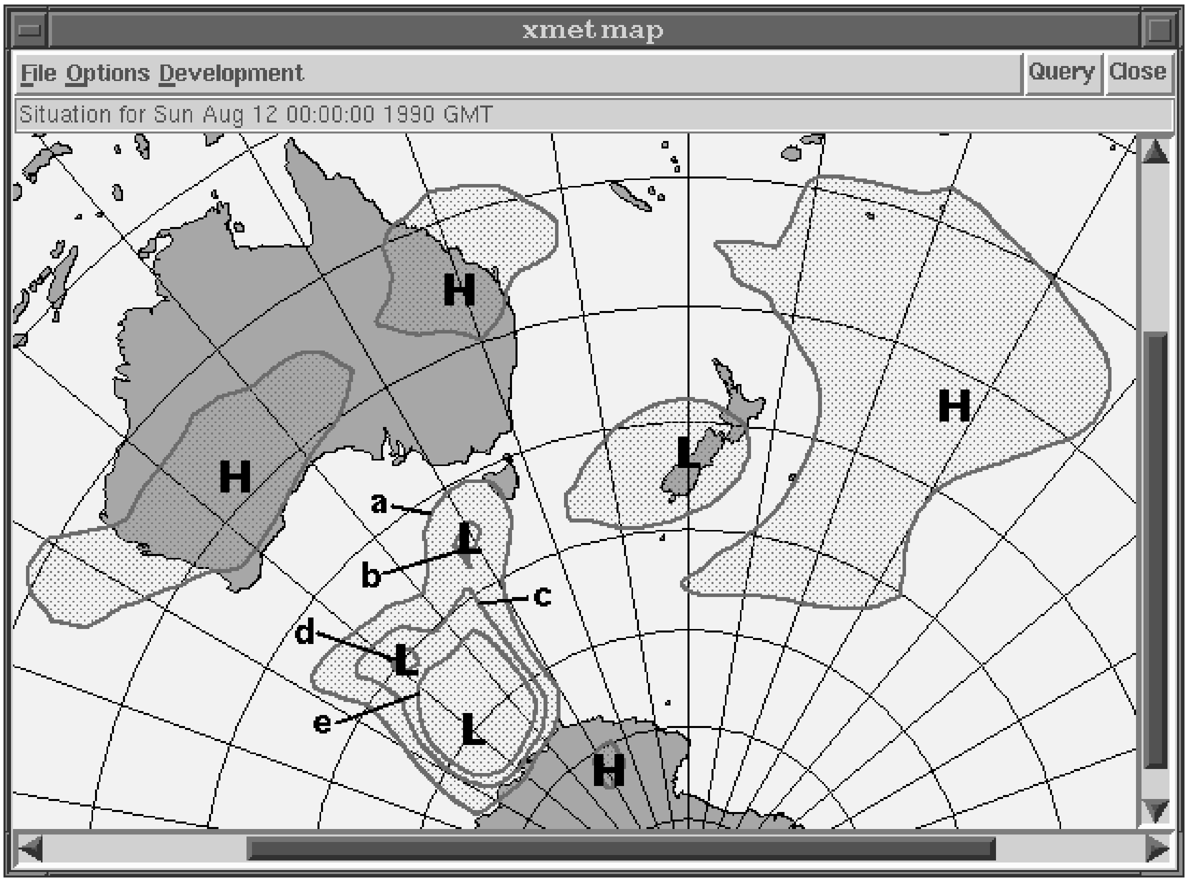
For purposes of pairwise matching, similarity assessment makes no distinction between regions internal to a cluster and the simple low pressure systems at the cluster's leaves. To compute the similarity between a low pressure region in a query and a corresponding region in a case, MetVUW Workbench computes the following five parameters.

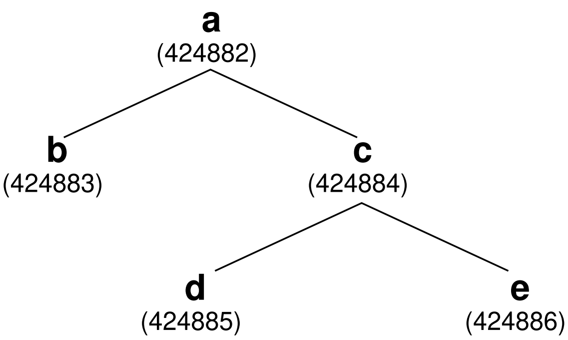
1. *Relative location of pressure minimums.* A measure of the displacement between the pressure minimums of a region in the query and a region in the case is computed. Longitudinal (east-west) displacements are usually of less concern than latitudinal (north-south) displacements, because weather systems tend to naturally progress from west to east, and small displacements in this dimension can be attributed to infrequency of sampling. Latitudinal displacements, on the other hand, are penalized more heavily because the observed weather at a given location is considerably affected by such deviations, and because the physical behaviour of weather systems is strongly conditioned by their latitude.

2. *Aspect ratio.* The aspect ratio of a low pressure system is the ratio of its lengths of the sides of its bounding box. Aspect ratio gives a cheap measure of the shape of a low pressure region.

3. *Density.* We define the density of a low pressure region to be the ratio of the area of its high-resolution polygonal approximation to the area of its bounding box. Density gives a second inexpensive measure of region shape.

4. *Aligned overlap.* The pressure minima of two low pressure systems to be compared are aligned, then the ratio of the area of the overlap between their bounding boxes to the combined area they cover is computed. This ratio gives a third way of computing similarity of shape.





Case Id	Time and Date	Extent	Description
424876	Aug 12 00:00:00 1990 GMT	(100.0,-80.0,210.0,-10.0)	automatically created situation for Sun Aug 12 00:00:00 1990 GMT

Reg. Id	Case Id	Parent Id	Class	Bounding Box	Peri. Pres.	Pres. Centre	Min Pres.
424877	424876	—	High	(168.7,-56.5,209.7,-17.4)	1016.5	(197.8,-32.6)	1017.0
424878	424876	—	High	(142.0,-29.8,159.2,-17.7)	1017.5	(149.7,-25.2)	1018.0
424879	424876	—	High	(136.3,-74.4,146.6,-69.8)	1018.0	(140.9,-72.4)	1018.5
424880	424876	—	High	(104.4,-33.4,138.1,-19.0)	1018.0	(123.7,-27.8)	1018.5
424881	424876	—	Low	(152.9,-49.5,177.6,-37.9)	998.5	(169.9,-42.4)	998.0
424882	424876	—	Low	(104.2,-67.2,145.8,-40.4)	994.0	(130.1,-50.0)	993.5
424883	424876	424882	Low	(137.2,-47.5,141.3,-43.6)	988.5	(139.6,-44.9)	988.0
424884	424876	424882	Low	(110.2,-66.1,136.2,-46.8)	988.5	(120.6,-55.0)	988.0
424885	424876	424884	Low	(119.4,-51.5,123.2,-49.5)	985.0	(121.3,-50.1)	984.5
424886	424876	424884	Low	(111.8,-65.4,134.5,-52.7)	985.0	(120.0,-60.0)	984.5

Figure 6: Representation of a typical case.

5. *Area.* The three previous parameters all facilitate comparisons of shape independent of a region's size. However, the relative size of low pressure regions to be compared is also important, so the area of the high-resolution polygonal approximation to each region is also considered as a separate parameter (even though it is also used to compute density).

These parameters are combined to compute a goodness of match between a pair of low pressure regions. Eight numbers are computed in all: the degree of match on relative location of pressure minima, the degree of aligned overlap, and the aspect ratio, density, and area of each of the regions to be compared. The goodness of match is computed as the sum of the first two parameters together with the ratios of the smaller to the larger value for each of the other parameters, using appropriate weights for each summand. Similarity assessment does not currently consider the intensity of low pressure systems; this needs to be fixed.

4.2 Assessing similarity of clusters

MetVUW Workbench computes an overall goodness of match between clusters in a query and a case by assessing both the goodness of match between individual low pressure regions and the degree of structural match between the clusters. Exactly how these assessments should trade off against one another is a difficult issue, and we are still experimenting with different alternatives. MetVUW Workbench currently employs a form of breadth-bounded depth-first search with successor ordering (Pearl, 1984) that appears to strike an appropriate balance.

We enforce the following structural constraints on the possible sets of correspondences between low pressure regions of a query and a case. First, the relation that the correspondences define must be one-to-one. That is, no region in a query can correspond to more than one region in the case, and no region in the case can correspond to more than one region in the query. Second, if a region R_q in the query corresponds to a region R_c in a case, then no second of R_q may correspond to an ancestor or sibling of R_c, and no descendant of R_c may correspond to an ancestor or sibling of R_q. A correspondence that violates these constraints is said to be *structurally inconsistent* with the correspondence (R_q, R_c).

An informal description of the matching algorithm is as follows. First, the goodness of match between each pair of low pressure regions in the query and the case is computed. Pairs whose similarity is beneath a certain threshold are not considered further in the search. The remaining pairs are sorted according to their goodness of match and stored on an ordered list L.

Next, a breadth-bounded backtracking search is performed through the space of possible correspondences between regions in the query and regions in the case. At each stage of the search, the best remaining pair (R_q, R_c) on L is selected and used to extend the current partial match. All pairs that are structurally inconsistent with (R_q, R_c) are removed from L. The search proceeds in a depth-first manner, bottoming out when L is empty. Each terminal node of the search represents a possible best match (set of pairs) between the query and the case. Upon backtracking, the previous value of L is restored (except that the pair most recently used to extend that node of the search is discarded), and the next best remaining pair on L is used to extend the partial match. At each level of the search, only the k best pairs are considered, for some small integer k. The parameter k constrains the breadth of the search, and can be tuned as needed. A small k produces a very fast search, but risks not finding a good match.

5 Case Selection

High quality case selection requires a choice of indices that achieve good recall without unduly sacrificing either precision or speed of case selection itself. This requires finding indices that closely mimic the behavior of similarity assessment. However, the available indices are constrained by the kinds of index mechanisms that Postgres provides.

Two kinds of design decision must be made to arrive at a procedure for case selection:

1. *Representation choice.* Which features of the full index label should case selection be sensitive to? For example, should case selection be sensitive to cluster structure or not? Should case selection consider the pressure minimum of a low pressure region, or the region's geographic extent, or both?

2. *Adjustable parameters.* If geographic extent is considered, then Postgres' R-tree indexing mechanism must be employed to achieve adequate efficiency, with its restriction to rectangular regions. How should suitable dimensions for rectangular boxes in a Postgres query be computed from the low-pressure regions present in a user's query? For example, should bounding boxes for these regions be employed, or larger rectangles? As we will see below, larger rectangles turn out to yield better performance. A parameter describing exactly how much larger to make these rectangles needs to be adjusted on the basis of empirical data.

In the next section, we describe several alternative approaches to case selection in more detail.

6 Evaluation

It should be clear from the discussion so far that developing representations and algorithms for case selection and similarity assessment necessarily involves a large number of design decisions, many of which should only be made on the basis of empirical evaluation.

There is an important sense in which designing a case retrieval system requires a kind of hill-climbing search, guided by empirical considerations. At each stage in the search, a range of possible choices is considered and evaluated, and the best one selected. To illustrate this process, we next briefly describe two preliminary experiments regarding retrieval of low pressure systems that we carried out as part of developing MetVUW Workbench's current algorithm for case selection.

Experiment 1: Representation choice for case selection

We evaluated three possible index representations for case selection involving queries that mention only low-pressure systems. These index representations are summarized in figure 7. The third column exemplifies actual queries to Postgres that case selection constructs in response to a query from the user involving a single low pressure system. Queries to Postgres are represented in Postquel, Postgres' query language.

Five representative queries were chosen and tested using each index representation. The average precision and recall of each was computed with respect to the "benchmark set" obtained by applying similarity assessment to the entire case base. The results of our analysis are presented in figure 8.

It is instructive to recount the history of design decisions that led us to consider these choices of index representation for case selection. Throughout the design process, our aim was to maximize precision without unduly sacrificing recall.

Method	Description	Sample Postquel query
1	The bounding boxes overlap	retrieve (r.situation) from r in region where r.classification = "low"::char16 and r.bounds && "(162.8,-53.8,183.6,-29.3)"::box
2	The pressure minimum of one system is contained in the bounding box of the other	retrieve (r.situation) from r in region where r.classification = "low"::char16 and (r.centre $\longrightarrow$ "(162.8,-53.8,183.6,-29.3)"::box or "(173.1,-41.4)"::point $\longrightarrow$ r.bounds)
3	Both the pressure minima are within the overlap of the bounding boxes	retrieve (r.situation) from r in region where r.classification = "low"::char16 and r.centre $\longrightarrow$ "(162.8,-53.8,183.6,-29.3)"::box and "(173.1,-41.4)"::point $\longrightarrow$ r.bounds

Figure 7: Index representations

Method	Precision	Recall
1	0.14	0.80
2	0.22	0.60
3	0.58	0.28

Figure 8: Results of experiment 1.

We started with the most simple-minded approach possible: retrieve a case in response to a query if each low pressure system in the query overlaps a low pressure system in the case (method 1 in figure 8). We quickly discovered that this strategy produces quite low precision. We therefore tried incorporating pressure minima into case selection in an effort to improve precision. As can be seen from figure 8, quite high precision was obtained for method 3 but at the cost of unacceptably low recall.

The results of this first experiment led us to explore new ways to improve recall while maintaining precision. Our next strategy was to modify methods 2 and 3 to retain sensitivity to pressure minima, but allow a degree of latitudinal and longitudinal displacement between systems in a query and corresponding systems in a case. Our rationale was as follows: similarity assessment also permits a certain amount of displacement, so if we also allow a little displacement in queries, case selection should mimic similarity assessment more closely. It is therefore reasonable to expect an improvement in recall without incurring a huge penalty for precision.

Experiment 2: Parameter choice—geographic displacement

In this experiment, we made two changes to the Postgres queries constructed by case selection to permit geographic displacements between low pressure systems in queries and cases. First, we introduced two new adjustable parameters *Lat* and *Long* that specify amounts to expand rectangular bounding boxes of low pressure systems in queries in the north-south and east-west directions respectively. The new queries permit a degree of geographical displacement between regions in queries and corresponding regions in cases. Second, we replaced each pressure minimum in the query with a rectangular region whose dimensions are *Lat* and *Long*, and we replaced each test of point containment involving this minimum with a corresponding test of overlap with the new region. It is easy to see that this partially relaxes the constraint that pressure minima in the query must be contained in a low pressure system in the case.

Method 2

Lat Long	**0°**	**4°**	**8°**
0°	0.60 0.22	0.68 0.18	0.75 0.14
5°	0.73 0.20	0.82 0.13	0.87 0.10
10°	0.82 0.17	0.90 0.10	−− −−
15°	0.82 0.14	−− −−	−− −−

Method 3

Lat Long	**0°**	**4°**	**8°**
0°	0.28 0.58	0.32 0.31	0.35 0.27
5°	0.45 0.56	0.53 0.37	0.57 0.27
10°	0.48 0.48	0.60 0.34	0.65 0.21
15°	0.53 0.41	0.68 0.28	0.73 0.18

Lat = Permitted latitudinal displacement
Long = Permitted longitudinal displacement

Figure 9: Results of experiment 2 (recall over precision).

We tried three different settings for *Long* and two different settings for *Lat*. The results are summarized in figure 9. We did not carry out tests for cells whose expected precision was less than 10%. The results for method 2 are largely negative: there appears to be a straightforward and relatively uninteresting tradeoff between precision and recall, and no choice of values for *Lat* and *Long* seems to yield results clearly superior to the methods of experiment 1. It appears that the requirement that only one pressure minimum be approximately contained in the bounding box of the region it is matched against does not sufficiently constrain case selection to produce good results.

The results for method 3 are more encouraging. Relatively high levels of both precision and recall are achieved for a number of settings for *Lat* and *Long*. For example, the setting $\{Lat = 4°, Long = 10°\}$ achieves the same recall as method 2 of experiment 1, but with significantly better precision.

7 Conclusions and Future Work

The two experiments describe above only consider a small region of the space of design decisions that have gone into MetVUW Workbench's case retrieval system. For example, we have not discussed evaluation of similarity assessment. Nevertheless, these experiments well illustrate the utility of a process of iterative design. More generally, we believe that the large number of decisions that go into the design of a case retrieval system, and the difficulty of assessing their impact *a priori*, make the design process an enterprise with a necessarily empirical component.

Earlier we used a metaphor of hill-climbing search to describe the design process. We are actively considering whether it might be possible to take this metaphor literally, and automate part of the design-evaluation cycle. Design decisions can be divided into two categories: choices of appropriate representational distinctions and "parameter tweaking," in which a good value for a numeric parameter is selected. Decisions as to representational distinctions cannot be easily automated, as formulating the available options appears to require both human intuition and AI expertise. Choosing appropriate values for numerical parameters, however, may be easier to automate.

In this paper, we have seen a number of adjustable numerical parameters: weights for matching of latitudinal and longitudinal displacement, aspect ratios, density, area, and the like when matching regions of low pressure; width of search in structure matching; and degree of allowed latitudinal and longitudinal displacement in case selection. Perhaps hill-climbing techniques from machine learning can be used to explore this parameter space in an attempt to optimize the choice of parameter values? We intend to explore this idea in our future research.

Acknowledgements

This research was carried out in collaboration with James McGregor of the Institute of Geophysics at Victoria University of Wellington. Thanks to Linton Miller for commenting on a draft of this paper, and to Erik Brenstrum of the Meteorological Service of New Zealand for useful discussions regarding the meteorologically significant features of low and high pressure systems. The Meteorological Service of New Zealand provided the situation maps in figures 3 and 4.

References

Domeshek, E. A. (1991). Indexing stories as social advice. In *Proceedings, Ninth National Conference of Artificial Intelligence*, volume 1, pages 16–21. AAAI Press.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM-SIGMOD Conference on Management of Data*, Boston, Massachusetts. ACM-SIGMOD.

Hunter, L. E. (1989). *Knowledge Acquisition Through Experience*. PhD thesis, Yale University.

Jacobs, P. S., editor (1992). *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates.

Jones, E. K. and Roydhouse, A. (1994). Intelligent retrieval of historical meteorological data. To appear in *AI Applications*. Also available as Technical report CS-TR-93/8, Victoria University of Wellington. URL: ftp://ftphost.comp.vuw.ac.nz/doc/vuw-publications/CS-TR-93/CS-TR-93-8.ps.gz.

Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, Reading, MA.

Roydhouse, A., Miller, L., Jones, E., and McGregor, J. (1993). The design and implementation of MetVUW Workbench version 1.0. Technical Report CS-TR-93/7, Victoria University of Wellington. URL: ftp://ftphost.comp.vuw.ac.nz/doc/vuw-publications/CS-TR-93/CS-TR-93-7.ps.gz.

Schank, R. C. (1982). *Dynamic Memory*. Cambridge University Press, Cambridge, England.

Schank, R. C. and Fano, A. (1992). A thematic hierarchy for indexing stories in social domains. Technical report, The Institute for the Learning Sciences, Northwestern University.

Stonebraker, M. and Rowe, L. (1986). The design of Postgres. In *Proceedings of the 1986 SIGMOD Conference on Management of Data*. ACM Press.