Nested Graph-Structured Representations for Cases

Luís Macedo[1,2], and Amílcar Cardoso[2]

[1] Instituto Superior de Engenharia de Coimbra, Quinta da Nora,
3030 Coimbra, Portugal
lmacedo@sun.isec.pt
[2] Centro de Informática e Sistemas da Universidade de Coimbra, Pinhal de Marrocos,
3030 Coimbra, Portugal
{macedo, amilcar}@eden.dei.uc.pt

Abstract. This paper describes an approach to representing cases as nested graph-structures, i.e., as hierarchically, spatially, temporally and causally interconnected nodes (case nodes), which may be themselves recursively described by other sets of interconnected nodes. Each case node represents a case piece (sub-case). An adjacency matrix may represent these nested graph-structured cases. Within our approach, new cases are constructed using an iterative context-guided retrieval of case nodes from multiple cases. In order to illustrate the expressiveness of this case representation approach, we discuss its application to the diagnosis and therapeutics of neurological diseases, to architectural design and to storytelling. Some issues that come out of this approach, like its contribution to the representation of cases of CBR and to integrate ordinary and creative reasoning, are discussed.

1 Introduction

The need to keep to a minimum the knowledge engineering effort required to construct case libraries, and the need of efficiency are the two main reasons to use simple case representations in some CBR systems. These simple case representations usually comprise two unstructured sets of attribute-value pairs or case features: the problem and the solution features [12]. There is no description of the relationships or constraints between the features of a case. Moreover, these simple case representations are characterised by having a low number of indexed features [36]. The retrieval simply involves the standard nearest neighbour algorithm.

However, the construction of CBR systems in complex real-world domains critically requires complex case representations. CBR systems for these domains are usually characterised by having a large problem space. As described by Watson and Perera [36], and Leake [16], the larger the problem space is, the more likely the case coverage is lower, and so, the more likely the case matching is poorer. Consequently, the system may propose distant and less useful solutions, which will require more adaptation effort.

Hierarchical-structured representations of cases [18, 36] aid overcoming the large problem space drawback as they provide the implementation of the *divide and conquer* approach, offering the capability to treat subparts of cases as full-fledged cases. This way they enable solving complex problems by recomposition of sub-solutions [21]: a large problem (or a large goal) is divided into several smaller sub-problems (sub-goals), which can be independently solved using CBR. This means that the problem space may be broken into sub-problem spaces, each one having less features than the higher level problem space. The benefit of considering cases as set of pieces, called *snippets* [28], instead of monolithic entities, can improve the results of a CBR system in that solutions of problems may result from the contribution of multiple cases. Therefore, they allow minimising the problems that appear when using parts of multiple monolithic cases, particularly, the lot of effort taken to find the useful parts in them.

However, conflicts may appear from the recomposition of sub-solutions. Context-guided retrieval has been proposed to overcome this problem [19, 36].

Case representations cannot reflect only the hierarchical decomposition relationship between the objects or case pieces that constitute a case. Actually, in complex real-world domains, although hierarchy is an important dimension to take into account, there are other kinds of relations between objects in a case, as, for example, spatial (especially in design) [8, 10, 37], temporal (especially in planning) [1], or simply causal explanations [31].

Graph-structured case representations, comprising objects and relations among them, are a suitable approach to deal with the complex case representation problem, since they allow the capability of expressing the relations between any two objects in a case, they allow the variation of the set of relations used in different cases, they allow the continuous addition of new relations to the set of relations used in a continuously updated case library, and they allow the implementation of both hierarchical and non-hierarchical case decomposition. Consequently, they provide a more flexible and higher expressive power than attribute-value representations. However, they have the problem of requiring complex retrieval mechanisms (e.g.: a structure similarity is usually needed) that causes significant computational costs and a hard case acquisition task. Actually, this is the main reason why some CBR systems have used representations that fall between graph-structured and unstructured representations. The research group of Maryland University [2] has proposed a parallel structure matching, and an automated case acquisition method to overcoming matching and case acquisition costs, respectively. Plaza [27] and Börner [8] have also proposed approaches to structure similarity measure.

In this paper we will focus on a nested graph-structured representation (NGSR) of cases. These are described by pieces (sub-cases - represented by the nodes) and a set of relations among them (represented by the edges), with the particularity that each one of those case pieces may embed another graph, and so on. Each one of these case pieces is considered, for indexing, matching, retrieving and validation purposes, as an individual case, which facilitates the reuse of parts of multiple cases to construct a new solution. We do not retrieve monolithic cases, but instead we construct new cases using an iterative context-guided retrieval of case pieces from multiple cases. An adjacency matrix representation for graphs is used to efficiently know whether or not a node is related (and how is related) with another node.

Our approach to case representation is presented in the next section. Three application domains illustrate it. In Section 3 we give an overview of the solution construction process. In section 4 we discuss some issues provided by our approach. At last, a conclusion about our work is made in section 5.

2 Case Representation

Within our approach, a case is represented by a graph-structure, comprising a set of interrelated case pieces: the case pieces are represented by the nodes, called case nodes (sub-case is another synonym), and the relations are represented by the edges. These cases are a kind of nested graph structures as their case nodes may be recursively described by another set of interconnected case nodes, i.e., another graph which we call the *internal context* (or the node graph) of the case node. This way, these nested graphs allow representing the decomposition of a case into sub-cases. It is worth notice that a case itself is also a node (a node of a higher graph-structure that represents a case of CBR, as we will discuss in section 4). The *internal context* of a case node comprises its sub-case nodes and the relations among them. It describes the relevant aspects of the case node, as for example its constraints, its functionalities, its behaviour and its structure (its constituent parts) [13]. It is worth notice that two case nodes may have *internal contexts* with a non-null intersection, i.e., they may be described by two different node graphs sharing a common set of case nodes. There may be four main types of relations between case nodes: hierarchical (e.g.: decomposed into, described by, etc.), spatial (e.g.: close to, touches, supported by, etc., [8; 10, 37]), temporal (e.g.: meets, during, overlaps, etc., [1]), or simply causal justifications or explanations (e.g.: cause, implies, explains) [31]. Notice that there may be more than one relation between two case nodes. Additionally, each relation may be directed or undirected. Several case nodes of the same case may explain the existence of another node in that case. We call the *external context* of the case node to the set of case nodes and relations that surrounds it. This means the *external context* of a case node *n* is also a graph (its neighbourhood graph) comprising the case nodes and the edges that surround it.

Figures 1, 3 and 5 present three illustrative examples of real-world cases from three different domains: diagnosis and treatment of neurological diseases (RECIDE [4, 5]), architectural design (e.g: FABEL project [12]) and storytelling (e.g.: MINSTREL [34], TALESPIN [23], SPIEL [9], [22]). Figures 2, 4 and 6 present their correspondent NGSR, showing evidences for the main features of NGSR described above, like the integration in a same case of spatial, temporal, causal and hierarchical links between case nodes, or the possibility of treating a set of case nodes as a unique case node (just like *propositions* in conceptual graphs [33]). Nested oval curves implicitly represent the hierarchical relations between case nodes[1].

[1] For the sake of simplicity and lack of space the cases are partially represented.

Patient Description:

Name: I. B. ; Sex: female; Age: 22

Diagnosis: Microadenoma (Prolactinoma)

Disease History:

Signals and Symptoms: Headaches;

Previous Therapeutics: 0.5 mg/day of Parlodel and 10 mg/day of Sibelium

Exams:

CAT (Computerised Axial Tomography:

NMR (Nuclear Magnetic Resonancy):

Date: 93/06/23 -> Description: The Saddle Diafragm is large and the Turkish Saddle is partially destroyed;

Blood Analysis:

Date: 92/01/27 -> Description: level of PRL hormone = 139.3 (very high); TSH=1.2 (normal) ; ….

Date: 92/03/22 -> Description: level of PRL hormone = 123 (high); TSH=1.2 (normal) ; ….

Therapeutics: 10 mg/day of Sibelium; 0.5 mg/day of Parlodel

Fig. 1. A real-world case from the neurological domain

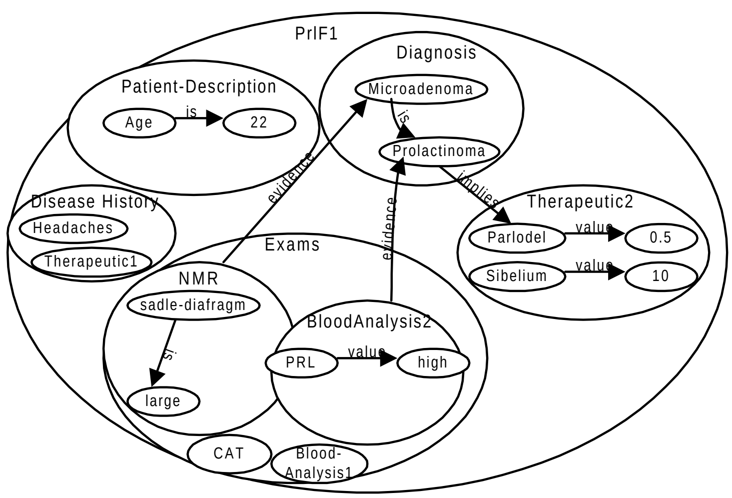

Fig. 2. The correspont NGSR for the case of Figure 1

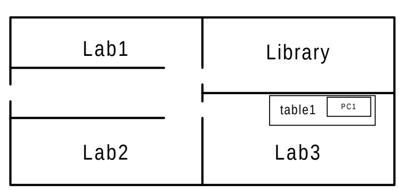

Fig. 3. A real-world case from the architectural domain

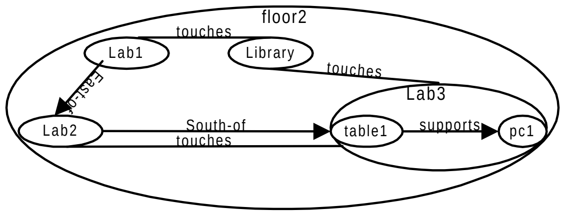

Fig. 4. The correspont NGSR for the case of Figure 3

(s1)There was a penguin who lived by himself on a floating iceberg in a cozy little igloo. (s2) One day a storm destroyed all the igloos around him. (s3) The penguin was truly sorry his friends have no home. (s4) So, he decided he must share his igloo with his friends. (s5) He invited everyone to spend the night in his house. (s6) All the animals accepted his offer to stay with him. (s7) His home got so full that the penguin slept outside.

Fig. 5. A fable story

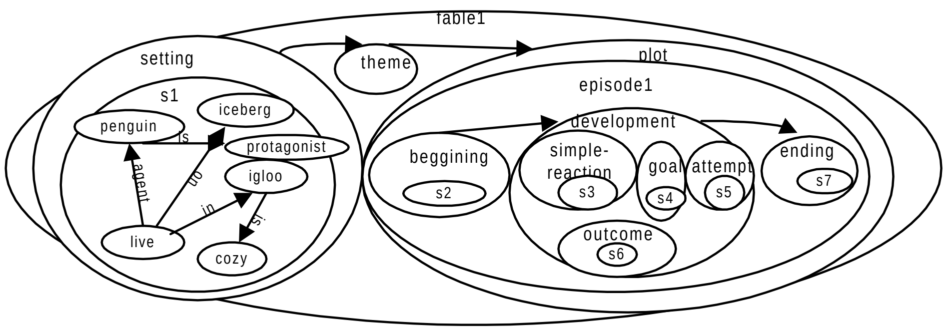

Fig. 6. The correspondent NGSRfor the case of Figure 5. s1, s2, s3, …, s7 are the sentences represented in the case of Figure 5[2]

Mapping the case nodes to a set of contiguous integers (0, |N|-1), where N is the number of case nodes of the case, we may adopt the adjacency matrix approach to represent the relations (the edges) between the case nodes (Figures 7 and 8). Notice that according to the theory of graphs an adjacency matrix is a |N|×|N| matrix of booleans. The value of the element *A(i,j)* of an adjacency matrix *A* (where *i* and *j* are integers) is either true (represented by the number one) or false (represented by a zero), depending upon whether or not node *j* is adjacent to node *i*. Remember also that a node *j* is adjacent to a node *i* if there is an edge from *i* to *j*. We adopt a slightly different adjacency matrix because we don't want to represent only the presence of an edge (or edges) between two case nodes, but also the relation(s) that the edge(s) rep-

[2] For more details about the language and story representation approach see [22, 26, 30].

resents (represent). Therefore, the value of the element *A(i,j)* is a set whose elements are the relation(s) between case node *i* and case node *j*. If that set is empty then the value of *A(i,j)* is 0. For example, a possible matrix and necessary mapping of the case nodes for the case of Figure 1 is portrayed in Figures 7 and 8.

The row and column *i* of the adjacency matrix efficiently give us the *external* and the *internal context* of the case node mapped to the integer *i*. For example, the *external context* of the case node 19 comprises the nodes 2, 10 and 20, related with it by the relations δ, α e β, respectively.

Case Node	**Integer**
prlf1	1
diagnosis	2
exams	3
NMR	10
saddle diafragm	11
microadenoma	19
prolactinoma	20

Relation	**Symbol**
described-by (decomposed-into)	δ
evidence	α
is	β
cause	χ

Fig. 7. An example of a mapping for the case of Figure 1

	1	2	3	.	.	10	11	.	.	19	20	.	.
1	0	$[\delta, \chi]$	δ	.	.	0	0 .	.	.	0	0	.	.
2	0	0	0	.	.	0	0	.	.	δ	δ	.	.
3	0	0	0	.	.	δ	0	.	.	0	0	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
10	.	.	.	.	.	0	δ	.	.	α	0	.	.
11	.	.	.	.	.	0	0	.	.	0	0	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
19	0	0	0	.	.	0	0	.	.	0	β	.	.
20	0	0	0	.	.	0	0	.	.	0	0	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.

Fig. 8. An adjacency matrix representation for the case of Figure 1 using the mapping of Figure 7

3 Overview of the Solution Construction Process

A problem proposed to the system is a partially complete solution, as it may comprise a set of possibly connected case nodes, which we call the query case nodes. The construction of a solution consists on completing that partially complete solution by iteratively retrieving case nodes from previous cases. For each query case, and starting by the broader one (the hierarchically higher), the system has to complete its *external* and *internal contexts* and fill them (i.e., construct their record description structure[3]).

[3] A case node may have a record structure associated with it to describe, for example, its type.

To do that the system retrieves a case node from memory according to an algorithm similar to the one presented in [19] (in CBR the most similar case node is usually retrieved). However, we do not use the similarity metric described there because, although it takes into account the context similarities, it also takes into account similarities between the addresses of the nodes (a code that uniquely represents the position of the nodes in a case), which is not represented anymore in the current representation approach. Instead of that similarity metric, we may apply one of the several structural similarity metrics like the presented by Börner et al. [8] (specifically based on the computation of the maximal clique), by Sanders et al [29] (based on an algorithm for finding sub-graph isomorphisms) or by Plaza [27] (based on the concept of antiunification of cases). Another structural similarity metric (our own) that we adopt is based on the error-correcting algorithms for sub-graph isomorphism detection usually used in Pattern Recognition [24]. It measures the structural similarity between the contexts of the candidate case nodes and those of the (possibly partially missing) query case node. The edges of the retrieved case node are then reused by (added to) the query case node of the partially complete solution. These edges follow from or to missing (or partially missing) case nodes of the *internal* and *external context* of the concerned case node. It is worth notice that these edges are not just pointers to missing case nodes, but also may embed suggestions for some aspects that those partially or completely missing case nodes should have (for example, their functionalities, their types, etc). These missing case nodes are then filled, using the algorithm mentioned above. The process stops when there is no more missing case nodes.

4 Discussion

Other works on hierarchical CBR are the stratified CBR technique presented in [7], the recursive hierarchical CBR of Déjà Vu system [32] and the Schank's Dynamic Memory [30]. Some systems like JULIA [15], PRODIGY/ANALOGY [35], and CAPlan/CbC [25], NIRMANI [36], etc., use hierarchical-structured representations for cases. CHIRON and CAPER [29], GREBE [6], SME [11], FABEL [12], Saxex [3], etc., are examples of CBR and analogical reasoning projects that have used graph-structured case representations.

Besides the benefits of graph-structured representations already presented in other related works such as presented in [29, 12], we additionally discuss here some other positive aspects that come out of our NGSR approach.

First, NGSR allows the representation of the decomposition of a case into sub-cases (sub-case nodes), with all the benefits that come from, as we mentioned above in the introduction. This means cases are stored as interconnected individual pieces (case nodes) facilitating the access to all useful case pieces from several cases, and thus improving the efficiency of retrieval. In contrary, CBR systems dealing with monolithic cases have two steps to access the useful parts of previous cases: they need to retrieve the whole case and then they take a lot of effort to find its relevant part(s).

An issue worth addressing is the case piece size, because CBR systems' efficiency and capability to solve new problems depend on that. It could be expected that a system dealing with smaller case pieces would be less efficient than one dealing with bigger ones (or with no case pieces at all), because of the greater number of retrieval operations that have to be performed. However, this drawback is overwhelmed by providing direct access to the case pieces in memory, avoiding unnecessary processing.

We also think that the capability of a CBR system to solve problems grows when the case piece size decreases: using smaller case pieces, we may dispose of a higher number of combinations to construct the solution.

The approach we presented constitutes a generic case representation tool to support several kinds of domains (diagnosis, planning and design). This is mainly because NGSR allows the representation not just of decomposition (or hierarchical) relations, but also of temporal, spatial and (simply) causal relations between objects of a case (even in a same case). A consequence of supporting the integration of those relations in a same case allows the representation of a considerable amount of information in a case, which obviously improves CBR.

Our retrieval process is context-guided, taking advantage of the benefits pointed by Watson and Perera [36]. Additionally, it is an iterative retrieval process which may be more efficient than usually graph retrieval, as it is confined to the problem of comparing *internal* and *external context* graphs of the case node, which are significantly smaller graphs than whole case graphs. Then the structure similarity algorithms (e.g.: sub-graph finding, etc.) may become less complex. However, empirical proof of this has not been done yet.

In our approach, we represent cases of CBR [17]. This way the reasoning steps (adaptations made, similarity parameters used, etc.) may be stored for further use. Each case of the case-base is explained by other cases. Therefore, the case-base stores information of how a specific case was constructed. The case-base it is a kind of history of the reasoning performed before. Considering that in our approach cases are themselves represented as nodes, and that NGSR allows the representation of relations between different grainsize case nodes, the case-base may be seen itself as a continuously updated case, since it is a network of interconnected nodes (case or sub-case nodes) (see Figure 9). The system may pick successful and unsuccessful ideas of reasoning strategies from previous cases of CBR, when dealing with similar reasoning problems (e.g.: which adaptation strategies were used when a specific case node was added to a particular case, which similarity parameters were used, etc.).

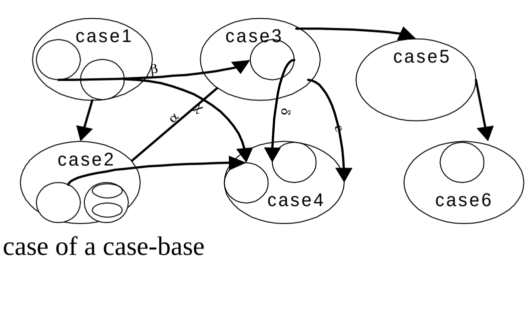

Fig. 9. A case of a case-base

As a consequence of the most aspects we have been pointing, NGSR provides a suitable approach to support creative reasoning (it is particularly appropriate for domains like storytelling and architectural design). This makes sense if we consider one of the main definitions of creativity, the combination theory, that says creativity consists on relating or combining previously unrelated things [19]. Actually, this idea may be easily transferred to the context of NGSR as relating previously unrelated case nodes. Using a process that does not retrieves the case node with the most similar context, but instead, for instance the second or third most similar, a case node may be placed in the new case in a context that is different from the context it had in the original case. This way new and even non-obvious combinations of case nodes may be constructed, although taking some cognitive risks that sometimes lead to bizarre solutions. Nonetheless, a required adaptation process may avoid that bizarreness. Moreover, creative reasoning basically requires taking cognitive risks in some degree. This idea is very related with the relaxing retrieval technique used by Schank to increase creativeness of CBR [31].

If the retrieval process is repeated several times, each time choosing a case node with a different degree of structural similarity [19], i.e., changing the ranking of the candidate case nodes, then a wide variety of solutions may be constructed to a same problem. This is a case-based computational implementation of the concept of divergent production of solutions [20], which, as claimed by the psychologist Guilford [14], is the foundation of the creative process, in contrast to convergent production used in ordinary problem solving. He defended that fluency, a creative ability deeply connected with divergent production, and measured by the number of solutions that one may give to a same problem, may play an important role in creativity. NGSR may strongly contribute to improve the fluency of CBR systems, as it provides the recombination of case nodes in several ways, through restructuring previous cases. Another two important creative abilities are analysis and synthesis [14], i.e., the ability to decompose things into pieces, and the ability to recompose them, respectively. As we previously explained, both abilities are also provided by NGSR, as well as by other hierarchical CBR approaches presented above. As claimed by Guilford, a flexible knowledge representation is another important issue in creative reasoning. NGSR provides that flexibility, allowing the representation of cases from disparate domains, as we have described in section 2. Furthermore, the case nodes may be considered with different contexts, and thus viewed in different ways.

Our approach integrates ordinary and creative case-based problem solving, since we use the same case representation, the same retrieval, reusing and revision processes, although the goals that guide these processes are different: while creative problem solving is appropriateness and originality-guided, ordinary problem solving is just appropriateness-guided. These distinct driven-goals imply some slight differences in those processes. For example, a slight distinction in the retrieval process is presented as follows. In creative problem solving, presented for instance in architectural design, storytelling or musical composition [18, 19], the most useful case pieces are not necessarily the ones with more similarity metric value. We think that the most useful case piece [15] in these cases is the one with: (i) higher similarity value; (ii) which gives originality to the new case; (iii) and that does not confront the coherence and meaningfulness of the new case. However, in domains such as diagnosis of neurological diseases, that originality-guided retrieval must be avoided. When solving a problem in a particular domain, the system must know whether or not originality (and which degree of originality) is pretended for the solution. It may pick that information from previous cases of CBR of that particular domain. For example, when solving a diagnosis problem it should take knowledge from previous cases of CBR diagnosis that tell it to use no originality.

The main drawback of this approach may be the required computational costs. However, research on graph theory is being made to overwhelm or at least decrease that.

5 Conclusions

We have presented an approach to representing cases as nested graphs: cases are split into hierarchically, spatially, temporally or causally interconnected pieces (nodes) that may be described by another set of interconnected pieces, and so on. An adjacency matrix may represent a graph. We use an iterative context-guided retrieval of case nodes. A structural similarity function compares the contexts of the query case node and of the candidate case nodes. This case representation approach allows the representation of cases from different types of domains (diagnosis, design and planning). Among other things, it supports the integration of ordinary and creative problem solving and the representation of cases of CBR.

Aknowledgements

We would like to thank Carla Gomes, physician at the Hospitais da Universidade de Coimbra, for her valuable contribution.

References

1. Allen, J.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM 26(11), (1983) 832-843
2. Andersen, W., Hendler, J., Evett, M., Kettler, B.: Massively Parallel Matching of Knowledge Structures. In: H. Kitane & J. Hendler (eds.): Massively Parallel Artificial Intelligence. AAAI/MIT Press, Menlo Park, CA, Cambridge, MA (1994)
3. Arcos, J., & Mántaras, R.: Perspectives: A Declarative Bias Mechanism for Case Retrieval. In: Leake, D., Plaza, E. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1266. Springer-Verlag, Berlin Heidelberg New York (1997) 279–290
4. Bento, C., Macedo, L., Costa, E.: Reasoning with Cases Imperfectly Described and Explained. In: Wess, S., Althoff, K., Richter, M. (eds.): Topics in Case-Based Reasoning.

Lecture Notes in Computer Science, Vol. 837. Springer-Verlag, Berlin Heidelberg New York (1994) 263-272.
5. Bento, C., Exposto, J., Francisco, V., Costa, E.: Empirical Study of an Evaluation Function for Cases Imperfectly Explained. . In: Wess, S., Althoff, K., Richter, M. (eds.): Topics in Case-Based Reasoning. Lecture Notes in Computer Science, Vol. 837. Springer-Verlag, Berlin Heidelberg New York (1994) 215-223
6. Branting, L.: Integrating rules and precedents for classification and explanation: automating legal analysis. Tech. Rep. AI90-146, Dept. of Computer Sciences, University of Texas at Austin (PhD Thesis) (1990)
7. Branting, L., Aha, D.: Stratified case-based reasoning: Reusing hierarchical problem solving episodes. In: Proceedings of the International Joint Conference on Artificial Intelligence. (1995) 384-390
8. Börner, K., Pippig, E., Tammer, E., Coulon, C.: Structural Similarity and Adaptation. In: Smith, I., Faltings, B. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Computer Science, Vol. 1168. Springer-Verlag, Berlin Heidelberg New York (1996) 58-75
9. Burke, R., Kass, A.: Retrieving Stories for Case-Based Teaching. In: Leake, D. (ed.): Case-Based Reasoning – Experiences, Lessons, & Future Directions. AAAI Press/MIT Press, Menlo Park, CA, Cambridge, MA, London (1996) 93-109
10. Coulon, C.: Automatic indexing, retrieval and reuse of topologies in architectural layouts. In: Tan, M., The, R. (eds.): The Global Design Studio – Proceedings of the 6th International Conference on Computer-Aided Architectural Design Futures. (1995)
11. Falkenhainer, B., Forbus, K., & Gentner, D.: The structure-mapping engine: algorithm and examples. Artificial Intelligence 41 (1990) 1-63
12. Gebhardt, F., Voβ, A., Gräther, W., Schmidt-Beltz, B.: Reasoning with Complex Cases. Kluwer Academic Publishers, Norwell, MA (1997)
13. Goel, A.: Representation of Design Functions in Experience-Based Design. In: Brown, D., Walderon, M., Yosnikawa, H. (eds.): Intelligent Computer Aided Design. Elsevier Science (1992)
14. Guilford, J.: Intelligence, creativity and their educational implications. Robert Knapp, San Diego, CA (1968)
15. Kolodner, J.: Judging Which is the “Best” Case for a Case-Based Reasoner. In: Case-Based Reasoning: Proceedings of a Workshop. Morgan-Kaufmann, San Mateo, CA (1989)
16. Leake, D.: CBR in Context: The Present and Future. In: Leake, D. (ed.): Case-Based Reasoning – Experiences, Lessons, & Future Directions. AAAI Press/MIT Press, Menlo Park, CA, Cambridge, MA, London (1996) 3-30
17. Leake, D., Kinley, A., Wilson, D.: A Case Study of Case-Based CBR. In: Leake, D., Plaza, E. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1266. Springer-Verlag, Berlin Heidelberg New York (1997) 371-382
18. Macedo, L., Pereira, F., Grilo, C., Cardoso, A.: Plans as Structured Networks of Hierarchical and Temporal Related Case Pieces. In: Smith, I., Faltings, B. (eds.): Advances in Case-Based Reasoning. Lecture Notes in Computer Science, Vol. 1168. Springer-Verlag, Berlin Heidelberg New York (1996) 234-248
19. Macedo, L., Pereira, F., Grilo, C., Cardoso, A.: Experimental Study of a Similarity Metric for Retrieving Pieces from Structured Plan Cases: its Role in the Originality of Plan Case Solutions. In: Leake, D., Plaza, E. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1266. Springer-Verlag, Berlin Heidelberg New York (1997) 575-586

20. Macedo, L. A Model for Creative Problem Solving based on Divergent Production of Solutions. In: Prade (ed.): Proceedings of the 13th European Conference on Artificial Intelligence. John Wiley & Sons, Ltd. (1998)
21. Maher, M, & Zhang, D.: CADSYN: using case and decomposition knowledge for design sinthesis. In: J. Gero (Ed.): Artificial Intelligence in Design. Butterworth-Heinmann, Oxford, UK (1991)
22. Mandler, J.: Stories, Scripts, and Scenes: Aspects of Schema Theory. Lawrence Erlbaum Associates, Hillsdale, NJ (1984)
23. Meehan, J.: The metanovel; Writing stories by computer. Tech. Rep. #74, doctoral dissertation. Yale University, Dept. of Computer Science (1976)
24. Messmer, B., Bunke, H.: A new algorithm for error-tolerant subgraph isomorphism detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (5), (1998)
25. Munõz-Avila, H., Huellen, J.: Retrieving Cases in Structured Domains by Using Goal Dependencies. In: Veloso, M., Aamodt, A. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1010. Springer-Verlag, Berlin Heidelberg New York (1995) 241-252
26. Norman, D., Rumelhart, D.: Explorations in Cognition. W. H. Freeman and Company, San Francisco (1977)
27. Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases. In: Veloso, M., Aamodt, A. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1010. Springer-Verlag, Berlin Heidelberg New York (1995) 265-276
28. Redmond, M.: Distributed Cases for Case-Based Reasoning; Facilitating Use of Multiple Cases. In: Proceedings of the 7th National Conference on Artificial Intelligence. AAAI Press/MIT Press, Menlo Park, CA, Cambridge, MA, London (1990)
29. Sanders, K, Kettler, B., Hendler, J.: The Case for Graph-Structured Representations. In: Leake, D., Plaza, E. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1266. Springer-Verlag, Berlin Heidelberg New York (1997) 245-254
30. Schank, R.: Dynamic Memory: A theory of learning in computers and people. Cambridge University Press, New York (1982)
31. Schank, R.: Explanation Patterns: Understanding Mechanically and Creatively. Erlbaum, Hillsdale, NJ (1986)
32. Smyth, B, Keane, M.: Design à la Déja Vu: Reducing the Adaptation Oberhead. In: Leake, D. (ed.): Case-Based Reasoning – Experiences, Lessons, & Future Directions. AAAI Press/MIT Press, Menlo Park, CA, Cambridge, MA, London (1996) 151-166
33. Sowa, J.: Knowledge Representation. PWS Publishing Company, Boston, MA (1995)
34. Turner, S.: The Creative Process: A Computer Model of Storyttelling and Creativity. Lawrence Erlbaum Associates, Hillsddale, NJ (1994)
35. Veloso, M.: Learning by Analogical Reasoning in General Problem Solving. Ph D thesis. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA (1992)
36. Watson, I., Perera, S.: The Evaluation of a Hierarchical Case Representation Using Context Guided Retrieval. In: Leake, D., Plaza, E. (eds.): Case-Based Reasoning Research and Development. Lecture Notes in Computer Science, Vol. 1266. Springer-Verlag, Berlin Heidelberg New York (1997) 255-266
37. Winston, P.: Learning Structural Descriptions from Examples. In: Brachman, L. (ed.): Readings in Knowledge Representation. Morgan Kaufmann, Los Altos, CA (1985)