The Knowledge Engineering Review, Vol. 12:1, 1997, 41–58

Survey on structure-based case retrieval

FRIEDRICH GEBHARDT*

GMD—German National Research Center for Information Technology, Institute for Applied Information Technology (FIT), Artificial Intelligence Department, D-53754 Sankt Augustin, Germany (e-mail gebhardt@gmd.de)

Abstract

The main components of case-based reasoning are case retrieval and case reuse. While case retrieval mostly uses attribute comparisons, many other possibilities exist. The case similarity concepts described in the literature that are based on more elaborate structural properties are classified here into five groups: restricted geometric relationships; graphs; semantic nets; model-based similarities; hierarchically structured similarities. Some general topics conclude this survey on structure-based case retrieval methods and systems.

1 Case-based reasoning with structured cases

People remember past events when making decisions. A German proverb says "A burnt child shies away from fire", an English one "Once bitten, twice shy". Trying to reach an unfamiliar place, one often follows first the known route to a place in the vicinity even if a shorter route to the destination exists. A good chess player remembers scores of games played by chess masters. If the car stalls, one thinks of failures with similar symptoms that have occurred in the past, and becoming sick many people try a cure that has helped before, although the cause may have been quite different.

Case-based reasoning tries to transfer such human reasoning methods to the computer. The examples just mentioned show already some characteristics. The solution of the old case may have to be adapted to the new situation, and it is not always optimal as in the route planning example. It may even be harmful as in applying the wrong cure.

Given a problem, the *query case*, one must first remember a suitable past situation, the *source case*. This is the case retrieval. Next one has to reuse the retrieved case. This is sometimes trivial; more often the source case has to be adapted to the new problem, that is to the query case.

Adaptation requires knowledge on when and how to adapt. If a correct domain theory can be applied, the adaptation leads to a correct result. More often, only limited knowledge is available and adaptation relies totally or in part on a continuity assumption: a solution for a similar source case will, after reasonable adaptations, perhaps not be the best solution for the query case, but at least a good one.

The whole field of case-based reasoning has been reviewed a couple years ago in this journal by Watson and Marir (Watson & Marir 1994; Marir & Watson, 1994) giving more than 500 references. Recently, cases with complex internal structures have gained increased attention; these are not specifically treated in the above mentioned review and will be investigated here.

* This research was supported by the Federal Ministry of Education, Science, Research and Technology (BMBF) within the joint project FABEL under contract no. 01IW104. Project partners in FABEL are GMD—German National Research Center for Information Technology, Sankt Augustin, BSR Consulting CmbH, München, Technical University of Dresden, HTWK Leipzig, University of Freiburg, and University of Karlsruhe.

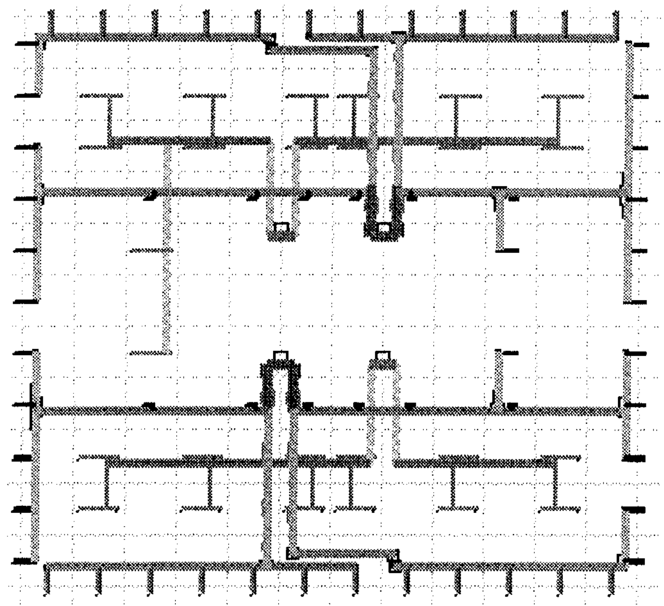

Figure 1 A complex case: the layout of supply air and return air in a storey of a school building. This case contains 551 design objects, some of them tiny. This example is taken from the FABEL project.

Most case retrieval systems base the search on the comparison of descriptors or of values for attributes. This procedure is so simple and fast and therefore so widespread that other possibilities are widely ignored. The text book by Janet Kolodner (1993) has many chapters on indexing and on retrieval based on indexing, but none on other retrieval methods, although some systems not relying on attributes are described. However, Aamodt & Plaza (1994), in their survey on case-based reasoning methods, recognize that a typical case has a certain complexity with respect to its internal organization.

For cases with complex structure, attribute-based retrieval is inadequate. Let us give two examples.

Given a case base of graphs, consider a query case in the form of a graph and the problem to find a source case that contains as large a subgraph of the query case as possible. Any descriptions of the source cases such as number of nodes and edges (of a given type, if appropriate) or number of nodes with a given number of edges will not help much.

In the design of technical buildings with high demands on installations (air, water, gas, chemicals, etc.) the essence of a case is not the number of service points (e.g., outlets) or the length of the supply lines or anything else that can be captured by attributes, but the geometrical constellation. Figure 1 shows the supply air and return air system of a storey as an example.

The survey is on retrieving complex cases. However, retrieval cannot be completely separated from reuse; after all, reuse is the final goal and retrieval only an important step in this direction. The suitability for reuse will finally determine the worth of retrieval.

2 Case similarities

Sometimes all cases in a case base are checked for applicability; in general a retrieval step selects one or a few source cases for further use. Retrieval is based on a judgement of the similarity of cases, e.g., on the identity or similarity of case attributes. As mentioned above, this attribute-based similarity is insufficient for complex cases.

There are many ways to define similarity between cases and organize retrieval accordingly. The common ground for these non-standard procedures is to exploit structures in the cases that go beyond the simple enumeration of attributes. But apart from this commonality there exists a large variety in the case retrieval systems presented in the literature. One could perhaps collectively call these approaches "structure-based case retrieval" or "structural similarity".

FABEL was a project to investigate ways of supporting design by case-based and model-based methods; its application area was architectural design. The project developed a bunch of case retrieval algorithms (including attribute-based modules not mentioned here). Work in that project has initiated a fairly comprehensive search for other systems employing structure-based similarities or related procedures, and thus led to this survey. A few systems found are omitted here because the information on details of interest was too scarce or the case retrieval was only a minor issue. Nevertheless, it is possible that some interesting systems have been overlooked.

Most authors use unspecific names for the cases in the case base such as old case, stored case, case in memory. We prefer the term *source case*, since this is the distinguishing knowledge source in case retrieval. The user's problem is formalized as a (perhaps incomplete) case, the *query case*, by other authors called search case, target case, new problem, input case, current query.

There are of course several ways to classify the retrieval systems—according to the field of application, the type of case structure or the class of search algorithm. We will use here primarily the case structure but with an empirical touch: we will not develop a theoretical classification scheme for conceivable systems but rather try to classify the existing ones. However, some case retrieval systems show characteristics of more than one class so that one could debate their best fitting classification.

Similarity is also involved in many procedures outside case-based reasoning. Some examples are: matching similar strings, finding known items in a picture, identifying items in a picture, clustering, data mining. Certainly some of the ideas there could be rephrased as case similarities and utilized for a case retrieval system, but since no working examples have been found, such procedures are neglected here.

This review includes some procedures that are not part of a case retrieval system but are interesting enough in the types of similarity they propose or apply. It omits on the other side document retrieval systems exploiting the structure of documents, which could be considered a special form of structured cases.

Of course, a review can only sketch the diverse methods presented. To learn how they work exactly, the primary literature has to be consulted, and often enough it leaves questions open.

A short review on similarity issues in philosophy, cognitive phychology and computer science including many references is contained in Jurisca (1994). Other recent survey articles on similarity and cognitive science are Goldstone (1994) and Medin et al. (1993). Some questions related to this survey are probably discussed in Tanaka (1990), but unfortunately in Japanese.

3 Restricted geometric relationships

3.1 General idea

Many kinds of cases that are not adequately handled using attributes are geometrical in nature or at least related to geometrical problems. Typical examples are arrangements of physical objects subject to some side conditions.

A reasonable measure for the similarity between geometrical arrangements is essentially the degree of fit or coincidence after appropriate transformations such as shifting, resizing or rotating.

Finding geometrical coincidences can sometimes be restated as a graph matching task, but the general graph matching problem is NP-complete, so that the time for computing a single similarity may be exceedingly high. Therefore it is essential to exploit any regularities that exist in the cases. Depending on the nature of the regularities, different similarity measures and different methods for finding the most similar case are needed. We will indicate now some typical situations.

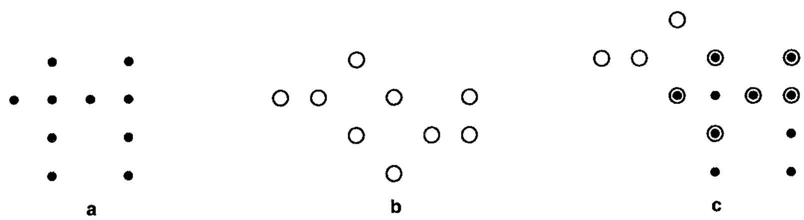

Figure 2 Figures (*a*) and (*b*) can be shifted such that six points coincide.

3.2 Discrete positions

A simple first problem type is represented by sets of points with the restriction that only certain discrete positions are admissible places. In particular, these places may be the points on a grid (evenly spaced lines in horizontal and vertical direction). The obvious similarity between two such sets of points is the maximal number of coincidences of both sets after shifting (and perhaps rotating by 90, 180 or 270 degrees or mirroring, depending on the problem domain). An example is shown in Figure 2.

This is a string matching problem: both sets can be represented as arrays of zeros and ones that can be matched like character strings; the time is at most proportional to the square of the array size, and is rather short if a logical AND function for memory arrays can be used.

The coincidence of grid points has been investigated to a certain degree in the FABEL project (Gebhardt, 1993), but was abandoned because it was not powerful enough for the domain.

There are many ways to modify the simple schemes. The points in the grid may have types; this is still a character matching problem now with more letters in the alphabet. The points may have weights (or sizes or densities).

A somewhat more complicated situation is investigated by Petrakis and Orphanoudakis (1993). The grid is artificially imposed on objects in the plane; the algorithm searches source cases with objects of matching type in grid points which have matching relative positions using a two-dimensional string search.

3.3 General positions

The second problem type is more general. A case may now consist of a set of typed objects in arbitrary positions in two- (or three-) dimensional space. The retrieval goal is to find a case where as many objects as possible agree in type and suitably chosen pairs of objects agree in their relative position, i.e., the distance between corresponding objects is minimized after moving appropriately (shifting and perhaps rotating). Here the similarity has two components, the number of objects with matching type and the geometrical distance between objects of a pair.

Scaling the distances may be permitted. Objects in the query case missing in the retrieved case may contribute more to the dissimilarity than the other way round. If there is a hierarchy of object types, objects of different, but related types may contribute to the similarity.

An example for the arrangement of typed objects assumes that there is at most one object of each type in a case so that the mutual assignment is trivial (Gudivada & Raghavan, 1995). Rotations and resizing are permitted; thus the only invariants are angles. The similarity used for this constellation is based on the angles between the connections of corresponding pairs of objects in both cases. In a first step, the average of these angles is computed, then the second case is rotated back by this amount. The similarity is a function of the remaining angles. An example is shown in Figure 3.

A complex problem in three-dimensional space is to find molecules of similar structure. This includes not only the positions of atoms but also their chemical properties. A set of related programs has been developed for this purpose (Artymiuk et al., 1992; Bures et al., 1994; Pepperrell, 1991; Bath et al., 1994). Similarity is based in a rather complicated way on the set of distances of an atom (optionally, of a certain type) to all other atoms in the molecule. Instead of atoms, protein

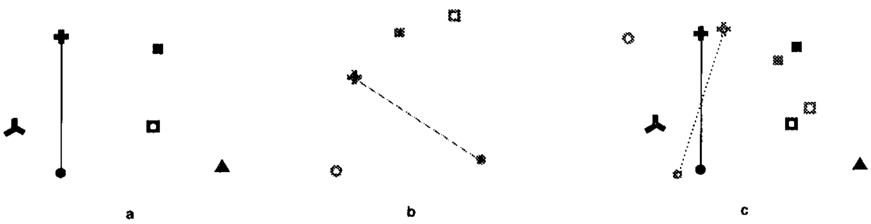

Figure 3 Figure (*b*) is rotated such that the angles between connections of corresponding objects in (*a*) and (*b*) become minimal. Here the connections between the cross and the full circle are shown including in their original positions (*a*) and (*b*) an angle of 57°, after rotation (in (*c*)) one of 17°.

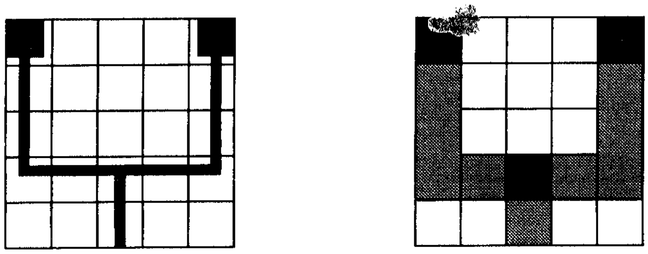

Figure 4 Simplified principle of ODM. The case is replaced by a matrix (here 5×5) of grey values standing for intervals of object densities.

substructures are used for comparing large protein molecules. These programs have been used with rather big case bases, in particular the Brookhaven Protein Data Bank with several hundred thousand molecules.

ODM (Object Density Matrix), also developed in the FABEL project, recognizes several classes of objects and compares the density of a class in geometric segments of a case (Coulon & Steffens, 1994; Coulon & Gebhardt, 1994). For this purpose, the minimal square containing the case is divided into 11×11 subfields, say, and the proportions of each field covered by objects of an object class are compared (see Figure 4). The subfields overlap somewhat to alleviate discontinuities in the resulting similarity.

Petrakis has a fixed number of typed objects (in his case, outlines of body, liver and spinal column) and a number of unknown objects (e.g., tumors) (Petrakis, 1993; Petrakis & Faloutsos, 1994). The known objects serve as a geometrical reference to the unknown objects. The cases are essentially decomposed into subcases with just one unknown object. Given a query case, the subcases most similar to each one of its unknown objects are retrieved and those originating from the same case are recombined.

A related problem, similarity of two pictures by means of the relative position of known icons, in the context of picture retrieval is investigated by Buehrer & Chang (1993).

3.4 Route finding

A third type of geometric problem is route finding. In a given environment, find the best path (shortest in distance or time or money) between two points. The case base contains known paths. The similarity is a function of the distances between corresponding end points or perhaps a function of the length of the path that can be reused (thus permitting to assemble the solution from several cases).

Since route planning is a special form of planning, similarities between some other types of plans may perhaps be handled in a comparable manner.

Variations arise from the available domain knowledge such as layout of a factory hall for robot routing, partial or complete street maps of a city or region, inclusion of time tables for connections between locations, construction systems that prescribe certain admissible paths for supply lines (as in the construction system ARMILLA used in FABEL (Haller, 1985)).

A square world of $2^n \times 2^n$ field contains obstacles (Branting & Aha, 1995). To avoid exponential search time, two ideas are combined. The case base contains known routes between certain fields. The world is successively coarsened by combining 2×2 fields into one which knows from which to which of its neigbours there exists a valid path. This hierarchical coarsening is used firstly for indexing the cases in the case base and secondly for finding paths between the desired end points and the end points of the retrieved case by means of a standard search algorithm.

A geographical problem is to find the best connection between two points in a city using a huge city map (in this case, Pittsburgh with 25 000 street segments) and some routes with weighted distances for route segments, the weights reflecting the comfort or discomfort in using that segment (Haigh & Shewchuk, 1994; Haigh & Veloso, 1995). The known segments are combined into a net with special properties (Delaunay triangulation). Thus one has in effect not a case base of routes but a net of route segments. Traversing this net is computationally easy. The city map is needed for missing pieces at the ends of the desired path and for intermediate stretches where the Delauney triangulation has created connections between two points that are not yet verified by a known route.

Another route finding system is described in (Fox & Leake, 1995). The similarity is composed of the distance between the starting points and the distance between the end points in both cases.

Geographical data play a role in two other systems that are otherwise hard to classify in our classification. Both of them deal with forest fires. In Rougegrez (1993), similarity is based on three time series: wind, relief and vegetation at the fire front. Little is known about treating the geographical data in the ongoing project CHARADE (Ricci et al., 1994).

3.5 Picture retrieval

One might expect that picture retrieval in large picture data bases provokes research in this direction. This is hardly the case. Picture retrieval is accomplished almost exclusively by means of descriptions as documented by survey papers (Cawkell, 1993; Layne, 1994, see also Enser, 1995). Descriptors may be derived attributes such as Hugh transforms of the boundary of an object (Hough & Arbor, 1962; Ballard, 1981; Ballard & Brown, 1982) or functions of the higher moments of a figure that are invariant against certain geometrical transformations (Flusser & Suk, 1993). There are a few systems where a sketch or an available picture can be used as part of the query.

For a special issue on syntactic pattern recognition including issues like structural pattern recognition, similarity of patterns and picture analysis see (Bunke & Sanfeliu, 1986). Many younger references are given in Shepherd et al. (1992). No doubt some of the methods described there and in other publications on picture recognition could be rephrased into novel similarity measures for the depicted objects.

The picture retrieval system QBIC (Flickner et al., 1995; Niblack et al., 1993) matches sketches by means of a special procedure working on reduced versions of the pictures (64×64 pixels), scanning the whole picture base, but otherwise it is completely attribute-based.

4 Graphs

4.1 General graphs

Whenever cases are represented as graphs, a natural similarity measure is the size of the largest matching subgraph. Alternatively, the common subgraph itself is the similarity. This is an example of a non-numeric similarity measure where two similarities (i.e., the similarities of a query case to two source cases) may or may not be comparable, "smaller" meaning here "subgraph of". It is also an example for a similarity between any number of objects, not necessarily two.

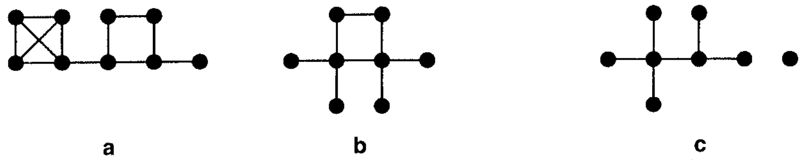

Figure 5 The largest common subgraph of graphs (*a*) and (*b*) is (*c*) with 6 edges and 8 nodes in common, among them an isolated node. If the similarity were to be the largest common connected subgraph, the isolated node would have to be deleted.

Graphs need not be the original form of the cases. If a case consists of many objects and various relations between them, it can easily be transformed into a labeled graph. Examples are geometric relations between physical objects or causal relations between events.

Finding matching subgraphs is an NP-complete problem. This is not the place for reviewing graph matching algorithms (Mehlhorn, 1984; Brandstädt, 1994; Gati, 1979).

Algorithms for graph or subgraph isomorphisms are in general based on state-space search with backtracking. There exist special algorithms for many subclasses of graphs such as planar graphs or graphs with a bounded number of edges in a node (bounded valence) where the computational complexity is polynomial.

In graph matching one has to distinguish graphs with directed or undirected arcs; nodes and edges may be typed (labeled).

If connectivity is not required, each node of the smaller graph can be associated to a node of the larger one; thus the number of matching nodes does not contribute to the similarity. Assuming the query graph is smaller than the source graphs, the similarity reduces to the number of matching edges (analogously for typed nodes and edges). Figure 5 shows a tiny example.

If one is interested in connected subgraphs only, the similarity is composed of the numbers of matching nodes as well as edges. Possibly nodes and edges of various types may receive different weights.

4.2 Retrieval systems for general graphs

In CaPER, the cases are labeled graphs. The main emphasis lies on utilizing highly parallel computers (a Connection Machine) (Kettler & Hendler, 1994; Sanders et al., 1995). Domain ontologies are exploited. Application fields are car construction (a toy base), transport logistics and protein sequencing.

The FABEL component Topo transforms the graph isomorphism problem into the algorithmically simpler problem of detecting cliques (completely connected subgraphs) (Coulon, 1995a, 1995b); however, the transformed graphs are much larger than the original ones. Originally, there is an edge between any two objects (nodes) within a certain distance; the type of the edge reflects the kind of connectivity (roughly speaking: touching, overlapping, lying inside or outside) in each of the three spatial dimensions. Topo finds easily the most similar subgraphs in a graph corresponding to several thousand objects with several hundred thousand topological relations. For this reason, it is not necessary to cut a building (the current application of Topo) into sizeable cases; the whole building is taken as a case.

FABEL has also implemented another general backtracking algorithm for finding the largest common subgraph (Tammer et al., 1995). Using depth search, it prunes branches in the search tree that cannot possibly lead to a better solution than the best one known so far. Additional implemented algorithms are tuned to special situations (connected installation graphs).

Bunke and Messmer (1994) introduce a similarity between graphs that weighs the transformations leading from one graph to the other; if, for instance, the deletion of nodes and edges has zero cost, the (unsymmetric) distance from a graph to a subgraph is 0.

The exponential search time for finding exactly matching typed subgraphs can be replaced by exponential storage space (neglecting the time to construct it) (Messmer & Bunke, 1995). A decision tree is built with the empty graph as its root. On each subsequent level of the tree, a node of the given graph is added together with its connections to the already processed nodes. The number of nodes can grow exponentially with the size of the graph but search time is quadratic. All cases in the case base can be incorporated into the same decision tree; the search time has the remarkable property of being independent of the size of the case base.

There is much redundancy in this decision tree; some refinements reduce the redundancy and the required storage space. In practice one would cut the tree off after a limited number of levels. Then the leaves point to cases that may or may not contain the query case but have at least a common subgraph corresponding to the number of levels in the tree.

4.3 Special types of graphs

Due to the NP complexity of the general graph matching problem, it is natural to exploit any available simplifications if the graphs are known to belong to special classes.

The graph isomorphism problem can be solved in linear time for planar graphs (Hopcroft & Wong, 1974) (although possibly with a rather large proportionality constant). This speeds up the retrieval of large graphs but does not help to find similar rather than isomorph ones.

Messmer and Bunke propose another division of graph algorithms, group theoretic concepts versus more practically oriented procedures mostly based on state-space search with backtracking (Messmer & Bunke, 1995).

5 Semantic nets

5.1 Activation spreading

In the methods of the present section we do not deal with separate cases in a case base; rather all the retrievable knowledge has been tied together into a large semantic net. Typically the nodes represent object types with inheritance relations and objects themselves, the edges (links) are various relations from `is-subtype-of`, `is-a` and `is-part-of` to domain-specific ones like `is-born-in`, `has-airline-connection-to` or `touches`.

Search in a semantic net is usually based on activation spreading (activation passing). Initially, some nodes get a numerical value for activation; these nodes correspond to the query case in other systems. The activation is iteratively passed on to nodes connected to activated nodes taking into account various properties such as the type of link and simultaneous activation from several nodes. Some provisions have to be taken that the activations do neither grow indefinitely nor are too low to produce meaningful results. In particular, the propagation functions may be non-linear. There may, or may not, be negative activations having the effect of inhibitors.

A connected set of nodes that after a while is sufficiently activated corresponds to a retrieved case. The structure of such a subpart may differ considerably from that of the query (the initial activation). Sometimes the result qualifies as source case only if it has a prescribed structure, which has to be completed by adding appropriate neighbours, if necessary.

The propagation of activations has in principle exponential complexity. If the activation of a node changes due to additional connections found during the process, this altered activation has to be passed on again. To reduce work, one may neglect activations below a threshold.

A lot of literature exists on semantic nets which is not reviewed here, but no comprehensive study on activation passing seems to exist. In semantic nets there are perhaps fluid boundaries from case retrieval to other retrieval or reasoning algorithms; the main distinction is that case retrieval demands some at least rudimentary definition of what constitutes a case within the net.

A small semantic net is used by the Fabel method ASM (ASsociative Memory method) (Gräther, 1994). It has only two types of nodes, one for cases and one for descriptors. Each case is connected

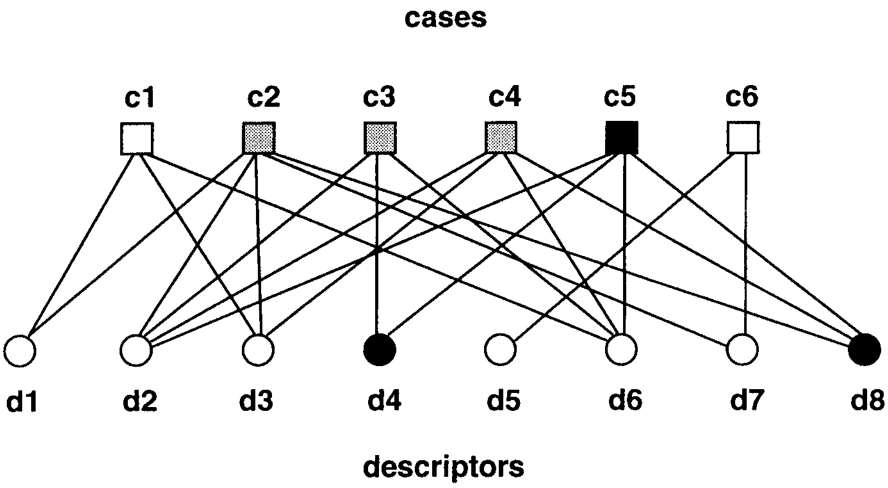

Figure 6 A small semantic net with only two types of nodes as used in ASM. Here the activation of descriptor nodes at retrieval time (*d*4 and *d*8) passes only one step. In other more complex nets, this is an iterative procedure.

to all its decriptors (see Figure 6). At retrieval time the descriptors of the query case are activated and the associative memory essentially finds the cases with the largest number of activated descriptors (actually, it is a little bit more complicated).

In the system BankXX, legal theories (formalizations of laws and former judgements) and legal cases are arranged in a net using several dozen different types of edges (among others, eight qualities of citations) (Rissland et al., 1995a; 1995b). Three different evaluation functions measure the applicability of the theory or case found to the new problem.

In an application for travel agencies (Burkhard & Lenz, 1996) information items of the domain as well as travel offers are represented by nodes of a semantic net. The offers are connected to the applicable information items and the latter ones are interconnected if they are semantically related, e.g., Spain to South Europe and to West Europe, beach and lake to swimming.

5.2 Activation spreading with colours

A generalization of the activation spreading algorithm as described so far is to use several kinds of activations in parallel; these are sometimes called "colours". Special rules govern the interactions between colours such as mutual strengthening or inhibition.

The effect of different colours of activations is investigated in a small system (22 cases for a bin packing problem, a rather specialized knapsack problem) called CRASH (Brown, 1994).

IDA, the Innovative Design by Analogy, sets out to apply known cases to semantically distant problems (Wolverton, 1994; Wolverton & Hayes-Roth, 1994): a solution in one area is to be transferred to a quite different application. For this purpose, cases in the semantic net have the components structure, behaviour and function. The "knowldge-directed spreading activation (KDSA)" looks for cases with high similarity in function to the query case, but low similarity in structure and behaviour. As an example, IDA tries to design a sprinkler irrigation system and finds the blood circulatory system from which it transfers design ideas.

5.3 Other procedures for semantic nets

Another approach using nets of cases assumes that neighbouring cases (connected nodes) have approximately the same similarity to any query case and that it makes sense to proceed to neighbouring cases with higher similarity, an adaptation of steepest ascent search to nets. To avoid getting stuck at a local maximum, this can be combined with a so-called taboo search where a move that annihilates a recent move is prohibited for n steps (n being constant or time dependent). This presupposes a net with directed edges of several types.

An example for this category is a system used for playing the game Abalone (Reiser & Kaindl, 1994). The net consists of game positions (with an evaluation) with edges to similar positions. "Similar" is defined by the distances between individual pieces on the board. Given a problem, all possible moves are generated, the cases most similar to the resulting situation are used for an evaluation and the best scoring move is selected. In the next step, the candidates for the most similar cases are taken from the neighbours of appropriate cases used in the former move.

6 Model-based similarities

6.1 General idea

In a way, all similarity measures are, or should be, model-based. It makes little sense to deal with graphs if all that matters in a particular domain are certain attributes. The present section collects examples where an elaborate domain model leads to somewhat specialized (if not ideosyncratic) similarity measures.

In a way, the systems to be described in this section are the left-overs that are too special to build more general classes. There is another tendency most of them have in common: retrieval and reuse cannot clearly be separated. This is not surprising. The typical purpose for searching an old case is its intended adaptation to a current problem. In complex situations the adaptation knowledge influences or determines the retrieval, and retrieving a good case may involve parts of the adaptation.

Adaptation presupposes a model of the domain; the natural measure for similarity is then the effort necessary for a successful adaptation. However, this effort is mostly known only afterwards and is therefore rather useless for selecting a case from the case base.

Nevertheless, the underlying model can guide the retrieval providing in a simple example the weights for various attributes in an attribute-based retrieval. The main task in developing a retrieval system lies then in finding easily recognizable features of a case that are good predictors for the effort of adaptation.

Although adaptation is a strong incentive for developing retrieval systems based on a deep domain model, it is not the only one. Others include diagnosis (in technical as well as medical fields) or prediction (e.g., whether, economical indicators, success of legal reasoning). Naturally the resulting similarities are as diverse as the domain models.

There is no obvious subdivision of the class of model-based procedures, the simpler ones tend to describe the cases and to rely on heuristics, while the more complex ones include case explanations and deeper domain theories. We will use the intended type of reuse, case description or case explanation, for distinction, although this criterion is not always quite clear.

6.2 Models for case description

In this section we collect systems that utilize somewhat elaborate models for constructing or describing cases.

Given a query case, the domain model may select features that are important for determining the similarity to source cases; these features then govern the retrieval. Thus the similarity function itself (not only the similarity value) depends on the query case. In the simplest situation, this leads to an attribute-based search with attribute weights depending on the query case. In more complicated situations, the domain model takes account of various dependencies between the features of the query case.

The exceptional feature of CADET is a bunch of transformations using domain knowledge that can be applied to the query (Sycara et al., 1992). Therefore this attribute-based system is loosely related to model-based search. CADET is used for engineering design.

Déjà Vu knows which attributes are unimportant for finding a similar case, for which important attributes there exists an adaptation tool and which other attributes might be affected by using it so that additional adaptations could become necessary (Smyth & Keane, 1994, 1995).

A classification algorithm is applied to tree-structured cases yielding abstract cases (subtrees) in SYN* (Börner & Faßauer, 1995; Börner, 1995). In the application, the trees are connection lines for air, water etc. from a supply (a trunk line) to the service points (outlets). In addition, the abstract case carries the union of all cut branches. At retrieval time, the query case is compared with this combined source case; the subtree extracted from the abstract source case to solve the query may contain parts of several original source cases.

Finding a protein in a protein case base that fits a given nucleotid sequence is a special kind of string matching (Shavlik, 1991). Since three nucleotids code for one amino acid, there are three frames to read the DNA string (shifted by one position); since the complementary DNA string might be the correct one, one also has to exchange the corresponding nucleotid pairs (A and T, C and G) reading the resulting string in opposite direction; since the sequencing process makes errors in about 1% of all nucleodids (substitutions, omissions), the similarity between a nucleotid sequence and a protein is not just the longest compatible sequence but is composed of non-overlapping sequences in the same direction of reading.

Quite a different situation is the similarity between well-formed logical formulas (Esposito et al., 1992). The similarity is measured by the chance of finding unifications and by the semantic distances in a domain model of the atoms involved.

Occasionally, the source case rather than the query determines which features are important. Now many or all cases in the case base have to be inspected and the computed similarity decides which one (or which ones) to investigate more closely. This is true in the following application.

A similarity between strings can be defined using the minimal length of encoding a query string by references to substrings of one or more source strings (plus portions of the query string that cannot be encoded). This has been applied to the four letter alphabet of nucleotid sequences (Milosavljević, 1993, 1996). Since each encoding is done in linear time in the length of the query string, it is used for searching similar genes in a genome bank.

6.3 Models for case explanation

The more complicated similarity functions in this chapter anticipate some of the subsequent work so that the boundary between retrieval and reuse becomes blurred. The underlying model serves not only for describing but also for explaining the case or the applicability of the source case to the present problem. This is particularly true for the system CASEY, described below.

A simple form of models for case explanation consists in an adaptation of the similarity assessment driven by the source case, perhaps just in adjusting weights in attribute-based retrieval depending on properties of the source case (rather than the query case). A similar strategy for story understanding in the system ACCEPTER is called constructive similarity assessment by Leake (1992).

In CLASH, source cases consist of facts representing the problem part, other facts representing the solution part and complete or partial explanations of the solution facts by the problem facts (Bento & Costa, 1994; Bento et al., 1995a, 1995b). Given a query case with facts for the problem and the desired solution, one is looking for source cases where this solution is derived or at least made plausible by the facts. The explanation takes in this situation the place of the "solution" that normally is requested in case retrieval. A related, but older development are the "footprints" used by (Veloso & Carbonell, 1991).

CASEY has a well-chosen case base of about two dozen patients with heart failures (Koton, 1988). Given a new patient, it searches for cases displaying similar symptoms; the weight of an attribute depends on the importance for explaining the diagnose in the retrieved case. CASEY then tries to reconcile any differences using an elaborate domain model, the Heart Failure Program. In this step, the diagnosis of the retrieved case plays a similar role as the problem facts in CLASH. If no known case is satisfactory, the new one is added to the case base after a complete analysis with the Heart Failure Program.

CASEY seems to be the only system that has been evaluated by a person other than its author (Aghassi, 1990). The remarkable result is: the error rate is untolerably high because similar cases do not necessarily require similar solutions. This is a severe warning of relying on plausible similarity notions without testing them in great detail.

In legal reasoning it is not only important to find good cases; there should also be no stronger counterexample. Thus a comparison between several cases (not just two) is involved. This type of legal reasoning is implemented in the tutoring system CATO in the field of trade secrets law (Ashley & Aleven, 1994).

Still other approaches consider the similarity of trees, in particular logical formulas; the similarity is essentially the largest common partial tree, possibly after performing some admissible transformations reflecting the semantics. A mere comparison of trees would fall into the chapter on graph searches, but the inclusion of transformations requires special models.

Syntactic and semantic similarity of a set of formulas is proposed in (Börner, 1994). The syntactic similarity is related to the least general antiunifier; the semantic similarity takes also into account transformations according to equational theories (these express the semantics).

A somewhat unusual application is the focus of CAESAR: a library of 30 mathematical subroutines (linear algebra) (Fouqué & Matwin, 1993). The query is stated in a Prolog-like language; retrieval uses deductions and inductions on terms of a thesaurus.

Similarities between sets of logical formulas using involved substitution rules are also needed in the Structure Mapping Engine SME (Falkenhainer et al., 1989). These similarities are applied to machine learning rather than case retrieval.

7 Heirarchically structured similarities

7.1 *True hierarchies*

Sometimes the similarity judgement is hierarchically structured. For instance, there must be agreement in certain features; depending on these features, the final similarity is computed in different ways in a second step. There is no exploitable similarity if the features considered in the first step do not match. The principle is sketched in Figure 7.

There may be more than two steps; one then obtains a decision tree with exact matches in the early nodes and possibly diverse similarity measures in the leaves.

In CAPLAN (Computer Aided Planning), used for cutting rotary-symmetric workpieces on a lathe, the decision tree reflects the manufacturing process, for instance the type of machines needed (Muñoz et al., 1995; Muñoz-Avila & Huellen, 1995). The final similarity is the estimated cost of

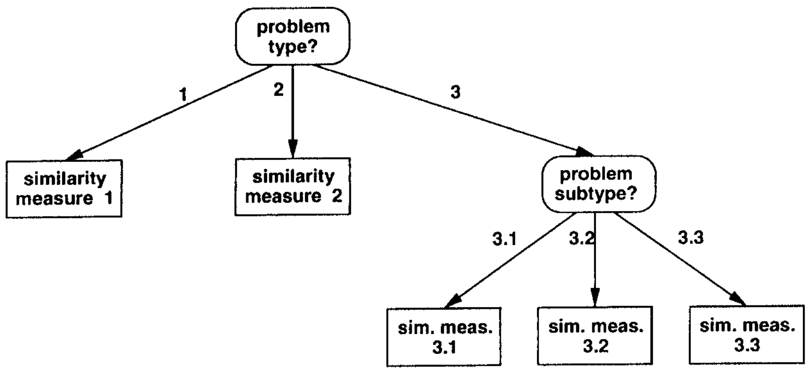

Figure 7 Example of a hierarchically structured similarity. The similarity measures in the leaves may be identical or not.

modifying the retrieved manufacturing plan. The modification uses detailed knowledge on machines, materials, manufacturing steps for clamping and cutting, possible and impossible sequences of steps and so on.

KICS (Knowledge-Intensive Case-based reasoning System) is applied to legal problems, in particular Scottish building law (regulations and former cases) (Yang et al., 1993, 1995). Building plans are to be checked on rule conformity. The hierarchy corresponds to legal theory, KICS resembles more a decision system than a case retrieval system. As a peculiarity, there are five different similarity measures from literal similarity (directly conforming to the rules) to implied similarity (using reasoning to make rule conformity plausible).

Explanation-based reasoning with cases as advocated in Bergman et al. (1994) is hierarchically structured into several levels of abstraction. A case is applicable if its preconditions match the preconditions of the query case on a certain abstraction level and if the operators of the source case (the explanation of that case) lead to the desired result when applied to the query case. The similarity is given by the lowest compatible abstraction level. If for the next lower level no case is available, the problem must be solved (explained) there directly but now within a largely reduced search space.

The cited article describes briefly two applications: MoCAS, Model-based Case Adaptation System for technical diagnosis (for a CNC machine with 110 components and 365 attributes) and PARIS, Plan Abstraction and Refinement in an Integrated System, a general planning system (e.g., for a lathe).

7.2 *Artificially imposed hierarchies*

Sometimes the domain theory leads in a natural way to a tree structure. Sometimes it is artificially generated by clustering the available cases, perhaps to avoid checking all of them at retrieval time. Then the border between clusters may be somewhat artificial and either the cases near the boundary should be stored in more than one part or a query case near a boundary (and a query case with uncertain classification due to missing data) should initiate searches in several branches.

The general case retrieval system ReMind is used in several fields for a hierarchical search (Barletta, 1994). For instance in a help desk application, the first decision level distinguishes major computer and operating systems, the following ones differentiate between types of application. More than one branch can be pursued in parallel.

A hierarchy of a different nature is used by Jagadish (1991) for retrieving similar shapes. Each shape is decomposed into a sequence of rectangles of decreasing size where the area of a rectangle may be added to, or subtracted from, the previous composition. Given a query shape decomposed in the same manner, all shapes in the case base are selected where the first rectangle nearly matches, among those the cases with nearly matching second rectangle are chosen etc. Some provisions are made to prevent a failure if two similar shapes would be decomposed in quite a different way; e.g., if a shape allows two nearly equally good decompositions, both of them are stored in the case base or used as a query case, respectively.

7.3 *Sequential combination*

In document retrieval one often uses a two-step procedure: fast and cheap preselection on the basis of keywords in the index and costly postselection among the candidates on the basis of the full document, for instance, for phrases or within-sentence operators if the index contains only the document numbers.

In case retrieval, several retrieval algorithms are rather expensive; then it makes sense to preselect candidates by a simple algorithm. This will often be an attribute-based search, but in principle any fast method may be used.

8 Complementary issues

8.1 *Non-specific methods*

An innovative method combining several retrieval algorithms has been developed in FABEL, ASPECT (Schaaf, 1995; Schaaf & Voß, 1995). The assumption is here that different measures contribute to the user's vision of similarity. Distances between cases in the case base according to several criteria are computed beforehand and stored if they exceed a threshold. The FABEL implementation uses ODM (mentioned in section 3.3) with several instantiations of its parameters and Topo (section 4.2), among others. At search time, the individual distances can be combined freely. The retrieval uses a novel idea. The distance of the query case to a selected source case is determined. For all stored cases near the selected case, a lower and an upper bound for the distance to the query case are computed using a triangular inequality; all these cases are "sunk" into the depths corresponding to the minimal distance. This is repeated for the next case in the case base with minimal lower bound (which is 0 at the beginning), i.e., a new case is 'fished' from the surface. The minimal and maximal distances of its neighbours are updated: the permissible interval is shrunk. Therefore this any-time algorithm is nicknamed "fish and shrink".

An older attempt to use distances between the source cases for retrieval is Shasha and Wang (1990).

A general procedure for improving search results is relevance feedback: the user evaluates some of the results as to their relevance and the retrieval system uses this information to refine the query. Relevance feedback has not been applied to the systems covered here, perhaps due to the mostly small case bases that do not require an interated search.

Most similarity functions take two cases as arguments. Some have the entire case base as a third argument, e.g., CATO. Certain similarities are applicable to sets of cases, in particular those that measure the largest common subcase. Similarities for sets of cases include some geometric relationships (coinciding points on a grid), smallest common subgraphs and the similarity of logical expressions based on anti-unification and possibly equational theories; see also Jantke (1994).

Fuzzy logic has not only been applied to document retrieval (Wong & Yao, 1995; Cross, 1994), but also to case retrieval (Jaczynski & Trousse, 1994; Bonissone & Ayub, 1993).

9 Conclusions

As we have seen, there is a large variety of similarity concepts in those systems where attribute search or attribute comparison do not suffice.

Most systems found in the literature use unique methods difficult to transfer to different situations. An exception is graph matching and perhaps case retrieval in semantic nets where activation spreading is a fairly general procedure.

Ideally, the case retrieval system finds the case or cases best suited for the overall goal, mostly case analysis (classification, diagnosis, decision support) or case synthesis (planning, configuration, design). However, to find out which case is best suited one usually has to try out all candidates—obviously not the desired solution. Thus a similarity measure is substituted for the suitability or the necessary effort. We have tried to characterize these similarities, but sometimes this is impossible: the search criteria are intricately interwoven in the search algorithm, in particular in some model-based systems.

The reports on existing case retrieval systems give very few hints on the quality of the search results; therefore this aspect has not been covered here. Assessment of retrieval results is already a problem with document retrieval, among other reasons because there is no clearcut boundary between relevant and irrelevant and because relevancy is a fairly subjective measure. The problems become harder with case retrieval; for a deeper discussion see Coulon and Gebhardt (1994, 1994a).

There has been some debate whether a case should in general be divided into a problem part and a solution part. The inspected systems rarely make such a distinction unless the particular application demands it (e.g. given two points, find the best connection between them). Even in diagnostics the diagnosis need not belong purely to the solution part: it can be used in the problem part as a suggestion to be verified or rejected as in CASEY.

There are at least two areas where a larger variety of case retrieval algorithms had been expected: geography and picture recognition. Apart from the route planning systems and two forest fire projects (section 3.4), no geographical case retrieval systems have been found. For a survey on cataloging digital geographic data, a step towards retrieval, see Frank (1994). Many picture recognition algorithms should lead to structural representations on various semantic levels that could be used for case retrieval, but no serious developments have become known.

References

Aamodt, A. and Plaza, E, 1994. "Case-based reasoning: foundational issues, methodological variations, and system approaches" *AI Communications* 7(1) 39–59.

Aghassi, DS, 1990, "Evaluating case-based reasoning for heart failure diagnosis" Technical Report MIT/LCS/TR-478, MIT, Cambridge, MA.

Artymiuk, PJ, Bath, PA, Grindley, HM, Pepperrell, CA, Poirrette, AR, Rice, DW, Thorner, DA, Wild, DJ, Willett, P, Allen, FH, and Taylor, R, 1991, "Similarity searching in databases of three-dimensional molecules and macromolecules" *J. Chemical Information and Computer Sciences* **32** 617–630.

Ashley, KD and Aleven, V, 1994, "A logical representation for relevance criteria" in Wess, S, Althoff, KD and Richter, MM, editors, *Topics in Case-Based Reasoning: First European Workshop, EWCBR-93, selected papers*: *Lecture Notes in Artificial Intelligence 837*, pp. 338–352. Springer.

Ballard, DH and Brown CM, 1982, *Computer Vision* Prentice-Hall.

Ballard, DH, 1981, "Generalizing the Hough transform to detect arbitrary shapes" *Pattern Recognition* **13** 111–122.

Barletta, R, 1994, "A hybrid indexing and retrieval strategy for advisory CBR systems built with ReMind" in Keane, M, Haton, J-P and Manago, M, editors, *EWCBR–94: Second European Workshop on Case-Based Reasoning*, pp. 49–58. AcknoSoft Press.

Bath, PA, Poirrette, AR, Willett, P and Allen FH, 1994, "Similarity searching in files of three-dimensional chemical structures: comparison of fragment-based measures of shape similarity" *J. Chemical Information and Computer Sciences* **34** 141–147.

Bento, C. and Costa, E, 1994. A similarity metric for retrieval of cases imperfectly explained" in Wess, S, Althoff, K-D and Richter, MM, editors, *Topics in Case-Based Reasoning: First European Workshop, EWCBR-93, selected papers*; *Lecture Notes in Artificial Intelligence 837*, pp. 92–105. Springer.

Bento, C, Exposto, J, Francisco, V and Costa, E, 1995. "Empirical study of an evaluation function for cases imperfectly defined" in Haton, J-P, Keane, M and Manago, M, editors, *Advances in Case-Based Reasoning: Second European Workshop, EWCBR–94*, pp. 33–44. Springer.

Bento, C, Macedo, L and Costa, E, 1995, "Reasoning with cases imperfectly described and explained" in Haton, J-P, Keane, M and Manago, M, editors, *Advances in Case-Based Reasoning: Second European Workshop, EWCBR–94*, pp. 45–59. Springer.

Bergmann, R, Pews, G and Wilke, W, 1994, "Explanation-based similarity: a unifying approach for integrating domain knowledge into case-based reasoning for diagnosis and planning tasks" in Wess, S, Althoff, K-D and Richter, MM, editors, *Topics in Case-Based Reasoning: First European Workshop, EWCBR–93, selected papers*; *Lecture Notes in Aritifical Intelligence 837*, pp. 182–196. Springer.

Bonissone, PP and Ayub, S, 1993, "Similarity measures for case-based reasoning systems" in Bouchon-Meunier, B, Valverde, L and Yager, RR, editors, *IPMU '92—Advanced Methods in Artificial Intelligence: 4th Internat. Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Palma de Mallorca*, pp. 161–172. Springer.

Börner, K. and Faßauer, R, 1995, Analogical layout design (Syn*)" in Börner, K, editor, *Modules for Design Support*, vol. 35 of *Fabel-Report*, pp. 59–68. GMD, Sankt Augustin.

Börner, K, 1994, "Towards formalizations in case-based reasoning for synthesis" in Aha, DW, editor, *AAAI–94 Workshop on Case-Based Reasoning*, pp. 177–181, Seattle, WA.

Börner K, 1995, "Conceptual analogy" in Aha, DW and Ram, A, editors, *AAAI 1995 Fall Symposium Series: Adaptation of Knowledge for Reuse*, pp. 5–11, November 10–12, Boston, MA.

Brandstädt, A, 1994, *Graphen und Algorithmen*, Teubner.

Branting, LK and Aha, DW, 1995, "Stratified case-based reasoning: reusing hierarchical problem solving episodes", in *IJCAI–95: Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 384–390. Montreal. Morgan Kaufmann.

Brown, M, 1994, "A memory model for case retrieval by activation passing", PhD thesis, University of Manchester.

Buehrer, DJ and Chang, CC, 1993, "Application of a reciprocal confluence tree unit to similar-picture retrieval", in Abel, D and Ooi, BC, editors, *Advances in Spatial Databases: Proc. Third International Symposium SSD '93*, pp. 437–442, Singapore. Springer.

Bunke, H and Messmer, BT, 1994, Similarity measures for structured representations" in Wess, S, Althoff, K-D and Richter, MM, editors, *Topics in Case-Based Reasoning: First European Workshop, EWCBR–93, selected papers; Lecture Notes in Artificial Intelligence 837*, pp. 106–118. Springer.

Bunke, H and Sanfeliu, A, 1986, "Introduction [to a special issue on syntactic pattern recognition]" *Pattern Recognition* **19** 249–253.

Bures, MG, Martin, YC, and Willett P. 1994, "Searching techniques for databases of three-dimensional chemical structures" in Eliel, EL and Wilen SH, editors, *Topics in Stereochemistry, vol. 21*, pp. 467–511. Wiley.

Burkhard, H-D and Lenz, M, 1996, "Case retrieval nets: Basic ideas and extensions", in Burkhard, H-D and Lenz, M, editors, *Fourth German Workshop on Case-Based Reasoning: System Development and Evaluation*, Informatik-Berichte, pp. 103–110. Humboldt-Universität.

Cawkell, AE, 1993, "Picture-queries and picture databases" *J. Information Science* **19** 409–423.

Coulon, C-H and Gebhardt, F. 1994, "Evaluation of retrieval methods in case-based design", Fabel-Report 24, GMD, Sankt Augustin.

Coulon, C-H and Gebhardt, F, 1994a, "Evaluation of retrieval methods in case-based reasoning" in Keane, M, Haton, J-P and Manago, M, editors, *EWCBR–94: Second European Workshop on Case-Based Reasoning*, pp. 283–291, Chantilly. AcknoSoft Press.

Coulon, C-H and Steffens, R, 1994, "Comparing fragments by their images" in Voß, A, editor, *Similarity Concepts and Retrieval Methods*, vol. 13 of *Fabel-Reports*, pp. 36–44. GMD, Sankt Augustin.

Coulon, C-H, 1995a, "Automatic indexing, retrieval and reuse of topologies in architectural layouts" in Tan, M and Teh, R, editors, *The Global Design Studio—Proceedings of the 6th International Conference on Computer-Aided Architectural Design Futures*, pp. 577–586, Singapore.

Coulon, C-H, 1995b, "General geometric and topologic retrieval and adaptation (TOPO)", in Börner, K, editor, *Modules for Design Support*, vol. 35 of *Fabel-Report*, pp. 33–45. GMD, Sankt Augustin.

Cross, V, 1994, "Fuzzy information retrieval", *J. Intelligent Information Systems* 29–56.

Enser, PGB, 1995, "Pictorial information retrieval", *J. Documentation* **51** 126–170.

Esposito, F, Semeraro, G and Malerba, D, 1992, "A syntactic distance for partially matching learned concepts against noisy structural object descriptions", *Int. J. Expert Systems* **4** 409–451.

Falkenhainer, B, Forbus, KD and Gentner, D, 1989, "The structure-mapping engine: algorithm and examples" *Artificial Intelligence* **41** 1–63.

Flickner, M, Sawhney, H, Niblack, W, Ashley, J, Huang, Q, Dom, B, Gorkani, M, Hafner, J, Lee, D, Petkovic, D, Steele, D and Yanker, P, 1995, "Query by image and video content: the QBIC system" *Computer* **28**(9) 23–32.

Flusser, J, and Suk, T, 1993, "Pattern recognition by affine moment invariants" *Pattern Recognition* **26** 167–174.

Fouqué, G and Matwin, S, 1993, "A case-based approach to software reuse" *J. Intelligent Information Systems* **2** 165–197.

Fox, S and Leake, DB, 1995, "Modelling case-based planning for repairing reasoning failures" in Cox, M. and Freed, M, editors, *Proceedings of the AAAI Spring Symposium on Representing Mental States and Mechanisms*, pp. 31–38. AAAI Press.

Frank, S, 1994, Cataloging digital geographic data in the information infrastructure: a literature and technology review" *Information Processing & Management* **30** 587–606.

Gati, G, 1979, "Further annotated bibliography on the isomorphism disease" *J. Graph Theory* **3** 95–109.

Gebhardt, F, 1993, "Rasterähnlichkeit" *Fabel-Report* 19, GMD, Sankt Augustin.

Goldstone, RL, 1994, "The role of similarity in categorization: providing a groundwork" *Cognition* **52** 125–157.

Gräther, W, 1994, "Computing distances between attribute-value representations in an associative memory" in Voß, A, editor, *Similarity Concepts and Retrieval Methods*, vol. 13 of *Fabel-Reports*, pp. 12–25. GMD, Sankt Augustin.

Gudivada, NV and Raghavan, VV, 1995, "Design and evaluation of algorithms for image retrieval by spatial similarity" *ACM Trans. Information Systems* **13** 115–144.

Haigh, KZ and Shewchuk, JR, 1994, "Geometric similarity metrics for case-based reasoning" in *Case-Based Reasoning: Papers from the 1994 AAAI Workshop, Seattle*, pp. 157–163. AAAI Press.

Haigh, KZ and Veloso, MM, 1995, "Route planning by Analogy" in Veloso, M and Aamodt, A, editors, *Case Based Reasoning Research and Development: First International Conference, ICCBR–95*, pp. 169–180. Springer.

Haller, F, 1985, ARMILLA—ein installationsmodell" Institut für Baugestaltung, Universität Karlsruhe.

Hopcroft, JE and Wong, JK, 1974, "Linear time algorithm for isomorphism of planar graphs" in *Conference Record of 6th Annual ACM Symposium on Theory of Computing*, pp. 172–184. Seattle, WA. ACM.

Hough, PVC and Arbor, A, 1962, "Method and means for recognizing complex patterns" U.S. Patent No. 3 059 654.

Jaczynski, M and Trousse, B, 1994, "Fuzzy logic for the retrieval step of a case-based reasoner" in Keane, M, Haton, J-P and Manago, M, editors, *EWCBR–94: Second European Workshop on Case-Based Reasoning*, pp. 313–321. AcknoSoft Press.

Jagadish, HV, 1991, "A retrieval technique for similar shapes" *SIGMOD Record* **20**(2) 208–217.

Jantke, KP, 1993, "Nonstandard concepts of similarity in case-based reasoning", in Bock, H-H, Lenski, W and Richter, MM, editors, *Information Systems and Data Analysis: Prospects—Foundations—Applications, Proceedings of the 17th Annual Conference of the GfKl, Univ. of Kaiserslautern*, pp. 28–43, Kaiserslautern. Springer.

Jurišica, I, 1994, "Context-based similarity applied to retrieval of relevant cases" Technical Report DKBS-TR-94–5, University of Toronto.

Kettler, BP and Hendler, JA, 1994, "Evaluating a case-based planning system" in *Case-Based Reasoning: Papers from the 1994 AAAI Workshop*, Seattle, pp. 157–163. AAAI Press.

Kolodner, JL, 1993. *Case-Based Reasoning* Morgan Kaufmann.

Koton, PA, 1988, *Using Experience in Learning and Problem Solving*, PhD thesis, MIT.

Layne, SS, 1994, Some issues in the indexing of images" *J. American Society for Information Science* **45** 583–588.

Leake, DB, 1992, "Constructive similarity assessment: using stored cases to define new situations" in *Proceedings of the 14th Annual Conference of the Cognitive Science Society*, pp. 313–318, Bloomington, IN. Erlbaum.

Marir, F and Watson, I, 1994, "Case-based reasoning: a categorized bibliography" *The Knowledge Engineering Review* **9** 355–381.

Medin, DL, Goldstone, RL and Gentner, D, 1993, "Respects for similarity" *Phychological Review* **100** 254–278.

Mehlhorn, K, 1984, *Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness* Springer.

Messmer, BT and Bunke, H, 1995, "Subgraph isomorphism in polynomial time" Technical Report IAM 95–003, Universität Bern, Institut für Informatik und angewandte Mathematik.

Milosavljević, A, 1993, "Discovering sequence similarity by the algorithmic significance method" in Hunter, L, Seals, D and Shavlik, J, editors, *Proceedings, First International Conference on Intelligent Systems for Molecular Biology ISMB–93*, pp. 284–291. AAAI Press.

Milosavljević, A, 1996, "Repeat analysis" in *Imperial Cancer Research Fund Handbook of Genome Analysis*, Chapter 13, 4. Blackwell Scientific.

Muñoz-Avila, H and Huellen, J, 1995, "Retrieving cases in structured domains by using goal dependencies" in Veloso, M and Aamodt, A, editors, *Case Based Reasoning Research and Development: First International Conference, ICCBR–95*. pp. 241–252. Springer.

Muñoz, H, Paulokat, J and Weß, S, 1995, "Controlling a non-linear hierarchical planner using case replay" in Haton, J-P, Keane, M and Manago, M, editors, *Advances in Case-Based Reasoning: Second European Workshop, EWCBR–94*, pp. 266–279, Springer.

Niblack, W, Barber, R, Equitz, W, Flickner, M, Glasman, E, Petkovic, D, Yanker, P, Faloutsos, C and Taubin, G, 1993, "The QBIC project: querying images by content using color, texture and shape" Research Report, IBM Research Division, Almaden Research Center, San Jose, California.

Pepperrell, CA, 1991, *An Investigation of Methods for Similarity Searching in Databases of Three-Dimensional Chemical Structures*. PhD thesis, University of Sheffield.

Petrakis, EGM and Faloutsos, C, 1994, "Similarity searching in large image databases" Technical Report CS-TR-3388, University of Maryland.

Petrakis, EGM and Orphanoudakis, SC, 1993, "A generalized approach for image indexing and retrieval based on 2-D-strings" Technical Report 103, ICS-FORTH, Heraclio, Crete.

Petrakis, EGM, 1993, *Image Representation, Indexing and Retrieval Based on Spacial Relationships and Properties of Objects*, PhD thesis, University of Crete.

Reiser, C and Kaindl, H, 1994, "Case-based reasoning for multi-step problems and its integration with heuristic search" in *Case-Based Reasoning: Papers from the 1991 AAAI Workshop*, Seattle, pp. 101–105, AAAI Press.

Ricci, F, Mam, S, Marti, P, Normand, V and Olmo, P, 1994, "CHARADE: a platform for emergencies management systems" Technical Report 9404–07, Istituto per la Ricerca Scientifica e Tecnologica, Povo.

Rissland, EL, Skalak, DB and Friedman, MT, 1995a, "BankXX: supporting legal arguments through heuristic retrieval" *J. Artificial Intelligence and Law* **4**.

Rissland, EL, Skalak, DB and Friedman, MT, 1995b, "Evaluating a legal argument program: the BankXX experiments" Technical Report UM-CS-1995–30, University of Massachusetts.

Rougegrez, S, 1993, "A case-based reasoning system independent of a representation of cases in terms of features" in *Case-Based Reasoning and Information Retrieval: Papers from the 1993 Spring Symposium*, Stanford, CA, pp. 98–104, AAAI Press.

Sanders, KE, Kettler, BP and Hendler, JA, 1995, "The case for structure-based representations Technical Report CS-TR-3468, University of Maryland at College Park.

Schaaf, JW and Voß, A, 1995, "Retrieval of similar layouts in FABEL using AspecT" in Tan, M and Teh, R, editors, *The Global Design Studio—Proceedings of the 6th International Conference on Computer-Aided Architectural Design Futures*, pp. 617–626. National University of Singapore.

Schaaf, JW, 1995, "'Fish and Sink': an anytime-algorithm to retrieve adequate cases" in Veloso, M and Aamodt, A, editors, *Case-Based Reasoning Research and Development: First International Conference, ICCBR-95, Proceedings*, pp. 538–547. Springer.

Shasha, D and Wang, T-L, 1990, "New techniques for best-match retrieval" *ACM Trans. Information Systems* **8** 140–158.

Shavlik, JW, 1991. "Finding genes by case-based reasoning in the presence of noisy case boundaries" in *Proceedings of the DARPA Case-Based Reasoning Workshop*, pp. 327–338. Morgan Kaufman.

Shepherd, TS, Uttal, W, Dayanand, S and Lovell, R 1992, "A method for shift, rotation, and scale invariant pattern recognition using the form and arrangement of pattern-specific features" *Pattern Recognition* **25** 343–356.

Smyth, B and Keane, MT, 1994, "Retrieving adaptable cases" in Wess, S, Althoff, K-D and Richter, MM, editors, *Topics in Case-Based Reasoning: First European Workshop, EWCBR-93, selected papers; Lecture Notes in Artificial Intelligence 837*, 209–220. Springer.

Smyth, B and Keane, MT, 1995, "Retrieval and adaptation in Déjà Vu, a case-based reasoning system for software design" in *Adaptation of Knowledge for Reuse: a 1995 AAAI Fall Symposium: Working Notes*, pp. 99–105, MIT, Cambridge.

Sycara, K, Navin chandra, D, Guttal, R, Koning, J and Narasimhan, S, 1992, "CADET: a case-based synthesis tool for engineering design" *Int. J. Expert Systems* **4** 157–188.

Tammer, EC, Steinhöfel, K, Schönherr, S and Matuschek, D, 1995, "Anwendung des Konzeptes der Strukturellen Ähnlichkeit zum Fallvergleich mittels Term- und Graph-Repräsentationen" *Fabel-Report* 38, GMD, Sankt Augustin.

Tanaka, E, 1990, "Structural distances and similarities" *Information Processing Society of Japan* **31** 1270–1279 (in Japanese).

Veloso, MM and Carbonell, JG, 1991, "Variable-precision retrieval in analogical problem solving" in *Proceedings of the DARPA Case-Based Reasoning Workshop*, pp. 93–106. Morgan Kaufman.

Watson, I and Marir, F, 1994, "Case-based reasoning: a review" *The Knowledge Engineering Review* **9** 327–354.

Wolverton, M and Hayes-Roth, B, 1994, "Retrieving semantically distant analogies with knowledge-directed spreading activation" in *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA, vol. 1, pp. 56–61. AAAI Press.

Wolverton, M, 1994, "Retrieving semantically distant analogies" PhD thesis, Stanford University.

Wong, SKM and Yao YY, 1995, "On modeling information retrieval with probabilistic inference" *ACM Trans. Information Systems* **13** 38–68.

Yang, SA, Robertson, D and Lee, J, 1993, "KICS: a knowledge-intensive case-based reasoning system for statutory building regulations and case histories" in *The Fourth International Conference on Artificial Intelligence and Law: Proceedings of the Conference*, Amsterdam, pp. 254–263. ACM.

Yang, S-A, Robertson, D and Lee, J, 1995, "Use of case-based reasoning in the domain of building regulations" in Haton, J-P, Keane, M and Manago, M, editors, *Advances in Case-Based Reasoning: Second European Workshop, EWCBR-94*, pp. 292–306. Springer.