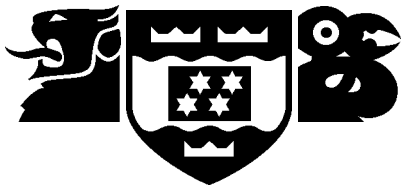
VICTORIA UNIVERSITY OF WELLINGTON



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

Intelligent Retrieval of Historical Meteorological Data

Eric K. Jones
Aaron Roydhouse

Technical Report CS-TR-93/8
November 1993

Abstract

Large repositories of historical data about the natural world present novel opportunities to assist decision makers and problem solvers. Past cases or scenarios that are similar to a problem of current interest can often provide valuable insights or suggest solutions to the problem.

We are constructing a case retrieval system that is intended to give meteorologists fast access to relevant past weather situations. The system is to serve as a "memory amplifier," allowing users to rapidly locate historical situations of interest. In particular, forecasters should be able to use the system to quickly retrieve situations that are similar to the current one in meteorologically significant respects, providing them with an additional source of information to supplement the output of numerical models. The case retriever is part of MetVUW Workbench, a system for intelligent retrieval and display of historical meteorological data.

In designing the MetVUW case retriever, we have addressed challenging problems in three areas: representation of spatial and temporal knowledge, case selection, and similarity assessment. We have also identified a number of lessons for the design of intelligent databases in natural resource domains.

Publishing Information

This report also appeared in *Proceedings of a Workshop on Artificial Intelligence and the Natural World*, Melbourne, Australia, November 1993, and has been submitted to *AI Applications*.

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.

Introduction

Large repositories of historical data about the natural world present novel opportunities to assist decision makers and problem solvers. Past cases or scenarios that are similar to a problem of current interest can often provide valuable insights or suggest solutions to the problem. This paper focuses on the intelligent retrieval of historical meteorological data to assist weather forecasters and other meteorologists.

We are in the process of constructing a case retrieval system that is intended to give meteorologists fast access to relevant past weather situations. The system is to serve as a "memory amplifier," allowing users to rapidly locate historical situations of interest. In particular, forecasters should be able to use the system to quickly retrieve situations that are similar to the current one in meteorologically significant respects, providing them with an additional source of information to supplement the output of numerical models. The case retriever is part of MetVUW Workbench, a system for intelligent retrieval and display of historical meteorological data (Jones and McGregor, 1993).

Each case is a slice of time (or a sequence of time slices) for which meteorological data is available. MetVUW Workbench currently supports the following types of data: laser disc video imagery, digital satellite imagery, time-tagged text descriptions, and numeric fields from the European Centre for Medium-range Weather Forecasting (ECMWF). Examples of numeric fields include pressure, temperature, relative humidity, wind speed, and relative vorticity. Existing components of MetVUW Workbench allow this information to be retrieved by date and time, full text search, or by scanning the laser disc. The case retriever is intended as an additional search mechanism for retrieving past situations in terms of high-level descriptions of their content.

It is important to emphasize that MetVUW Workbench is intended to empower rather than replace a human expert. Forecasters at the Meteorological Service of New Zealand (MSNZ) currently engage in a "map session" every day at 10am, at which they compare competing prognoses to arrive at a consensus forecast. We expect that MetVUW Workbench system will provide valuable input to these discussions. We envisage that forecasters will follow three steps when using the case retriever:

1. Determine the meteorological processes and systems that appear to be driving the current weather situation.
2. Formulate a query to the case retriever in terms of these processes and systems.
3. Evaluate and display the cases that the system returns. These past cases can be used to provide evidence for or against the prognosis of numerical models, to suggest possible outcomes that the forecasters might not have considered, or to help adjudicate between alternative forecasts.

For the case retriever to be useful to forecasters, it must satisfy four criteria:

1. **Expressivity.** The query language must be expressive enough to represent all queries of interest. Forecasters should be able to formulate queries in terms of high-level descriptions of the meteorological systems and processes that appear to be relevant in the current situation.
2. **Ease of use.** Queries specify fairly elaborate patterns of temporal and spatial information. Forecasters operate under severe time pressure, so a user interface is required that makes it easy to enter this information.
3. **Speed of retrieval.** The system must rapidly retrieve cases in response to queries.
4. **Quality of output.** As far as possible the system should retrieve only cases that closely match the query, and a large proportion of the cases in memory that do match the query should be retrieved.

No existing historical database of meteorological data meets all four of the above criteria, particularly the first. Information is typically only indexed by date and time, which is insufficient to the needs of meteorologists. In many cases, the date and time of items of interest are not known. Forecasters, however, need to be able to retrieve the past situations that are most closely related to the current situation, irrespective of when they occurred. It follows that new methods are required to store historical data in terms of their meteorological content.

Techniques from artificial intelligence and case based reasoning can be usefully applied to three aspects of the design of a suitable retrieval system:

- **Knowledge representation.** The query language must allow users to describe high-level features of past weather situations as they vary in both space and time. It must be possible to conveniently describe objects such as low and high pressure systems, ridges and troughs, and jet streams. It should also be possible to represent properties of these objects as they develop over time, such as the track of a centre of low pressure. A suitable query language must thus address non-trivial problems of spatial and temporal representation.
- **Case Selection.** Not only should it be possible to represent all queries of interest, it must also be possible to quickly retrieve past cases from memory that plausibly match these queries.
- **Similarity assessment.** Past situations seldom exactly match a query; moreover, certain kinds of mismatch typically have much less meteorological significance than others. A smart partial matcher is required to sensibly distinguish good matches from bad, and to rank past situations by goodness of match.

In the remainder of this paper, we describe our approach to the design of intelligent case bases of this kind, focusing on problems that have arisen in the areas of knowledge representation and similarity assessment. The problem of case selection is discussed only briefly.

We begin with an overview of our approach, then we discuss key problems of knowledge representation and similarity assessment. Next, we enumerate some important implementation issues. We conclude with broader morals for the design of intelligent databases in natural resource domains.

Overview

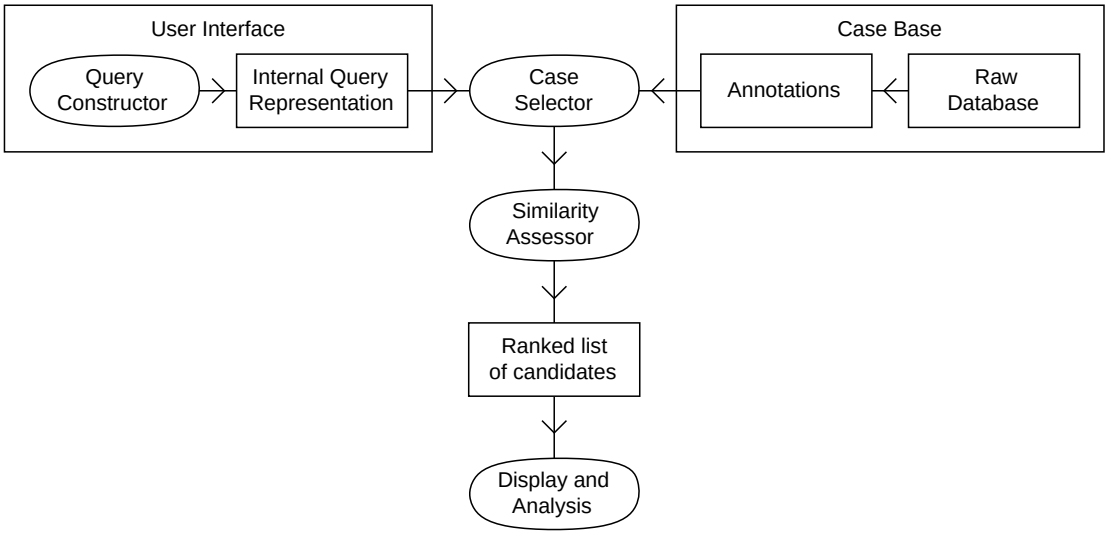


Figure 1: Information flow in the case retriever

The MetVUW case retriever can be thought of as an intelligent front end to a conventional relational database. The database stores all of the system's information about past weather situations: ECMWF fields, satellite imagery, document collections, and so forth. The case retriever provides access to this database by way of queries expressed in a high level vocabulary suited to the needed of forecasters. A block diagram depicting information flows in the case retriever is shown in figure 1.

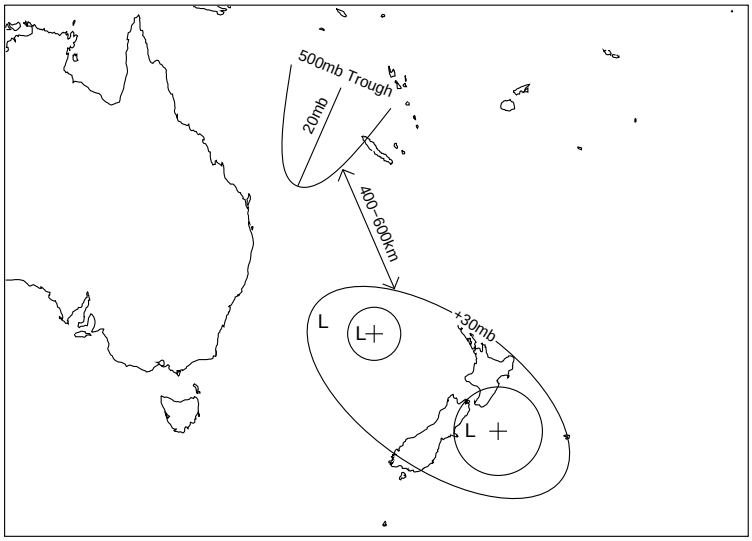


Figure 2: Typical query

Queries are entered using a point-and-click graphical user interface called the *query constructor*. The canvas for constructing queries is an area of the globe depicted by a coastline map and optional lines of latitude and longitude. The user selects from a palette of graphical objects, which can then be placed on the map and sized as appropriate. This kind of graphical user interface meets our criteria of expressivity and ease of use (criteria 1 and 2, above), as it greatly simplifies the entry of spatial and time-varying features of a query. A typical query is depicted in figure 2. The query requests a low-pressure system with an upper-level trough in the pressure field situated to the northwest at a

distance of 400–600km. The "+30mb" marking indicates that there is a 30mb pressure difference between the contour so labelled and the lowest pressure at any point within it.

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 from the relational database. Our basic approach to case selection leans heavily on the underlying machinery of the relational database, as discussed in the section on implementation, below.

Cases are stored in the database in terms of representations of high-level features of the weather situation they describe, in much the same vocabulary as the queries themselves. These representations are computed off line, from the raw ECMWF data and satellite imagery. We call these representations *annotations* (of the raw data). As far as possible, annotations 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.

Once a collection of past cases has been retrieved, the cases 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.

The MetVUW retriever thus implements a two-stage retrieval mechanism. Case selection uses the efficient retrieval machinery of the relational database to quickly identify a manageable number of candidate cases that need to be explicitly considered; these cases are then filtered further using a more costly but accurate process of similarity assessment. This mixture of conventional database retrieval and knowledge-intensive matching techniques results in a retrieval mechanism that is both fast and high quality, meeting the design criteria of speed of retrieval and quality of output (criteria 3 and 4, above).

Representing Queries and Annotations

Queries are constructed using graphical objects such as points, vectors, and regions, which denote high-level meteorological features. These features can be divided into three types: 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 selection and similarity assessment. In this paper, rather than attempting a survey of the full range of available representations, we restrict our attention to some of the hard problems of spatial and temporal representation involved in representing high and and low pressure systems. These problems can be conveniently grouped into four categories: shape, clusters, histories, and spatial relations. We now discuss each of these in turn.

Shape

When matching static descriptions of high and low pressure systems to past cases, it is important to identify those aspects of their shape and geographical extent that experts consider to have important meteorological implications.

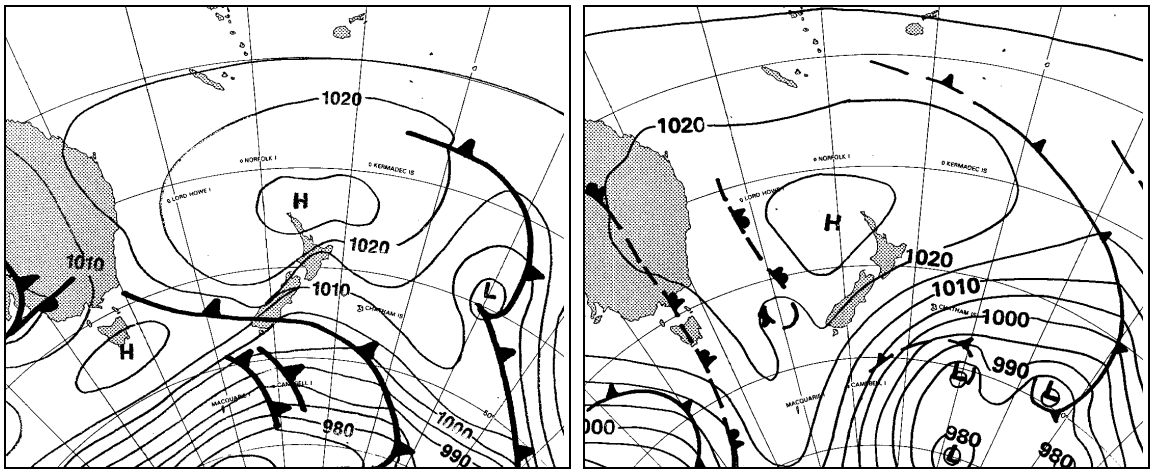


Figure 3: Two high pressure systems with different shapes

It turns out that the relevant aspects of the shape of high pressure systems are quite different from those of low pressure systems. 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 ridges. To capture this shape information, we encode high pressure systems in terms of a pinwheel of convergent axes. Each axis is labelled with a qualitative boundary point describing the outer limit of the system along that axis. These boundary points jointly define a polygonal region that approximates the geographical extent of the high pressure system, as depicted by the shaded area in figure 4.

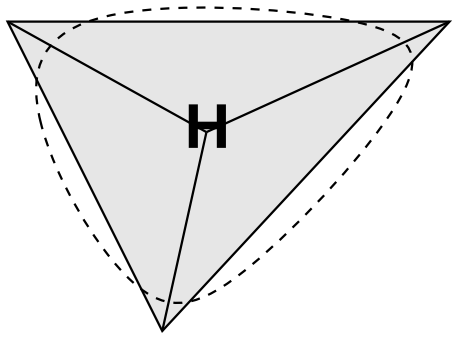


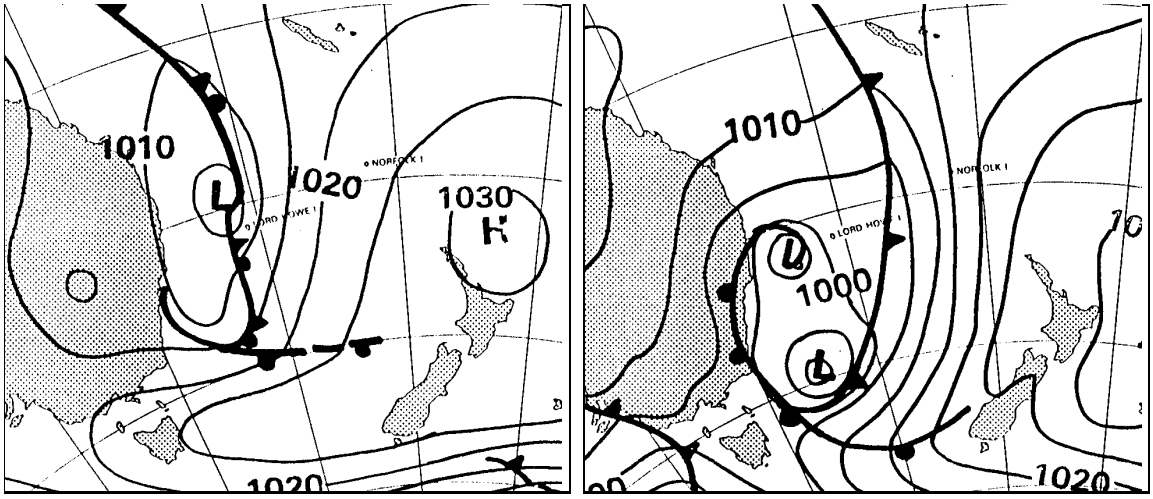
Figure 4: Representation of a high-pressure system

The case selector employs a rectangular bounding box that encloses this polygon to quickly locate cases in memory that plausibly match one another. The similarity assessor uses the polygonal shape representation to help assess the goodness of match between corresponding high pressure systems in a query and a candidate case. This involves computing the sum of the angular displacements between each axis of the high pressure system in the query and the closest-matching axis of the high pressure system in the case, and considering the degree of overlap between the regions enclosed by each system.

Simple low pressure systems—that is, systems with a single pressure minimum—turn out to have less interesting shape properties than high pressure systems. Only the geographical extent of these systems and the location of their pressure minima are considered meteorologically significant by forecasters. Simple low pressure systems are approximately circular, so the graphical user interface depicts them as circles centred on the pressure minimum. The corresponding internal representation used for case selection and similarity assessment is a square that bounds this circle. Although simple low pressure systems have uninteresting shape, they are often clustered together into groups of two or more that overlap. These clusters have non-trivial shape properties, as we now discuss.

Clusters

Clusters of low pressure systems often come into being and evolve in interesting ways. For example, a simple low pressure system may spawn a second subsidiary system (figure 5), or two existing simple systems may join into a single large cluster (figure 6). The entire region covered by a cluster is likely to experience adverse weather, especially near centres of low pressure. It is therefore important to encode the shape of clusters in a way that makes clear the location of regions of low pressure.



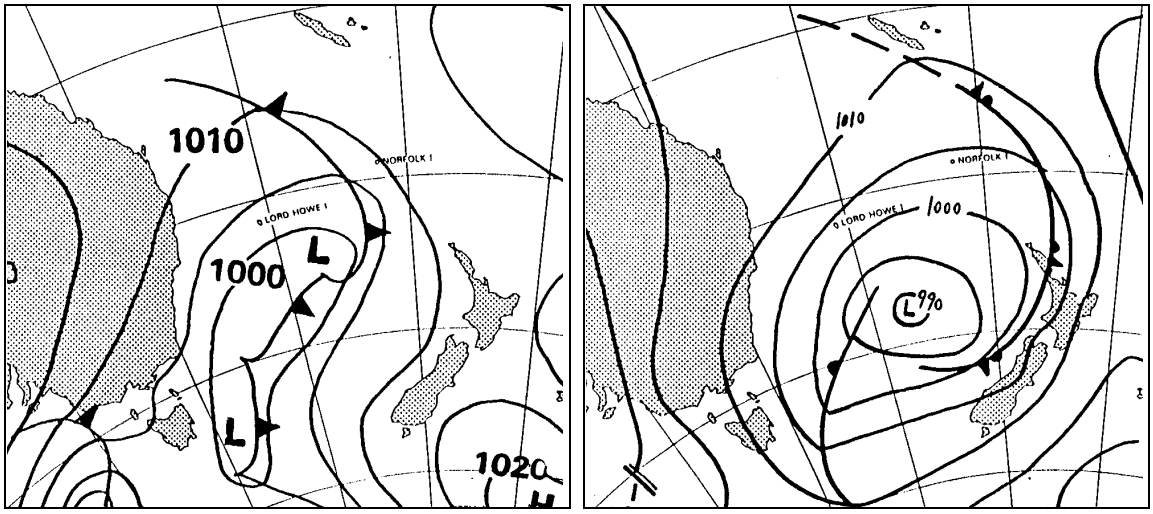
Midnight, 3 August, 1990 NZST Midnight, 4 August, 1990 NZST

Figure 5: Low pressure system over the Tasman Sea spawning a second system

In particular, it is insufficient to simply record each individual low pressure system participating in the cluster and not explicitly represent the cluster at all. Such a representation is poorly suited to the needs of case retrieval: many queries would fail to clearly match a relevant cluster. Consider, for example the query and cluster shown in figure 7. The query (indicated in shaded grey) matches the cluster well in geographical extent, but poorly matches each of its subsidiary low pressure systems. It is therefore important to represent clusters as separate entities in there own right. In fact, we encode simple low pressure systems as a degenerate species of cluster with minimal internal structure.

We are still experimenting with different representations for clusters, but our current scheme is as follows. A cluster is represented as a tree whose leaves are simple low pressure systems and whose internal nodes are 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 former cluster that completely encloses the contours of the latter.

The graphical user interface depicts clusters as arbitrary-shaped ellipses. Internally, however, a simpler symbolic representation in terms of rectangles is employed, to facilitate case selection and similarity assessment. As shown in figure 8, the shape of a cluster is approximated by a rectangle



Midnight, 22 August, 1990 NZST Midnight, 23 August, 1990 NZST

Figure 6: Two low pressure systems joining into one

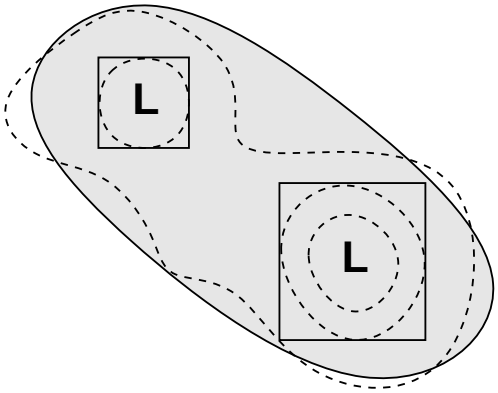


Figure 7: A query (shaded) that poorly matches subsidiary low pressure systems in a cluster

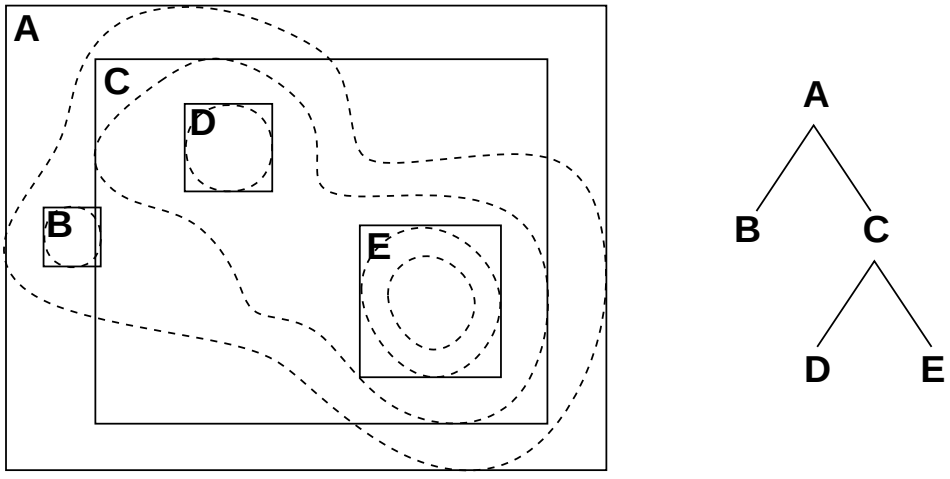


Figure 8: Cluster of low pressure systems approximated by rectangles

aligned horizontally and vertically that bounds the regions covered by subclusters. During case selection, features of the top-level cluster in a query are used as indices for retrieving other top-level clusters from memory. Similarity assessment computes a more detailed match based on the internal structure of the cluster.

Histories

Not only is it important to retrieve state descriptions of high and low pressure systems, it is also necessary to be able to specify patterns of their development through time. For example, it is important to be able to track the movement of centres of low pressure and changes in their intensity. It follows that the annotations of cases in memory must be able to efficiently support queries of this kind.

This is a non-trivial problem, because low and high pressure systems develop and mutate over time. For example, we have seen that low pressure systems frequently spawn subsidiary lows, giving rise to a cluster, while a pair of lows can collapse into a single low pressure system. It is therefore frequently impossible to identify a low pressure system in one image with any single system in a temporally adjacent image.

However, it is usually possible to establish a causal correspondence between weather systems in temporally adjacent data sets. We represent this correspondence by recording *parents* and *children* of each weather system. A parent (child) of a weather system occurring in one data set is a causal predecessor (successor) of that system in a temporally adjacent data set. It is easy to use the resulting causal graph to answer queries about the behaviour of weather systems over time: queries can be answered by finding a suitable path through this graph.

Spatial relations

Queries typically mention more than one meteorological system. In many cases, the position of these systems relative to one another is as important or more important than their absolute location. Relative location of two systems is especially relevant when one of them exerts a causal influence over the other. For example, a so-called *blocking high* can impede the westward progress of a low pressure system to its west, or an upper level pressure trough can act to intensify a low-pressure system to its southeast.

It must 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 dynamic relations between systems in a certain spatial configuration. One conceptual model, for example, encodes the idea of "blocking high." Recognition of such schemas is triggered from the symbolic query representation in a forward chaining manner. Schema recognition causes the system to automatically suggest augmenting a query with one or more new spatial relations. Spatial relations are used to guide both case selection and similarity assessment.

Further Features of High and Low Pressure Systems

In the foregoing, we have discussed only the most challenging problems for the representation of high and low pressure systems, all of which involve representing spatial or temporal properties of these systems. A number of other features of the systems are also represented in addition to ones we have discussed. These include the pressure values as pressure minima and maxima, and the difference in pressure between a system's centre and its boundary.

Similarity Assessment

The similarity assessor is handed a query and a set of cases retrieved by the case selector. Similarity assessment sorts the cases by their degree of partial match to the query; those that match too poorly are discarded. Each case retrieved by the case selector comes with a set of associations between features of the query used to retrieve it and annotations in the case. These correspondences form the starting point for partial matching.

To match a high or low pressure system in a query to its corresponding annotation, a centre point is first determined for each. The centre point for high pressure systems is the location of the pressure maximum. The centre point for low pressure systems is the centre of the outermost bounding rectangle in its representation. Next, corresponding centre points of the shape representations are aligned with one another and the degree of partial match between the shapes is computed. As described above, the axes of high pressure systems are considered as well as the system's geographical extent. For simple low pressure systems, geographical extent is the only feature of the shape that matching considers. The internal structure of clusters is matched recursively.

As part of similarity assessment, an average geographical displacement between the features in the query and corresponding features of the input is computed and contributes to downgrading the degree of partial match. Displacements of individual features from the average displacement are also calculated. East-west displacements are usually of less concern than north-south displacements, because weather systems tend to naturally progress from east to west, and small displacements in this dimension can be attributed to the infrequency of sampling. North-south displacements, on the other hand, are more problematic because the weather over New Zealand is heavily affected by such deviations, and because the physical behaviour of weather systems is strongly conditioned by their latitude.

The degree of match between spatial relations in the query and the data is also computed; all such features contribute to the goodness of match between the query and a case.

Finally, a number of other features of weather systems such as the absolute values of pressure minima and maxima are also compared during the match.

All of the factors described above are combined to arrive at an overall goodness of match. Exactly how these factors are to be combined has yet to be determined, and we expect to have to perform many tests to determine the appropriate contribution of each factor.

Implementation

In implementing our representation and retrieval mechanisms, we apply both traditional and up-to-the-minute database management techniques. Our representation of weather systems introduces a number of spatial data types not typically found in commercial database systems, such as points, rectangles, and polygons. While these types could be simulated using standard data types such as integers, the operators and indexing methods that databases typically provide are insufficient for efficient retrieval using these more complex types. To provide a flexible environment for working with spatial and temporal representations, we are using Postgres, a research database management system under development at the University of California Berkeley.

Postgres is a relational database that allows users to define their own data types. New data types can be constructed by providing suitable input and output functions, written in a general programming language. These functions can be dynamically linked with the database engine. Arbitrary operators on these types can also be specified as additions to Postgres' query language, and the query optimiser treats them in a sensible way.

In particular, custom negation, join, and sort operators can be specified for each data type, and time and space costs can be defined for each. These costs are employed by the Postgres query optimiser. For example, consider a user defined type to represent geographic regions, and comparison operators that compare the area of regions. If we declare the "area less than or equal" operator AREA_LE to be the negator of the "area greater than" operator AREA_GT then Postgres can automatically optimise the query

```
retrieve (low.minima) where not (low.region AREA_GT 40000)
```

by transforming it to the equivalent but more efficient query

```
retrieve (low.minima) where low.region AREA_LE 40000
```

The transformed query is more efficient because it requires only one pass over the data, whereas the original query requires two passes: one to select those cases whose area is greater than 40000 units, and a second pass to select all cases that were not selected on the first pass.

An especially important efficiency consideration is the availability of appropriate indexing methods for user-defined types. Although Postgres does not currently support user-defined indexing methods, it does provide an unusually wide choice of built-in methods.

Of particular interest to us, Postgres provides an R-tree indexing method that organises spatial data for efficient retrieval (Guttman, 1984). Two-dimensional regions are approximated by the smallest rectangle that contains them. This indexing method allows the system to quickly retrieve all regions that overlap a given system or are contained in it. Case selection relies heavily on this indexing method to retrieve features of weather situations that have geographical extent, such as high and low pressure systems.

Postgres includes an object model and a rule system. The object model supports unique tuple identity, and efficient tuple access, and is useful in implementing efficient case selection analogous to traversing slot-filler links in a frame-structured knowledge base.

The rule system is a declarative language capable of recursion, as well as forward and backward chaining. We use forward chaining rules to automatically maintain temporal relations between meteorological features. For example, inserting a new feature into a time slice causes a search for parents and children of that feature in adjacent time slices. We use backward chaining rules to carry out recursive queries as single database operations. To query whether two weather systems are causally related requires a recursive query for all non-trivial cases. For example, to find whether a given low pressure system is the parent of a low pressure cluster occurring several time slices later requires following parent-child links through intermediary time slices. We could implement this query by walking the parent-child links with a series of database queries. Instead, we implement this more efficiently using a single database query that invokes appropriate backward-chaining rules.

Conclusions

We are still in the process of implementing the case retrieval system described in this paper as an addition to MetVUW Workbench. Even at this early stage, however, we believe that several useful lessons can be drawn from our experiences in its design and implementation:

1. **Use high level features**. For experts to fully exploit the potential of databases in natural resource domains, they must be able to issue queries for relevant data using a high-level vocabulary that closely parallels the way they talk about the domain. In the case of meteorology and weather forecasting, this vocabulary should include ways to describe high and low pressure systems, ridges and troughs, the jet stream, and other standard features of the weather.

2. **Work closely with domain experts**. Our cooperation with MSNZ has been invaluable in helping us to determine which high level features of weather systems are relevant, such as the orientation of the ridges in high pressure systems. In the longer term, we expect that continued close interaction will facilitate acceptance of the completed system within the organisation.

3. **Annotations encode domain-specific expertise**. The kind of system we are constructing is not a general purpose database: instead, it is specifically tailored to the domain of meteorology. The graphical query language and underlying symbolic representations encode only aspects of weather systems that experts consider meteorologically significant. Indeed, we expect the completed case retriever to automate much of the domain-specific meteorological expertise needed to determine whether a past case matches a given high-level description. We expect that attempts to apply the approach outlined here to other natural resource domains will likewise have to address their own issues of domain-specific representation.

4. **Use off-the-shelf components**. The great bulk of MetVUW Workbench has been built using off-the-shelf components that are freely available on the Internet. For example, we use an efficient public-domain package called LQ-text to implement full text search. We use the Postgres relational database to store all data; case selection and similarity assessment are layered on top of this system. Off-the-shelf components have allowed us to rapidly build a prototype that can be expected to scale to a very large database.

5. **Build intelligent tools, not automated problem solvers**. Artificial intelligence technology has progressed to the point where intelligent tools for organising and displaying large quantities of data can be constructed with comparative ease. We are constructing a tool of this kind. This new generation of tools focuses on those tasks that computers do best, namely searching and processing vast quantities of data. As far as possible, they avoid the much harder task of automatic decision making. The rapid rise in popularity of case-based decision aids in industry (Kolodner, 1991) is strong evidence for the success of this approach. Like the MetVUW case retriever, these decision aids employ a large memory of past cases and carefully designed domain-specific representations. These systems deliberately avoid the "deep" domain theories often employed in rule-based expert systems, as such theories have often proved very difficult to engineer in practice.

Finally, to the best of our knowledge the MetVUW case retriever is unique in the domain of meteorology: we know of no existing system that can retrieve past weather situations from high level descriptions. We therefore believe that our system sets a valuable precedent for the design of intelligent database applications in natural resource domains.

Acknowledgements

Thanks to Linton Miller for assisting with the preparation of this report, and to the Meteorological Service of New Zealand for providing the situation maps in figures 3, 5, and 6.

References

Fong, Z. (1986). The design and implementation of the Postgres query optimizer. Master's thesis, University of California, Berkeley.

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, Mass. ACM-SIGMOD.

Jones, E. K. and McGregor, J. (1993). MetVUW: A multimedia teaching aid. In *Proceedings of the International Conference on Computer-Aided Learning and Distance Learning in Meteorology*, Boulder, Colorado.

Kolodner, J. L. (1991). Improving human decision making through case-based decision aiding. *AI Magazine*, 12(2):52–68.

Stonebraker, M. (1987). The design of the Postgres storage system. In *Proceedings of the International Conference on Very Large Data Bases*, Brighton, England.