Proceedings of the ICCBR 2012 Workshops

Luc Lamontagne and Juan A. Recio-García (Editors)

Preface

We are pleased to present the proceedings of the workshops of the Twentieth International Conference on Case-Based Reasoning (ICCBR 2012) held in Lyon in September 2012. Over the last two decades, ICCBR workshops have been a vibrant forum where exploratory and novel CBR contributions were made by AI researchers. This year's program makes no exception and includes the papers of three workshops exploring various areas of CBR research:

- Case-Based Reasoning in the Health Sciences is a well established workshop investigating how CBR techniques and methodologies can contribute to this application domain and demonstrating new CBR systems developed for medical applications.
- Traces for Reusing Users Experience (TRUE) is a new workshop theme exploring how recorded episodes of human activities can be captured as cases and exploited to build reasoning systems.
- Process-oriented CBR is the second edition of a workshop dedicated to the integration of CBR techniques with process-oriented techniques like task planning, business process management and workflow management.

In addition, this volume includes research summaries of the talks given during the fourth Doctoral Consortium, a successful event where students can present their PhD project and obtain feedbacks to help them progress in their research.

We would like to thank all those who contributed to the success of this workshop program, including the presenters and authors of the workshop papers along with the program committees for the quality and timeliness of their reviews. In particular, our thanks go to the workshop organizers for their support and cooperation in the preparation of the program.

We very much appreciated the collaboration of the program chairs Ian Watson and Belen Díaz-Agudo. Our special thanks go to the conference chair Amélie Cordier for her unwavering support and to Béatrice Fuchs for her precious help in the reproduction of the proceedings.

We hope that the participants will enjoy this year's workshop program and that these proceedings will inspire discussions and lead to innovative research in the future.

Luc Lamontagne and Juan A. Recio-García September 2012

Table of Contents

TRUE: Traces for Reusing Users' Experiences – Cases, Episodes, and Stories

The ICCBR 2012 Doctoral Consortium

Process-oriented Case-based Reasoning

Workshop at the
Twentieth International Conference on
Case-Based Reasoning
(ICCBR 2012)

Lyon, France
September, 2012

Mirjam Minor and Stefania Montani (Eds.)

Co-Chairs

Mirjam Minor
University of Trier, Germany

Stefania Montani
University of Piemonte Orientale, Italy

Programme Committee

David Aha, Naval Research Lab, USA
Pedro A. González-Calero, Complutense University of Madrid, Spain
David Leake, Indiana University, USA
Hector Muoz-Ávila, Lehigh University, USA
Ashwin Ram, Georgia Institute of Technology, USA
Juan A. Recio-García, Universidad Complutense de Madrid, Spain
Barry Smyth, University College Dublin, Ireland
Barbara Weber, University of Innsbruck, Austria

Preface

The second workshop on Process-oriented Case-based Reasoning (PO-CBR) is dedicated to address the challenges of integrating Case-Based Reasoning (CBR) with process-oriented research areas, like Business Process Management (BPM) and workflow management, in business/software process support. BPM is a set of activities aimed at defining, executing, monitoring and optimizing business processes (BP), with the objective of making the business of an enterprise as effective and efficient as possible, and of increasing its economic success. Such activities are highly automated, typically by means of the workflow technology. BPM activities, and BP optimization in particular, may ask the enterprise to be able to flexibly change and adapt the predefined process schema, in response to expected situations (e.g. new laws, reengineering efforts) as well as to unanticipated exceptions and problems in the operating environment (e.g. emergencies). The agile workflow technology is the technical solution which has been invoked to deal with such adaptation and overriding needs. In order to provide an effective and quick workflow change support, many agile workflow systems share the idea of recalling and reusing concrete examples of changes adopted in the past. To this end, Case-based Reasoning (CBR) has been proposed as a natural methodological solution.

While the contributions of last years workshop discussed mainly solutions for basic PO-CBR issues like retrieval, adaptation and monitoring of business processes this years workshop goes one step further by raising efficiency, maintenance, and learning issues for process-oriented cases. An invited talk [Montani] on the optimization of processes will provide insights into a PO-CBR system that has been successfully applied in the medical domain for a few years. A learning approach for similarity in order to manage sets of process cases [Gunawardena and Weber] is followed by two papers on retrieval efficiency [Bergmann et al., KendallMorwick et al.], which both implement a MAC-FAC approach in different domains by incidence. A case study on applying a PO-CBR system in logistics concludes the workshop. We wish to thank all who contributed to the success of this workshop, especially the authors, the Program Committee, and the editors of the workshop proceedings!

Mirjam Minor　　September 2012
Stefania Montani

Case-based Optimization of Medical Processes

Stefania Montanii

DISIT, Computer Science Institute, Università del Piemonte Orientale, Alessandria, Italy

Abstract. Healthcare institutions are increasingly facing pressure to reduce costs, while, at the same time, improving the quality of care. In order to reach such a goal, healthcare administrators and expert physicians need to evaluate the services the institution provides. Service evaluation requires to analyze medical processes, which are often automated and logged by means of the workflow technology.

Specifically, by medical processes we mean clinical guidelines and clinical pathways. These processes typically capture the actions which will *most commonly* represent best practice for *most patients*, *most of the time*. Nonetheless, high individual variability, local resource constraints and treatment obsolescence may lead to the need of revising/redesigning the process, if optimization goals have to be fulfilled. Business Process Management techniques can be adopted to this end.

In particular, the *agile workflow* technology can be relied upon for adaptation. As it is well known, many agile workflow systems share the idea of recalling and reusing concrete *examples of changes* adopted in the past. In this context, Case-based Reasoning (CBR) can thus be exploited as a natural methodological solution.

We will discuss how to adopt a case-based approach to support medical process optimization.

Indeed, the main issues characterizing medical processes (i.e.: (1) high individual variability, (2) strong need to fulfill safety requirements, and (3) presence of many "temporal" (e.g. synchronization) challenges) make a CBR approach particularly well suited, but non trivial. Issue (1) strongly justifies the exploitation of episodic knowledge, rather than an effort towards knowledge generalization/abstraction. Issue (2) makes a "pure" retrieval approach (or, at least, an implementation in which case adaptation is not automated) very meaningful, since the human decision maker can not be substituted by an automated reasoning system. However, e.g. issue (3) leads to the need for complex similarity measures, able to deal with temporal information.

In conclusion, medical process optimization involves non-trivial CBR research topics, such as the study of new similarity measures, structured cases retrieval, textual case retrieval, and proper case base maintenance/library organization choices. Lessons learned in this field could then be exported/adapted to other domains.

Scaling similarity-based retrieval of semantic workflows

Ralph Bergmann, Mirjam Minor, Mohd. Siblee Islam, Pol Schumacher, and Alexander Stromer

University of Trier - Department of Business Information Systems II
D-54286 Trier, Germany
[bergmann|minor|islam|pol.schumacher|s4alstro]@uni-trier.de

Abstract. This paper presents an approach for scaling the retrieval of semantic workflow cases. Similarity-based graph-matching approaches have been used in our previous work for the retrieval of semantic workflows. However, their high computational complexity makes it difficult to scale the approach to case bases with a size of more than a few hundred cases. However, many application areas of semantic workflows like scientific workflows or cookery workflows involve a large amount of semantic workflows to be reused. We propose a novel two-step retrieval method for workflows, inspired by the MAC/FAC (many are called, but few are chosen) approach proposed by Forbus et al. An additional computationally efficient retrieval step (MAC stage) is introduced prior to the graph-based retrieval (FAC stage) to perform a pre-selection of potentially relevant cases. It is based on a feature representation of the workflows automatically derived from the original graph-based representation. In the paper, we briefly introduce our previous work on the semantic workflow retrieval and then we describe the pre-selection step in more detail. An evaluation with case bases from the cooking domain has been performed. It demonstrates scalability towards case bases of up to 15000 cases.

1 Introduction

In the past few years, Case Base Reasoning (CBR) systems have significantly improved their ability to deal with large numbers of cases. Indexing techniques like decision trees [1, 2], kd-trees [3], or case retrieval nets [4] are often applied for a large case base to achieve retrieval results within milliseconds. However, if the representation of cases is complex, the retrieval time can be very high, because a complex representation requires complex similarity measures, which are computationally expensive. Workflows require a complex case representation for retrieval. Bergmann and Gil [5] propose a graph-based approach to represent and retrieve workflow cases. The graph-based retrieval is computationally expensive as the similarity computation involves a kind of graph matching. Current experiments have shown that it works sufficiently fast only for cases bases containing less than 200 cases. Today, websites are a source of procedural knowledge, which can be represented as workflows [6]. We can create a large case base using these workflows for CBR applications. The graph-based retrieval is approaching its limits with respect to computational complexity. However, indexing is hard to apply to the graph-based retrieval as discussed in [5], because sets and mapping functions are used in many similarity functions.

As a consequence, we developed a novel, two-step retrieval for workflows inspired by the MAC/FAC ("Many are called, but few are chosen") approach proposed by Forbus, Gentner and Law [7]. An additional retrieval step (MAC stage) prior to the graph-based retrieval mentioned above (FAC stage) is the solution to our problem. Features are extracted from the original graph representation in order to pre-select cases in a computationally cheap retrieval step during the MAC stage. For our experiments, we implemented this approach with the CAKE framework [8].

The remainder of the paper is organized as follows: The MAC/FAC approach for workflows is described in Section 2 including a brief recall of the graph-based retrieval. Experimental results are discussed in Section 3. A conclusion is drawn in Section 4.

2 A MAC/FAC Approach to Workflow Retrieval

Traditionally, workflows are "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [9]. In addition, tasks exchange certain *products*, which can be of physical matter (such as ingredients for cooking tasks) or information. Tasks, products and relationships between the two of them form the *data flow*. Broadly speaking, workflows consist of a set of *activities* (also called *tasks*) combined with *control-flow structures* like sequences, parallel or alternative branches, and loops. Tasks and control-flow structures form the *control-flow*. Today, graph representations for workflows are widely used in process-oriented CBR. In this paper we build upon the workflow representation using semantically labeled graphs developed by Bergmann & Gil [5], which is briefly summarized in section 2.1. This graph representation enables modeling related semantic similarity measures which are well inline with experts assessment. Specific heuristic search algorithms for computing the semantic similarity for graphs have been developed, but their scalability with growing case bases is quite limited. This is caused by the inherent computational complexity of graph similarity.

To overcome this problem, we investigate a retrieval method based on the MAC/FAC [7] idea, similar to what was proposed by Leake and Kendall-Morwick [10, 11]. The basic idea behind MAC/FAC is very simple: it is a two-step retrieval approach that first performs a rough pre-selection of a small subset of cases from a large case base. This pre-selection is the MAC stage ("Many Are Called"), which is performed using a selection method which is computationally efficient even for large case bases. For example, cases may be stored in a relational data base and the pre-selection can be performed by an SQL query [12]. Then, the second step called FAC phase ("Few Are Choosen") is executed, which only uses the pre-selected cases to perform the computationally expensive similarity computation. This method improves the retrieval performance, if the MAC stage can be performed efficiently and if it results in a sufficiently small number of pre-selected cases that allows applying the complex similarity measure for retrieval.

The major difficulty with MAC/FAC retrieval in general is the definition of the filter condition of the MAC stage. Since cases that are not selected by the MAC stage will not occur in the overall retrieval result, the completeness of the retrieval can be easily violated if the filter condition is too restrictive. Hence, retrieval errors, i.e., missing cases will occur. On the other hand, if the filter condition is less restrictive, the number of pre-selected cases may become too large, resulting in a low retrieval performance. To balance retrieval error and performance, the filter condition should be a good approximation of the similarity measure used in the FAC stage, while at the same time it must be efficiently computable to be applicable to a large case base in the MAC stage.

We address this problem by proposing an additional feature-based representation of workflows, which is automatically derived from the original graph-based representation. This representation thus simplifies the original representation while maintaining the most important properties relevant for similarity assessment. The MAC stage then selects cases by performing a similarity-based retrieval using a feature-based similarity measure. This similarity measure will partially use the local similarity functions of the graph-based retrieval but in a more simple manner, ignoring the structural properties of the workflow graph. The resulting feature-based retrieval method is thus more efficient. A further important property of this realization of the MAC stage is that the number of selected cases can be easily controlled. Therefore, we introduce a parameter we call *filter size* s, which specifies the number of cases resulting from the MAC stage. Hence, the MAC stage retrieves the s-most similar cases using feature-based retrieval. The choice of the filter size determines the behavior of the overall retrieval method with respect to retrieval speed and error in the following manner: the smaller the filter size, the faster the retrieval but the larger the retrieval error will become. Hence, an appropriate choice of the filter size is important.

We now introduce our approach in more detail. Next, the basic ideas and the notation used in the graph-based retrieval described by Bergmann and Gil [5] are revisited. Then, the feature-based workflow representation and the related similarity measure used in the MAC phase are described.

2.1 Graph-based Retrieval

We represent a workflow as a directed graph $W = (N, E, S, T)$ where N is a set of nodes and $E \subseteq N \times N$ is a set of edges. Nodes and edges are annotated by a type from a set Ω and a semantic description from a set Σ. Type and semantic description are computed by the two mapping functions $T\colon N \cup E \to \Omega$ and $S\colon N \cup E \to \Sigma$, respectively. The set Ω consists of the types: *workflow node, data node, task node, control-flow node, control-flow edge, part-of edge* and *data-flow edge*. Each workflow W has exactly one workflow node. The task nodes and data nodes represent tasks and data items, respectively. The control-flow nodes stand for control-flow elements. The data-flow edge is used to describe the linking of the data items consumed and produced by the tasks. The control-flow edge is used to represent the control flow of the workflow, i.e., it links tasks with successor tasks or control-flow elements. The part-of edge represents a relation between the workflow node and all other nodes. Σ is a semantic meta data language that is used for the semantic annotation of nodes and edges. In our work we treat the semantic descriptions in an object-oriented fashion to allow the application of well-established similarity measures.

Based on this representation, Bergmann & Gil [5] introduced a framework for modeling semantic workflow similarity. It is based on a local similarity measure for semantic descriptions $sim_\Sigma : \Sigma^2 \rightarrow [0,1]$ that must be formulated for nodes and edges.

The similarity of a query workflow QW and a case workflow CW is then computed by means of a legal mapping $m : N_q \cup E_q \rightarrow N_c \cup E_c$, which is a type-preserving, partial, injective mapping function of the nodes and edges of the query workflow to those of the case workflow. For a particular mapping m the overall workflow similarity $sim_m(QW, CW)$ is computed by a particular aggregation of the local similarity values modeled using sim_Σ (for details see [5]). The overall workflow similarity $sim(QW, CW)$ is then determined by the best possible mapping of that kind, i.e.,

$$sim(QW, CW) = \max\{sim_m(QW, CW) | \text{ legal mapping } m\}.$$

As a consequence of this definition, the computation of the similarity requires the systematic construction of such mappings m, which is the cause for the computational complexity of this approach.

2.2 Feature-based Retrieval

The MAC phase for the proposed retrieval approach is based on a feature-based representation of workflows. A feature-based case base $CB' = \{CW'_1, \ldots, CW'_n\}$ is computed offline, i.e., prior to performing the retrieval. Therefore, each case CW'_i is derived from the corresponding case CW_i of the original graph-based case base CB. In the representation of a feature-based case CW', two types of features are considered: *semantic features* and *syntactic features*. A vector V_{sem} represents the semantic features derived from the workflow graph, while a vector V_{syn} represents the syntactic features, thus $CW' = (V_{sem}, V_{syn})$.

Currently, two semantic features are considered. The first feature is related to the data nodes and is represented by a set $D \subseteq N$. The second feature is related to the task nodes and is represented by a set $A \subseteq N$. Hence, $V_{sem} = (D, A)$ with

$$D = \{n \in N | T(n) = DataNode\}$$

$$A = \{n \in N | T(n) = TaskNode\}$$

These features (together with the related semantic description of the nodes in D and A) can be considered an abstraction of the overall graph, as the linking of the nodes is completely ignored.

The syntactic features, however, are simple numerical features that together build a kind of profile reflecting the size of the graph. Hence, V_{syn} is defined as $V_{syn} \in \mathbb{R}^f$, with f being the number of features. These features reflect the number of the various components the graph consists of. Currently, the derived features are: the number of data flow nodes, number of task nodes, number of control flow nodes, the number of data flow edges and the number of control flow edges.

To perform the MAC/FAC retrieval for a given query workflow QW the related feature-based representation QW' of the query is derived in the same manner as for cases in the case base. The similarity measure sim' that compares a query $QW' = (V_{sem_q}, V_{syn_q})$ with a case $CW' = (V_{sem_c}, V_{syn_c})$ is further specified as follows: For both vectors, separate similarity functions are specified. The computed similarity values are then aggregated into the overall similarity. For the two semantic features D and A, the local similarity measure sim_Σ modeled for the graph-based retrieval is used again, but without applying any mapping. Let's assume, $D_q = \{d_{q_1}, d_{q_2}, ..., d_{q_n}\}$ and $D_c = \{d_{c_1}, d_{c_2}, ..., d_{c_m}\}$. The measure sim_Σ is used to assess the similarity between each pair of nodes (d_{q_i}, d_{c_j}). Based on this, a local similarity measure for D is specified as follows:

$$sim'_\Sigma(D_q, D_c) = \Phi\left(\begin{pmatrix} sim_\Sigma(S_q(d_{q_1}), S_c(d_{c_1})) & \cdots & sim_\Sigma(S_q(d_{q_1}), S_c(d_{c_m})) \\ \vdots & \ddots & \vdots \\ sim_\Sigma(S_q(d_{q_n}), S_c(d_{c_1})) & \cdots & sim_\Sigma(S_q(d_{q_n}), S_c(d_{c_m})) \end{pmatrix}\right)$$

Here, Φ is an aggregation function specified for the matrix (s_{ij}) as follows:

$$\Phi((s_{ij})) = \frac{1}{n} \cdot \sum_{i=1}^{n} max\{s_{ij} \mid j = 1..m\}$$

Hence, for each data node in the query, the best matching data node in the case is selected. Their similarity is aggregated into the overall similarity for D. This is obviously still a kind of mapping, but it is less constraint with respect to the mapping m computed in the graph-based approach, because each node is mapped independent of the mapping of the other nodes and independent of any linking. Thus, the computed similarity is an upper bound for the similarity of the nodes in D that can be achieved by the best mapping m in the graph-based retrieval. The local similarity measure $sim'_\Sigma(A_q, A_c)$ for A, the set of task nodes, is specified analogously. Again, the computed similarity can be considered an upper bound for the similarity of the task nodes.

In addition, the similarity of the syntactic features is considered. Here, we apply a standard similarity measure $sim' : \mathbb{R}^2 \rightarrow [0, 1]$. In order to aggregate the local similarity values into the global similarity, feature weights are considered for the features in V and for the semantic features D and A. Lets assume, $W = (w_1, \ldots, w_f)$ is a vector of feature weights for the corresponding features in

$V = (v_1, \ldots, v_f)$ and w_d and w_a are the feature weights for D and A, respectively. Then, the global similarity between the query and the case for feature-based retrieval is specified as follows:

$$sim'(QW', CW') = \frac{w_d \cdot sim'_\Sigma(D_q, D_c) + w_a \cdot sim'_\Sigma(A_q, A_c) + \sum_{i=1}^{f}(w_i \cdot sim'(v_{q_i}, v_{c_i}))}{w_d + w_a + \sum_{i=1}^{f} w_i}$$

The selection of cases $CW_1, ..., CW_s$ during the MAC phase is performed by a similarity-based retrieval from CB' using the similarity measure $sim'(QW', CW'_i)$. Thereby, the s most-similar cases are retrieved (s is the filter size).

3 Experimental Results

The benefits of our approach are demonstrated by means of some experiments in the cooking domain. Cooking recipes are represented in the form of workflows created by an automated extraction procedure [6]. We measure the retrieval time and the retrieval error for the MAC/FAC approach parameterized by

– the size of the case base,
– the filter size s, and
– the number of retrieval phases (the entire MAC/FAC retrieval vs. the MAC stage only vs. the FAC stage only).

The retrieval error is measured by the percentage of cases missing in the retrieval result of the MAC/FAC approach with the retrieval result of the un-filtered graph-based approach (the FAC stage only) as a base line. Broadly speaking, the retrieval error measures how many cases we are loosing during the MAC phase that are considered to be relevant by the more sophisticated retrieval method of the FAC phase.

We investigated the following hypotheses:

H1. The retrieval time of the feature-based retrieval (MAC stage only) is significantly shorter than the retrieval time of the graph-based retrieval (FAC stage only).
H2. For a certain filter size s, the retrieval time of the MAC/MAC approach is sufficiently shorter compared to the graph-based retrieval (FAC stage only) while the retrieval error is significantly low.

Three case bases with the sizes of 200 cases, 2000 cases, and 15000 cases have been extracted from various cooking recipe websites by using the procedure, described in [6]. Complete recipes are considered and all recipes are chosen randomly. Most of the recipes consist of few tasks and few ingredients used as input for the tasks. Currently, we are only extracting cooking workflows with sequential control-flow.

The experiments are performed for two hundred queries, which are identical for the three case bases.

The extracted features have been specified as introduced in Section 2.2: Only the names of ingredients and tasks have been extracted as semantic features. Furthermore, we do not yet consider an ontology to derive similarity functions for this domain, which can cover the huge number of tasks and ingredients in our large case base. Hence, for this work, the Levenshtein distance measure and standard similarity measures are used for the local similarity computations for the semantic syntactic features.

We implemented our MAC/FAC retrieval approach and run the experiment in the CAKE framework [8]. We run the experiment on Windows 7 Enterprise 64-bit, using an Intel i7 CPU 870 @ 2.93GHz and 8.00 GB ram.

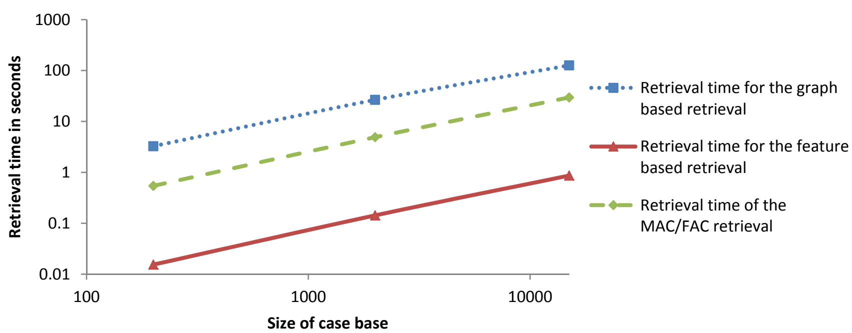


Fig. 1: The average retrieval time of the graph-based, the feature-based, and the MAC/FAC-retrieval for the filter size s set to 12 % of the size of the case base.

The experimental results are shown in Figure 1 to Figure 4. In the first experiment (see Figure 1), the filter size s has been specified without loss of generality by 12 percent of the size of the particular case base.

We can see from the squares and triangles depicted in Figure 1 that the retrieval time for the feature-based retrieval is significantly shorter than the retrieval time for the graph-based retrieval for the three case bases considered. This clearly confirms hypothesis **H1**.

Furthermore, the figure shows that the retrieval time for the MAC/FAC-retrieval (depicted by diamonds) is also significantly shorter than the retrieval time of the graph-based retrieval. We conducted further experiments with a variable filter size s for each of the three case bases. Figure 2 to Figure 4 depict the average retrieval errors and average retrieval times for these runs. The results

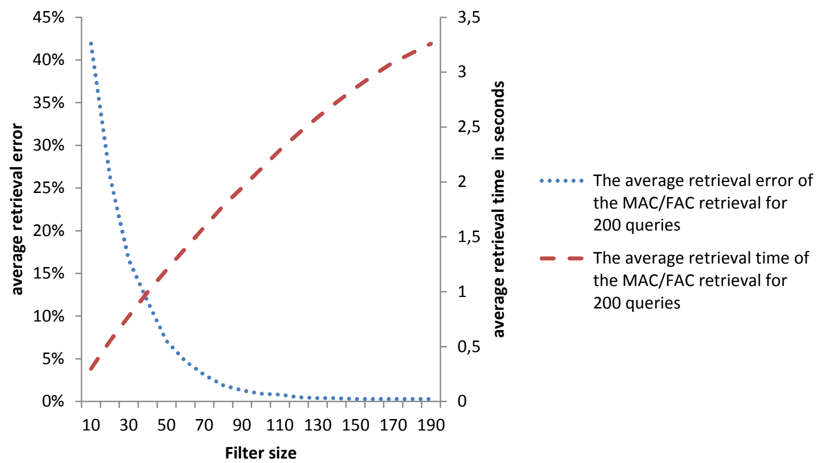


Fig. 2: The average retrieval error and average retrieval time for the case base with 200 cases with different filter sizes s.

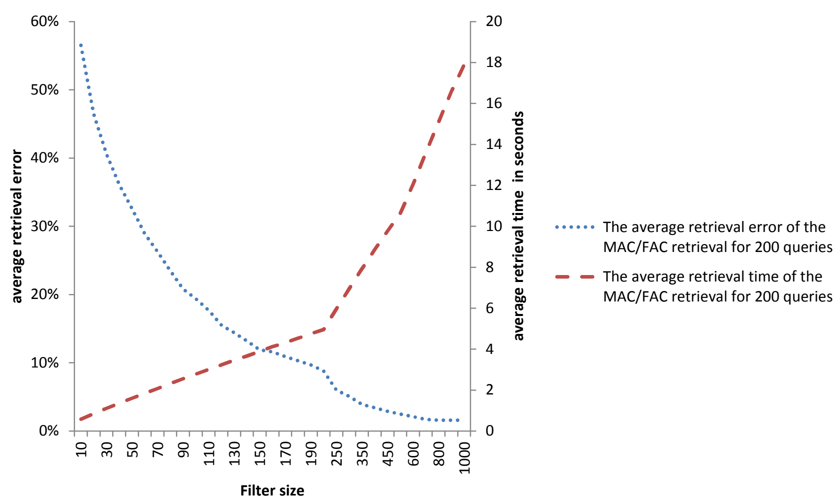


Fig. 3: The average retrieval error and average retrieval time for the case base with 2000 cases with different filter sizes s.

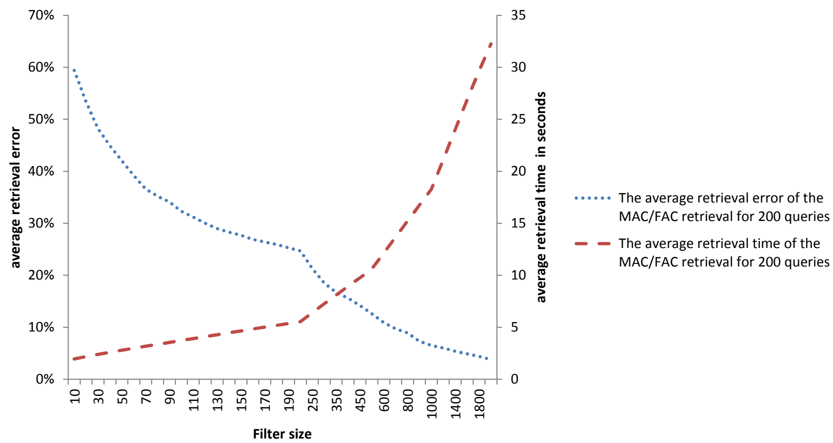


Fig. 4: The average retrieval error and average retrieval time for the case base with 15000 cases with different filter sizes s.

illustrate that for a certain filter size the retrieval time is significantly shorter while the retrieval error is sufficiently low. This confirms our hypothesis **H2**.

4 Conclusion

We presented a new MAC/FAC approach to scale the similarity-based retrieval of semantic workflows. A similar method was proposed by Leake and Kendall-Morwick [10, 11], but they use a different filter method in the MAC phase. Our approach is based on a feature-based representation of workflows, which includes properties that are relevant for the similarity assessment. In our current experiments, we just considered data and task nodes represented as sets as well as some ad-hoc features representing the size of the workflow. Even with this representation we were able to show that the retrieval time can be significantly reduced without introducing a very high error rate. A more elaborated definition of features with related local similarity measures will probably lead to a better performance. Currently, we don't apply any indexing of the features to improve the retrieval speed of the MAC stage. Due to the use of the two set features A and D, the straight-forward application of an existing indexing method is not possible. However, we feel that case-retrieval nets could be extended to be able to cover those features as well. Also methods for optimizing the feature weights and the filter size would be useful. Both issues will be addressed in our future work. Also more detailed empirical evaluations are necessary, involving other domains and more sophisticated ontologies and similarity measures.

5 Acknowledgements

This work was funded by the German Research Foundation, project number BE 1373/3-1.

References

1. Jarmulak, J., Craw, S., Rowe, R.: Self-optimising cbr retrieval. In: Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence. (2000) 376–376
2. Jarmulak, J., Craw, S., Rowe, R.: Genetic algorithms to optimise cbr retrieval. In: Lecture Notes in Computer Science. (2000) 159–194
3. Wess, S., Althoff, K.D., Derwand, G.: Using k-d trees to improve the retrieval step in case-based reasoning. In: Lecture Notes in Computer Science. (1994) 167–181
4. Burkhard, H.D.: Extending some concepts of CBR – foundations of case retrieval nets. In: Case-Based Reasoning Technology. Springer (1998) 17 – 50
5. Bergmann, R., Gil, Y.: Retrieval of semantic workfows with knowledge intensive similarity measures. In: Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011, Springer (2011) 17–31
6. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: WWW'12 Workshop Proceedings, ACM (2012)
7. Forbus, K.D., Gentner, D., K., L.: Mac/fac: A model of similarity-based retrieval. Cognitive Science **19**(2) (1995) 141 – 205
8. Bergmann, R., Freßmann, A., Maximini, K., Maximini, R., Sauer, T.: Case-Based support for collaborative business. In: Advances in Case-Based Reasoning, 8th European Conference, ECCBR 2006, Fethiye, Turkey, September 4-7, 2006, Proceedings. Volume 4106., Springer (2006) 519–533
9. Workflow Management Coalition: Workflow management coalition glossary & terminology. http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf (1999) last access on 05-23-2007.
10. Leake, D.B., Kendall-Morwick, J.: Towards Case-Based support for e-Science workflow generation by mining provenance. In: Advances in Case-Based Reasoning, 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings. (2008) 269–283
11. Kendall-Morwick, J., Leake, D.: A toolkit for representation and retrieval of structured cases. In: Proceedings of the ICCBR 2011 Workshops. (2011) 111–120
12. Schumacher, J., Bergmann, R.: An efficient approach to Similarity-Based retrieval on top of relational databases. In Blanzieri, E., Portinale, L., eds.: Advances in Case-Based Reasoning, 5th European Workshop, EWCBR 2000, Trento, Italy, September 6-9, 2000, Proceedings. Volume 1898 of Lecture Notes in Computer Science., Springer (2000) 273–284 The original publication is available at www.springerlink.com.

On Tuning Two-Phase Retrieval for Structured Cases

Joseph Kendall-Morwick[1] and David Leake[2]

[1] Computer Science Department, DePauw University, Greencastle, IN 46135, U.S.A.
[2] School of Informatics and Computing, Indiana University Bloomington, IN 47408, U.S.A.
josephkendallmorwick@depauw.edu, leake@cs.indiana.edu

Abstract. Process-oriented CBR systems often use structured cases. Consequently, they commonly require effective retrieval methods for such cases, especially when dealing with large processes and/or large case bases. Achieving good performance often depends on sophisticated two-phase retrieval methods which combine an initial and comparatively inexpensive retrieval process, to winnow the cases to be considered, with a more expensive strategy that ranks the remaining cases. Examples of such processes have been shown to provide good retrieval results with limited retrieval time, but studies have focused primarily on overall performance, rather than on how the individual contributions of each phase interact to affect overall performance. This misses an opportunity to determine optimal choices for the component algorithms, in light of the system task and case base characteristics, to tune the system for specific task needs. This paper examines the performance of the two individual phases of two-phase retrieval, demonstrating characteristics of their interaction and providing general lessons for how to design and deploy two-phase retrieval systems.

1 Introduction

In Process-Oriented Case-Based Reasoning, retrieval performance plays a crucial role. CBR systems supporting process-oriented reasoning may deal with a wide range of processes, such as workflows, provenance traces, and plans, all of which require the use of structured cases. These cases generally represent process information in the form of labeled or attributed graphs. Graph matching algorithms have high computational cost, and their performance is constrained by theoretical limits: aspects of the graph matching problem are NP-complete. As a result, heuristic methods may be needed to speed up retrieval, even for small case bases, and especially for interactive applications requiring rapid response time. Greedy algorithms are often applied for these purposes, but depending on the complexity of the data, may not be optimal choices. Beyond the issue of how to compare a query case to a case in the case base is the issue of which cases should be compared. An exhaustive case-by-case comparison wastes much of the allotted computation time comparing irrelevant cases to the query.

Thus achieving desired retrieval performance may require both speeding up the case comparisons which are done and avoiding costly comparisons wherever possible. A common strategy for reducing the number of costly comparisons, dating back to early structure-based retrieval systems such as MAC/FAC [1], is a two-phase retrieval process. In such approaches, an initial "coarse-grained" and comparatively inexpensive retrieval process is used to winnow the cases to be considered. A more expensive "fine-grained" strategy is then applied to the pool of candidate cases selected by the coarse-grained strategy. PO-CBR researchers have noted the potential value of this strategy for scaling up case base sizes [2, 3]. For example, in one comparison, the Phala CBR system, required about 15 minutes to apply its similarity metric to each case in a moderately sized case-base containing about 350 workflow cases, but required only a few seconds on the same hardware to perform the same search with 2-phased retrieval and appropriate indexing for the first phase.

Two-phase retrieval processes enable irrelevant cases to be quickly dismissed and relevant cases to be more closely examined. For time-limited tasks, this strategic use of time can result not only in greater speed, but in greater accuracy, by focusing effort. The use of indexing in 2-phased retrieval makes it possible to add and remove cases with relatively little maintenance. It is flexible in that it can support multiple similarity measures simultaneously, to be applied as needed. In addition, there is also a large and growing body of research and software which can be leveraged when implementing a 2-phased retrieval system for PO-CBR.

Two-phase methods also have some potential drawbacks—for example, the recall of the retrieval process will decrease if the coarse-grained strategy misses relevant cases—but for many tasks have proven useful compared to single-phase approaches. Two-phase methods have been applied both in process-oriented CBR [2] and in graph database systems [4–7].

The overall performance of two-phase retrieval systems has been studied in many task contexts (e.g., [8]). However, little study has been devoted to a key factor in how to design such systems: the individual contributions of the two phases. Analyzing such factors is important because it affects the characteristics to be sought in the design of each phase.

This paper examines the design questions underlying development of two-phase retrieval systems and their ramifications on the performance of the two individual phases of two-phase retrieval. The analysis suggests the value of tailoring overall two-phase retrieval choices to case base characteristics: that is, in addition to any domain-specific choices for retrieval processes within each phase, a two-phase retrieval strategy must be evaluated and the selectivity of each phase tuned for the cases it is intended to work with. Our evaluation provides insight into what the important parameters and performance metrics for 2-phased retrieval are, how changes in these parameters can affect performance, and how properties of the case base can complicate these relationships.

The paper begins with brief background on two-phase retrieval. It then provides motivations for evaluating Phase 1 separately from Phase 2 and how this can be performed, and next presents an evaluation focused on understanding the ramifications of different levels of selectivity in the two phases of the two-phase process, as well as outlining a process for answering key questions in the design of such a system. The evaluation is performed using the Structure Access Interface system (SAI), a retrieval system developed for process-oriented CBR [2]. The paper closes with observations and opportunities for future work.

2 Two-Phase Case Retrieval

Gentner and Forbus's MAC/FAC model, an early two-phase retrieval approach, was inspired by humans' capability to access structured knowledge quickly and soundly [1]. Gentner and Forbus argue that human retrieval speed implies that human retrieval must employ a fast, coarse-grained, massively parallel process which need not fully reflect structural constraints, and that the soundness implies the existence of a more thorough and precise process, which weighs structural constraints. The MAC/FAC algorithm models this behavior by performing retrieval in two sequential phases. The MAC (Many are Called) phase involves comparing candidate cases (structures) through flat representations of features of their structural content. The FAC (Few are Chosen) phase chooses a range of the top ranked cases from the MAC phase and re-orders them according to a more precise and computationally complex similarity metric. The smaller set of the top ranked cases from the FAC phase is returned.

The addition of the FAC Phase can dramatically speed up retrieval by using a $O(log\ n)$ time lookup of cases indexed by their features. Only the cases deemed most likely to be similar will then reach the FAC Phase and have their structure examined and mapped in a more costly process (potentially requiring computation of graph isomorphism).

More recently, many other systems have applied two-phased retrieval approaches. The use of such approaches is a strong trend in the growing research area of graph databases [9]. Many graph database systems use a two-phased approach in which graphs are indexed by structural features used for an initial coarse-grained retrieval phase. During this phase, structural details of the graph are not loaded into memory, but are instead inferred through comparison of their association with flat indices representing structural summaries. Examples of graph database systems employing such a technique are GraphGrep [7], G-Hash [4], Periscope/GQ [5], and gIndex [6]. Additionally, the SAI system joined this collection as a graph database system specifically tailored to the needs of PO-CBR researchers [2].

3 Evaluating Phase 1 and Phase 2 Individually to Understand Ramifications of Window Sizes

This section analyzes how each phase effects various performance metrics. The first subsection outlines the original MAC/FAC approach, the second subsection outlines how this approach can be adapted to address time constraints to accommodate interactive systems, and the third subsection outlines a method for comparing performance of each phase to determine which must be tuned when addressing performance problems.

We use the following terminology for two-phase retrieval: *Phase 1* refers to the coarse-grained retrieval process aimed at efficiently selecting a subset of cases for more complete consideration, and *Phase 2* to the more computationally complex fine-grained ranking process. The maximum number of cases Phase 1 provides to Phase 2 is the Phase 1 Window, shortened to *Window 1*. The number of cases returned from the Phase 2 process for further processing by the CBR system is called the Phase 2 Window, shortened to *Window 2*.

Effects of Limiting Phase 1 Retrieval by Similarity: In Phase 1 of the original MAC/FAC model, cases are filtered according to the number of features they share with the query structure. The structure in memory sharing the most features with the query is selected, along with any others matching at least 90% as many features. In this sense, the MAC Phase works as a filter, drastically reducing the number of cases in contention to be matched against the query, and making the matching problem feasible for a more computationally expensive structure mapping comparison. Graph databases use a similar approach, in which features may be mined and used to approximately summarize structural details. These features are recorded as indices for the cases, enabling Phase 1 to retrieve a complete set of potentially relevant cases (cases which share many indices with the query) with sub-linear complexity with respect to the number of cases in the case-base. For such a Phase 1 process, the number of matches provided for consideration by Phase 2 depends on two main factors:

– The average size of clusters of cases with high similarity to each other
– The distinguishing power of the features selected for filtering

Applying Phase 2 processing to all cases within 10% of the best match provides some assurance as to the quality of the result from Phase 2. For instance, if 99% of the time the top-ranked case according to the ranking of Phase 2 has a 90% or greater match with the query based on Phase 1 features, then Phase 2 will output the top case in the case-base over 99% of the time. Unfortunately, providing this assurance requires that there be no constraint on the number of cases considered by Phase 1, and thus, there is also no constraint on the amount of time Phase 2 may require. We can avoid this problem by setting a limit on Window 1.

Using a Fixed Retrieval Window to Bound Run Time: If response time is important, as in an interactive system, we can set a bound on run time by limiting the number of cases brought forth from the inexpensive Phase 1 to the significantly more expensive Phase 2. In the MAC/FAC model, cases are judged against each other relative to the degree to which they matched features within the query graph. We note rather than simply using this as a filter, it can be used to generate a ranking. For example, Phase 1 can create a nearest-neighbor ranking of the cases in the case-base according to the similarity of their features to those of the query, with Window 1 set to the maximum number of cases Phase 2 can process

Ideal	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Ranking 1	5	4	3	2	1	15	14	13	12	11	10	9	8	7	6
Ranking 2	1	2	3	4	6	5	7	8	9	10	11	12	13	14	15

Fig. 1. Example Rankings

within the time allowed for Phase 2. Phase 1 will no longer be returning every case within a given similarity threshold of the query; as a result, the expected response time for a query can be bounded. Window 1 can be adjusted for the desired trade-off between response time and the similarity of the retrieved cases to the query.

Comparing Phase 1 and Phase 2 Rankings for Credit/Blame Assignment: Evaluation of two-phase retrieval algorithms has traditionally been performed by examining performance of the system as a whole. The problem with this approach is that, if the desired case is not retrieved, it fails to identify which component of the system is most responsible for failure. Either phase may be responsible: Phase 1 may fail to rank the most relevant cases into the top cases in Window 1, making it impossible for the Phase 2 algorithm to find the most relevant cases. Similarly, Phase 2 may be provided with the most relevant case, but fail to rank it higher than other cases provided from Phase 1. To determine which bears more of the blame, we can compare the rankings from Phase 1 and Phase 2. Assuming that the finer-grained strategy of Phase 2 can be used as a "gold standard," increased difference between the rankings suggests blame for Phase 1. However, comparing similarity rankings, particularly in 2-phased retrieval, is less straightforward than might be expected. The following paragraphs examine how such a comparison should be done for each retrieval phase.

Evaluating Phase 1 retrieval: In a study of case ranking for single-phase retrieval, Bogaerts and Leake [10] provide methods for comparing the results of a single-phase retrieval system to an ideal ranking of cases, to evaluate different retrieval algorithms. Adding a second phase to the retrieval process adds complications which may produce unexpected results in isolated cases. However, we have conducted an experiment which suggests that methods which are effective in comparing rankings for single-phased case retrieval systems may also be the most effective for comparing phase 1 and phase 2 rankings. This section first describes the complications which raise questions about assessing rankings for two-phase retrieval, and then sketches our experimental result.

For an example of how the problem of comparing phase 1 and phase 2 rankings can be complicated, suppose that Phase 1 happens to always return the top Window 1 cases of the ideal ranking, but in the opposite order. If Window 1 is large, this ranking scores very low when evaluated for single-phase retrieval. However, the end result of using this ranking for Phase 1 of the two-phase algorithm will be identical to the result of Phase 1 returning all cases in the ideal order: Phase 2 will re-rank the cases, so all that matters is that Phase 2 be provided with the correct set of Window 1 cases.

Another issue, addressed for single-phase systems by Bogaerts and Leake, concerns what they dub "the k-boundary" problem. For reasons of space, we cannot describe this in detail, but it arises if a set of cases with the same similarity to the query straddles the retrieval window boundary, so that only some of the cases are provided to the next step. In such cases, obvious methods of assessing retrieval quality may provide counter-intuitive results, making it difficult to determine which ordering is best. Depending on the similarity metrics used, this problem may arise frequently in both phase 1 and phase 2. For instance, we have used a simple matched feature count for phase 1 similarity assessment, which very frequently results in ties. For phase 2, we have used a covered edge count as a similarity metric, which often results in ties for small graphs.

Comparing ordering may also result in other anomalies for two-phase processes. For example, the Phase 1 ranking of the top Window 1 cases in opposite of ideal order will also score lower than some rankings which do not contain all of the top Window 1 cases. In such cases, the lower-scoring ranking will actually provide better performance than the higher scoring ranking! In this sense, traditional rank quality measures are not guaranteed to be a good measure for the effectiveness of Phase 1 ranking in every instance. Figure 1 illustrates such a case when Window 1 is 10 and Window 2 is 5. The Phase 2 ranking will score lower than ranking 1 with a traditional weighted rank measure, yet ranking 2 will yield better results than ranking 1, meaning that the traditional rank measure did not properly identify which ranking was superior.

However, such anomalies are generally unlikely to arise. In the prior example, sufficiently increasing or decreasing Window 1 or Window 2 eliminates the anomaly. In general, if the ranking generated by Phase 1 is a good approximation of the ranking generated by Phase 2, the combined algorithm will be more likely to locate the most similar cases. To test whether considering rank order when comparing Phase 1 and Phase 2 rankings provides better information about the culpability of Phase 1 versus Phase 2 when the system fails to retrieve desired cases, we conducted experiments with a two-phase retrieval system in which the Phase 1 ranker was identical to the Phase 2 ranker, except for a predetermined random error introduced in Phase 1. We created 2 Phase 1 rankers, each with differing amounts of error. We used two rank measures to compare the rankings generated by each of these Phase 1 rankers: one measure which considered the ordering as part of the quality measure, and one which only considered the presence of the desired cases in the Window 2 set of cases (a recall measure). We ran 1000 tests on a synthetic dataset, with Window 1 sizes of 20 and 25 and Window 2 sizes 5 and 10, with error differences from 0.01 to 0.30 and base error of 0.1. We then computed rank quality for both Phase 1 rankers with both measures, iteratively, until enough judgments were made to indicate which ranker had a higher error to a statistically significant degree. In all trials, the measure which considered order detected a statistically significant difference between the performance of the two phase 1 rankers faster than the recall measure, and only required an average of 5.5% of the tests recall required to find a statistically significant difference with a Z test using 99% confidence intervals. Thus considering the order in which phase 1 ranks cases provided better information on the source of retrieval problems.

4 An Approach to Tuning Two-Phase Retrieval Method Window Settings

In this section, we outline a process by which a system designer can answer basic questions about how to implement and deploy a 2-phased retrieval subsystem. Specifically, we will examine the trade-offs involved in choosing Window 1 and Window 2 settings. We examined this question by implementing a simple two-phased retrieval algorithm within the SAI framework and applying it to two different datasets. The test retrieval algorithm indexes cases by creating indices representing two features from two different nodes linked by an edge. This is a simplification of the indexing strategy used by GraphGrep, which indexes graphs by extracting paths of such linked features of arbitrary length. The structural comparison component is a modified greedy algorithm.

Step 1: Assessing the Dataset. The first step in our process is to assess relevant properties of the case-base itself for prevalence of distinguishing features. Our first test dataset consists of scientific workflows downloaded from the myExperiment website [11]. We have frequently used this dataset to evaluate the Phala CBR system for assisting authorship of scientific workflows [12]. In myExperiment the number of unique features is relatively large in comparison to the number of nodes within cases in the case-base (specifically, there are 0.65 unique features per node). This results in high performance for our simple retrieval strategy, since the uniqueness of features leads to a larger index space and more distinguishing indices (for 341 graphs, there are 1627 unique indices).

myExperiment			PubChem		
Window 1	Memory	Time	Window 1	Memory	Time
5	21573	1095	5	20142	5997
8	31768	1688	8	31146	10316
10	37581	1965	10	38811	13296
15	51053	2660	15	54403	19232
20	62741	3130	20	68887	25295
25	71527	3438	25	83618	30688
30	79996	3771	30	96774	35105

Table 1. Resources Used per Window 1 Size

myExperiment				PubChem			
Window 1 / Window 2	3	5	10	Window 1 /Window 2	3	5	10
5	88.96	84.23		5	58.20	57.50	
8	92.27	89.01		8	59.83	59.62	
10	93.08	89.98	88.61	10	61.52	61.49	59.91
15	94.34	91.90	90.68	15	65.40	65.70	64.08
20	94.67	92.36	91.21	20	69.86	70.26	68.60
25	94.68	92.46	91.33	25	73.40	73.87	72.17
30	94.97	92.93	91.62	30	77.01	77.49	75.72

Table 2. Average Rank Similarity

Step 2: Determining Performance Needs. The next step is to determine what is needed or desired in terms of performance. Performance of the system can be measured in many ways other than the problem-solving accuracy of the CBR system itself. Response time is important in interactive systems and constraints on the hardware used (e.g., memory limitations) are always relevant. To examine the impact the choice of indexing strategy has for a variety of window settings, and also to examine the effectiveness of our indexing strategy, we generated rankings for both phases with the myExperiment dataset.

The results of this experiment are listed in the left-hand portions of Tables 1 and 2. Table 1 lists the response time of the entire retrieval process and the RAM consumed during retrieval. Response time is indicated in milliseconds and RAM is indicated by an implementation-independent unit. Actual RAM consumption is a function of this variable which also includes a constant factor for the size of the specific representation of structural elements in a specific implementation. Table 2 indicates how similar the ranking for the top Window 2 cases produced by Phase 1 is to the ranking produced by Phase 2, using a rank measure as described in the previous section.

Figure 2 shows that, as Window 1 size increases, RAM used and retrieval time also increase. This is expected, but what is more interesting is relating this data to the data in Table 2, which in turn shows that rank quality improves as either Window 2 decreases or Window 1 size increases. Specifically, we note the conflict between these qualities and the trade-off that results. This illustrates that values of Window 1 and Window 2 must be chosen carefully to achieve the optimal result for the system's specific goals.

Step 3: Examining the Impact of the Dataset. We used a second dataset to examine how the properties of a dataset bear on decisions for implementing a retrieval system. The second dataset was 100 cases from the PubChem library of chemical structures [13]. Feature variety was much smaller, resulting in 0.0057 features per node and only 8 unique indices. Tables 1 and 2 show poor Phase 2 performance compared to that for the first data set, which we ascribe to to the small number of indices produced by our simple indexing strategy. The tables also show the same trends noted for the myExperiment dataset.

5 Conclusions and Future Work

Our experiments illustrate that evaluation of each phase in a two-phased retrieval process may uncover which of the two components is directly contributing to a lack of system performance. Furthermore, they show that choice of the Window 1 setting can have a dramatic impact on several competing performance benchmarks, so consideration of this design parameter by a system designer is important to achieve the optimal balance between performance metrics.

We have also outlined a set of considerations for implementing 2-phase retrieval, taking these trade-offs into account, including key questions to examine. We believe that there are many other important questions to ask during this process, and we intend to elaborate these in future work. Beyond the feature to node ratio, there are many properties of case-bases which we expect to correlate with high or low performance for different indexing strategies (such as the size of the case base and its rate of growth, the average structural similarity of the cases, and others). We seek to study these characteristics by expanding the number and type of datasets we use in our experiments.

We also seek to study how these case-base properties predict performance of various indexing strategies. Specifically, we aim to implement the complete GraphGrep indexing strategy in order to study the effect of index size on the performance metrics for retrieval. We believe examination of choices affecting the size of indices, the range of index values, or the generality of indices should and will comprise an additional step in our approach to analyzing and implementing 2-phased retrieval systems.

We note that two-phase retrieval is not the only alternative for speeding up structured case retrieval; for example, other viable techniques include clustering [14], cover trees [15], and fish and shrink [16]. While a comparative analysis of each of these techniques is beyond the scope of this paper, a thorough study comparing the drawbacks and benefits of these strategies is warranted. We believe the SAI framework presents an opportunity to realize such an analysis.

Finally, we seek to expand the performance metrics we consider in this process to include the time taken to store or index cases and the amount of long-term storage used, as well as to identify any other key questions not considered within this paper in our future work.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. OCI-0721674 and by a grant from the Data to Insight Center of Indiana University.

References

1. Gentner, D., Forbus, K.: MAC/FAC: A model of similarity-based retrieval. In: Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, Chicago, IL, Cognitive Science Society (August 1991) 504–509

2. Kendall-Morwick, J., Leake, D.: A toolkit for representation and retrieval of structured cases. In: Proceedings of the ICCBR-11 Workshop on Process-Oriented Case-Based Reasoning. (2011)
3. Bergmann, R., Gil, Y.: Retrieval of semantic workflows with knowledge intensive similarity measures. In Ram, A., Wiratunga, N., eds.: ICCBR. Volume 6880 of Lecture Notes in Computer Science., Springer (2011) 17–31
4. Wang, X., Smalter, A., Huan, J., Lushington, G.H.: G-hash: towards fast kernel-based similarity search in large graph databases. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. EDBT '09, New York, NY, USA, ACM (2009) 472–480
5. Tian, Y., Patel, J.M., Nair, V., Martini, S., Kretzler, M.: Periscope/gq: a graph querying toolkit. Proc. VLDB Endow. **1** (August 2008) 1404–1407
6. Yan, X., Yu, P.S., Han, J.: Graph indexing: a frequent structure-based approach. In: Proceedings of the 2004 ACM SIGMOD international conference on Management of data. SIGMOD '04, New York, NY, USA, ACM (2004) 335–346
7. Giugno, R., Shasha, D.: Graphgrep: A fast and universal method for querying graphs. Object recognition supported by user interaction for service robots **2**(c) (2002) 112–115
8. Han, W.S., Lee, J., Pham, M.D., Yu, J.X.: igraph: a framework for comparisons of disk-based graph indexing techniques. Proc. VLDB Endow. **3**(1-2) (September 2010) 449–459
9. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Comput. Surv. **40** (February 2008) 1:1–1:39
10. Bogaerts, S., Leake, D.: Formal and experimental foundations of a new rank quality measure. In: Proceedings of the 9th European conference on Advances in Case-Based Reasoning. ECCBR '08, Berlin, Heidelberg, Springer-Verlag (2008) 74–88
11. Goble, C.A., De Roure, D.C.: myexperiment: social networking for workflow-using e-scientists. In: Proceedings of the 2nd workshop on Workflows in support of large-scale science. WORKS '07, New York, NY, USA, ACM (2007) 1–2
12. Leake, D., Kendall-Morwick, J.: Towards case-based support for e-science workflow generation by mining provenance information. In: Proceedings of the Ninth European Conference on Case-Based Reasoning, Springer (2008) 269–283
13. Bolton, E.E., Wang, Y., Thiessen, P.A., Bryant, S.H.: PubChem: Integrated Platform of Small Molecules and Biological Activities. Annual Reports in Computational Chemistry **4** (2008)
14. Kapetanakis, S., Petridis, M., Ma, J., Knight, B., Bacon, L.: Enhanching similarity measures and context provision for the intelligent monitoring of business processes in CBR-WIMS. In: Proceedings of the ICCBR-11 Workshop on Process-Oriented Case-Based Reasoning. (2011)
15. Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: Proceedings of the 23rd international conference on Machine learning, New York, ACM Press (2006) 97–104
16. Schaaf, J.: Fish and shrink. a next step towards efficient case retrieval in large scaled case bases. In Smith, I., Faltings, B., eds.: Advances in case-based reasoning, Berlin, Springer Verlag (1996) 362–376

Reasoning with Organizational Case Bases in the Absence Negative Exemplars

Sidath Gunawardena, Rosina O. Weber

iSchool at Drexel, Philadelphia, USA
{sg349, rw37}@drexel.edu

Abstract. Organizational case bases may lack negative exemplars and have multiple solutions to one problem, making it difficult to learn weights for reasoning. Case bases in typical Process-Oriented Case-Based Reasoning (POCBR) contexts are organizational, thus inheriting those problems. This paper describes an approach to identify a subset of cases from an organizational case base that meets the criterion that similar problems have similar solutions. This subset is then used to characterize classes, establishing positive and negative exemplars that are then used to learn weights for reasoning with the entire case base. We apply this approach to three organizational case bases, showing significant improvements in accuracy with weights learned with this approach in case bases without negative exemplars.

Keywords: organizational case bases, process-oriented case-based reasoning, processes, workflows, negative exemplars, learning, weights

1 Introduction

The focus of CBR towards tasks, processes, and workflows widens the applicability and usefulness of the CBR methodology. This focus led to the rise of Process-Oriented Case-Based Reasoning (POCBR) systems [10], [16]. POCBR systems are typically organizational in that cases are included because they are relevant to the organization the system is designed to support. The decision of which cases to include does not take into account the purpose of reasoning. This results in case bases that are difficult to use for reasoning.

CBR systems deployed in organizational contexts aim to solve a variety of problems. When POCBR systems are used for managing and modeling workflows, the goal is to find one workflow sufficiently similar to a workflow in use to be adapted. In these problems, it is hard to determine what makes one workflow more similar to another to assign weights to represent relative relevance. Identifying negative exemplars is also hard because given enough adaptation knowledge, any workflow can be considered similar.

When POCBR systems are used for recommendation (e.g., e-commerce, expert locator systems), case bases include characteristics of entities. This is another class of problems where there may be no negative exemplars. Conceptually it is hard to say whether a combination of characteristics is not suitable or it simply n ever happened. This issue may also be present in other uses of POCBR systems such as prediction and simulation, where the goal is to identify a similar workflow to reuse its sequence or next step.

The absence of negative exemplars can be very problematic as it prevents the use of the feedback algorithms typically recommended for learning weights [1]. For this reason, we present a systematic approach for organizational case bases. The approach takes a case base and reduces it to a subset that meets the criterion that similar problems have similar solutions. In this process, it eliminates boundary cases and some diverse cases. The approach creates clusters that are used as classes, allowing the distinction between positive and negative exemplars as cases that, respectively, belong or not to a class. This resulting subset of cases organized in classes enables the learning of weights to represent relative relevance of individual features. Subsequently, boundary and diverse cases can be incorporated again into the case base.

In the next section we present a general description of our method. In the following section, we describe a study where we apply the method to three case bases. The study shows that our method leads to learning weights that result in average accuracies that are equivalent to alternative methods when negative exemplars are available. In the absence of negative exemplars, our proposed method leads to learning weights that result in average accuracies that are significantly higher than when no weights are used. We then summarize some related work, and conclude with a few remarks on the implications of the results and future work.

2 Method

2.1 Introduction

We want to overcome the problem of lack of negative exemplars and the presence of diverse and boundary cases to learn weights that represent the relative relevance of features. We seek to identify core cases: sets of cases where either the problem parts or the solution parts meet a certain threshold of similarity. In this approach, a boundary case does not have sufficient cases near it to be a core case. A diverse case is a core case whose solution (or problem) is not sufficiently similar to the solutions (or problems) of other core cases belonging to the same class. While such cases are valuable as they promote diversity, they violate the CBR tenet that similar problems have similar solutions and inhibit learning a set of consistent weights. **Fig. 1** represents cases in problem-solution pairs (P_i, S_j). The three cases {P_2, S_2}, {P_5, S_5} and {P_6, S_6} are examples of core cases. {P_4, S_4} is a boundary case as there are not enough cases similar to its problem or solution. $\{P_1, S_1\}$ and $\{P_3, S_3\}$ are diverse cases as they violate the principle that similar problems have similar solutions. These cases may overlap or be completely disjoint.

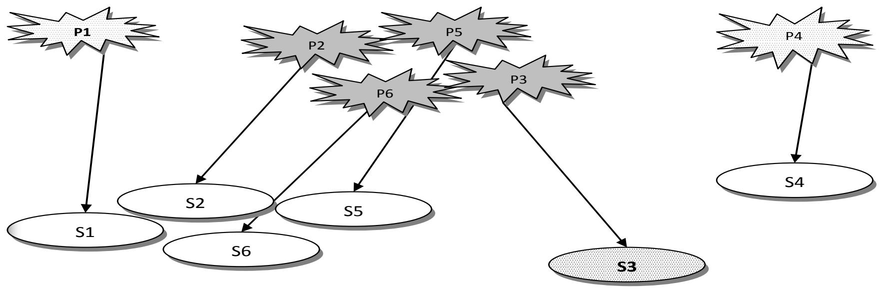


Fig. 1. Examples of core, diverse and boundary cases

Our method is motivated by the CBR assumption that similar problems have similar solutions [7]. By clustering cases based on problems and solutions we identify cases that meet this criterion by those that violate it. The resulting clusters are treated as classes where positive exemplars belong and negative exemplars do not. By removing cases that do not comply with these classes, we eliminate boundary cases and some diverse cases. In absence of domain knowledge, we assume that two entities (e.g., problems, solutions) are similar when they share common features of a given representation [16].

2.2 General Methodology

We illustrate this methodology in **Fig. 2**. Step 1 is comprised of two independent phases where cases are clustered both on the problems and also on the solutions. In the Step 2 we utilize the clusters to learn negative exemplars and in Step 3 we evaluate the quality of the weights generated by both clustering phases by applying them to assess the average accuracy of the entire case base. The weights learned after the clustering phase that result in higher accuracy on the overall case base are recommended for adoption.

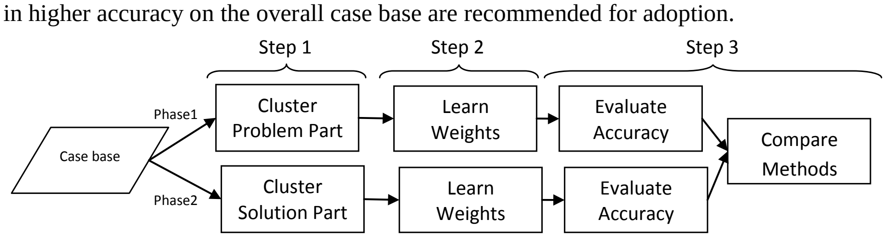


Fig. 2. General Methodology

Step 1 identifies groups of cases based on the problem space and on the solution space. We cluster cases based on both problem and solution because we do not know beforehand which will produce the better clusters overall (Fig. 3a and 3b). We use a density clustering algorithm as it removes outliers, so it removes boundary cases and some diverse cases. The resulting clusters will then be used as classes in Step 2.

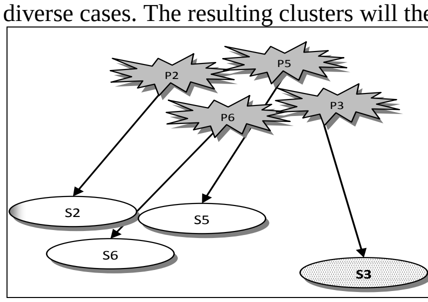


Fig. 3a. Clustered cases based on problems

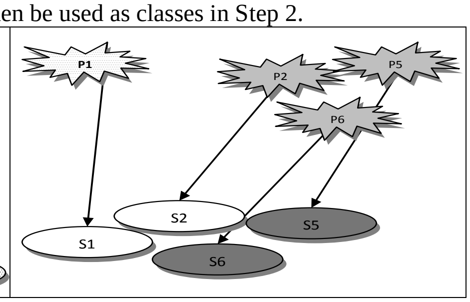


Fig. 3b. Clustered cases based on solutions

Step 2 is where the resulting classes enable learning of weights. Any feedback algorithm with performance bias can be used for learning weights for classifiers. The clusters are the reference outcome. The learned weights are then used in the CBR system.

Step 3 is the evaluation step. The evaluation is done on the entire case base, which includes cases removed during clustering. We use cross-validation to determine the resulting average accuracy for each set of weights learned from each phase in Step 1. In other words, there are two evaluations, one that uses weights learned based on clustering the problems (Phase 1), one that uses weights learned based on clustering the solutions (Phase 2). The evaluation computes average accuracy using a suitable similarity measure. The more accurate weighting scheme is to be adopted.

2.3 Further Aspects

The clustering is run separately on problem and solution parts. We note that if a case is removed in both it is a boundary case and if only in one it is a diverse case. The resulting clusters depend on the specifics of the data, that is, whether problems or solutions are more similar to each other. It should be noted that if there are a sufficient number of diverse cases so that they form their own core cluster, then they will not be removed.

An important aspect of this method is that it uses the uncertain concept that similar problems have similar solutions. The uncertainty inherited from this concept is embedded in its results. It is not the goal of the method to eliminate this uncertainty but rather to make the dataset usable. In the next section we describe an example using the method.

3 Applying the Method

We wish to investigate the following hypotheses:
H1. In the absence of negative exemplars, clusters of cases can be used as classes to identify negative exemplars to learn weights that lead to significant improvements in average accuracy in organizational case bases over feature counting.
H2. In the presence of negative exemplars, clusters of cases used as classes to identify negative exemplars to learn weights lead to average accuracy in organizational case bases equivalent to when the actual negatives are used.

3.1 Data

We apply our proposed method to three organizational case bases (**Table 1**). Casebase 1 describes a process at a high level with the solution being its specific implementation. This case base does not have negative exemplars. Casebase 2 describes a collaboration of entities and also lacks negative exemplars. It describes a recommendation problem. Casebase 3 describes a process with the solution representing its success or failure. This is the only case base with negative exemplars is a binary classification problem.

Table 1. Summary of case base characteristics

	No. of Cases	No. of Features	Has Negative Exemplars	Problem	Solution
Casebase 1	254	3	No	Abstract Process	Specific Process
Casebase 2	198	3	No	Collaboration Seeker	Recommended Collaboration
Casebase 3	88	23	Yes	Description of Process	Success/Failure

3.2 Experiments

When applying our proposed method, we use a density clustering algorithm, DBSCAN [4]. The clusters are used to learn weights with a genetic algorithm, and the performance of the learned weights is evaluated via a Leave-one-out Cross-Validation (LOOCV) on each case base.

Step1. For Casebase 1 & 2, we cluster based on both problems and solutions. For Casebase 3, which describes binary classification cases, solutions are either 0 or 1, so we only cluster on the problem part of the cases. Table 2 shows the resulting number of cases and number of clusters.

Table 2. Step 1: Clustering cases based on problems and solutions

	Casebase 1 254 cases		**Casebase 2 198 cases**		**Casebase 3 88 cases**	
	# of cases	# of clusters	# of cases	# of clusters	# of cases	# of clusters
Phase 1: Clustering on problems	129	40	189	8	31	11
Phase 2: Clustering on solutions	77	29	124	10	NA	

Step 2. The clusters now provide us with positive and negative exemplars of the classes they represent. We use the ability to correctly make this classification as the fitness function to learn weights via a genetic algorithm. Two hundred randomly generated sets of weights represent the chromosome, where each individual weight can be thought of as a gene. The algorithm is run for 1000 iterations. In all iterations the better performing chromosomes (sets of weights) have a greater chance of contributing their genes (weights) towards the next generation. To reduce the likelihood of being stuck in a local maximum we introduce a 5% of mutation where instead a random gene is inserted.

Step 3. In this step, we evaluate the quality of the weights. For this we use the learned weights and assess accuracy for the entire case base based on the subsets of cases determined in Step 1. We now present those results in **Table 5** to 5. To evaluate the statistical significance of the experiments, we conducted separate one way within-subjects ANOVAs followed by post-hoc analysis with Tukey's Honest Significant Difference test, with alpha = 0.5. Table 3 shows the average accuracy for Casebase 1.

Table 3. Casebase 1, Average accuracy using LOOCV, * significant difference at $\alpha = 0.05$

Similarity	**Feature counting**	**Weights From Problem Clusters**	**Weights From Solution Clusters**
kNN =1	23%	24%	28%*

Casebase 1 has 254 cases and no negative exemplars. We cluster on both the problem parts and solution parts. We learn two sets of weights based on these clusters. Accuracy is measured based on a gold standard, where for 81 cases of the 254, the next best solution is determined manually. This gold standard is used as basis for LOOCV run on the 81 cases, where only the top scoring result is selected. Where there are ties, one of the tied results is chosen randomly. This process is repeated 10 times for each set of weights, and the average is presented here. The results show a statistically significant improvement (α =0.05) when using the weights from the solution clusters.

Table 4. Casebase 2, Average accuracy using LOOCV, * significant difference at $\alpha = 0.05$

Similarity	Feature counting	Weights from negatives from problem clusters	Weights From negatives from Solution Clusters
Sim1	63%	63%	67%*
Sim2	66%	65%	67%*
Sim3	63%	64%	67%*

Casebase 2 has no negative exemplars. The similarity functions are based on the level of abstraction. It is clustered on both the problem and solution parts. The learned weights are compared to using no weights. Accuracy is measured as the edit distance between the recommended solution and the solution of the removed case. **Table 4** shows statistically significant improvement (α =0.05) when using the weights from the solution clusters.

Table 5. Casebase 3, Average accuracy using LOOCV, * significant difference at $\alpha = 0.05$

Similarity	Feature counting	Weights learned from actual negatives	Weights learned from negatives from clusters
kNN =1	80%	85%	84%
kNN =3	81%	92%	92%

The results in Table 5 show no significant difference between weights learned from actual negatives and weights learned from negatives from clusters. The resulting average accuracy when actual negatives are used are equivalent to average accuracy resulting when negatives are defined based on interpreting clusters as classes.

3.3 Results & Discussion

Table 6. H1. Absence of negative exemplars

Case base	Negative Instances	Improvement over Feature Counting?	Difference is Statistically Significant
Casebase 1	No	Yes	Yes
Casebase 2	No	Yes	Yes

Table 6 supports H1, "In the absence of negative exemplars, clusters of cases can be used as classes to learn weights that lead to significant improvements in average accuracy in organizational case bases over feature counting."

Table 7. H2. Presence of negative exemplars

Case base	Negative Instances	Comparable to using actual Negative Exemplars?	Difference is Statistically Significant
Casebase 3	Yes	Yes	No

Table 7 supports H2, "In the presence of negative exemplars, clusters of cases used as classes to identify negatives to learn weights lead to average accuracy in organizational case bases equivalent to when the actual negatives are used."

Our results show that learning weights from clustering on the solutions and removing boundary and diverse cases can lead to a significant increase in the accuracy when compared to using no weights. When this method is applied where negative exemplars exist, it produces comparable results to standard methods. The weights produced from clustering on the problems do not increase accuracy from not using weights. Thus, it is the set of core cases resulting from clustering on solutions that better meets the requirement that similar problems have similar solutions.

4 Related and Background Work

[1] has discussed the use of weight learning methods with (i.e., *performance bias*) and without feedback for classifiers (i.e., *preset bias*). Our datasets are not necessarily being used for classification tasks. The problem we face is the lack of negative exemplars. Previous work on reasoning without negative exemplars focuses mainly on single-class learning and classification problems where there is a large collection of unlabelled data and a small collection of positively labeled data, e.g., web pages and DNA sequences. Typical methods iterate a two-step process where the first step learns heuristics to identify negative instances that can then be used in the second step by a classifier such as SVM [15], [17] or Naïve-Bayes [2],[8]. [6] generate artificial negative exemplars via induction, where all feasible cases that do not exist are used as negatives. Because organizational case bases may need diverse solutions, we do not know if a process that is not present is a negative exemplar or one that does not exist or one that has not yet been tested.

Process cases are complex and learning models of the entire problem space can lead to overgeneralization. Clustering has been using in process mining to subdivide the problem space so multiple models can be learned [5], [14]. We select density clustering, as it is recommended for remove outliers [13]. Density clustering does not determine a centroid or number of clusters, but creates cluster of a specified density. We use DBSCAN [4], but do not make a thorough review or assessment of density clustering methods we simply demonstrate our approach.

Works that explored the CBR requirement of similar problem being associated with similar solutions include [9] who show that boundary cases can also affect performance of classification systems and to improve classification accuracy suggesting valid cases may need to be removed. This property of CBR is also investigated by [3] who shows that some cases can also be a liability to the case base by promoting misclassification. Other approaches employ singular value decomposition [12] to reduce dimensionality, or use a threshold based on verified cases [11] to determine 'good' cases.

5 Concluding Remarks

POCBR case bases are typically gathered because they are pertinent to an organization and not due to their potential contribution to reasoning. This organizational orientation produces case bases that may lack negative examples and include boundary and diverse cases. Our proposed method takes a set of cases and reduces it to a set that, within a reasonable distance, have similar problems with similar solutions.

We study three case bases, two without and one with negative exemplars. Our method leads to learning weights that result in average accuracies that are equivalent to an alternative method when negative exemplars are available. In the absence of negative exemplars, we note that there are no alternative systematic methods and thus we compare the resulting accuracy of the case bases using weights learned from our method to using no weights. The results for both datasets produce significantly higher average accuracy.

This work is a first step towards dealing with real world datasets from organizations that have many experiences to contribute, mostly being processes or workflows. The approach discussed here can also benefit datasets that do not necessarily represent processes; but like business processes, are gathered by one organization and may be difficult to learn due to the lack of negative exemplars. It is suitable for learning weights when systems are being designed, and also for systematic maintenance.

Among the next steps we want to investigate the impact of removing more cases after clustering. For example, when clustering based on the problems, we will remove cases whose solutions to not fit well the clustering organization provided by the problems. We will also investigate different clustering algorithms and their potential for this task.

6 Acknowledgments

The authors are supported in part by the U.S. EPA Science to Achieve Results (STAR) Program and the U.S. Department of Homeland Security Programs, Grant # R83236201. We would like to thank our reviewers for their detailed and insightful feedback.

7 References

1. Aha, D.W. (1998). Feature weighting for lazy learning algorithms. In H. Liu & H. Motoda (Eds.) Feature Extraction, Construction and Selection: A Data Mining Perspective. Norwell, MA: Kluwer.
2. Denis, F., Gilleron, R., Tommasi, M., (2002). Text classification for positive and unlabeled examples. In: IPMU 2002, pp. 1927–1934.
3. Delany, S.J. (2009). The Good, the Bad and the Incorrectly Classified: Profiling Cases for Case-Base Editing. ICCBR 2009: 135-149
4. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Evangelos Simoudis, Jiawei Han, Usama M. Fayyad (eds.). Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press:Menlo Alto. pp. 226–231.
5. Ferreira, R., Zacarias, M., Malheiros, M., and Ferreira, P. (2007) Approaching Process Mining with Sequence Clustering: Experiments and Findings. BPM: 360-374
6. Goedertier, S., Martens, D. , Vanthienen, J., Baesens, B. (2009). Robust Process Discovery with Artificial Negative Events. Journal of Machine Learning Research 10: 1305-1340
7. Leake, D. B., ed. (1996). Case-Based Reasoning: Experiences, Lessons, and Future Directions. Menlo Park, CA: AAAI Press/MIT Press, Menlo Park, CA.
8. Liu, B., Lee, W. S., Yu, P., and Li, X. (2002). Partially supervised classification of text documents. ICML-02.
9. Massie, S., Craw, S., Wiratunga, N. (2007). When Similar Problems Don't have Similar Solutions. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp.92-106 . Springer, Heidelberg
10. Minor, M., Bergmann, R., Görg, S., and Walter, K. (2010). Towards Case-Based Adaptation of Workflows. In I. Bichindaritz and S. Montani (Eds.): ICCBR 2010, LNAI 6176, pp. 421–435, 2010. Springer-Verlag Berlin Heidelberg
11. O'Mahony, M., Hurley, N. and Silvestre, G. (2006). Detecting noise in recommender system databases. In IUI 2006. ACM, New York, NY, USA, pp. 109-115
12. Symeonidis P. (2007). Content-based Dimensionality Reduction for Recommender Systems, Proceedings of the 31st Conference of the German Classification Society (GfKl'2007), Freiburg.
13. Richer, M. M. and Weber, R. O. (2012). Case-based reasoning: a textbook. Berlin: Springer. In press.
14. Veiga, G., Ferreira, D. (2009). Understanding Spaghetti Models with Sequence Clustering for ProM. Business Process Management Workshops: 92-103
15. Wang, C., Ding, C., Meraz, R. F. and Holbrook, S. R.. (2006). PSoL: a positive sample only learning algorithm for finding non-coding RNA genes. Bioinformatics, 22(21):2590–2596.
16. Weber, B., Wild, W., Breu, R. (2004). Cbrflow: Enabling adaptive workflow management through conversational case-based reasoning. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 434–448. Springer, Heidelberg
17. Yu, H., J. Han, J. and Chang,, K. C.-C. (2004). PEBL: Web page classification without negative examples, IEEE Trans. Knowl. Data Eng., vol. 16, no. 1, pp. 70–81, Jan.

An Evaluation of the CBR-WIMS Architecture for the Intelligent Monitoring of Business Workflows using Case-Based Reasoning

Stelios Kapetanakis[1], Miltos Petridis[2]

[1]School of Computing and Mathematical Sciences, University of Greenwich, Maritime Greenwich Campus, Old Royal Naval College, Park Row, Greenwich, London SE10 9LS, UK
[2] School of Computing, Engineering and Mathematics, University of Brighton, Moulsecoomb Campus, Lewes road, Brighton BN2 4GJ, UK
s.kapetanakis@gre.ac.uk, m.petridis@brighton.ac.uk

Abstract. CBR-WIMS is a framework that implements the Case-based Reasoning (CBR) process to enable the intelligent monitoring of business workflows. The framework uses the SOA paradigm to store and index a set of business workflow execution event traces. It also allows transparent service interfaces to enterprise system components that orchestrate and monitor business workflows. The CBR component employs similarity measures to retrieve workflow execution cases similar to a given target case. This enables the reuse of associated knowledge about the workflow execution into the new target case. This paper presents the CBR-WIMS approach and architecture and illustrates its features through its application to two real-life enterprise systems. The paper examines the portability and robustness of the CBR-WIMS architecture and provides an evaluation of its suitability through an analysis of the experience gained from the two enterprise systems application case studies.

Keywords: Case-based Reasoning, Business Workflows, Systems Architecture

1 Introduction

Modern businesses and organisations use business processes for a variety of purposes. This is due to the constantly evolving operational environments while there is also a need for ensuring formalism and auditability in the business process procedures. They define strictly the internal hierarchical structures as well as setup in a transparent way the relationships among internal and external stakeholders [11]. Business processes offer an agile and flexible way to apply a set business strategies towards a desired goal as well as offer an organisational overview in a leveraged way. Expert managers can examine these processes following a bottom-up or top-down approach, focusing on areas of operational significance.

Due to their wide usability and acceptance business processes are being increasingly subject to artificial management and monitoring. Several business process management systems (BPMS) have been implemented following standards that assure the suitable formalism in the representation of business processes. The Business Process Management Notation (BPMN) is a standard of wide acceptance developed to provide a graphical representation of workflow based business processes [1]. The OASIS WS-BPEL is a key execution language, annotating the behaviour of business processes [2]. The XML Process Definition Language (XPDL) provides a standardised format to interchange definitions between workflow vendors [3]. Several other standards like UML Activity Diagrams, BPEL4WS, EPC, and YAWL exist, allowing business process modelling based on different operational perspectives.

Business process standards have been embraced by Service Oriented Architectures (SOAs) allowing the provision of abstracted software web services, loosely connected with requesters. This web service provision through SOA has increased the flexibility and adaptability of BPM systems [4].

With the maturity of adequate representation standards, business processes have ensured their wide acceptance. However, modern business environments are event-driven, and generate a large number of events. As a result a challenge emerges for the effective monitoring and interpretation of a business process instance, based on its generated events. Typically when a business process runs, it generates events that could appear in a general form (name, time-stamp); including though a significant amount of relevant environmental information (detailed name, type, nature, origin, involved workflow stakeholders, further related time-stamps, etc.). The volume of the generated (event-related) data can vary from system to system depending on the system's size, involved operations, the current configuration and the event generation policy followed (trigger an event on every update, on warning, on alert, on severe alert, the list can be expanded further depending on the investigated system).

Event traces generated by a system may be clear and straight-forward when small in number. However, the difficulty in understanding a specific operational statue rises significantly when the numbers of events grow as well as with the underlying system complexity. In many cases the presence of uncertainty, when some of the event information is incomplete or vague can make the job of human auditors harder. This can be a common phenomenon in business workflows that deal with human resources since several events may be not be captured [5] (such as phone calls, unofficial meetings, corridor chats, etc.). In such cases, although the business process can be represented with high precision, its monitoring can be rather difficult, since the system may not be able to identify the relevance of some captured information.

In addressing the above issues, a CBR approach could be elaborated [5] for the intelligent monitoring of workflows based on available past experience. The system could be incorporated within the defined business process and liaise with its associated information drawn from past workflow executions. Based on this information the system could provide relevant diagnosis and explanation [6] on what could be the actual situation faced.

This paper presents the architecture of CBR-WIMS, a CBR framework developed for the intelligent monitoring of agile business workflows. Section 2 provides a background regarding CBR, work conducted about workflows monitoring as well as the first system used as a case study; section 3 presents the adopted approach for the intelligent workflow monitoring; section 4 presents the architecture of the monitoring framework; section 5 presents an evaluation of the CBR-WIMS based on a new business process monitoring system used as a case study, showing the framework's efficiency and wide reusability of its software components. Finally the conclusion section summarises the work conducted and presents some possible future research paths.

2 CBR and workflows monitoring

In order to achieve an effective monitoring of business processes, a variety of approaches can be adopted. When a business process is being monitored, the current series of executed events shows the actual state of the business workflow. This state can be analysed and be compared with existing past knowledge in an attempt for the efficient monitoring of the workflow. The perspective behind such approach is based on the fact that usually similar problems have similar solutions. Case-based reasoning can be regarded as the closest series of techniques to the above perspective following the four R's standard CBR process model [7]. CBR has been proposed as a possible approach to the reuse and adaptation of workflows based on graph representation and the application of structural similarity measures [8]. For the measurement of similarity among business processes Dijkman et al [9] have investigated algorithms focused on tasks and the control flow relationships among them. Work has also been conducted on the comparison of process models based on behavioural observations within the context of Petri-nets [10]. Similarity measures for structured representations of cases have also been proposed within the context of CBR [12] and have been applied to real life applications requiring reuse of past available knowledge as applied to rich structure-based cases [13], [14].

CBR seems a natural way to allow the effective monitoring of business processes. Existing CBR frameworks provide generic services to assist with the application of CBR on various application areas. An example of this is jColibri [15], an open-source CBR framework aimed at integrated applications that contain models of general domain knowledge. myCBR [16] is another framework providing rapid prototyping of CBR product recommender systems. The above provide effective modelling of applications but do not offer mechanisms for monitoring business processes or environment adaptation according to the investigated process's needs.

A CBR approach towards the intelligent monitoring of business workflows has been proposed and shown to be efficient in monitoring real workflows [5], [6], [17]. This approach can provide workflow monitoring based on similarity measures and a knowledge repository of past cases. Cases consist of several attributes such as event traces, actions taken as well as any available temporal relationships. The characteristics of cases can be represented in terms of a graph used for estimating similarity.

3 CBR-WIMS

CBR-WIMS is a generic architecture and framework developed for the intelligent monitoring provision of business processes [17]. CBR-WIMS provides smooth integration between the CBR knowledge repository and a BPM workflow system. The architecture can deal with the orchestration and choreography of a business process and also provide workflow monitoring. The approach promotes the intelligent management of processes among businesses and organisations. Previous work has shown encouraging results in terms of monitoring been compared to expert managers of specific workflows [5]. The framework can provide a reasoning mechanism for the assistance of workflow managers while dealing with complex situations or situations containing some uncertainty.

Earlier work [17] has shown how the framework's architecture allows smooth integration with a given business process adapting itself to the requirements of the given domain. That work conducted using the CBR-WIMS architecture focused primarily on:

- The adaptation of the architecture on top of a live system forming an integrated case study evaluation.
- Enhancement of the architecture to provide explanation and visualisation facilities required to make the resulting integrated system fully effective.
- Adding elements of inter-connections among business process and reusability in terms of architecture enhancements.

The proposed architecture has now been tested and evaluated with its application on two real-life enterprise systems. The first system is a quality assurance enterprise system used in past work [5], [6], [17]. The second system relates to a Box Tracking System (BTS) which provides temporal tracing of physical items (boxes) within the context of prescribed workflows. The second case study involves advanced temporal complexity deriving from the operational complex context of the workflow. This case study involves certain elements of uncertainty due to the large number of possible errors and alterations that can take place outside of the scope of the workflow monitoring system.

4 Generic business process monitoring

CBR-WIMS is a generic framework and architecture for the intelligent monitoring of Business processes. The motivation behind its design and implementation was to offer a resilient collection of tools that can easily be integrated with an existing business process operational environment. The system should be able to model the rules and limitations of the business process, identify data correlations, actors' roles and activities [17], as well as monitor the execution of the workflow.

4.1 Similarity measures

In order to be able to compare different execution traces, CBR-WIMS defines the processes in terms of graphs. Similarity measures can be applied afterwards using graph similarity algorithms based on the Maximum Common Sub-graph (MCS) [14]. Sequences of events can be represented as graphs following their temporal execution traces. The relevant information extracted from the available event logs can be represented with the use of the general time theory based on points and intervals [18].

4.2 CBR-WIMS Architecture

The CBR-WIMS system is based on a flexible component-oriented architecture, allowing the collaboration, addition and adaptation of existing components based on the anticipated business process environment. The architecture can be seen at Figure 1 below. The full architecture of the system is described in detail in previous published work[17].

CBR-WIMS contains a collection of services that can deal with an imported business process and assist in its effective monitoring. The architecture consists of a set of core software apparatuses that comprise the monitoring backbone and a set of flexible components that can be adapted to a given business process.

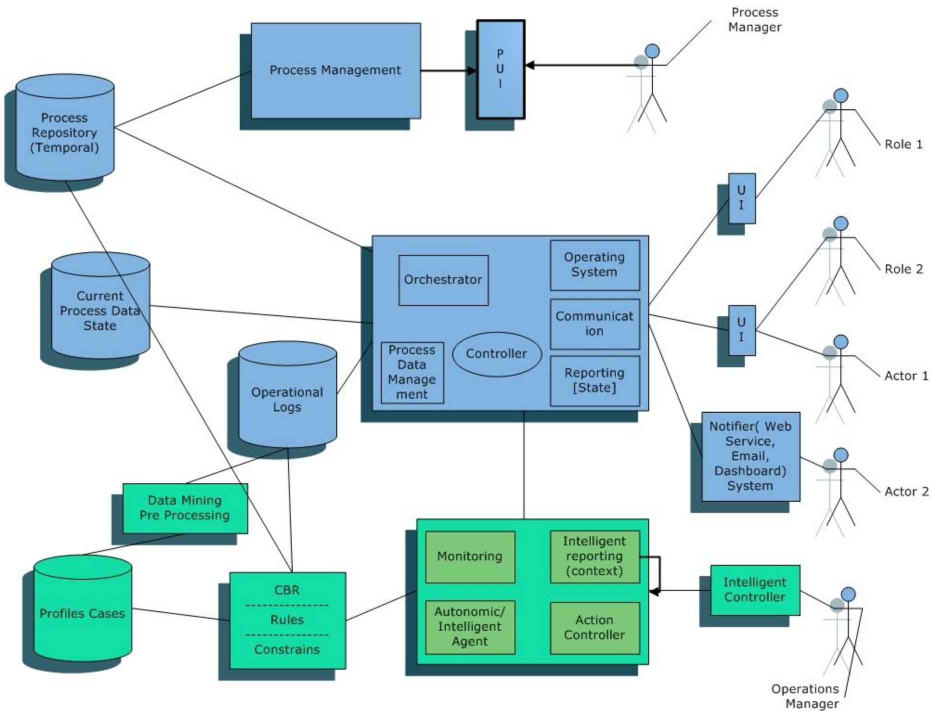


Fig. 1: CBR-WIMS Architecture

The framework is generic. As a result every attempt to use it should start with a process of setting up rules and constraints for a particular formal definition of the business process. The widely accepted standards of BPMN and XPDL are used to import the process business rules and constraints as well as "*sketch*" their operational environment. Based on that, the framework afterwards will be called upon to attempt the process monitoring.

Further to the setup of rules and constraints, CBR-WIMS can adjust its main components in order to be able to deal with the monitored business process. The kernel of the system provides a regulatory context that outlines the overall process. However, the rest of the components are being changed depending on the environmental set. Figure 2 shows the way the framework changes along systems.

For the needs of the BTS system, CBR-WIMS has been subject to a number of modifications. A new Adaptor component has been created in order to create the relevant monitoring parts that can establish monitoring hooks to the BTS workflow traces. The available logs from the investigated system where parsed using a modified version of the Parser component and were inserted to the WIMS as cases, constituting and populating its case-base. The Adaptor component contains all the communication channels between the investigated system and the monitoring framework ensuring smooth integration of the latter on top of the second's executed traces. The CBR component was not subject to changes, although its case base was populated with different data from what it contained for the previous experiments [5], [6]. Its Similarity measures calculation component was also not affected since its generic methods for the calculation needs were populated explicitly by the Adaptor component.

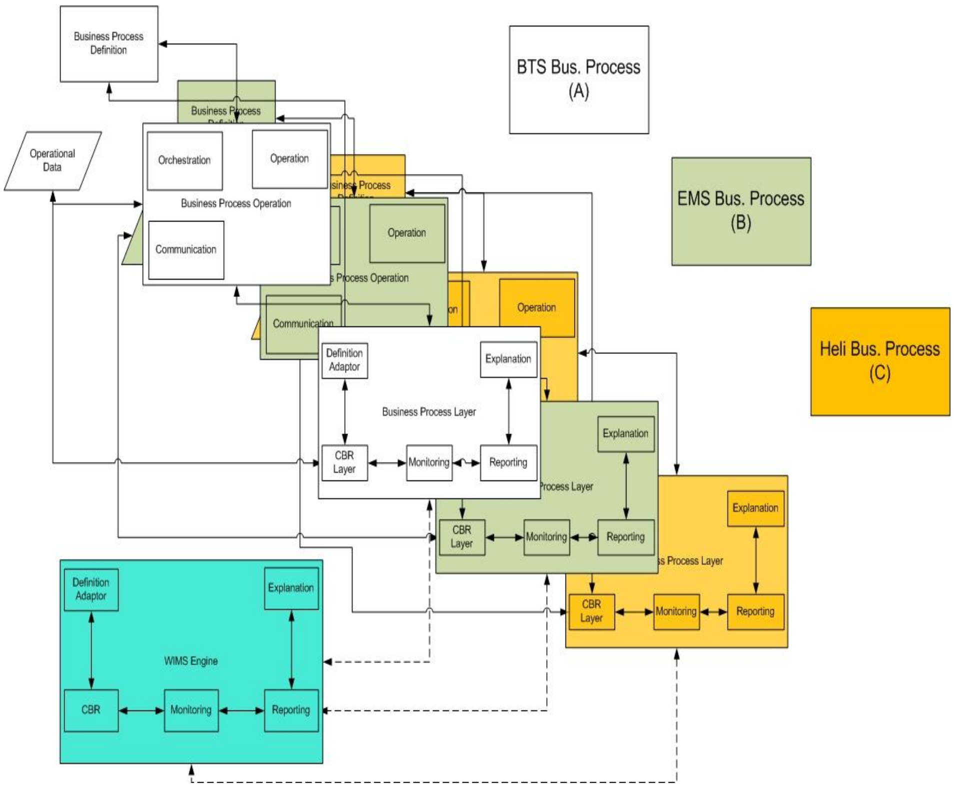


Fig. 2: Architectural layers depending on the investigated business processes

In order to be able to persist its generic characteristics CBR-WIMS does not allow adaptations to core apparatuses when dealing with new business processes. Instead, it requires a semantic ontology to be defined for any investigated system and be populated to the explicit adaptor created for the monitoring purposes. This approach avoids the need for unnecessary adaptations in the system and preserves its integrity and reusability characteristics. A more specific picture that shows the system adaptations for the BTS case study can be seen at Figure 3.

In contrast to the system’s generic core functions and services, the Results, Reporting and Explanation components have been adapted in order to meet the requirements stemming from the specific monitored business process management system. These components using the framework’s visualisation and report compilation libraries, but should also be able to assist the business process experts by responding in their own “*language*” as well as following the explicit characteristics of the system.

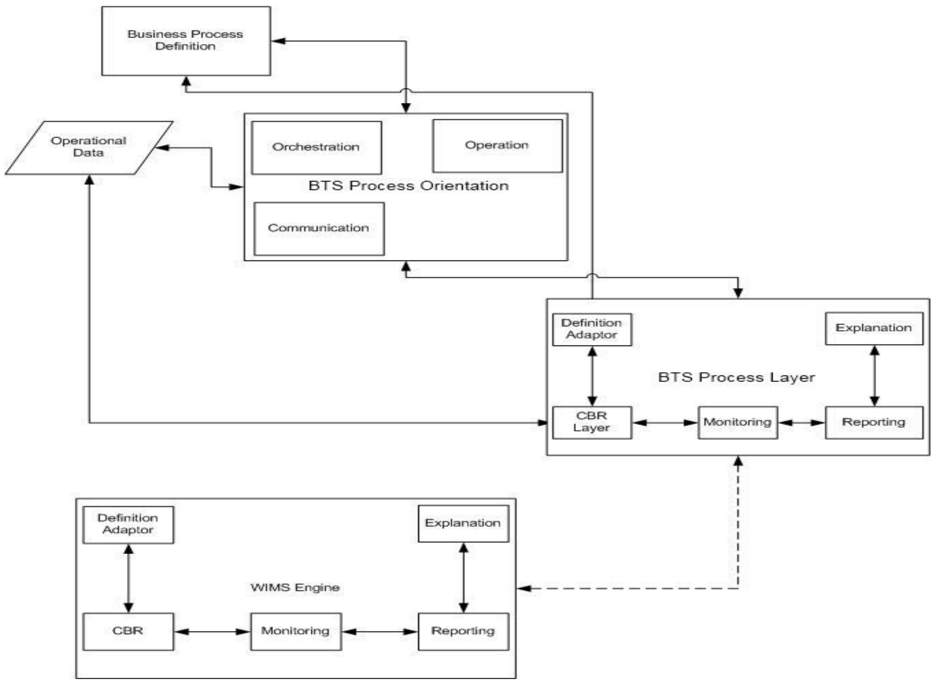


Fig. 3: CBR-WIMS adaptation layer for the needs of BTS system

5 Evaluation

The CBR-WIMS architecture has been first implemented on an educational enterprise system (EMS) [5] that tracked quality assurance transactions and authorisations. That initial evaluation showed the suitability of the CBR-WIMS architecture for the effective monitoring of business workflows. However, that original evaluation did not establish the the suitability of the CBR-WIMS architecture as a generic architecture and framework across different types of workflow management systems. Since the original evaluation of the CBR-WIMS framework, a number of the features that ensure its portability and abstraction have been further implemented and enhanced. The evaluation presented in this system concentrates on these specific aspects of the CBR-WIMS architecture and framework. The BTS system was chosen as significantly different from the EMS system. This is both in terms of the nature of the monitored workflows, mainly in terms of the types of events and actors involved and in terms of the different types of enterprise technologies and architectures of the two workflow management systems. For the needs of the new system, several modifications have taken place in terms of the architecture dealing with the system's unique characteristics.

5.1 The BTS case study

In order to evaluate the generic characteristic of the proposed architecture, the monitoring on Box Tracking System has been investigated. BTS is an on-line archive management system that deals with the transfer and archiving of boxes between its owned warehouse and its customer premises. Due to the operational nature the system can initiate and host large numbers of transactions that could be spread over time, depending on the customer needs. The number of boxes that could be included in a particular transaction can vary in size (from a rather small number to a rather large one), adding to the given system's complexity. The increased complexity also propagates a higher possibility for errors on either side (company or clients). Box deliveries are being conducted via the BTS owned vehicles. All transactions take place in real-, or near-real- time, and are being monitored via the online system.

System authorised users (managers) are usually monitoring the actual flow of the workflow execution. However, due to the nature of the transactions usually problems are being identified from the customers' side, stating a drawback for the effective monitoring of the business process.

The intelligent monitoring of the executed workflows is built to provide early warning, explanation and context to problems that may emerge at a relatively early stage of developing problems.

5.2 Integration with BTS

In order to evaluate the generic characteristics of the proposed architecture a set of experiments were conducted aiming to verify the adaptability of the architecture to a newly incorporated system. For the evaluation purposes in previous conducted experiments [17], an educational enterprise system was used (EMS) and CBR-WIMS was called to establish a monitoring overview. The system was able to establish similarity measures among workflow execution traces, and present the results to system experts in a visualised way along with elements of textual explanation.

For the needs of these experiments the code structure was reused, although the bare enterprise systems were significantly different from both the implementation and operational perspective. The EMS system [5], [6] is an educational Quality Assurance management system, containing consecutive versions of ASP layers backed up by Access databases. BTS on the other hand is a box management system, comprising of several C-sharp (C#) modules on top of a SQL-server database. CBR-WIMS was able to tackle the integration issues in a rather speedy approach, achieving a smooth, data-integral merge with both systems. Cases of the new system were fed to the case base following the existing proprietary system format. Similarity measures were performed afterwards in a rather trifling time-span. Indicatively the full integration procedure of the system along with the database, that contained more than seventy three thousands of box instances (73,000), lasted approximately eighty (80) work-hours, a rather limited amount of time for operations of that level.

A simple experiment conducted on top of the integrated system included 180 journeys set as the case base. Each journey included a range of parameters indicating certain box states. The MCS [14] was applied to the given cases estimating the similarity measures among them. A sample of 30 journeys was afterwards randomly selected and was redirected for monitoring from both the CBR-WIMS and the system's experts. The experts were able to identify several cases of "*interesting*" behaviour based on the pattern followed within the case. CBR-WIMS was able to identify such patterns and then extrapolate whether a case was of "*interest*" or not based on its closest neighbours' ranking. The system was able afterwards to visualise the similarities between investigated cases and neighbours as well as provide explanation for the provided results. The results provided were able to be shown to system experts without major amendments in WIMS reporting and explanation modules since the flexibility of the architecture allowed their effortless extraction.

CASES	CBR-WIMS	Expert's classification
Correctly classified "*no clear status*" workflows	**6 – 66.7%**	**9**
Missed cases	**3 – 33.3%**	**0**
False positives	**4 – 19%**	**0**
Correctly classified normal workflows	**17 – 81%**	**21**

Table 1: WIMS classification on BTS cases.

System experts were called afterwards to comment on the findings shown by the CBR-WIMS. Since the focus of the experiment was not on the quantitative aspect of monitoring but on the qualitative one, the experts were shown the results and were called to comment on the system's efficiency. It was acknowledged that CBR-WIMS offers a flexible architecture, offering rapid integration to an existing business process due to its high design modularity. Table 1 below summarises the results among the experts' monitoring and CBR-WIMS findings. From the results it can be seen that CBR-WIMS can significantly contribute to the experts' monitoring assistance.

6 Conclusions

This paper has presented the application of CBR-WIMS, an intelligent monitoring platform for agile business workflows. The system was shown to be efficient in monitoring workflow execution cases as well as reusable in terms of its components adaptation. Major advantage of the CBR-WIMS system is its generic architecture which allows a simple step-by-step integration to existing systems using SOA architecture. This paper has presented its application on two enterprise systems real-life systems coming from different operational disciplines. The system has been proven effective in dynamically assisting human experts in workflow monitoring tasks. Future work will focus more on the operational metrics when applied to different business process environments as well as the in-depth adaptation of the system. The aim is towards integration on current systems. Finally, further work will be concentrated to the investigation of the reusability limits of past experience among different workflows.

References

1. Business Process Management Initiative (BPMI): BPMN 2.0: OMG Specification, January, 2011, http://www.omg.org/spec/BPMN/2.0
2. OASIS: BPEL, The Web Services Business Process Execution Language Version 2.0, http://www.oasis-open.org/apps/org/workgroup/wsbpel.
3. Workflow Management Coalition (WfMC): XPDL 2.1 Complete Specification (Updated Oct 10, 2008), http://www.wfmc.org/xpdl.html
4. Hill, J. B., Sinur, J., Flint, D., Melenovsky, M. J.: Gartner's Position on Business Process Management. Gartner, Inc. (2006)
5. Kapetanakis, S., Petridis, M., Knight, B., Ma, J.,Bacon, L. : A Case Based Reasoning Approach for the Monitoring of Business Workflows, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, LNAI (2010)
6. Kapetanakis, S., Petridis, Ma, J., Bacon, L.: Providing Explanations for the Intelligent Monitoring of Business Workflows Using Case-Based Reasoning. In Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.) Proceedings of the Fifth International workshop on Explanation-aware Computing ExaCt (ECAI 2010). Lisbon, Portugal
7. Aamodt A., Plaza E.: Case-based reasoning; Foundational issues, methodological variations, and system approaches. AI Communications, vol. 7, no. 1, pp. 39-59 (1994)
8. Minor, M., Tartakovski, A. and Bergmann, R.: Representation and Structure-Based Similarity Assessment for Agile Workflows, in Weber, R., O. and Richter, M., M. (eds.) CBR Research and Development, Proceedings of the 7th international conference on Case-Based Reasoning, ICCBR 2007, Belfast, NI, UK, August 2007, LNAI 4626, pp 224-238, Springer-Verlag (2007)
9. Dijkman, R. M., Dumas, M., Garcia-Banuelos, L.: Graph matching algorithms for business process model similarity search. In Dayal, U., Eder, J. (eds.), Proc. of the 7th Int. conference on business process management. (LNCS, Vol. 5701, pp. 48-63). Berlin: Springer (2009)
10. van der Aalst, W., Alves de Medeiros, A. K., Weijters, A.: Process Equivalence: Comparing two Process Models Based on Observed Behavior, In Proc. Of BPM 2006, vol 4102 of LNCS, pp 129-144, Springer, (2006)
11. Regev G., Wegmann A.: Regulation Based Linking of Strategic Goals and Business Processes, Workshop on Goal-Oriented Business Process Modelling, London (2002)
12. Bunke, H., Messmer, B.T.: Similarity Measures for Structured Representations. In: Wess, S., Richter, M., Althoff, K.-D. (eds.) Topics in Case-Based Reasoning. LNCS, vol. 837, pp. 106-118, Springer, Heidelberg (1994)
13. Mileman, T., Knight, B., Petridis, M., Cowell, D., Ewer, J.: Case-Based Retrieval of 3-D shapes for the design of metal castings in Journal of Intelligent Manufacturing, Kluwer. 13(1): 39-45 (2002)
14. Wolf, M., Petridis, M.: Measuring Similarity of Software Designs using Graph Matching for CBR, In workshop proceedings of AISEW 2008 at ECAI 2008, Patras, Greece (2008)
15. Recio-García, J. A., Sánchez-Ruiz, A. A., Díaz-Agudo, B., González-Calero, P. A.: jCOLIBRI 1.0 in a nutshell. A software tool for designing CBR systems, in: Proccedings of the 10th UK Workshop on Case Based Reasoning, CMS Press,University of Greenwich (2005)
16. Stahl, A., Roth-Berghofer, Th.: Rapid prototyping of CBR Applications with the Open Source Tool myCBR. Künstliche Intelligenz, 23(1):34–37 (2009)
17. Kapetanakis, S., Petridis, M., Ma, J., Knight, B.: CBR-WIMS, an Intelligent Monitoring Platform for Business Processes. In Petridis, M. (ed.): Proceedings of the 15th UK CBR workshop, pp. 55-63. Cambridge: CMS press (2010)
18. Ma, J., Knight, B.: A General Temporal Theory, the Computer Journal, 37(2), 114-123 (1994).

CBR in the Health Sciences

Workshop at the
Twentieth International Conference on
Case-Based Reasoning
(ICCBR 2012)

Lyon, France
September, 2012

Stefania Montani, Cindy Marling and
Isabelle Bichindaritz (Eds.)

Co-Chairs

Stefania Montani
University of Piemonte Orientale, Italy

Cindy Marling
Ohio University, USA

Isabelle Bichindaritz
University of Washington, USA

Programme Committee

Agnar Aamodt, NTNU, Norway
Klaus-Dieter Althoff, University of Hildesheim, Germany
Peter Funk, Malardalen University, Sweden
Alec Holt, University of Otago, New Zealand
Jean Lieber, Loria, University of Nancy, France
Beatriz Lopez, University of Girona, Spain
Stefan Pantazi, Conestoga College Institute of Technology, Canada
Petra Perner, Institute of Computer Vision and Applied Computer Sciences, Germany
Luigi Portinale, University of Piemonte Orientale, Italy
Olga Vorobieva, I. M. Sechenov Institute of Evolutionary Physiology and Biochemistry, Russia

Preface

The research community working on health sciences applications of case-based reasoning (CBR) has been very active recently, as evidenced by special issues of premier AI journals, as well as by books of edited collections on the topic. After a short break, the community meets again at the International Conference on Case-based Reasoning (ICCBR) this year, and many new ideas and system descrip- tions are collected in the proceedings of this workshop. This event is the eighth in a series of successful workshops, co-located with different ICCBR/ECCBR conferences. The first seven were held at ICCBR-03, in Trondheim, Norway, at ECCBR-04, in Madrid, Spain, at ICCBR-05, in Chicago, USA, at ECCBR-06 in Oludeniz, Turkey, at ICCBR-07 in Belfast, Ulster, at ECCBR-08 in Trier, Germany, and at ICCBR-09 in Seattle, USA.

The workshop is organized in two tracks this year: a regular track on methodological as well as more applicative contributions of CBR research in health care; and a special track devoted to system overview presentations, which showcase specific CBR systems.

Five papers are to be presented within the regular track. They deal with time series retrieval in hemodialysis [Montani et al.], knowledge inference through a combination of AI techniques for cardiac arrhythmias diagnosis [Khelassi and Chikh], graph-based case representation for supporting frontotemporal dementia protocols [Juarez et al.], multiple sensor data fusion and analysis for patient classification [Begum et al.], and case-based patient classification by exploiting time series features and personalized health profiling [Ahmed et al.].

Five papers will showcase systems, and are related to diabetes management [Mar- ling et al.], stress management [Ahmed et al.], end-of-life cancer care [Elvidge], can- cer prognosis [López et al.] and long-term follow-up of oncological patients [Bichindaritz].

The ten papers report on the research and experience of twenty-seven authors working in six different countries on a wide range of problems and projects, and illus- trate some of the major trends of current research in the area. Overall, they represent an excellent sample of the most recent advances of CBR in the health sciences, and promise very interesting discussions and interaction between the major contributors in this niche of CBR research.

Stefania Montani September 2012
Cindy Marling
Isabelle Bichindaritz

Extending a time series retrieval tool to deal with sub-series matching: an application to the hemodialysis domain

S. Montani, G. Leonardi, A. Bottrighi, L. Portinale, P. Terenziani

DISIT, Sezione di Informatica, Università del Piemonte Orientale, Alessandria, Italy

Abstract. The problem of retrieving time series similar to a specified query pattern has been recently addressed within the Case Based Reasoning (CBR) literature. Providing a flexible and efficient way of dealing with such an issue is of paramount importance in medical domains, where many patient parameters are often collected in the form of time series. In the past, we have developed a framework for retrieving cases with time series features, relying on Temporal Abstractions. With respect to more classical (mathematical) approaches, our framework provides significant advantages. In particular, multi-level abstraction mechanisms and proper indexing techniques allow for flexible query issuing, and for efficient and interactive query answering. In this paper, we present an extension to such a framework, aimed at supporting sub-series matching as well. Indeed, sub-series retrieval may be crucial in medical applications, when the whole time series evolution is sometimes not of interest, while critical patterns to be search for are only "local". Moreover, their relative order, but not their precise location in time, may be known, and an interactive search, at different abstraction levels, may be of great help for the medical decision maker. The framework is currently being applied to the hemodialysis domain.

1 Introduction

Several real world applications require to capture the evolution of the observed phenomenon over time, in order to describe its behavior, and to exploit this information for future problem solving. This issue is particularly relevant in medical applications, where the physician typically needs to recall the clinical history that led the patient to the current condition, before prescribing a therapy; actually, the pattern of the patient's changes is often more important than her/his final state. The need for capturing the phenomenon's temporal evolution emerges even more strongly when a continuous monitoring of the patient's health indicators is required, such as in chronic diseases, or when control instruments that automatically sample and record biological signals are adopted, such as in hemodialysis and Intensive Care Units. In these applications, (many) process features are naturally collected in the form of **time series**, either automatically generated and stored by the control instruments (as e.g. in hemodialysis), or obtained by listing single values extracted from temporally consecutive situations (as e.g. the series of glycated hemoglobin values, measured on a diabetic patient once every two months).

Interpreting time series features on screen or on paper can be tedious and prone to errors. Physicians may be asked to recognize small or rare irregularities in the series itself, or to identify partial similarities with previous situations, such as retrieving the past occurrence of a similar trend behavior in a given feature, independently of its actual values. Such identification may be extremely relevant for patient care, but may also require a significant amount of expertise in the specific field [16]. A user-friendly and flexible retrieval support is therefore strongly desirable in the medical domain.

The problem of retrieving time series similar to a given query pattern has been recently addressed specifically within the Case Based Reasoning (CBR) [1] literature. In particular, various case-based retrieval works dealing with cases with time series features have been recently published. Referring to the medical domain, we can mention the works in [14–16]. In these applications, time series retrieval resorts to a mathematical transform, able to preserve the distance between two time series (or to underestimate it). However, mathematical transforms have several limitations, since they usually work in a black box fashion with respect to end users, lacking of flexibility, clear interpretation and interactivity.

More recently, we have introduced a novel framework [13], resorting to a different technique for time series retrieval, namely **Temporal Abstractions** (TA) [18, 3, 17, 12]. Rather interestingly, TA have been extensively resorted to in the literature, especially in the medical field, from diabetes mellitus [19, 2], to artificial ventilation of intensive care units patients [11] (see also the survey in [21]), but typically with the aim to solve a data interpretation task [18], and not as a retrieval support facility. To the best of our knowledge, our framework represents the first effort in the direction of supporting data interpretation as well as case exploration and retrieval.

TA are not the only methodology for reducing dimensionality by transforming a time series into a sequence of symbols. Actually a wide number of symbolic representations of time series have been introduced in the past decades (see [4] for a survey). However, some of them require an extremely specific and hard to obtain domain knowledge [7]. Rather interestingly, Lin [9] has introduced an alternative to TA, capable to deal with such issues, in which intervals are first obtained through Piecewise Constant Approximation [8], and subsequently labeled with proper symbols. However, the approach in [9] is not as simple as TA, which allows a clear interpretation of the qualitative description of the data provided by the abstraction process itself. As a matter of fact, such a description is often easier to understand for end users (e.g. clinicians [20]), and easily adapts to domains where data values that are considered as normal at one time, or in a given context, may become dangerously abnormal in a different situation (as in medicine, due to disease progression or to treatments obsolescence). And, of course, the ease and flexibility at which knowledge can be managed and understood by experts is an aspect that impacts upon the suitability and the usefulness of decision support systems in practice.

In summary, our framework shares with other methods the dimensionality reduction aspect (in both cases the original raw time series is significantly reduced in size), but it is also characterized by a strong degree of flexibility, interactivity, and computational efficiency (see section 2 for some details). Moreover, also the interpretation of the resulting transformation (from raw data to TA) is clear and user-friendly. However, the user is forced to express only queries that cover the overall temporal duration of the time series.

The goal of the present work is the one of addressing this additional issue, which is of great usefulness in many real world domains. In particular, sub-series retrieval may be of paramount importance in medical applications, when the whole time series evolution is sometimes not of interest, while critical patterns to be search for are only "local". Moreover, their relative order, but not their precise location in time, may be known, and an interactive search, at different abstraction levels, may be of great help for the medical decision maker. We are currently developing extensions focusing on these needs, that are described in section 3. The work is exemplified referring to the field of hemodialysis. Finally, section 4 reports on our conclusions and future work directions.

2 A flexible time series retrieval framework

As sketched in the Introduction, in recent years we have defined a framework [13] for time series retrieval, which exploits the TA methodology to reduce time series dimensionality, and multi-dimensional index structures to make retrieval efficient.

TA allow to map huge amounts of temporal information to a compact representation, by aggregating adjacent time series points sharing a common behavior (e.g. the same qualitative level, the same trend direction) into a single interval, labeled by a proper symbol. This technique not only summarizes the original longitudinal data, but also highlights meaningful behaviors in the data in a clear symbolic high level view. Specifically, in [13] we support *multi-level abstractions* of the original data. Time series values can be abstracted (and queried) at finer or coarser detail levels, according to two dimensions: a taxonomy of symbols, and a taxonomy of time granularities. For instance, a taxonomy of trend symbols can be introduced (see figure 1 - left), in which the symbol I (increase) is further specialized into I_W (weak increase) and I_S (strong increase), according to the slope. As regards time granularity, a series of two adjacent intervals of I_W, each one with a duration of half an hour, can be merged into a single I_W interval, with a duration of 1 hour. In order to abstract along the temporal dimension, a function for scaling up from two or more symbols expressed at a specific time granularity to a single symbol expressed at the coarser one must be provided. We call this function the *up* function: $up(X_1, X_2, \dots X_k) = Y$ means that a sequence of symbols $X_1, \dots X_k$ (specified at the same time granularity) is abstracted into the symbol Y at the coarser time granularity. Such an abstraction is a domain-

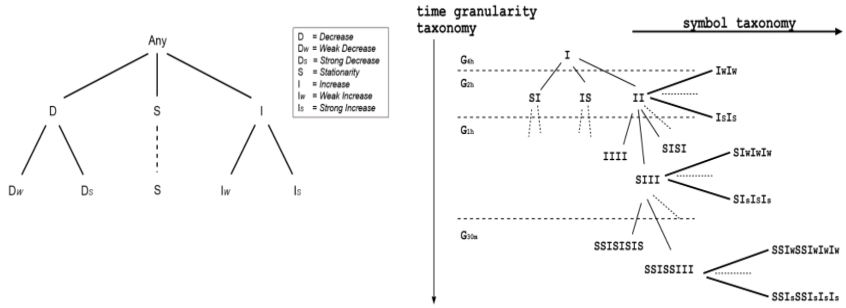


Fig. 1. An example symbol taxonomy (left). An example multi-level orthogonal index structure (right). The symbol taxonomy is articulated into two levels (where e.g. I stands for increase, I_w and I_s for weak increase and strong increase respectively; S stands for stationarity - and cannot be further specialized). Time granularity can be progressively refined from 4 hours to 30 minutes.

dependent procedure. In order to retain the maximal generality, our framework allows one to freely define the *up* function. However, we impose some general constraints, such as "persistence" (the result of coarsening two granules with the same symbol x is a larger granule still labeled as x) and "monotonicity" (ordering among symbols, if any, is preserved by the *up* function) - see [13]. Once abstracted, time series can be retrieved, relying upon a proper *distance* definition, able to measure the similarity between symbols in the taxonomy. Since our approach is general, it is parametric with respect to the choice of the distance function. We just enforce that the distance of each symbol from itself is zero and the fact that distance must be "consistent" with symbol ordering (if any) - see [13].

In order to increase *retrieval efficiency*, in [13] we have implemented an indexing strategy, exploiting orthogonal indexes built on the basis of the two taxonomies mentioned above.

In particular, we resort to a forest of index structures, providing a flexible indexing of cases at different levels of the symbol and/or time granularity taxonomies. The root node of each index structure is represented by a string of symbols, defined at the highest level in both dimensions. An example, taking as a root the I (increasing trend) symbol, is provided in figure 1 - right. Here, the root node I is refined along the time dimension (which is taken as the leading dimension) from the 4 hours to the 2 hours granularity, so that the nodes II, IS and SI stem from it, provided that $up(I, S) = I$ and $up(S, I) = I$ - where S stands for stationarity. Further refinements in time then take place, until reaching the 30 minutes granularity. From each node of the leading dimension structure, another index can stem, built according to the symbol taxonomy dimension, keeping the time granularity abstraction level always fixed. Such an index develops orthogonally with respect to the leading dimension. Observe that indexes are often incomplete (in order to reduce memory occupancy), but could be refined on demand.

In summary, time series features in our cases are processed by means of TA, executed at the ground level according both to the symbol taxonomy and to the time granularity one. Queries on such data can then be issued at any level of detail, according to both dimensions. Indexes allow for a quick (and interactive) query answering.

However, in the approach in [13] the user is forced to express only queries that cover the overall temporal duration of the time series. In the next sections we will discuss how we are extending the framework in order to deal with sub-series matching as well.

3 Extending the framework to deal with sub-series retrieval

Despite the flexibility of the framework we have implemented, and the positive experimental results we have obtained [13], our work does not allow users to query for sub-series, i.e. to express queries in the form of (a set of) substrings, to be matched to the time series contained in the database, without knowing in advance their precise location on the time axis. On the other hand, such a functionality would be very useful in many real world domains. In particular, we are currently working in the field of hemodialysis, in which we collected some case studies that would strongly benefit of this extension. In the next sections, we will first introduce a motivating example. Then, we will move to a description of how sub-series to be retrieved can be expressed in our query language. Finally we will discuss on how sub-series retrieval is implemented.

3.1 Sub-series retrieval in the hemodialysis domain: a motivating example

Hemodialysis is the most widely used treatment for End Stage Renal Disease, a severe chronic condition which, without medical intervention, leads to death. Hemodialysis relies on a device, called hemodialyzer, which clears the patient's blood from catabolites, to re-establish acid-base equilibrium and to remove water in excess. On average, hemodialysis patients are treated for four hours three times a week. Each single treatment is called a hemodialysis session, during which the hemodialyzer collects several variables, most of which are in the form of time series. Considering a case as a hemodialysis session, cases are thus characterized by time series features, that we pre-process by means of TA. In particular, we will focus our attention on cases with a single time series feature, for the sake of clarity: namely, diastolic pressure. Diastolic pressure is a very powerful indicator for evaluating water reduction from the patient's blood during a session. The reduction of water causes a constant decrease of the blood pressure. This behavior is correct and, even if it can sometimes cause minor problems to the patient (e.g. light head spinning), it is necessary. However, in certain conditions, the reduction of water is not constant, but can be characterized by stationarity periods and (sudden) increasing or decreasing trend episodes. In particular, problems arise when the pressure remains stationary for almost half of the session, which means that no water reduction takes place. Then (sharp) decreasing or increasing episodes can take place, destabilizing the cardiovascular system of the patient. Clinical studies state that intradialytic increase in diastolic pressure can complicate the management of hypertension in hemodialysis patients. Furthermore, diastolic pressure increasing trend is associated with greater 2-year mortality in these patients [6]. We are thus particularly interested in looking for cases in which an initial stationarity episode is followed by one or more episodes of increasing trend. The sum of these episodes lengths does not need to cover the whole session duration (i.e. 4 hours). Moreover, the precise location of the episodes on the time line is not very relevant: we just want to find cases in which stationary diastolic pressure is followed by increasing pressure, provided that their distance in time is included in a specific numerical interval.

3.2 Sub-series queries

Through the system interface, the user can build a query, composed by one or more substrings to be searched for, at a specific time granularity, and at a specific level in the symbol taxonomy (see section 2).

The user query is mapped by the system to the following format:

$< granularity, level >$

$< (min_1, max_1)SUB_1(min_2, max_2)SUB_2...(min_k, max_k)SUB_k(min_{k+1}, max_{k+1}) >$

where:

- $< granularity, level >$ represent the chosen time granularity and symbol taxonomy level;
- SUB_j represents a substring (i.e. a sequence of symbols expressed at the given time granularity and symbol taxonomy level);
- (min_j, max_j) represents the minimal and maximal delay between the two items (i.e. substrings, or time series starting/ending point) it is placed between.

As an example, the query

$< 30m, 2 >< (0, 2)SS(0, 2)I_w(1, 5) >$

where every symbol is expressed at the 30 minutes granularity and at the most specific level in the symbol taxonomy (level 2, see figure 1 - left), addresses the problem illustrated in section 3.1, asking to look for 1 hour of stationarity, that may begin immediately after the session start, or up to 1 hour later (i.e. the pattern $(0, 2)SS$). The stationarity episode has to be followed by half an hour of weakly increasing trend, that may start immediately after the stationarity episode itself, or up to 1 hour later (i.e. the pattern $(0, 2)I_w$); finally the whole series must end in a time period between half an hour and two and a half hours (i.e. the final pattern $(1, 5)$), in order to cover the whole 4 hours duration.

It is worth noting that such a query is intensional: it allows to express at a high level, in a compact and user-friendly way, a very large set of extensions. Indeed, without relying on our framework, the user would be forced to issue a distinguished query for every possible delay value, also providing all possible combinations of symbols for each of the un-interesting parts of the overall time series (e.g the tail portion). Moreover, queries should be expressed at the ground level, according both to the time granularity, and to the taxonomy of symbols. The problem is obviously combinatorial, with respect to the possible delay values and symbols. We thus believe this would not be feasible in practice. Observe that, in our framework, the consistency of the temporal constraints can be automatically checked before answering the query [5]. Also note that a string covering the overall time series duration (e.g. 4 hours in the hemodialysis domain) is just a special case of sub-series query.

3.3 Sub-series query answering

If the time granularity and the symbol taxonomy level chosen in the query exist in one of the indexes, we apply a classical pattern matching algorithm on the nodes at the given level. Indeed, pattern matching algorithms that match regular expressions (e.g. [22, 10]) are very well suited for our task.

On the other hand, it may happen that the query symbol taxonomy or granularity level are missing in the indexes, as they can be incomplete. If this situation holds, we need to properly abstract the query, in order to compare it to the index nodes. To this end, we have to preliminarily generate all the extensions of the query, each for every possible delay value. In fact, a query written as in section 3.2 corresponds to a whole set of queries, each one obtained by choosing a specific delay value in each of the (min_j, max_j) intervals. In the temporal dimension, duration (e.g. delay length) is expressed by using a proper number of symbols at the given granularity (e.g. 2 symbols to cover a 1 hour duration, if working at the 30 minutes granularity). Therefore, every query in such a set has to be made explicit as a string, containing as many "dummy" symbols $*$ as needed, to cover the corresponding delay length (where the dummy symbol is chosen because we don't care about what kind of pattern characterizes the time series in that temporal location).

Such queries are still only partially explicit, because they contain dummy symbols. However, the *up* function introduced in section 2 can take dummy symbols as an input. The result of abstracting (one or more) dummy symbol(s) is obviously domain dependent, and has to be defined by experts in the system set up phase. Then, as second step, the partially explicit queries can undergo a proper abstraction process (by resorting to the *up* function), in order to be mapped to nodes that exist in the indexes. (Simplified) pattern matching can finally be applied. In this situation, therefore, the first step of query answering consists in making query extensions (partially) explicit. This is done by means of the algorithm illustrated in figure 2.

```
1.   SetOfString ExplicitQuery(query,minLength,maxLength){
2.   output ← { }
3.   FOR EACH e ∈ query{
4.        temp ← { }
5.        IF (e is a delay) {
6.             stringdummyset ← build(min(e), max(e))
7.             IF (output = { }) THEN
8.                  temp ← stringdummyset
9.             ELSE
10.                 FOR EACH i ∈ output
11.                      FOR EACH m ∈ stringdummyset{
12.                           i ← (i concat m)
13.                           IF (NumberOfSymbols(i)= maxLength) THEN
14.                                temp ← temp ∪ {i}
15.                           }
16.            }
17.       ELSE
18.            IF (e is a substring) {
19.                 IF (output = { }) THEN
20.                      temp ← {e}
21.                 ELSE
22.                      FOR EACH i ∈ output{
23.                           i ← (i concat e)
24.                           IF (NumberOfSymbols(i) = maxLength) THEN
25.                                temp ← temp ∪ {i}
26.                           }
27.                 }
28.       IF (temp = { }) THEN
29.            RETURN { }
30.       ELSE
31.            output ← temp
32.       }
33.  FOR EACH i ∈ output
34.  IF NumberOfSymbols(i) < minLength  THEN
35.       output ← output - {i}
36.  RETURN output
37.  }
```

Fig. 2. Pseudo-code of the algorithm for converting a query in the intensional form into all its extensions with respect to time delay values.

The algorithm takes in input the query (*query*), in the format described in section 3.2, and two integers (*minLength* and *maxLength*), which correspond to the minimal and maximal length of all the time series to be created (according to the specific domain characteristics). It generates a set (*output*) containing all the extensions of the input query with respect to delay values. Specifically, all the items in *output* will be strings of symbols expressed at the chosen time granularity and symbol taxonomy level, and will be composed by as many symbols as needed to cover a time duration contained between *minLength* and *maxLength* (strings which are too long or too short will be excluded, see lines 13, 24 and 34). In detail, the algorithm processes the elements in *query* one at a time. At every intermediate step of the main cycle (lines 3-32) the algorithm analyses the current query element *e*. It exploits the current content of *output*, containing the already partially generated strings (excluding the contribution of *e*), and sets the variable *temp*, in which it expands all the *output* strings to consider the contribution of *e* as well. Finally, it overwrites *output*, by saving the content of *temp*. If the current element *e* is a delay (i.e. it is in the form (min_j, max_j), see section 3.2), it generates all the strings made by $*$ symbols, whose length is included between min_j and max_j (i.e. *min(e)* and *max(e)* in the pseudo-code of figure 2). Specifically, such strings are generated through a call to the *build* procedure, and saved into *stringdummyset* (line 6). Then, every string in *stringdummyset* is concatenated to all the elements already saved in *output* (lines 10-16). On the other hand, if the current element e is a string (i.e. it is in the form SUB_j, see section 3.2), it is concatenated to all the elements already saved in *output* (lines 22-26). For instance, given that *query*$=< (0,2)SS(0,2)I_w(1,5) >$, *minLength* has to be set to 0 and *maxLength* has to be set to 8 (to cover the whole 4 hours duration by operating at the 30 minutes granularity). The content of *output* is then calculated as follows:

$(SSI_w *****$, $*SSI_w ****$, $**SSI_w ***$, $SS*I_w ****$, $*SS*I_w ***$, $**SS*I_w **$, $SS**I_w ***$, $*SS**I_w **$, $**SS**I_w*)$

Observe that the problem of generating query partial extensions is obviously combinatorial in nature. However, in the hemodialysis application the time granularity taxonomy is quite limited in height, so that we do not have to deal with a very large imprecision in delays. This makes the algorithm efficient in practice.

4 Conclusion

In this paper, we have presented an extension to a time series retrieval framework we defined in the past, which was based on multi-dimensional Temporal Abstractions. Such an extension allows us to deal with sub-series retrieval, while maintaining the characteristics of flexibility of the former version.

In our opinion, this further improvement increases the usefulness and usability of the tool, especially as regards its application in medical domains.

The framework is currently being applied to the hemodialysis domain, where we plan to conduct an extensive experimental work.

In the future, we will also consider how to optimize computational time, e.g. by introducing heuristics or user preference mechanisms, in order to prune some of the query extensions since the beginning.

References

1. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. *AI Communications*, 7:39–59, 1994.
2. R. Bellazzi, C. Larizza, P. Magni, S. Montani, and M. Stefanelli. Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. *Artificial Intelligence in Medicine*, 20:37–57, 2000.
3. R. Bellazzi, C. Larizza, and A. Riva. Temporal abstractions for interpreting diabetic patients monitoring data. *Intelligent Data Analysis*, 2:97–122, 1998.
4. C.S. Daw, C.E. Finney, and E.R. Tracy. Symbolic analysis of experimental data. *Review of Scientific Instruments*, 2002-07-22, 2001.
5. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Knowledge Representation*, pages 61–95. MIT Press, London, 1991.

6. Jula K. Inrig, Uptal D. Patel, Robert D. Toto, and Lynda A. Szczech. Association of blood pressure increases during hemodialysis with 2-year mortality in incident hemodialysis patients: A secondary analysis of the dialysis morbidity and mortality wave 2 study. *American Journal of Kidney Diseases*, 54(5):881 – 890, 2009.
7. S. Kadar, J. Wang, and K. Showalter. Noise-supported travelling waves in sub-excitable media. *Nature*, 391:770–772, 1998.
8. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263–286, 2000.
9. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of ACM-DMKD*, San Diego, 2003.
10. R. McNaughton and H. Yamada. Regular expressions and state graphs for automata. *IRE Transactions on Electronic Computing*, 9:39–47, 1960.
11. S. Miksch, W. Horn, C. Popow, and F. Paky. Utilizing temporal data abstractions for data validation and therapy planning for artificially ventilated newborn infants. *Artificial Intelligence in Medicine*, 8:543–576, 1996.
12. S. Montani, A. Bottrighi, G. Leonardi, and L. Portinale. A cbr-based, closed loop architecture for temporal abstractions configuration. *Computational Intelligence*, 25(3):235–249, 2009.
13. S. Montani, G. Leonardi, A. Bottrighi, L. Portinale, and P. Terenziani. Supporting flexible, efficient and user-interpretable retrieval of similar time series. *IEEE Transactions on Knowledge and Data Engineering*, (accepted for publication: DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/TKDE.2011.264).
14. S. Montani, L. Portinale, G. Leonardi, R. Bellazzi, and R. Bellazzi. Case-based retrieval to support the treatment of end stage renal failure patients. *Artificial Intelligence in Medicine*, 37:31–42, 2006.
15. M. Nilsson. Retrieving short and dynamic biomedical sequences. In *Proc. 18th international florida artificial intelligence research society conference–special track on case-based reasoning*. AAAI Press, 2005.
16. M. Nilsson, P. Funk, and N. Xiong. Clinical decision support by time series classification using wavelets. In C.S. Chen, J. Filipe, I. Seruca, and J. Cordeiro, editors, *Proc. Seventh International Conference on Enterprise Information Systems (ICEIS'05)*, pages 169–175. INSTICC Press, 2005.
17. L. Portinale, S. Montani, A. Bottrighi, G. Leonardi, and J. Juarez. A case-based architecture for temporal abstraction configuration and processing. In *Proc. IEEE International Conference on Tools with Artificial Intelligent (ICTAI)*, pages 667–674. IEEE Computer Society, Los Alamitos, 2006.
18. Y. Shahar. A framework for knowledge-based temporal abstractions. *Artificial Intelligence*, 90:79–133, 1997.
19. Y. Shahar and M.A. Musen. Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine*, 8:267–298, 1996.
20. M. Stacey. Knowledge based temporal abstractions within the neonatal intesive care domain. In *Proc. CSTE Innovation Conference, University of Western Sidney*, 2005.
21. P. Terenziani, E. German, and Y. Shahar. The temporal aspects of clinical guidelines. In A. Ten Teije, S. Miksch, and P. Lucas, editors, *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, 2008.
22. K. Thompson. Regular expression search algorithm. *Communications of ACM*, 11:419–422, 1968.

Cognitive Amalgam with a Fuzzy sets and case based reasoning for accurate cardiac arrhythmias diagnosis

Abdeldjalil Khelassi[1], Mohammed Amine CHIKH[2]

[1] Informatics department, Abou Bakr Belkaid University,
[2]Biomedical Lab, Abou Bakr Belkaid University,
BP 230 Tlemcen, Algeria.
Khelassi.a@gmail.com, ma_chikh@yahoo.fr

Abstract. In This paper a cognitive amalgam for inferring from distributed and heterogeneous knowledge bases by using a set of cognitive agents is presented. This cognitive amalgam is developed by combining some intelligent approaches and algorithms, as well as case based reasoning, rule based reasoning, distributed reasoning and fuzzy sets to meet the needs of medical applications and improve their efficiency and transparency. Through the system criteria and some empirical experiments, applied to a data set extracted from the international MIT-BIH Electrocardiogram (ECG) records, it is concluded that the developed system achieves such average accuracies and performances better than most of the cited state-of-the-art approaches.

Keywords*:* Case Based Reasoning, Distributed reasoning, Fuzzy sets cardiac arrhythmia recognition, ECG.

1 Introduction

The integration of knowledge for multiple sources is an important aspect of automated reasoning systems. When different knowledge bases are used to store knowledge provided by multiple sources, we are faced with the problem of integrating multiple knowledge bases: under these circumstances, we are also confronted with the prospect of inconsistency.
The Case Based Reasoning CBR is an intelligent approach inspired from many disciplines. It draws the human reasoning model. It consists of using the prior expertise to resolve a new problem [1]. The distributed case based reasoning is a variant of CBR in which the reasoning is distributed through a set of agents and the cases through a set of case bases [2].
Fuzzy models are widely used in many areas like expert systems, pattern recognition, or system modeling. Fuzzy models are especially useful when facing inter-disciplinary problems, where the presentation of a purely mathematical description to the domain expert is not desirable. Unfortunately, this transparency aspect is neglected in many approaches that can be found in the literature. Aiming at a high

approximation quality, some authors tend to allow any transformation of the fuzzy sets, ending up with a fuzzy model that is not interpretable any longer. [3]
In this paper a medical application, in which a set of cognitive agents for inferring from a set of cases bases and knowledge bases in order to recognize the cardiac arrhythmias from a set of features measured from the Electrocardiogram ECG is presented. This application is developed to improve the accuracy and the performance of the reasoning systems as well as to enrich the transparency of the reasoning.
In this work, the employed approaches are described, namely the case based reasoning, the fuzzy sets. The cardiac arrhythmias and the ECG are also presented. The reasoning process in the developed decision support system IKCBRC is explained. In order to prove the quality of the contribution some empirical experiments with data taken from the MIT-BIH database [4] are performed.

2 The case based reasoning

The Case base reasoning systems are widely used to diagnose human disease many health sciences application are cited in [33]. The case based reasoning CBR approach is also widely and successfully applied in many other domains as games, recommendation systems, information retrieval, industrial applications and others. It represents a good and easy method of knowledge extraction, discovery and modeling.
The case is a contextualized piece of knowledge representing an experience that teaches a lesson fundamental for achieving the goals of the reasoning system [5]. The CBR is an intelligent approach inspired from many disciplines. It draws a human reasoning model [6]. It consists of using the prior expertise to resolve a new problem. This expertise is stored as a set or collection of cases called cases base. Each case represents one problem associated with its solution. The main idea of case based reasoning is that two similar problems have the same solutions or the solution can be generated from similar problems.
According to Richter [7], the knowledge container is a collection of knowledge that is relevant to many tasks. For rule-based systems, for instance, one can easily identify facts and rules as important knowledge containers. For CBR systems, Richter describes four knowledge containers: vocabulary, similarity measures, adaptation knowledge, and case base. They are depicted in Figure.1.

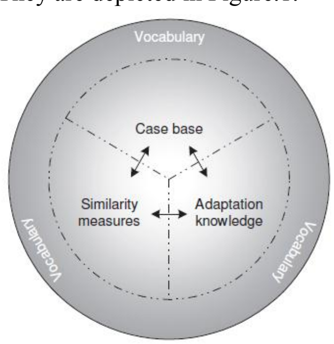


Figure 1: The four knowledge containers of a CBR system

Vocabulary container: the set of attributes, entities and structures used to represent the cases (problems and solutions). It can be characterized as the language words used to talk about the domain. It is also called the ontology knowledge container.
Case knowledge contains the past structured experience which will be exploited by the system. In other words, it is situation-specific knowledge obtained from the past situations in problem solving.
Similarity measure container: general knowledge required to select or to retrieve similar cases to be reused in a particular problem situation.
Adaptation knowledge: general knowledge needed to allow an efficient reuse of retrieved cases. It takes a form of Heuristics and algorithms used to modify the solution and to evaluate their usability for the new situations.
The distributed case based reasoning consists of distributing the reasoning through a set of autonomic agents and we talk about distributed reasoning, and the cases through a set of case bases and we talk about distributed case base. This notion increases the flexibility of the reasoning system and improves the performance and speedup of the reasoning because the reasoning is distributed through a set of local reasoning sub-systems. There are many works related to the distribution of reasoning like [8][9][10][11][12] but each one has its proper realization and strategy. Research efforts in the area of distributed CBR concentrate on the distribution of resources with the intent of improving the performance of CBR systems. Although the phrase distributed CBR can be used in a number of different contexts. [2]

3 Fuzzy sets

The fuzzy sets [13] generalize the classical sets by considering the membership as a graded concept. The membership degree of an element x to a fuzzy set A denoted by $\mu_A(x)$, takes a value in the interval [0,1].
Definition1: The support of a fuzzy set A is the crisp set that contains all the elements of X that have nonzero membership grades in A.

$$Supp(A)=\{x\epsilon X, \mu_A(x)>0\}$$

Definition2: The core of a normal fuzzy set A is the crisp set that contains all the elements of X that have the membership grades of one in A.

$$core(A)=\{x\epsilon X, \mu_A(x)=1\}$$

Definition3: The boundary is the crisp set that contains all the elements of X that have the membership grades of $0< \mu_A(x) <1$ in A.

$$Bnd\ (A)=\{x\epsilon X, 0< \mu_A(x) <1\}$$

Definition4: Having two fuzzy sets A and B based on X, then both are similar if:

$$Core\ (A)=Core\ (B) \text{ and } Supp(A)=Supp(B)$$

Definition5: If the support of a normal fuzzy set consists of a single element x_0 of X, which has the property Supp(A)=Core(A)={x0}, Then this set is called a singleton and the membership function is called triangular function.

Fuzzy models have been widely and successfully used in many areas such as data mining [16], data analysis [14] and image processing [15]. Also it has been successfully integrated with other approaches like artificial neuronal networks (neuro-

fuzzy approach) in [14] and [18] and rule based systems (Fuzzy Logic) in [15, 16] and case based reasoning in [28,29,30,35, 36].
Traditionally, fuzzy rules and subsets are generated from human expert knowledge or by using some machine learning algorithms [17] or heuristics, which brings about good high-level semantic generalization capability. On the other hand, some researchers have made efforts to build fuzzy models from observational data, leading to many successful applications such as [22] and [18]. Also, more and more efforts have been made to approach the problem of interpretability and transparency of data-driven by fuzzy models in [19], [20] and [21].

4 Cardiac arrhythmias

The cardiac arrhythmias are a group of conditions in which the electrical activity of the heart is irregular or is faster or slower than normal. Some arrhythmias are life-threatening medical emergencies that can cause cardiac arrest and sudden death. Others cause aggravating symptoms, such as an awareness of a different heartbeat, or palpitation, which can be annoying. Some are quite small and normal.

4.1 The Electrocardiogram ECG

Up till now, the main technique for detecting the cardiac arrhythmias is the electrocardiogram ECG signal. The ECG is a signal produced by an electrocardiograph, which records the electrical activity of the heart over time. Through its waves duration and axes values they recognize the abnormal heart beat and its type which indicates the cardiac arrhythmia (the disease). There are more than forty cardiac arrhythmias each one being characterized by some rules about the measures extracted from the ECG of the patient. The elementary unit of the Electrocardiogram ECG is the beat. As shown in Figure 2, it contains six waves (P, Q, R, S, T and U) which can also be considered as four parts: the P wave, the complex QRS, the T wave and the U wave. From the duration and axes of its waves, the cardiologists can conclude the normal case and the abnormality of the heart. For more information about the ECG read [24].

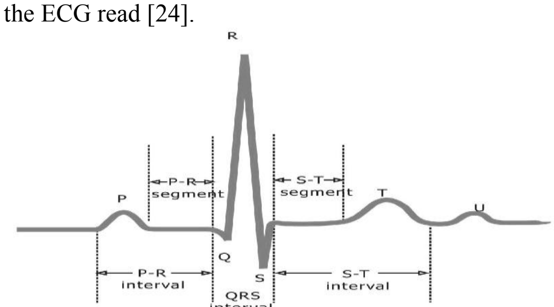


Figure 2: The ECG parameters

4.2 Electrocardiogram characterization

The ECG signals used in this work are recordings collected between 1975 and 1979 by the laboratory of BIH arrhythmia (Beth Israel Hospital) in Boston in the United States, which is known as the MIT-BIH data base [4]. The ECG signals are sampled at a frequency of 360 Hz. Two or more cardiologists have made the diagnosis for these various records and they have annotated each cardiac cycle. The extracted dataset contains some normal beats characterized by 11 features also it contains some cardiac arrhythmias as well as premature ventricular contraction PVC and other arrhythmias. The data is obtained and calculated using an algorithm developed and implemented in the LISI laboratory at the University of Rennes 1. This algorithm is based on the technique introduced by Pan J. and Tompkins W.J [23]. The extracted data set contains the parameters of 745 heartbeats, the training data represent 27% and the test represents 54 %, the training and the test data are independent. There are any missing measures in the data set.

4. The decision support system KI-DCBRC

The proposed decision support system for cardiac arrhythmias, organized as shown in Figure 3, contains a distributed case base where with each arrhythmia is associated one case base containing some cases selected in the learning step. As described in section 1, the case based reasoning systems contain four knowledge containers, in the developed system these containers are distributed in a set of cognitive agents via their local goals. The system contains also a domain knowledge base which contains a set of rules describing the explicit knowledge of the cardiologists. The developed system generates a log file which contains a trace of the reasoning process for enriching the transparency of the decision support system.

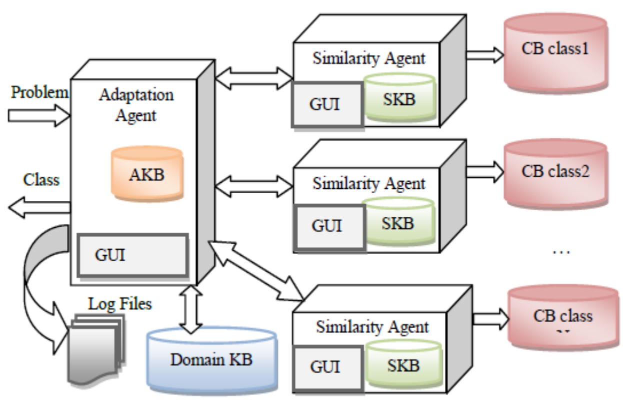


Figure 3: The conceptual model of the classification system KI-DCBRC. SKB, AKB: Similarity, Adaptation Knowledge Base. GUI: Graphic User Interface

The integration of knowledge from multiple sources is an important aspect of automated reasoning systems. When different knowledge bases are used to store knowledge provided by multiple sources, one faced with the problem of integrating multiple knowledge bases: under these circumstances, we are also confronted with the prospect of inconsistency. To avoid these problems, the adaptation knowledge base which contains some rules for ensuring an accurate decision by selecting the more accurate decision generated from the cognitive agents and by weighing their responses in case of conflict, is proposed.

4.1 Fuzzy similarity measures

After some experiments and comparative study, presented in [1], [25] and [26], between some similarity functions and strategies, a novel fuzzy similarity measures model, is introduced. The proposed model is developed for increasing the accuracy of the system. It combines the local-global similarity functions and the fuzzy sets theory. It also generates not just the traditional response (the class) but it generates the Unknown response if the similarity agents generates a high degree of membership in the unknown set it generates also non similar response when the similarity agents generate a high degree of membership in a non-similar set. Three fuzzy sets similar S, not similar N and unknown U with the triangular membership functions μ_S, μ_N and μ_U, have been defined.

$$\mu_s(x) = \begin{cases} 0 \; if \; x \leq a \\ \frac{x-a}{1-a} \; if \; x > a \end{cases} \quad (1)$$

$$\mu_n(x) = \begin{cases} 0 \; if \; x \geq b \\ \frac{b-x}{b} \; if \; x < b \end{cases} \quad (2)$$

$$\mu_u(x) = \begin{cases} 0 \; if \; x \leq b \; ou \; x > a \\ \frac{x-b}{0.5-b} \; if \; x > b \; and \; x \leq 0.5 \\ \frac{a-x}{a-0.5} \; if \; x < a \; and \; x > 0.5 \end{cases} \quad (3)$$

The triangular function for representing the fuzzy sets, where the variable x represents the global similarity measures between the query and the case, is used. The support of the fuzzy sets is defined intuitively by using the agents GUI or by using a machine learning algorithm. The similarity agent computes the global similarity between the query and the cases by using the selected function (sigmoid, exponential, linear or the threshold).

For example the sigmoid similarity function is defined as:

Let Q be the query and C the case, qi ci : the attribute number i respectively of the query and the case, D denotes the space of case characterization models.

Sim: DxD →[0,1]

$$sim(Q,C) = \sum_{i=1}^{N} w_i (1/(1 + e^{(\delta(q_i,c_i) - \theta)/\alpha})) \quad (4)$$

Where N is the number of attributes, w_i the weight of the attribute Ai, the parameters α and θ characterize the detour point of the function and δ(qi,ci) represents the distance function defined as:

σ:DxD→IR

$$\sigma(qi, ci) = \begin{cases} -\ln(c) - \ln(q) & \text{for } q, c > 0 \\ -\ln(-c) - \ln(q) & \text{for } q, c < 0 \\ \text{Undefined} & \text{else} \end{cases} \quad (5)$$

4.2 Features weights definition

To define the features weights which indicate the importance degree of each feature, a machine learning algorithm is used for each similarity agent: The gradient descent. With the following performance function:

Let q, c1 and c2 from the same class.

$$P: D_D \times D_D \times D_D \rightarrow [0, 1]$$

$$P(q,c1,c2) \rightarrow P(q,c1,c2)=|1-|\ sim(q,c1)-sim(q,c2)\ |\quad (6)$$

Figure 5 describes the importance degree of each feature after executing the machine learning algorithm computed by each similarity agent with a learning base which contains 200 annotated cardiac beat 100 from each class (normal and PVC), each similarity agent have used the learning instances (c1,c2) from the same associated class.

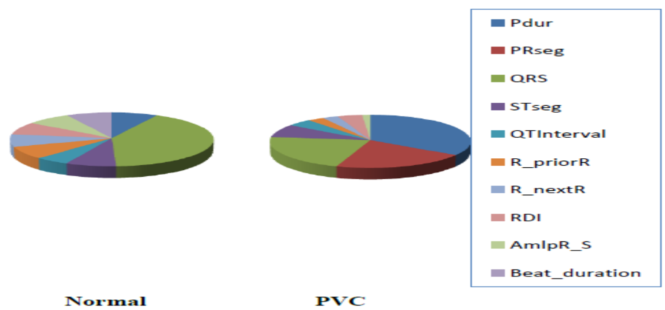


Figure 4: The degree of importance of the attributes defined by using the gradient descent algorithm.

The obtained results presented in Figure 4 represent a minor similarity with the knowledge extracted from the experts knowledge (the applied rules in medicine) for example in the PVC class they consider just the P_duration attribute which will be null and there is no consideration of importance degree in the normal class where they put just a general rules for the normal beat recognition. Also these results explain that the QRS_duration attribute is more significant in the normal class and explain that the rest of the features are less important but useful in the class recognize. And for the PVC class there are three significant features (Pduration, P_Rsegment and the QRS duration).

4.3 Case bases learning

We have applied the Case Base Learning algorithm CBL1 and CBL2 defined by D. Aha [27] for the cases bases learning and optimization in strategic games applications. The first algorithm consists of retaining all classified cases in the case base, and the second consists of retaining just the less similar cases for optimizing the case base. After some experiments by using all parts of the training base, we have obtained a case base with just 14.13% of the training base which mean increasing the performance by 85.87%, with the same rate of accuracy.

4.4 Results

With the obtained case bases and features weights we have applied the decision support system IKCBRC for a test data set which contains 400 heart beat 100 normal 100 PVC and 200 from other classes (Left Bundle Branch Block and Right Bundle Branch Block), we have varied the local similarity function by using the exponential in some experiments and the sigmoid in other ones with the same data. Also for the comparison we have done some experiments with just the domain knowledge base. The figure 6 represents the correctly recognized beats for each combination and the rate of correct classification for each strategy.

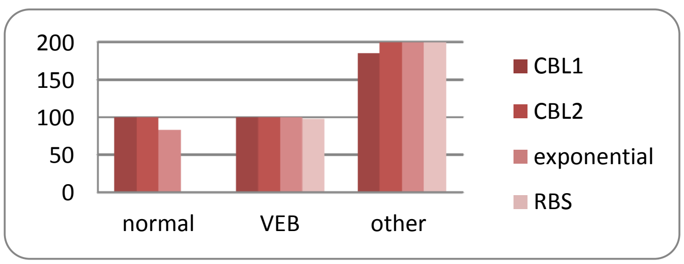


Figure 5: Recognized queries by using the sigmoid function with CBL1 and CBL2 and the exponential function with CBL2 and the rule based system.

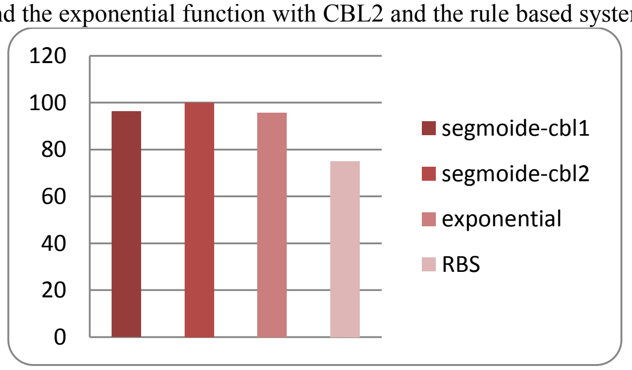


Figure 6: Rate of correct classifications for deferent experiments.

5. Discussions

In these experiments deferent strategies are tested. In the first experiment the CBL1 is applied with the fusion between Rule Based Systems and Fuzzy sigmoid similarity function for 96.25% as a rate of correct classification. In the second experiment 100% is obtained with the same strategy of the first one but by using the case base generated by the CBL2 algorithm. These results prove that the CBL2 algorithm optimizes not only the performance but also the accuracy of the system.

Also, generally, the rate of correct classification by using the sigmoid is better than that obtained with the exponential. This means that the sigmoid function is better than the exponential function in terms of accuracy but not from the point of view of complexity. And finally when we have used just the Rule Based System which represents the domain knowledge modeled, from the expert's knowledge, we have obtained just 75%.

Other works combine the fuzzy approach and the Case based reasoning cited in [36] we cite also here [30] in which the traditional case base paradigm by the Fuzzy Logic concepts in a flexible and extensible component-based architecture is incorporated. Also, [29] enforce the case based reasoning by a fuzzy logic system. We cite also [28] in which they introduce a fuzzy model for the representation of a CBR system. We can cite also [34] an interesting medical application for individual stress diagnosis with fuzzy similarity matching. These contributions integrate the fuzzy systems in the CBR systems for responding to the uncertainty problem each one represents an original contribution in the context of its contribution, the originality in our approach reside in the combination of the fuzzy sets with the global-local similarity measures for generating three similar responses, not similar and unknown. The obtained results prove that the proposed combination ensure the needed precision in the medical applications.

Also, others, in the cardiac arrhythmias diagnosis, different approaches and intelligent techniques for the classification and automatic recognizing, are applied. Among these can mentioned [31] and [33] which apply the Artificial Neuronal Network (ANN) and [32] and [33] have used the Fuzzy approach. On the other hand, researchers have used Support Vector Machines as in [31] and other intelligent approaches.

To compare two applications, not only the rate of correct classification, which is 100% achieved in this work as in [33], is needed but also other criteria such as the transparency and explanation for more interpretability and maintainability are to be used. These are incorporated in the presented system but are weak in the above cited works or do not exist, as it is the case in the ANN application.

The cognitive amalgam with the specialized cognitive agents reduces the conflict which is not considered in the other works. The modularity of the system ensures the scalability for integrating more classes at any time of exploration. The unknown response is very useful for the detection of abnormal beats which proves the efficiency of our contribution.

6. Conclusion

In this work we have presented a crucial medical application in which we have developed a novel decision support system which applies cognitive amalgam with case based reasoning, rule based reasoning distributed reasoning and fuzzy sets to meet the critical criteria in the medical domain. The empirical experiments trough a significant data and many alternatives of computing demonstrate that our investigation produces a significant progress in terms of accuracy, transparency and performance. These progresses appear as a consequence of the combination of some strong and smart approaches in which our system inherits the majority of their advantages and some minor disadvantages as the complexity of knowledge modeling in the rule based systems.

7. Acknowledgment

This work is realized for the purpose of obtaining a doctoral degree. We would like to thank so much the persons who have shared with us some useful information or provided us with invaluable guidance. We would like also to give special thanks to Dr Tomas Roth-Bergopher, Pr D. Aha and Pr Klaus-Dieter Althoff for their advices in ICCBR‟10 Doctoral consortium.

References

[1] A.KHELASSI, M A CHIKH "Automatic Recognition of Cardiac arrhythmias using a Distributed Case-Based Reasoning Classifier" Collogue international sur l‟Optimisation et les Systèmes d‟Information, Algeria, 2008.

[2] Plaza Enric; McGinty Lorraine Distributed case-based reasoning. The Knowledge engineering review, Vol. 20, N. 3, pp. 261-265, 2006.

[3] F.Hiippner,al Obtaining Interpretable Fuzzy Models from Fuzzy Clustering and Fuzzy Regression, fourth international conference on knowledge based intelligent engineering & Allied technologies, UK 2000

[4] MIT-BIH arrhythmia database [http://www.physionet.org/physiobank/database/mitdb/]

[5] J. Kolodner "Case-Based reasoning" Morgan Kaufmann Publisher 1993.

[6] Aamodt, E. Plaza Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. IOS Press, Vol. 7: 1, pp. 39-59, 1994.

[7] Richter, M. M. The Knowledge Contained in Similarity Measures. Invited Talk at International Conference on Case-Based Reasoning, Berlin: Springer ICCBR‟95. 1995.

[8] M V Nagendra Prasad, "Distributed Case-Based Learning," icmas, pp.0222, Fourth International Conference on Multi-Agent Systems (ICMAS'00), 2000

[9] Plaza, E & Onta˜n´on, S "Ensemble case-based reasoning: Collaboration policies for multiagent cooperative CBR" in Aha, DW & Watson, I (eds.), Proceedings of the Fourth International Conference on Case-Based Reasoning, Berlin: Springer, pp. 437–451. 2001

[10] Leake, D & Sooriamurthi, R "Automatically selecting strategies for multi case base reasoning" in Craw, S & Preece, A (eds.), Proceedings of the Fifth European Conference on Case-Based Reasoning, Berlin: Springer, pp. 204–219, 2002..

[11] Watson, I& Gardingen, D "A distributed case-based reasoning application for engineering sales support" in Dean, T (ed.), Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, volume 1, San Francisco, California: Morgan Kaufmann, pp. 600–605, 1999.

[12] Hammond, K, Burke, R & Young, B "The FindMe approach to assisted browsing," IEEE Expert 12(4):32–40, 1997.

[13] Zadeh, L. Fuzzy Sets, Information and Control, vol. 8, pp. 338-353, 1965.
[14] Er, M.J., Zhou, Y.: Automatic generation of fuzzy inference systems via unsupervised learning. Neural Networks 21(10), 1556–1566 ,2008.
[15] Nakashima, T., Schaefer, G., Yokota, Y., Ishibuchi, H.: A weighted fuzzy classifier and its application to image processing tasks. Fuzzy Sets and Systems 158, 284–294, 2007.
[16] Mohamadi, H., Habibi, J., Abadeh, M.S., Saadi, H.: Data mining with a simulated annealing based fuzzy classification system. Pattern Recognition 41(5), 1824–1833, 2008.
[17] F. Herrera, M. Lozano and J. L. Verdegay, A learning process for fuzzy control rules using genetic algorithms, Fuzzy Sets and Systems Volume 100, Issues 1-3, 16 November 1998, Pages 143-158
[18] Harris, C.J, Hong, X., Gan,Q.:Adaptive Modelling, Estimation and Fusion from Data – A Neurofuzzy Approach, Springer (2002)
[19] Casillas, J., Cordon, O. , Herrera,F., L. Magdalena.: Interpretability Improvements to Find the Balance Between Interpretability-accuracy in Fuzzy Modelling: An Overview, in J. Casillas, O. Cordon, F. Herrera, and L. Magdalena (Eds.), Interpretability Issues in Fuzzy Modeling, Vol. 128 of Studies in Fuzziness and Soft Computing, Springer, 3-24(2003)
[20] Zhou, S.-M., Gan, J.Q.: Interpretability Improvement of Input Space Partitioning by Merging Fuzzy Sets Based on An Entropy Measure, Proc. of IEEE Int. Conference on Fuzzy Systems (FUZZ-IEEE2004), Budapest, Hungary287-292, July (2004)
[21] Zhou, S.-M., Gan.,J.Q.: Improving the Interpretability of Takagi-Sugeno Fuzzy Model by Using Linguistic Modifiers and A Multiple Objective Learning Scheme, Proc. of the Int.Joint Conf. on Neural Networks (IJCNN2004), Budapest, 2385-2390 (2004)
[22] Gan, Q., Harris, C.J.: Fuzzy Local Linearization and Local Basis Function Expansion in Nonlinear System Modelling, IEEE Trans. on Systems, Man and Cybernetics - Part B, Vol. 29, No. 4, 559-565 (1999)
[23] Pan J. et Tompkins W.J. A real time QRS detection algorithme IEEE Trans. Biomed. Eng. Vol. 23(4) pp. 230-236, (1985).
[24] J. Bensaid, 2004, Électrocardiogramme normal de l‟adulte, EMC - Cardiologie-Angéiologie, Vol. 1, N. 1, pp. 2-10.
[25] A.KHELASSI, M A CHIKH «Distributed classification using Intensive Knowledge Case Based Reasoning" Conference International des Technologies d‟Information et de Communication 2009 Setif Algérie.
[26] A.KHELASSI "Fuzzy Similarity Based classification VS Intensive Knowledge Case Based classification" Collogue Intrnational sur les Nouvelles Techniques Immuno-Cognitives dans les Réseaux Informatiques NTICRI09 Oran Algerie 2009.
[27] Aha, D. Case-Based Learning Algorithms. In Proceedings of the DARPA Case-Based Reasoning Workshop, pages 147–158. Morgan Kaufmann(1991).
[28] Voskoglou, M.G. Fuzzy Sets in Case-Based Reasoning; Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09. Sixth International Conference on Issue 14-16 Aug. 2009 Volume: 6 page(s): 252 - 256
[29] Meijun Yang1, Qiang Shen "Reinforcing fuzzy rule-based diagnosis of turbomachines with case-based reasoning" International Journal of Knowledge-Based and Intelligent Engineering Systems Volume 12, Number 2 / 2008 page 173-181
[30] Kareem S. Aggour, and al „‟SOFT-CBR: a self-optimizing fuzzy tool for case-based reasoning‟‟ 5th ICCBR Pages: 5-19 2003
[31] M. Moavenian, Hamid Khorrami „‟ A qualitative comparison of Artificial Neural Networks and Support Vector Machines in ECG arrhythmias classification‟‟ Expert Systems with Applications Volume 37, Issue 4, April 2010, Pages 3088-3093
[32] ChikhM.A, O.Behaddada. « A PVC beats recognition using fuzzy classifier, Journal of Mechanics in Medicine and Biology JMMB Vol. 10 N° 2 (2009) pp 327-339.
[33] Yüksel Özbay and al "Integration of type-2 fuzzy clustering and wavelet transform in a neural network based ECG classifier" Expert Systems with Applications Volume 38, Issue 1, January 2011, Pages 1004-1010
[34]I. Bichindaritz, C Marling, Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine 36 (2), 127-135.
[35]Begum, S., Ahmed, M. U., Funk, P., Xiong, N. and Von Schele, B. (2009), A CASE-BASED DECISION SUPPORT SYSTEM FOR INDIVIDUAL STRESS DIAGNOSIS USING FUZZY SIMILARITY MATCHING. Computational Intelligence, 25: 180195.
[36] R Weber "Fuzzy set theory and uncertainty in case-based reasoning" Eng Int Syst (2006) 3: 121–136 2006 CRL Publishing Ltd

System Overview: The 4 Diabetes Support System

Cindy Marling[1], Razvan Bunescu[1], Jay Shubrook[2] and Frank Schwartz[2]

[1] School of Electrical Engineering and Computer Science
Russ College of Engineering and Technology
Ohio University, Athens, Ohio 45701, USA
marling@ohio.edu, bunescu@ohio.edu

[2] The Diabetes Institute
Heritage College of Osteopathic Medicine
Ohio University, Athens, Ohio 45701, USA
shubrook@ohio.edu, schwartf@ohio.edu

Abstract. This paper presents an overview of the 4 Diabetes Support System (4DSS). The 4DSS is a hybrid case-based reasoning system that aims to help patients with type 1 diabetes on insulin pump therapy achieve and maintain good blood glucose control. These patients and their professional health care providers have access to a plethora of blood glucose data. However, there is a paucity of automated data analysis to interpret this data and make it actionable. This paper describes how the 4DSS seeks to fill this gap by providing intelligent decision support for diabetes management.

1 Introduction

There are an estimated 346 million people worldwide who have diabetes [13]. Approximately 20 million of them have type 1 diabetes (T1D), the most severe form, in which the pancreas fails to produce insulin. Because insulin is an essential hormone needed to convert food into energy, T1D patients depend upon external supplies of insulin. T1D patients at the Diabetes Institute at Ohio University are treated with insulin pump therapy. An insulin pump continuously infuses the patient with basal insulin. The patient may instruct the pump to deliver additional insulin boluses to account for meals or blood glucose excursions.

While diabetes can not yet be cured, it is actively managed through blood glucose (BG) control. Good BG control is known to help delay or prevent long-term diabetic complications, including blindness, amputations, kidney failure, strokes, and heart attacks [3]. Effective BG control entails vigilant self-monitoring of BG levels. T1D patients prick their fingers from 4 to 6 times a day and use glucometers to measure their blood. They may also wear continuous glucose monitoring (CGM) devices, which produce BG readings every 5 minutes.

BG monitoring data is relayed to physicians, who must manually interpret it to find BG control problems and recommend appropriate therapeutic adjustments. While voluminous BG data contributes to data overload for physicians, data concerning life events that impact BG levels is not routinely maintained. Physicians may feel, paradoxically, that they have too much data and yet not enough data at the same time.

The 4 Diabetes Support System (4DSS) is a hybrid case-based reasoning (CBR) system that aims to assist T1D patients and their physicians by detecting problems in BG control and suggesting personalized therapeutic adjustments to correct them. It represents work in the CBR in the Health Sciences tradition [1]. While the 4DSS is still a research platform, commercialization efforts are currently underway [6]. The purpose of this paper is to provide a succinct system overview. Additional 4DSS references are available [4, 5, 7, 8, 10–12].

2 Overview

Figure 1 shows a graphical overview of the 4DSS. The system is data driven. Blood glucose data comes from glucometers and CGM devices. Insulin data comes from the patient's pump. The patient uploads BG and insulin data to Medtronic's proprietary CareLink system [9], where it is extracted and transferred to the 4DSS database. The patient manually enters data about life events that impact BG levels, including food, exercise, sleep, work, stress and illness. Originally entered via computer-based browsers, life-event data is now entered via smart phones.

The situation assessment module scans patient data. Traditionally in CBR systems, situation assessment begins with a static description of the current problem. It is different in this domain, because BG control problems continue over time, and because patients are not necessarily aware of problems when they occur. The 4DSS situation assessment module has three major components: problem detection, glycemic variability classification, and BG prediction. These components were built using rule-based reasoning, machine learning algorithms, and time series prediction techniques, giving the 4DSS its hybrid character.

The problem detection component contains 18 rule-based routines that incorporate physician strategies for finding problems in patient data. At a high level, these routines look for problems involving: (1) hyperglycemia, or high BG, which contributes to long-term diabetic complications; (2) hypoglycemia, or low BG, which may result in severe immediate reactions, including weakness, dizziness, seizure or coma; (3) fluctuations between hyper- and hypoglycemia; and (4) lapses in diabetes self-care.

The glycemic variability classication component assesses problems involving BG fluctuation. It detects the problem of excessive glycemic variability, which is believed to presage long-term complications caused by oxidative stress. When expert rules proved inadequate for detecting this problem, machine learning algorithms, including multi-layer perceptrons and support vector machines, were introduced. These algorithms classify 24-hour BG plots by variability level to match physician gestalt perception of such plots. This 4DSS component has stand-alone clinical applicability beyond its role in 4DSS situation assessment.

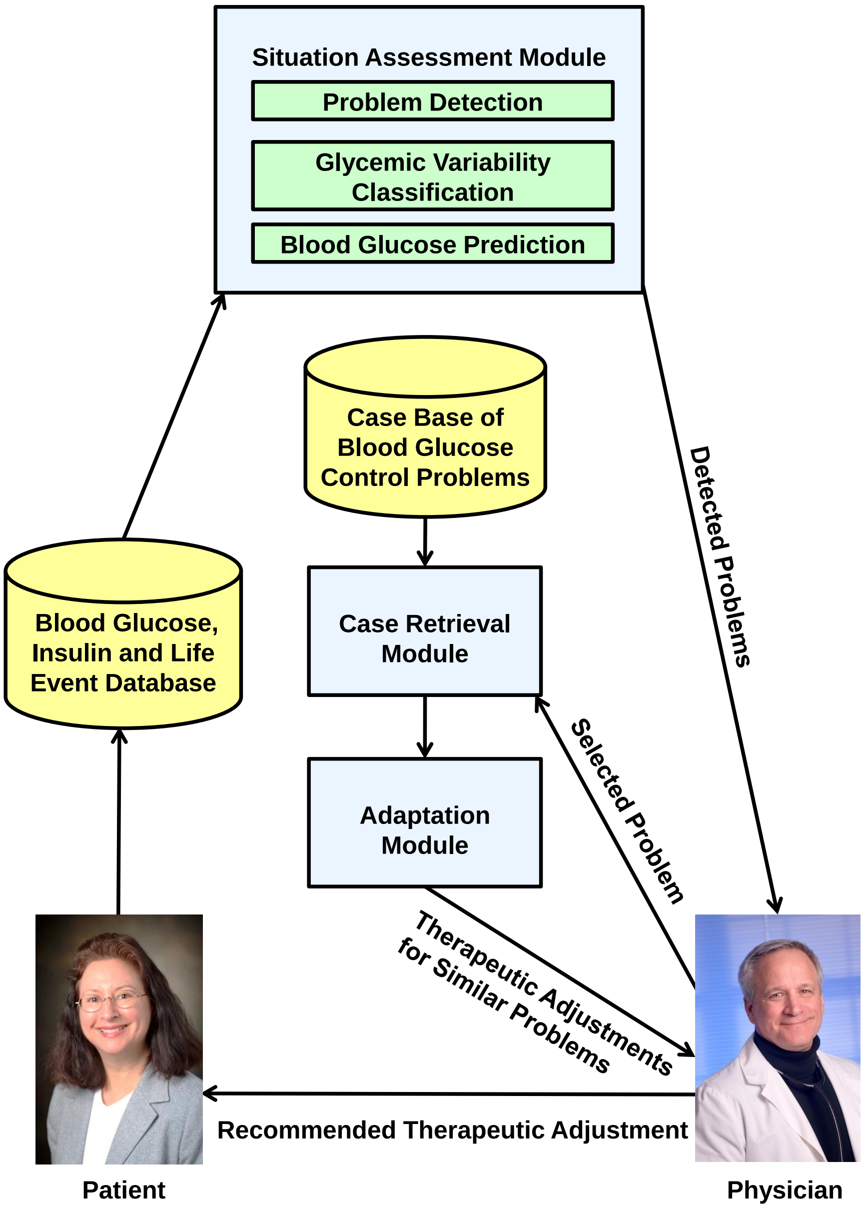


Fig. 1. Overview of the 4 Diabetes Support System

The BG prediction component, currently under construction, incorporates time series prediction techniques to anticipate problems before they occur. While BG data is not currently available in real time and must be scanned retrospectively, we are preparing for its near-term availability. Predicting problems even 30 minutes in advance would give patients time to take preventative actions. This 4DSS component also has a role beyond the 4DSS, enhancing patient safety through eventual incorporation in medical devices.

The situation assessment module reports the problems it finds to a physician, who must then select problems of interest. A selected problem triggers the case retrieval module of the 4DSS. The case retrieval module obtains the most similar cases from the 4DSS case base.

The case base includes 80 cases, each of which contains a specific BG control problem experienced by a T1D patient, a physician's recommended therapeutic adjustment for that problem, and the clinical outcome for the patient after making the therapeutic adjustment. These cases were compiled during clinical research studies in which: (1) T1D patients contributed BG, insulin and life event data; (2) physicians manually identified BG control problems and recommended solutions to patients; (3) patients made the recommended therapeutic adjustments (or not); and (4) physicians examined subsequent patient data to determine the efficacy of the solutions.

To retrieve the most similar cases from the case base, the case retrieval module employs a traditional two-step process. First, a subset of potentially similar cases is identified, and then the nearest neighbors are selected from that subset. In the first step, cases are partitioned by problem type. In the second step, a standard nearest neighbor metric is used. Domain specific similarity functions are combined with empirically determined weights to obtain an overall score for each case. Cases scoring above a similarity threshhold are forwarded to the adaptation module.

The adaptation module personalizes a solution from a retrieved case to fit the situation of a current patient. It begins with the most similar case, but if the solution in that case is not adaptable, it considers the next most similar case, and so on. A solution is a therapeutic adjustment composed of one or more actions that a patient can take. During adaptation, individual actions may be deleted or modified. For example, one possible action is to have a bedtime snack. If the current patient is already having an adequate bedtime snack, this action could be removed from the recommendation. In other situations, the advice could be modified so that the patient eats more or less food before bed, eats a different type of food before bed, or has a snack at a different time of day.

The adapted therapeutic adjustment is relayed to the physician as decision support. The physician decides whether or not to relay the recommendation to the patient. It has long been a goal to provide low-risk advice directly to patients, in real-time, as well as to their physicians. However, this must remain a future goal until the safety and efficacy of the system is proven through clinical trials and approval is obtained from governmental regulatory agencies.

3 Evaluation

Each component of the 4DSS has been evaluated. These evaluations have provided proof of concept, illuminated system strengths and weaknesses, and guided system development. Note that a definitive clinical trial, assessing system impact on patient outcomes, remains to be conducted.

The problem detection component was evaluated after the first and second 4DSS clinical studies. In the first evaluation, a panel of diabetes practitioners rated a sampling of problem detections, and in the second, each patient's physician rated all of the problem detections for the patient. In the first test, 77.5% of problem detections were rated as correct [4], while in the second, 97.9% were rated as correct [11].

The glycemic variability classification component was also evaluated twice. Here, ten-fold cross validation was used to determine the accuracy, sensitivity and specificity of each potential classifier, where correctness is defined as matching physician classifications. In an early test, a naive Bayes classifier matched physicians 85% of the time [8]. The current best classifier, a multi-layer perceptron, has accuracy, sensitivity and specificity of 93.8%, 86.6%, and 96.6%, respectively [12].

The BG prediction component, which is still under construction, is evaluated by examining the difference between predicted BG values and actual BG values. We compute the root mean square error (RMSE), as well as domain specific metrics, for different BG models. Here, evaluation is intertwined with model construction, as the RMSE and other metrics drive model refinement.

The case retrieval module was evaluated by a panel of diabetes practitioners after the first and second clinical studies. Leave-one-out testing was used to provide a sampling of case retrievals for evaluation. In the first test, evaluators rated the retrieved cases as similar to test cases 80% of the time and rated the retrieved solutions as beneficial for test patients 70% of the time [4]. In the second test, they rated retrieved cases as similar 79% of the time and retrieved solutions as beneficial 82% of the time [8].

The adaptation module was evaluated by showing physicians sample problems, with both original and adapted solutions, and eliciting feedback on a questionnaire. Physicians rated the original solutions as being fine without adjustment 47% of the time, needing minor adjustment 40% of the time, and needing major adjustment 13% of the time. They judged the adapted solutions to be better than the original solutions 83% of the time [2].

4 Acknowledgments

This work is supported, in part, with funding from the National Science Foundation under the Smart Health and Wellbeing program (award IIS-1117489). We gratefully acknowledge support from Medtronic, the Ohio University Russ College Biomedical Engineering Fund, the Ohio University Heritage College of Osteopathic Medicine Research and Scholarly Affairs Committee, and the Ohio

University Diabetes Research Initiative. We would also like to thank the participating patients, research nurses, and graduate research assistants for their contributions to this work.

References

1. Bichindaritz, I.: Case-based reasoning in the health sciences: Why it matters for the health sciences and for CBR. In: Althoff, K.D., Bergmann, R., Minor, M., Hanft, A. (eds.) Advances in Case-Based Reasoning: 9th European Conference, ECCBR. pp. 1–17. Springer, Berlin (2008)
2. Cooper, T.: Case Adaptation for an Intelligent Decision Support System for Diabetes Management. Master's thesis, Ohio University, Athens, Ohio (2010)
3. Diabetes Control and Complications Trial Research Group: The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. New England Journal of Medicine 329(14), 977–986 (1993)
4. Marling, C., Shubrook, J., Schwartz, F.: Case-based decision support for patients with type 1 diabetes on insulin pump therapy. In: Althoff, K.D., Bergmann, R., Minor, M., Hanft, A. (eds.) Advances in Case-Based Reasoning: 9th European Conference, ECCBR. pp. 325–339. Springer, Berlin (2008)
5. Marling, C., Shubrook, J., Schwartz, F.: Toward case-based reasoning for diabetes management: A preliminary clinical study and decision support system prototype. Computational Intelligence 25(3), 165–179 (2009)
6. Marling, C., Wiley, M., Bunescu, R., Shubrook, J., Schwartz, F.: Emerging applications for intelligent diabetes management. AI Magazine 33(2), 67–78 (2012)
7. Marling, C., Wiley, M., Cooper, T., Bunescu, R., Shubrook, J., Schwartz, F.: The 4 Diabetes Support System: A case study in CBR research and development. In: Ram, A., Wiratunga, N. (eds.) Case-Based Reasoning Research and Development: 19th International Conference on Case-Based Reasoning, ICCBR 2011 Proceedings. pp. 137–150. Springer, Berlin (2011)
8. Marling, C.R., Shubrook, J.H., Vernier, S.J., Wiley, M.T., Schwartz, F.L.: Characterizing blood glucose variability using new metrics with continuous glucose monitoring data. Journal of Diabetes Science and Technology 5(4), 871–878 (2011)
9. Medtronic: Carelink personal software (2012), http://www.medtronicdiabetes.net/products/carelinkpersonalsoftware, http://www.medtronicdiabetes.net/products/carelinkpersonalsoftware, accessed May, 2012
10. Schwartz, F.L., Shubrook, J.H., Marling, C.R.: Use of case-based reasoning to enhance intensive management of patients on insulin pump therapy. Journal of Diabetes Science and Technology 2(4), 603–611 (2008)
11. Schwartz, F.L., Vernier, S.J., Shubrook, J.H., Marling, C.R.: Evaluating the automated blood glucose pattern detection and case-retrieval modules of the 4 Diabetes Support System. Journal of Diabetes Science and Technology 4(6), 1563–1569 (2010)
12. Wiley, M., Bunescu, R., Marling, C., Shubrook, J., Schwartz, F.: Automatic detection of excessive glycemic variability for diabetes management. In: Proceedings of the Tenth International Conference on Machine Learning and Applications (ICMLA-11). pp. 148–154. IEEE Press, New York, NY (2011)
13. World Health Organization: Diabetes (2012), http://www.who.int/mediacentre/factsheets/fs312/en/index.html, accessed May, 2012

Supporting Frontotemporal Dementia Protocols by CBR Graph Models *

Jose M. Juarez[1], Beatriz Garcia-Gonzalez[1], Jose Palma[1], Maria M. Antequera[2], Begoña Martinez[2], and M. Carmen Antunez[2]

[1]Computer Science Faculty – Universidad de Murcia – Spain
{jmjuarez|b.garciagonzalez|jtpalma}@um.es
[2] Dementia Unit – Hospital Universitario Virgen de la Arrixaca – Spain

Abstract Neuropsychological protocols and neurological tests are essential in Alzheimer's and Frontotemporal dementia diagnosis. However, Dementia Units need to coordinate a large number of protocols to evaluate each patient. Therefore, the patient history consists of a large set of interrelated clinical acts. The relational aspects of these acts can be represented by means of computer models, such as computerised clinical guidelines. However, even if activity flows are correctly modelled, their automatic identification and combination is an open problem. Some effort has been made in Case-Based Reasoning (CBR) when cases are complex structures, and similarity measures and adaptation methods, usually tailored for specific languages (clinical workflows) have been proposed. However, most activity flow models have an underlying graph structure. Indeed, some authors represent cases by means of labelled direct graphs. Nevertheless, labelled graphs find it difficult to represent most of the information encapsulated in clinical acts or routing nodes of activity flows. Therefore, new general CBR models based on graphs are required to deal with activity flow independence and data-intensive activities. In this work, we propose a general CBR model based on attributed-labelled graphs to represent general structured cases. We also present GRACE, a tool based on the model, used to support the design of Frontotemporal dementia protocols, obtaining over 70% of accuracy in our experiments.

1 Introduction

Dementia is an umbrella term covering a variety of diseases and conditions. The administration of a comprehensive neuropsychological protocol including several tests increases the certainty about accurate clinical diagnosis [13]. Despite Alzheimer's Disease (AD) is the prevalent dementia, many diseases causing dementia (such as Frontotemporal dementia) show overlapped symptoms, and patients could suffer from more than one dementing pathology. Dementia Units (DUs) are multidisciplinary health services where neurologists, neuropsychologists and clinical psychologists evaluate and treat the patient simultaneously. The patient history is, therefore, a set of interrelated clinical acts, combining protocols and tests from different clinical disciplines. Neuropsychological protocols and neurological tests are time consuming, producing negative effects on the patients and their families. A suitable selection of protocols and an

* This work is funded by TIN2009-14372-C03-01 (MEC) and 15277/PI/10 (Seneca) projects

early diagnosis means an important reduction of time and human resources. However, this is not a simple task since even a 1-day stay patient could imply more than 30 different neuropsychological protocols (e.g. Minimental and Germcide) apart from other neurological tests and scores (e.g. CT/EEG tests or Blessed/IDDD scores).

DU teams usually make decisions based on previous experiences, resource limitations and medical criterion. Some effort has been made in Case-based reasoning (CBR) for DU decision support. In Auguste Project, the authors present a prescription support prototype for AD patients [9]. According to [6], three main internal case structures can be considered when the vector representation is insufficient: hierarchical, networks or flow structures. However, graphs underlie most of the structured case models, in particular digraphs, labelled graphs and relational graphs. Graphs comparison is a challenging problem. From the theoretical point of view, the graph matching problem includes exact matching (graph isomorphism and sub-graph isomorphism), maximum common subgraph, and inexact graph matching [8].

On the one hand, specialised CBR techniques have demonstrated their usefulness in reusing activity flow structures. A retrieval approach based on graph edit distance was proposed when cases can be regarded as workflow schemata (a kind of labelled graph) [10]. In [5], a clinical workflow retrieval technique is proposed, retrieving workflow executions (interval sequences) focusing on the temporal dimension. However, all the abovementioned methods share a common dependence on the activity flow model (e.g. workflow languages, etc.).

On the other hand, general models for representing structured cases can provide domain independent retrieval or reuse methods, such as the generalised CBR models of [1,12]. In this line, general models of graph-based CBR are proposed in [3,7]. In [4] cases are represented by a labelled directed graph (labelled vertex and arcs). From a practical perspective, efforts made in object-oriented CBR have demonstrated its suitability for representing complex pieces of knowledge [2]. However, its use for modelling activity flows is not a simple issue since clinical acts and relations are data-intensive.

Therefore, when CBR systems are required for managing complex structures, the use of an attributed-labelled graph model is a promising aproach, since it is expressive enough to satisfy these requirements, providing a case structure independently of the activity flow model.

In this work, we propose a theoretical attributed-labelled graph model to represent domain knowledge, cases and the CBR cycle. We illustrate the application of this model in the DU domain.

The reminder of this paper is organised as follows. In Section 2 we propose a theoretical CBR model based on attributed-labelled graphs. Section 3 presents the GRACE tool and our experiments in the DU domain. Finally, Section 4 concludes this paper with a discussion and suggests future work.

2 Graph CBR Model

A digraph D consists of a non-empty set of elements (N), called nodes, and a set of ordered pairs of the same ($A \subseteq N \times N$) called arcs. That is: $D = (N, A)$. From a computational perspective, graph structures have been traditionally used for information representation. Therefore, the digraph definition has been frequently extended by adding labels and attributes to their vertex and arcs.

2.1 Domain Model

Definition 1. *A **label set**, L, is the finite non-empty set of all labels that, given a domain, can be assigned to the components of D. In our case, $L = L_c \cup L_t$, with $L_c \cap L_t = \emptyset$, where L_c and L_t are the sets of all labels representing general concepts and terms, respectively. We denote the empty labels of general concepts and terms by $\varepsilon_c \in L_c$ and $\varepsilon_t \in L_t$ respectively.*

Concept labels refer to high level concepts for a given domain ontology, while term labels represent characteristics or attributes of concepts. For example, in the medical domain, $diagnosis$ can be a concept label and $dose$ a term label, while ε_c and ε_t are used when labels are not required or they are unknown.

Definition 2. *A **value set**, V, is the finite set of all possible values that, given a domain, can be assigned to general concepts and/or terms. In our case, $V = V_c \cup V_t$ where V_c stands for the set of values that can be assigned to general concepts and V_t stands for the set of values that can be assigned to terms. We denote the empty values of concepts and terms values by $\tau_c \in V_c$ and $\tau_t \in V_t$.*

The semantics of τ_c and τ_t depend on the designer and the domain (see example of Dementia Unit).

Definition 3. *The **attribute set**, Φ, is the set of all possible label-value pairs. Formally speaking, $\Phi = \Phi_c \cup \Phi_t$, with $\Phi_c = L_c \times V_c$ being the **general concept attribute set**, and $\Phi_t = L_t \times V_t$ being the **general terms attribute set**.*

Example: Dementia Unit domain *In the Dementia Unit domain, we can define the label set L where L_c (a set of concepts related to general elements of the health record) can be L_c={demographics, NRL, NPS, visit, order} and L_t={sex, age, EEG, MRT, day}, where NRL and NPS are neurological and neuropsychological studies respectively. The values are V_c={ psychiatricsuspect, noconclusive, τ_c } and V_t={ male, female, normal, abnormal, 1, $\cdots$, 60, $\cdots$, 100, τ_t }. The attribute set is defined by Φ_c={ (demographics, τ_c), (NRL, psychiatricsuspect), (NPS, noconclusive), (order, τ_c), (visit, τ_c), $\cdots$,(ε_c,τ_c) } and Φ_t={ (sex, male), (sex, female), (EEG, normal), (EEG, abnormal), (MRT, normal), (MRT, abnormal), (age, 1), $\cdots$,(age, 100), (day, 1) $\cdots$,(ε_t,τ_t)}. Here the semantic of the concept-attribute (order, τ_c) means that an order (label) requires no value to be defined, whilst (ε_t,τ_t) is used when you refer to an unknown term-attribute.*

2.2 Attributed-Labelled Graph Model

Definition 4. *In a given domain, a **valid concept attribute set**, $\bigtriangledown$, is a set of valid pairs of a general concepts attribute set, $\bigtriangledown \subseteq \Phi_c$.*

Definition 5. *In a given domain, a* ***valid terms attribute set****, $\triangle$, is a set of valid pairs of a terms attribute set, $\triangle \subseteq \Phi_t$.*

Definition 6. *A* ***concept labelling set****, $\mathcal{L}$, is a set of valid pairs (g, c), with $g \in N \cup A$ and $c \in \bigtriangledown \cup \triangle$. That is,*

$$\mathcal{L} \subseteq (N \times (\bigtriangledown \cup \triangle)) \cup (A \times (\bigtriangledown \cup \triangle)).$$

In other words, a labelling set constrains the general domain model, specifying the valid concepts, terms, attributes and their map with nodes and arcs. For practical issues, in our model, each node or arc usually has assigned to it a single concept-attribute element.

At this point, and taking into account previous definitions, we can formally define an attributed-labelled graph.

Definition 7. ***Attributed-labelled graph (ALG)****. An ALG, G is a 5-tuple consisting of a non-empty* ***set of nodes*** *N,* ***a set of arcs*** *A, $A \subseteq N \times N$, a* ***valid general concept attribute set*** *$\bigtriangledown$, a* ***valid terms attribute set*** *$\triangle$ and a* ***labelling set*** *$\mathcal{L}$. That is,*

$$G = (N, A, \bigtriangledown, \triangle, \mathcal{L}).$$

As can be deduced from definition 7, ALG is a particular type of graph in which each node and arc is associated to a general concept with its respective value (this can be the empty value) and a set of term-value pairs.

In order to simplify mathematical notation in the following sections, the $label$ function is introduced.

Definition 8. ***Label function****. Let be $n \in N$, $a \in A$ a node and an arc of a ALG. The* ***label function*** *of $x \in N \cup A$ is a set of labels so that,*

$$label(x) = \{(l, v) \in \Phi | (x, (l, v)) \in \mathcal{L}\}.$$

Example: Dementia Unit patient *We define an attributed-labelled graph for the DU domain, following the domain model example of Section 2.1. The set of nodes and arcs of the graph are $N = \{n_1, n_2, n_3\}$ and $A = \{a_1, a_2\}$ where $a_1 = (n_1, n_2)$ and $a_2 = (n_2, n_3)$. The valid labels, values and attributes are:*

- *$\bigtriangledown$=ϕ_c.*
- *$\triangle$=$\{$ (sex, male), (age, 40), $\cdots$, (age, 100), (day, 1), (EEG, normal), (MRT, abnormal), (ε_t,τ_t)$\}$.*
- *$\mathcal{L}$=$\{$ (n1,$\{$(demographics,τ_c),(sex, male), (age, 60)$\}$), (n2,$\{$ (NRL, psychiatricsuspect), (EEG, normal), (MRT,abnormal) $\}$), (n3,$\{$ (NPS, noconclusive), (ε_t, τ_t)$\}$), (a1,$\{$ (visit, τ_c), (day, 1)$\}$), (a2,$\{$ (order,τ_c), (ε_t, τ_t)$\}$) $\}$.*

Figure 1 (in bold) shows a graphical representation of $G_T = (N, A, \bigtriangledown, \triangle, \mathcal{L})$.

We denote by $\mathcal{G}$ the set of all possible attributed-labelled graphs, while $\mathcal{N}$ and $\mathcal{A}$ are the space of nodes and arcs, respectively. Without losing generality, hereinafter, we will use the term graph when we refer to an attributed-labelled graph (definition 7).

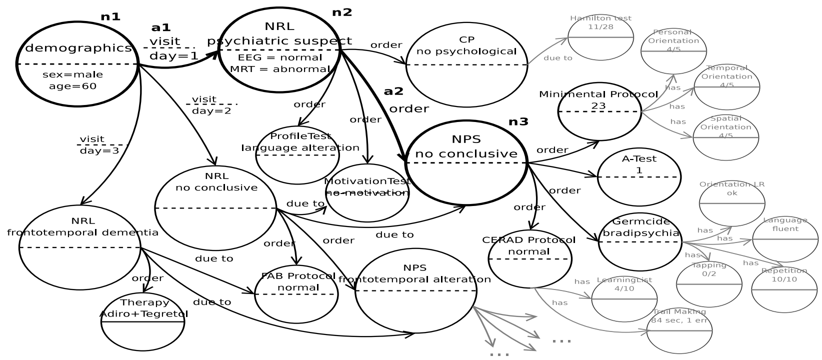


Figure 1. Example: extract of a Frontotemporal dementia patient.

Graph Operations Several operations can be performed on digraphs in order to form new ones.

Let $G_1 = (N_1, A_1, \bigtriangledown_1, \bigtriangleup_1, \mathcal{L}_1)$ and $G_2 = (N_2, A_2, \mathcal{L}_2, \bigtriangledown_2, \bigtriangleup_2, \mathcal{L}_2)$ be attributed-labelled graphs.

Definition 9. *(Union) The union of graphs G_1 and G_2 is*

$$G_1 \cup G_2 = (N_1 \cup N_2, A_1 \cup A_2, \bigtriangledown_1 \cup \bigtriangledown_2, \bigtriangleup_1 \cup \bigtriangleup_2, \mathcal{L}_1 \cup \mathcal{L}_2).$$

Definition 10. *(Intersection) The intersection digraph of G_1 and G_2 is*

$$G_1 \cap G_2 = (N_1 \cap N_2, A_1 \cap A_2, \bigtriangledown_1 \cap \bigtriangledown_2, \bigtriangleup_1 \cap \bigtriangleup_2, \mathcal{L}_1 \cap \mathcal{L}_2).$$

2.3 CBR Model

CBR systems are based on three essential components: the case structure, the case base and the CBR cycle engine.

Definition 11. ***All cases set****, $\mathcal{C}$, is a set of triples (cases), where each triple is composed of a graph, that represents a problem, a graph that describes the solution and an outcome value to qualify the degree of fulfilment of the solution of the problem. Formally:*

$$\mathcal{C} = \mathcal{G} \times \mathcal{G} \times (V \cup \{\lambda\}),$$

where λ denotes that the case is not valid for any reason. A case is denoted by $C = (g_p, g_s, o) \in \mathcal{C}$ where g_p and g_s are called the problem and the solution graph, respectively.

Examples of set V can be V=\{*completelySolved, partiallySolved, barelySolved*\} or $V = \mathbb{R}^+$, being $\mathbb{R}^+$ the set of positive real numbers.

Definition 12. ***All case bases set*** *$\mathcal{CB}$ is the set of all possible case bases. That is:*

$$\mathcal{CB} = \wp(\mathcal{C}),$$

where $\wp$ is the power set. A particular case base is expressed as $CB \in \mathcal{CB}$.

Definition 13. *A* ***case similarity function*** *is a normalised function that quantifies the similarity between two graphs. That is:*

$$S : \mathcal{G} \times \mathcal{G} \rightarrow [0, 1].$$

Given $g_i, g_j \in \mathcal{G}$ problem graphs, $S(g_i, g_j) = 1$ means that they are identical and $S(g_i, g_j) = 0$ if they are completely different. When S is used to state the similarity between two cases, the similarity between their respective problem graphs is calculated. The details of function S depends on the application domain. Section 3.1 shows some illustrative examples.

In the following, functions representing CBR cycle steps are formally defined.

Definition 14. ***Retrieve function****. Given a problem graph, g_{in} and a case base CB, the retrieval function* ***Retrieve*** *provides a subset of CB composed of cases similar to the problem graph g_{in}. Formally:*

$$Retrieve(g_{in}, CB) = \{(g_j, g_k, o) \in CB | S(g_{in}, g_j) \geq criteria_r\},$$

with $criteria_r$ being the minimum similarity value to consider two graphs (or cases) as similar. An evident consequence is that $|CB| \geq |Retrieve(g_{in}, CB)|$.

Definition 15. ***Adapt function****. Given a problem graph g_{in} defining a case c_{in} and a set R of cases belonging to a case base CB, all similar to c_{in} for a given criteria $cirteria_r$ ($R = Retrieve(g_{in}, CB)$), the* ***Adapt*** *function builds a solution graph, g_a, for problem graph g_{in}, from the solution graphs of cases in R. That is:*

$$Adapt(g_{in}, R) = g_a \text{ so that } consistent(g_{in}, g_a),$$

with $consistent$ being a domain-dependent function that is true if g_a fulfils some constraints to consider it a valid solution graph for g_{in} and false otherwise.

Different $Adapt$ functions can be designed, depending on the domain. Since case solutions are graphs, union and intersection operations can be considered (see Section 2.2).

Definition 16. ***Reuse function****. Given a problem graph g_{in}, describing the problem of the case c_{in}, and a set of cases R, retrieved from a case base CB ($R = Retrieve(g_{in}, CB)$), the* ***Reuse function*** *obtains a new case, where its problem graph is g_{in}, its solution graph is obtained adapting the solution graphs of the cases from R and the outcome component depends on a domain criteria. Formally,*

$$Reuse(g_{in}, R) = (g_{in}, Adapt(g_{in}, R), criteria_a(g_{in}, g_a)),$$

with $criteria_a(g_{in}, g_a)$ being a domain dependent function ($criteria_a : \mathcal{G} \times \mathcal{G} \rightarrow V \cup \{\lambda\}$) that determines the grade in which g_a can be considered as a solution of g_{in}.

Definition 17. ***CBR query function***. *Given a problem graph g_{in} and a case base CB, the function CBR query,* ***Q****, returns a new case, solving g_{in} according to the cases in CB. That is:*

$$Q(g_{in}, CB) = Reuse(g_{in}, Retrieve(g_{in}, CB)).$$

Definition 18. ***Retain function***. *Given a case c and a case base CB, function* ***Retain*** *adds (retains) the case c into CB if a specific domain criteria ($criteria_{ret}$) is fulfilled. That is:*

$$Retain(c, CB) = \begin{cases} CB \cup \{c\} & \textit{if } criteria_{ret}(c, CB) \\ CB & \textit{otherwise} \end{cases}$$

where $criteria_{ret}(c, CB)$ is a function ($criteria_{ret} : \mathcal{C} \times \mathcal{CB} \rightarrow \{true, false\}$) that is true if the case is added to CB. If the case outcome is λ (case not valid), the case is not retained.

Note that the model presented is a general CBR model based on graphs; therefore $criteria_a$, $criteria_r$ and $criteria_{ret}$ are strictly domain dependent functions that must be defined for each particular scenario.

3 Applications of ALG model in a Dementia Unit

3.1 Grace Tool

In this work, we present GRACE (GRAph Case Environment), a tool for developing and evaluating graph-based CBR systems. GRACE implements the CBR cycle based on the ALG model described in Section 2. GRACE has the following goals: (1) to provide a fast CBR prototype builder for graph based CBR systems in medical domains (interfaces) and (2) to provide a tool for experimenting and validating similarity and adaptation techniques for graph-based CBR systems.

GRACE architecture is organised in 4 layers: the user interface (providing a visual representation of graphs and charts), the logic level (main modules of the system: editors, import functionality, querying, etc.), the reasoning engines (case-based, rule-based and experiment engines) and finally the persistence level. All the information produced in GRACE (domain terminology, case base, rule base, project meta-data and experiments) is stored in XML files.

Retrieval The Retrieval step in GRACE focuses on similarity between the query case and the case base elements. The set of retrieved cases depends on the criteria selected: the K most similar cases (K-NN) or those which raise a given similarity threshold.

Cases are graph-structured and similarity depends on their topology and on their information encapsulated in labels. In this work we consider 2 case similarity functions: (1) based on the similarity of labels of nodes and arcs (see Appendix for further details) and (2) an adapted version of the editing distance proposed in [11].

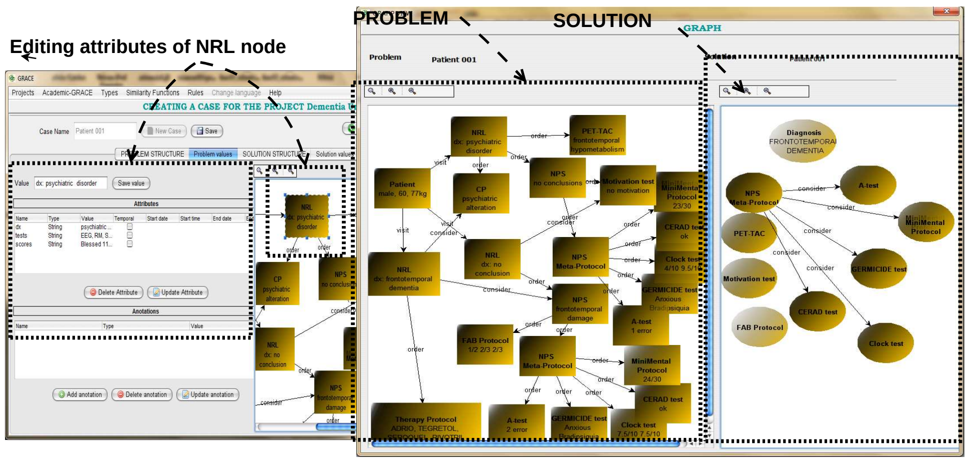


Figure 2. GRACE tool: case visualization of a Frontotemporal dementia patient.

Reuse, Revise & Retain GRACE provides different approaches to adapt the solutions of the retrieved cases; in particular, the null adaptation, the graph-based adaptation and the rule-based adaptation. Because GRACE is based on the ALG model, each solution retrieved is a graph. Therefore, graph operators such as union and intersection for labelled graphs are included. GRACE also includes a simple rule-based system to develop a knowledge-based adaptation step. To this end, GRACE implements a rule editor and a rule engine. Each rule antecedent describes the presence or absence of nodes and arcs of a solution graph, while the consequent part commands arcs and nodes be added or removed from the graph (adapted solution graph).

The revise step implemented in GRACE is semi-automatic. The outcome case (incoming problem graph, adapted solution graph and outcome) is shown to the expert for validation or modification.

Finally, the expert decides whether the solved outcome case is included in the case base ($criteria_{ret}$ function), the retain step of the CBR cycle.

Experimenter The experimenter module supports the evaluation of the case base and the CBR system configuration (similarity functions, weights, adaptation methods, etc.).

This module implements an experimenter engine, allowing the user to select between the following validation techniques: hold out, cross validation and leave one out. Once the experiments have concluded, GRACE provides a report showing a summary of the most relevant statistics, including the confusion matrix, hits, error rate, accuracy, specificity, sensitivity and precision.

3.2 Experiments in Dementia Unit

Early identification of dementia between AD and Frontotemporal dementia (FD) [13] is a recurrent problem of DUs. Our experiments focuses on decision support by retrieving

	(1)S: edit [11] + $Adapt$: null	(2)S: Attribute + Adaption: null	(3)S: Attribute + $Adapt$: union	(4)S: Hybrid + $Adapt$: null
Specificity	**0.7**	0.619	0.4	**0.7**
Sensitivity	0.663	**0.7**	0.333	**0.7**
Accuracy	0.625	0.625	0.375	**0.75**
Precission	**0.737**	0.541	0.255	0.735

Table 1. Classification experiments considering Frontotemporal dementia class.

similar patients' records in order to propose a particular configuration of protocols and clinical actions.

NPS protocols are ordered after a diagnosis suspect; however the evaluation of a protocol design is costly since it must be carefully supervised by clinicians. Therefore, the experiments were organised in 3 phases: (1) to built the Case Base from the patients' health records; (2) to choose the best system configuration using an automatic validation process considering the system as a classifier; and (3) to evaluate the (best configured) system to support the protocol design.

Using GRACE, clinicians modelled 8 cases selected from more than 50 health records, suffering from AD (4 cases) and FD (4 cases). During Case Base building process, some improvements, suggested by the DU team, were considered to adapt GRACE interface. The example provided in Section 2 (see Figure 1) is a simplified description of real case problem in a DU. Figure 2 depicts the complete case (problem and solution). The problem of the case is composed of basic demographic information, the results of neuropsychological protocols (FAB, Motivation tests, Minimental, Clock drawing, Germcide, etc.), therapy protocols, reports of neurologists (NRL), neuropsychologists (NPS) and clinical psychologists (CP), and their relation (temporal and semantic relations). The solution of the case is a graph, where nodes represent the protocols used to support the diagnosis.

An empirical study of system configurations was performed. To this end, we consider the system as a classifier, where the class is the diagnosis node of the case solution (AD or FD). A leave-one-out validation strategy has been adopted. The following system configurations obtained the best results to classify FD: (1) the topological similarity function proposed in [11], and null adaptation (i.e. using the solution of the nearest problem retrieved); (2/3) weighted Euclidean distance and null/union adaptation (see appendix); (4) a hybrid similarity function weighted by node information and null adaptation. Table 1 depicts a summary of the experiments carried out.

Finally, in order to evaluate the capacity of the system to support NPS protocols, members of the DU team query the system with 20 new patients using the best configuration (option 4). The suggested protocols were handmade evaluated, obtaining an overall accuracy of 73%. Most accurate CBR system configuration combines topological and weighted attributes similarity. It is worth mentioning that best results are obtained when high weighted attributes are gender, age and GDS (a general dementia degree) and, regarding the topology, subtrees from NPS nodes were also highly scored.

4 Conclusions

In this work we describe a general graph model for CBR system design and its application in a DU. The proposal comprises the domain, the ALG and the CBR models. We also present GRACE, a tool aimed at supporting graph-based CBR systems, based on the ALG model. Finally, we describe our experince in supporting protocol design in a DU, obtaining over 70% of accuracy. Thanks to its graphical interface, clinicians are able to model patient cases (graph-case edition) from the health records, after relatively few time. Another relevant aspect of GRACE is its experiment engine. In essence, the GRACE tool is designed for experimenting and evaluating CBR techniques based on graph structures. In practice, the experimenter engine has been sucesfully used to evaluate different system configurations (retrieval and adaptation methods) in order to tune the most accurate system for supporting protocol design, obtaining some promising results. Future works will focus on proposing novel topology-based similarity functions using the ALG model in Ambient Assisted Living and dementia problems and their development supported by the GRACE tool.

References

1. A. Aamodt and E. Plaza. Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, 1994.
2. Ralph Bergmann and Armin Stahl. Similarity measures for object-oriented case representations. In *Proc. of EWCBR'98*, pages 25–36. Springer-Verlag, 1998.
3. H. Bunke and B. T. Messmer. Similarity Measures for Structured Representations. In *Proc. of EWCBR'93*, pages 106–118. Springer Verlag, 1994.
4. Pierre-Antoine Champin and Christine Solnon. Measuring the similarity of labeled graphs. In *Proceedings of the 5th international conference on Case-based reasoning: Research and Development*, ICCBR'03, pages 80–95, Berlin, Heidelberg, 2003. Springer-Verlag.
5. Carlo Combi, Matteo Gozzi, Barbara Oliboni, Jose M. Juarez, and Roque Marin. Temporal similarity measures for querying clinical workflows. *Artif. Intell. Med.*, 46:37–54, May 2009.
6. Padraig Cunningham. A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 21:1532–1543, 2009.
7. J. Lieber and A. Napoli. Using Classification in Case-Based Planning. In *Proceedings of ECAI'96*, pages 132–136. John Wiley & Sons, Ltd., 1996.
8. R. Marcondes, E. Bengoetxea, I. Bloch, and P. Larrañaga. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Patt. Recog.*, 38(11):2099–2113, 2005.
9. Cindy Marling and Peter Whitehouse. Case-based reasoning in the care of alzheimer's disease patients. In *Proc. of ICCBR'01*, pages 702–715. Springer-Verlag, 2001.
10. M. Minor, A. Tartakovski, and R. Bergmann. Representation and structure-based similarity assessment for agile workflows. In *Proceedings of ICCBR'07*, LNCS 4626, pages 224 – 238. Springer, 2007.
11. R. Raveaux, J.C. Burie, and J.M. Ogier. A graph matching method and a graph matching distance based on subgraph assignments. *Pattern Recogn. Lett.*, 31:394–406, April 2010.
12. Armin Stahl. Learning similarity measures: A formal view based on a generalized cbr model. In *ICCBR'05*, pages 507–521, 2005.
13. Alexandra J. Walker, Susanne Meares, Perminder S. Sachdev, and Henry Brodaty. The differentiation of mild frontotemporal dementia from alzheimer's disease and healthy aging by neuropsychological tests. *International Psychogeriatrics*, 17(1):57–68, 2005.

Appendix: example of similarity measure using the ALG model

In the following, we describe an example of a *problem graph similarity* used in GRACE, based on the model proposed in Section 2. In particular, the similarity of two problem graphs can be defined, according to definition 13, as follows:

Definition 19. ***Problem graphs similarity****. Given two problem graphs* g_1 *= (*N_1*,* A_1*,* $\bigtriangledown_1$*,* $\bigtriangleup_1$*,* $\mathcal{L}_1$*) and* g_2 *= (*N_2*,* A_2*,* $\bigtriangledown_2$*,* $\bigtriangleup_2$*,* $\mathcal{L}_2$*), the similarity between them is defined by the* S *function:*

$$S(g_1, g_2) = \frac{\omega_N S_N(N_1, N_2) + \omega_A S_A(A_1, A_2)}{\omega_N + \omega_A},$$

where $\omega_N, \omega_A \in (0, 1]$ *are the node and arc weights, respectively, and* S_N *and* S_A *functions that calculate the similarity of node and arc sets of* g_1 *and* g_2*.*

Definition 20. ***Node set similarity****. Given two node sets,* N_1 *and* N_2*, the* ***node set similarity****,* S_N*, is defined as follows:*

$$S_N : \mathcal{N} \times \mathcal{N} \rightarrow [0, 1]$$
$$S_N(N_1, N_2) = \frac{1}{|N_1||N_2|} \sum_{\forall n_i \in N_1, n_j \in N_2} S_n(n_i, n_j).$$

Function S_n is the node similarity measure (used by S_N) and is a weighted Euclidean distance of their general concepts and terms.

Definition 21. ***Arc set similarity****. Given two arc sets* A_1 *and* A_2*, the* ***arc set similarity****,* S_A*, is defined as follows:*

$$S_A : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$$
$$S_A(A_1, A_2) = \begin{cases} 0 & \textit{if } \alpha = \emptyset \\ \dfrac{1}{|\alpha|} \displaystyle\sum_{(a_i, a_j) \in \alpha} S_a(a_i, a_j) & \textit{otherwise} \end{cases}$$

where $\alpha = A_1 \times A_2$*.*

That is, the arc set similarity is zero if one arc set has no edges; otherwise, the function combines arc similarity measures. Function S_a is the arc similarity measure (used by S_A) and is a weighted Euclidean distance of their general concepts and terms.

A System Overview of a Case-Based Reasoning System for Care Planning of End-of-Life Cancer Care

Krista Elvidge[1]

[1] Health Informatics Lab, Dalhousie University, Canada
Kelvidge@dal.ca

Abstract. In end-of-life cancer care, nurse clinicians strive to deliver care to minimize pain, manage debilitating symptoms, and improve patients' quality of life as they approach end of life. Unfortunately, existing literature suggests that suboptimal symptom management remains the reality for many end-of-life cancer patients, resulting in unnecessary suffering and a diminished quality of life. To improve symptom management in end-of-life care, research suggests that nurse clinicians require access to contextually relevant, medical knowledge resources at the point-of-care. We present a clinical decision support system (CDSS) that employs case-based reasoning which enables nurse clinicians to make informed clinical decisions based upon past patient cases in end-of-life cancer care. We evaluate the clinical validity of the system output with physicians in palliative care.

Keywords: case-based reasoning, care planning, clinical decision support system, palliative care

1 Introduction

Clinical decision-making is a complex task that demands healthcare providers to make clinical decisions that are often based upon imperfect, subjective patient medical information. The decision-making process can be improved when healthcare providers utilize explicit and tacit knowledge leveraged from existing medical knowledge resources. Collectively, explicit, or evidence-based standards of care, and tacit, or experiential knowledge acquired from past patient cases represent a valuable asset from which healthcare providers can simultaneously consult to inform the decision-making process, thus leading to improved patient care planning and the delivery of best practice. In palliative medicine, clinical decision-making is a fundamental element in the delivery of effective pain and symptom management for end-of-life cancer patients; however, palliative care often presents healthcare providers with unique patient cases that challenge their expertise to make highly proficient decisions.

1.1 Problem Domain of Palliative Care

In palliative medicine, advanced-cancer patients commonly experience a multitude of symptoms, which are often ineffectively managed and consequently have a profound impact on their physical and psychosocial health. With deteriorating functional performance and inability to perform many activities of daily living, they suffer diminished sense of well-being and quality of life as they near end-of-life [1]. Fortuitously, research has suggested that effective symptom management for advanced cancer could be achieved for more than 90% of patient cases if healthcare providers were to proficiently use existing knowledge resources – specifically, explicit and tacit knowledge resources that are present in the palliative healthcare environment that can assist with clinical decision-making and care planning [2-3]. Despite the advantages of using existing knowledge resources to assist with clinical decision making for the delivery of excellent symptom management, present practice reveals a knowledge gap which has been attributed to inadequate education and practical training in symptom management, a paucity of standards of care and evidence-based practice for end-of-life cancer care, and virtually nonexistent capture of records of care delivery to a knowledge repository for future usage in support of highly-proficient clinical decision-making [2, 4-7].

2 Overview of Our Knowledge Management Solution

In the knowledge intensive field of palliative medicine, a vast amount of information can be derived from various medical knowledge resources [8]. Collectively, tacit and explicit medical knowledge present a valuable asset that can improve clinical decision-making for symptom management, and ultimately affect the quality of care delivered to end-of-life cancer patients [8, 10-11]. To this end, we have adopted a medical knowledge management framework, which consists of several processes to support the capture, storage, retrieval, sharing, and management of tacit and explicit medical knowledge in palliative care. Our approach enables healthcare providers to (a) leverage tacit, experiential knowledge, and explicit, evidence-based knowledge, originating from various medical knowledge resources; and (b) represent this knowledge in the form of structured cases within a case-based reasoning (CBR) system to support clinical decision-making for effective symptom management in palliative care. The overall objective is to enrich clinical decision-making with the integration of explicit and tacit medical knowledge resources that currently exist in palliative medicine.

2.1 Knowledge Capture and Representation

We extracted *tacit, experiential knowledge* in the form of patient cases from an original dataset that consisted of 276 records of patients who had been admitted to the Victoria Hospice Society, Palliative Care Unit. The selection criteria for the extracted patient cases were (a) presence of advanced cancer, and the (b) presence of at least one symptom from the Edmonton Symptom Assessment Scale. Following this, we selected appropriate care interventions from symptom management guidelines that could be executed to manage the presented symptoms for each case. These interventions or case solution statements represent the seven care tasks which comprise the standard palliative care delivery process: patient assessment, pharmacological considerations, pharmacological therapies, non-pharmacological therapies, patient activity, psychosocial considerations, and education.

2.2 Knowledge Storage and Retrieval

This knowledge was archived into a knowledge repository or case base, in the form of structured cases (patient cases with recommended care solutions) within a case-based reasoning CDSS. An inference engine and user interface were developed to enable healthcare providers to: (a) retrieve knowledge stored within the knowledge repository, (b) reuse case knowledge for clinical decision-making in palliative care, and (c) store new case knowledge, ensuring the preservation of valuable knowledge assets from the palliative medicine community.

3 Case-Based Reasoning System Architecture

Our case-based reasoning system incorporates a case library, comprised of 276 patient cases and their simulated case solutions, a case-based reasoning engine, and a web-based user interface. The design of the presented case-based reasoning system was motivated by Aamodt and Plaza's (1994) CBR cycle, whereupon the case-based reasoner must perform four sequential processes: case retrieval, case reuse, case revision, and case retention. Below, we provide the design details of how these tasks were accomplished in our CBR system.

3.1 Case Representation

In our CDSS, case-based reasoning is used to support clinical decision-making for pain and symptom management in cancer care. Patient cases are represented by several feature attributes that describe a problem (e.g., mild pain) and the solution to the problem. In this system, the feature attributes that formed the problem description are: gender, age, cancer type, palliative performance scale (PPS), palliative performance scale-level, and the symptoms that were experienced by advanced cancer patients while in the Victoria Hospice Society (VHS) palliative care unit. For this research, simulated case solutions were derived using collaborative care plans and evidence-based symptom management guidelines. These case solutions were vetted by physicians in palliative care services, Capital District Health Authority. The solution attributes (care intervention categories) which formed case solutions were: patient assessment, pharmacological considerations, pharmacological therapy, non-pharmacological therapy, patient activity, psychosocial considerations, and patient/family education.

Care Task	Symptom	Solution Statement
Patient Assessment	Pain	Identify pain intensity using ESAS: Mild pain (ESAS 1-3), Moderate pain (ESAS 4-6), Severe pain (ESAS 7-10).
Pharmacological Therapy	Pain	Acetaminophen (325-650 mg PO or PR every 4-6 hrs)

Fig. 1. Exemplar Case Solution Statement

3.1.1 Assigning Importance Values to Case Features

An importance value, or weight, was assigned to each feature attribute of the case description to ensure that the most relevant set of cases which are similar to a new case would be retrieved from the case library.

3.2 Case Retrieval

In this CBR system, healthcare providers are able to interact with the system via a web-based user interface. Users enter the feature attributes of a new patient case on an electronic web form, and then perform a search query to retrieve the 10 most similar patient cases that are stored in the case library. The resultant set of cases is presented to the user in order of descending similarity. During case retrieval, similarity computation is calculated by applying the nearest-neighbor algorithm to find the most similar cases from the case library. A second algorithm was developed to compare the overall performance of the nearest-neighbor algorithm with a combined inductive decision tree/nearest-neighbor algorithm.

In our presented system, we combined an inductive decision tree with a nearest-neighbor algorithm to optimize the efficiency of the CBR system. The inductive decision tree is used to retrieve a selection of indexed cases for subsequent similarity matching by the nearest-neighbor algorithm, which applies a similarity metric to return ranked cases in descending order of similarity to the target case [13]. The ID3 algorithm is used to induce a decision tree by ranking attributes according to their importance in classifying the data. This algorithm ranks attributes using an entropy measure, and is first applied to the complete corpus of cases to determine the attribute with the highest entropy. This attribute becomes the top-most decision point of the tree and is referred to as the root node. Subsequently, this process is applied recursively down the tree until all cases have been classified and indexed in leaves or until there are no further attributes to incorporate [12, 13].

3.3 Compositional Adaptation

In medical case-based reasoning, case adaptation continues to be the main challenge to the development of a CDSS which applies the complete CBR method. Compositional adaptation is predicated upon combining the most salient solution components from multiple past cases to derive a final composite solution which is more representative of a user's target case query. This adaptation technique requires that (a) solution components be independent, or mutually exclusive, and (b) few conflicts exist between solution components. We explored the concept of compositional adaptation, and how this adaptation technique could be applied to our existing CBR system.

4 Evaluating Clinical Validity of CBR System Output

The adoption of a CDSS into existing clinical workflow ultimately depends upon the involvement of its intended audience during system development and the evaluation process. For this reason, it was important to enlist the participation of key informants from palliative care, thereby leveraging their domain expertise, and ensuring that clinicians in palliative care accept the overall concept and efficacy of the proposed CDSS. In addition, key informants provided input on the efficacy and clinical validity of the system output.

During the clinical validation process, the key informants formulated five patient cases that would typically be seen in palliative care, using a *template for patient case description* provided by the researcher. Upon creating patient cases, the key informants accessed the web-based research interface where they were presented with a web-form to input each of their simulated case descriptions for a search query. In the query process, the system retrieves similar patient cases from the system database along with their corresponding case solutions. Next, the key informants evaluated the clinical validity of the retrieved case solutions (system output). They examined the case solutions for clinical appropriateness, deficiencies, contraindications of medications or procedures, erroneous care protocols, and whether these case solutions represented local best practice and clinical practice guidelines.

Upon completion of the clinical validation process, the key informants completed a qualitative structured feedback questionnaire which consisted of open-ended questions to evaluate the concept of implementing the proposed CBR system for end-of-life cancer care, case feature attributes (case description and solution attributes), and the efficacy and clinical validity of the system output.

Key informants suggested that a CDSS such as this may be beneficial in assisting palliative care nurses with decision-making for unique palliative care cases, at the point of care. Participants agreed that the CDSS incorporated the most relevant patient attributes to be considered in a patient case, but also suggested additional attributes that could enhance the clinical problem description and case solution - allergies, DNR status, pain classification, oral complications, therapeutic relationships, referrals to other services, spiritual assessment/care. Further recommendations were that case solutions be short prompts, minimal page navigation, and that the output be contained within one page. The results of this qualitative evaluation process will be taken into consideration for future development of our CDSS, and the resulting recommendations will be applied to both the case structure and the case solutions stored in the case library, as deemed appropriate.

5 Concluding Remarks

In this paper, we presented a CDSS for healthcare providers who are involved in planning and delivering end-of-life cancer care. Our CDSS incorporates tacit, experiential knowledge that is mapped to explicit, evidence-based standards of care (collaborative care plans, symptom management guidelines) currently used in the care-delivery process for patients with advanced cancer, and is represented in the form of structured cases within a case-based reasoning system to support clinical decision-making to improve the delivery of effective pain and symptom management in palliative care. Clinical decision support systems that integrate case-based reasoning provide healthcare providers with a decision-making tool that emulates the problem-solving approach typically used by healthcare providers in clinical decision-making. There is general agreement among healthcare providers that case-based reasoning offers many advantages for organizing experiential and domain knowledge that is generated in the complex and knowledge-intensive field of medicine. With CBR, healthcare providers interact with a CDSS to retrieve historical patient cases from a corpus of cases, similar to a new, unsolved patient case. They can then review the resultant set of retrieved cases to support their clinical decision-making for the new, unsolved patient case, thus leveraging tacit knowledge from past experiences.

We investigated an alternative presentation format to display system output using a compositional adaptation approach. With this method, the user is presented with a final composite solution, derived from combining the most salient solution components from multiple past cases. Although this adaptation method has proven advantageous for several case-based reasoning systems, further investigation is required to determine whether this technique is appropriate for our medical case-based reasoning system.

Acknowledgments. We would like to acknowledge our partnership with the Victoria Hospice Society, University of Victoria, and Capital Health District Authority.

References

1. Dudgeon D, et al. Managing continuity through collaborative care plans: A study of palliative care patients. http://www.chrsf.ca 2007 June.
2. Foley K, Gelband H. Improving Palliative Care for Cancer. Washington, D.C.: National Academy Press (2001)
3. Bruera E, Pereira J. Recent developments in palliative cancer care. Acta Oncologica (1998) 37(7): 749-757.
4. Weinstein S. Integrating palliative care in oncology. Cancer Control (2001) 8(1): 32-35.
5. Woodruff R. Palliative Medicine: Evidence-Based Symptomatic and Supportive Care for Patients with Advanced Cancer. South Melbourne, Australia: Oxford University Press, (2005)
6. Anderson K, Cleeland C. Cancer Pain Assessment and Management. Cambridge: Cambridge University Press (2003)
7. Bruera E. Improving the care of terminally ill cancer patients and their families. Journal of Clinical Oncology (2005) 23(30): 7378-79.
8. Bichindaritz I, Marling C. Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine (2006) 36: 127-135
9. Macura R, Macura K. Case-based reasoning: Opportunities and applications in health care. Artificial Intelligence in Medicine (1997) 9: 1-4
10. Babka O, Whar S. Case-based reasoning and decision support systems. Paper presented at the IEEE International Conference on Intelligent Processing Systems (1997) October; Beijing, China.
11. Bellazzi R, Montani S, Portinale L, Riva A. Integrating rule-based and case-based decision-making in diabetic patient management. Paper presented at the 3rd ICCBR (1999) July; Monastery, Germany.
12. Althoff K, et al. Case-based reasoning for medical decision tasks: The INRECA approach. Artificial Intelligence in Medicine (1998) 12(1): 25-41
13. Craw S. Introspective learning to build case-based reasoning (CBR) knowledge containers. Proceedings of the IAPR International Workshop on Machine Learning and Data Mining in Pattern Recognition (2003) July; Germany

System Overview: Breast Cancer Prognosis Through CBR

Beatriz López[1], Albert Plà[1], Carles Pous[1], Pablo Gay[1], and Joan Brunet[2]

[1] University of Girona,
Campus Montilivi, P4 Building,
Girona, E17071, Spain,
{beatriz.lopez,albert.pla,carles.pous,pablo.gay}@udg.edu
[2] Institut Català d'Oncologia
Avda. França s/n
Girona, E17007
Jbrunet@iconcologia.net

Abstract. This paper is concerned with the development of a Case-Based Reasoning decision support system for breast cancer prognosis based on clinical data. The research we have been carried out during these years yielded a specific tool for medical prognosis, eXiT*CBR. The system brings close the CBR technique to the physicians, making its parameterization and results viewing fast and simple. The obtained results are demonstrated to be better than the ones given by the standard statistical techniques commonly used by the physicians.

Keywords: Case-based reasoning, Experimentation supporting tool, Breast Cancer Prognosis

1 Introduction

Breast cancer is one of the priority interest on health care [4]. Designing decision support systems for predicting its likely outcome (prognosis) is a key issue to apply preventive medicine. Although most decision support tools focus on image analysis, like mammographies, that reduce mortality up to the 30-40% [1], most biopsies analyzed are benign. Therefore, developing systems based on other clinical data is important to reduce annoying and costly biopsies.

Our research is concerned with the development of a decision support system for breast cancer prognosis based on clinical data. Physicians gather several patient data from clinical studies, that comprehend general patient characteristics (age, sex), as well as food habits, toxic habits (smoking, alcohol), drug ingesta, X-Ray expositions, and others. Several variables are gathered from the clinical studies, and the physicians wish to learn about the relationship among data and the illness. Learning a model from data is a difficult task, due to the huge amount of variables to be handle, and often, the low number of cases available in proportion to variables. On the other hand, case-base reasoning offers the possibility of providing an individual prognosis assessment based on the similarities among patients without the need of learning an explicit model.

Our research concerns the development of such a case-base reasoning decision support system for breast cancer prognosis. We have been working for years in the application of case-based reasoning in the breast cancer preventive field, yielding a specific tool for medical prognosis, eXiT*CBR. In this paper we provide an overview of the tool and its application to breast cancer, as well as we review the different results obtained as so far.

2 Overview

eXiT*CBR is a framework focused on helping engineers to design their CBR tools, specially in medical domains. The first exit*CBR version was used to build a CBR tool for developing medical prognosis applications. Then, it was extended as a new platform adding different working modes that allow the collaboration of different CBR systems, crating a team of agents who cooperate towards a common solution. This allows the tool to be used as a distributed environment in which different medical teams cooperate to achieve a prognosis, assigning a CBR agent to each medical source of information.

The first version of the tool offered a modular and heterogeneous framework to combine different CBR techniques according to the medical application requirements. In order to specify the required steps for constructing CBR experiments, the tool contains several artifacts which are chosen using a configuration file. The structure of this configuration file contains the information regarding the data and processes of the CBR system: information related to the localization of the dataset used in the experiment; the preprocessing steps which are divided in discretization, normalization and feature selection; the definition of the CBR main core, also divided in according to the case metrics and measures, the way the missing information is treated, the retrieve and the reuse algorithms defined; and the statistics which the experiment will generate (ROC curves, sensitivity and specificity measures, etc.).

When a physician needs to create a new CBR model for breast cancer diagnosis, thanks to the modular structure of eXiT*CBR (Figure 1), he can create it and test it from the main platform. First of all, given a database containing the case information, eXiT*CBR offers the opportunity to generate a knowledge base in a suitable format for the tool and to divide the case database into different data sets in order to train and test the CBR core. Then, using a configuration file editor, the physician can define the CBR model which is divided in 3 main blocks: the pre-process, the model generation (which includes the 4 phases of the CBR) and the experimentation. In the pre-process block, the physician can choose the source knowledge base, the attributes which will be used in the model and how will they be treated (discretization, kind of normalization, etc.). Then, in the model generation step, the physician must define the main core of the CBR. In this block the retrieve, reuse, revise and retain methods and their parameters (metrics, thresholds, etc.) can be chosen from a list containing the available techniques. Finally, in the experimentation block, the physician can determine if he wants to evaluate the performance of the chosen methods using

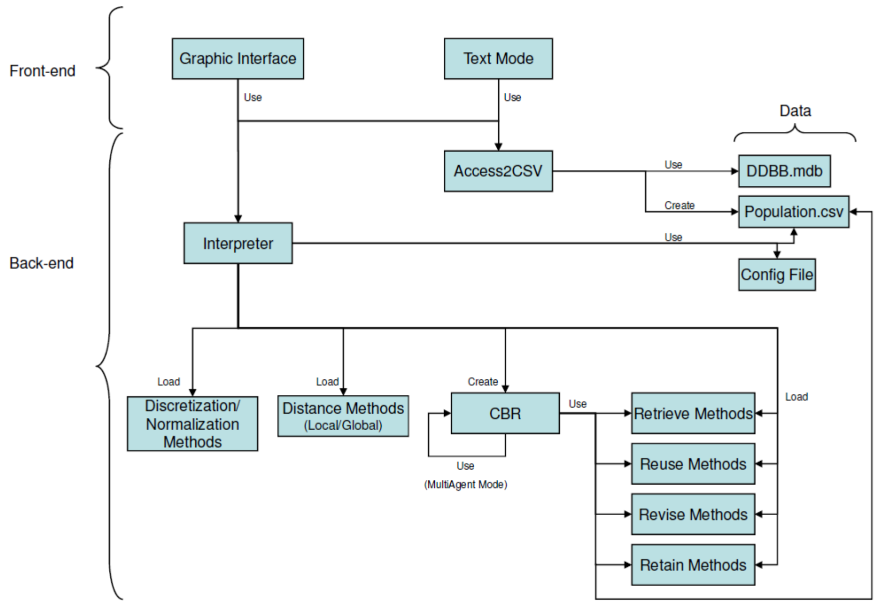


Fig. 1. Tool Architecture

statistical data (ROC Curves, accuracy, precision and recall, etc.) or if he wants to diagnose a new case.

Currently, the tool has been extended with distributed multi-agent capabilities, so different medical collectives can collaborate in the prognosis as an ensemble. In this new version, a different knowledge database is assigned to each agent so they become experts in different fields (all the CBR based agents are trained with a different set of examples and are focused on a different ensemble of attributes). This strategy can also help to diagnose by using different databases obtained from different source, giving also the chance of using distributed databases. As each agent provides its own opinion, a coordinator agent is needed to make a final decision based in a weighted voting schema and the confidence of each diagnose.

When a new case needs to be prognosed, the coordinator agent delivers the case to each CBR agent which contains a different knowledge base with different cases or a different expertise. Each agent computes its own result following a case based reasoning behavior. First, they evaluate the similarity between the broadcasted case and the instances they have stored in their own knowledge base. Then, a specific amount of cases (defined in the configuration file) are retrieved and fed to the reuse stage. Due to boosting characteristics, in the reuse stage all the agents follow the same method. After reuse, each agent has its own prognostic (0 for negative/healthy cases and 1 for positive/ill samples) and a confidence δ in its prognostic.

Once that all the agents return their prognostics, the task of the coordinator agent is to determine the final answer. The coordinator, in a previous learning stage, has assigned a weight to each agent concerning their reliability (other authors consider this weights as trusts [2]), and using the weights and confidences, the coordinator agent accumulates all the information regarding positive diagnoses and to negative cases independently. Finally, if positive results overcome the negative ones, the case is labeled as positive (ill), otherwise as negative (healthy).

3 Evaluation

The system has been tested, both, on UCI medical data (Breast Cancer Wisconsin Dataset)[1] and on a private database provided by physicians [2]. This last one consists of 871 cases, 628 corresponding to healthy women and 243 to women with breast cancer. There are 1199 attributes for each case. They correspond to people's habits (smoker or not, diet, sport habits, etc.), disease characteristics (type of tumor, etc.), and gynecological history. A complex database helps to illustrate the capabilities of the system when real, complex data are dealt with.

Although several tools currently support breast cancer prognosis, the system described on this paper can cope and combine different type of data, such as genetic, family and epidemiologic information. None of other systems satisfies the current needs of medical users. The Gail model, for example, includes some epidemiological data and a single piece of data regarding family information. Other models, such as Clauss model, uses more in-depth family information, while others focus on genetic information, like the BRCAPro program does. In general, the available systems deals with a single model. Conversely, the approach proposed with a CBR system allows medical users to tackle all the data at once, in an integrated way, dealing with different patterns implicit in the data to provide a diagnosis.

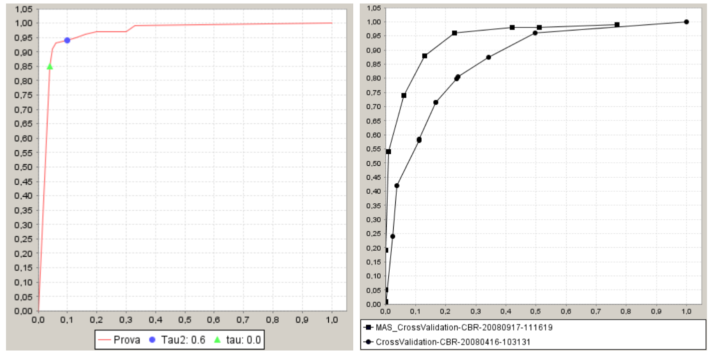
The system has been proven to provide better results than previous cited methods as it improves the number of True Positives (TP). For example, in Figure 2.a, when the system is used and tuned for the public Wisconsin Breast Cancer data base, an AUC (Area Under Curve) of 0.961 with a ratio of TP=0.95 (hit rate) and FP=0.1 (false alarms) can be derived for a threshold $\tau = 0.6$.

On the other hand, when tested with the private database, its performance is reduced since the given database has a lot of missing values. However the system can deal with numerical, categorical, string, incomplete and missing data thanks to its modular architecture, wich uses the appropiate metric function acording to the attribute value and therefore is able to produce a quite good result. Figure 2.b gives a comparison when the system is used as single CBR and as a MAS approach.

The result shows an AUC of 0.856 for single CBR and an AUC of 0.941 for the MAS approach. So, in this case the MAS approach produces better results.

[1] http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[2] Database provided by Hospital de la Santa Creu i Sant Pau (Barcelona)



(a) (b)

Fig. 2. (a) Results with Wisconsin database. (b)Private database with single and Multi Agent CBR.

As a conclusion, the system is more accurate than other methodologies used by physicians for the prognosis. Also, it is important to mention that the system possesses a navigation tool that makes it tremendously useful when tuning the system and comparing the results.

No less important is to remind that the system allows to introduce family information, relating how the disease is propagated through the genealogical tree, together with the genetic information of its individuals. This is a powerful information that normally physicians use in conjunction with other clinical data to get an outlook for the patients.

4 Conclusions

In this paper we have presentd the overview of our system for case-based reasoning applied to breast cancer prognosis. In this research it is important to highlight the synergies generated in a multidisciplinary team composed by physicians and technicians. On the one hand, physicians have now a system for breast cancer prognosis based on case-based reasoning. On the other hand, the aim of defining a general-specific tool, so that the insights on the breast-cancer development system can be transferred to other illnesses, has concluded in the development of a specific tool for medical prognosis, exit*CBR. Moreover, the tool has been integrated and hybridized with other information systems (as pedigree tools) and artificial intelligence techniques (as genetic algorithms), trying to fulfill several medical data formats.

However, there is a weakness in case-based reasoning difficult to overcome: the inability of producing explicit models. Physicians usually prefer explicit models, probably because their aim is to identify and design drugs or treatments for the learn patient patterns. To overcome such a situation, the development of more sophisticated case-based reasoning system that includes treatment options could be a good starting point. Another good direction is the hybridization of sequence learning, which will open in a near future a new dimension for patient data management, and in which we have started to have good results on medical equipment diagnosis [3].

References

1. Bilska-Wolak, A., Floyd Jr., C.: Development and evaluation of a case-based reasoning classifier for prediction of breast biopsy outcome with BI-RADSTM lexicon. Medical Physics 29(9), 2090–2100 (2002)
2. Birk, A.: Boosting cooperation by evolving trust. Applied Artificial Intelligence 14, 769–784 (2000)
3. Compta, M., López, B.: Integration of sequence learning and cbr for complex equipment failure prediction. In: In: A. Ram and N. Wiratunga (Eds.): ICCBR 2011, LNAI 6880. p. 408422 (2011)
4. Viladiu, P., Izquierdo, A., Marcos, R., Vilardell, L.: A breast cancer case-control study in girona, spain. endocrine, familial and lifestyle factors. European Journal of Cancer Prevention (5), 329–335 (1996)

System Overview on the Clinical Decision Support System for Stress Management

Mobyen Uddin Ahmed, Shahina Begum and Peter Funk

School Of Innovation, Design and Engineering,
Mälardalen University, PO Box 883 SE-721 23, Västerås, Sweden
{firstname.lastname}@mdh.se

Abstract. There is an increased need for Clinical Decision Support Systems (CDSS) in the medical community as ICT technology is increasingly used in hospitals as more and more patient data is stored in computers. A CDSS has the potential to play a vital role and bring essential information and knowledge to the clinicians and function as a second opinion in their decision-making tasks. In this paper, a CDSS in stress management is presented where the CDSS can help the clinicians in order to diagnosis and treat stress related disorders. As a foundation for the CDSS, the Case-Based Reasoning (CBR) approach has been used as a core method of the system. The systems also combine other techniques from artificial intelligence in a multimodal manner, such as fuzzy logic, rule-based reasoning and textual information retrieval. In this paper we review our experiences and research efforts while developing the CDSS. The performance of the CDSS shows that the system can be useful both for trainee clinicians as an expert and as well as for senior clinicians as a second option. Moreover, the observation shows the current developments, and pros and cons of the CBR approach.

1 Introduction

A Clinical Decision Support System (CDSS) that can aid experts in the decision making process of selecting treatment and individualize treatments for patients is therefore increasingly important. CDSS that bear similarities with human reasoning are often more easily accepted by physicians [3, 4, 6, 7, 8, 16 and 19]. Hence, CDSSs that based on this approach are believed to play an increasingly prominent role in tomorrow's health care.

In this paper, a case study on a CDSS for stress management is presented. Here, the case-Based Reasoning (CBR) approach is used as a core technique since it learns from experience in order solves a current situation [1, 5 and 11]. The advantages of CBR in the medical domain have been identified in several serves i.e. in [12] [17]. However, several other AI techniques combined with CBR approach such as Fuzzy Logic (FL) is incorporated in order to calculate the similarity between two cases, which handles vagueness and uncertainty which is inherent in much of human reasoning [13] [16]. Moreover, a fuzzy rule-based classification scheme handles the so called boot strap problem. That is, a part of the research work has made an effort to improve the performance of the stress diagnosis task when there are a limited number of cases [20]. In the stress management domain, reliability of the system for decision-making tasks is further improved through textual Information Retrieval (IR) with ontology [14]. A three phase computer-assisted sensor-based system for treatment including biofeedback training in stress management is proposed in [15].

1.1 Problem in the Medical Domains

In the stress management application domain, Finger Temperature (FT) is a popular measurement used by some clinicians to determine stress. Medical investigations have shown that FT has a correlation with stress for most people [18]. However, FT changes are so individual due to health factors, metabolic activity etc as shown in fig. 1. Interpreting/analysing FT and understanding large variations of measurements from diverse patients require knowledge and experience.

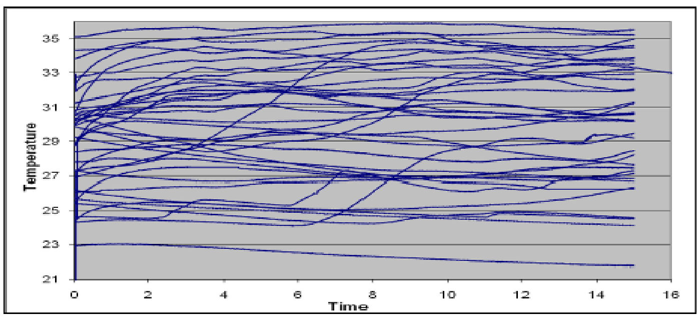


Fig. 1. FT sensor signals measurement samples are plotted.

Moreover, the effect of FT changes is very individual and there are some other factors such as the patient's feelings, behaviours, social facts, working environments and lifestyle which also plays a role in the diagnosis of stress. Besides the sensor measurements, such information can also be collected using text and Visual Analog Scale (VAS) input. VAS is a measurement instrument (a scale ranging between 0 and 10) which can be used to measure subjective characteristics or attitudes. So, CDSSs in this domain should be capable of dealing with textual information besides biomedical sensor signals.

Biofeedback is today a recognised treatment method for a number of physical and psychological problems. Stress is a more complex area for biofeedback as a treatment and different patients have very different physical reactions to stress and relaxation. In the stress area, a clinician commonly supervises patients in biofeedback and together with the patient they make individual adjustments to measurement and treatment. Since there are large individual variations when looking at FT, it is a worthy challenge to find a computational solution to apply in a computer-based system. Thus, one of the main goals of this research is to propose methods or techniques for a multipurpose-oriented CDSS i.e. a system that supports in the diagnosis and treatment of stress.

2 Overview of the Systems

In stress diagnosis, the structure of the cases contains two parts as most common practice in CBR 1) problem description and 2) solution. A protocol comprises different conditions in 6 steps, they are as follows: baselines, deep breath, verbal stress, relax, math stress, and relax [2], which is used during data collection. Moreover, patients also provided their feelings, behaviours, social facts, working environments and lifestyle information in free text. Thus, the problem description is formulated with a feature vector where 25 features in total are used [13]. Among the 25 features, 19 features are extracted from FT sensor signal [3], 5 selected features which reflect a subject's perception (i.e. VAS) and 1 textual feature [14]. The solution part of cases contains case classification i.e. levels of stress and expert confidence. The levels of stress are denoted as VeryRelaxed, Relaxed, Normal/Stable, Stressed and Very Stressed and the confidence level are denoted as High, Medium and Low. The steps that combine the measurements presented in Fig. 2 that is the FT sensor signal measurement and textual information are collected, features are extracted, and a new case is formulated using extracted features and finally send to the CBR cycle for stress diagnosis.

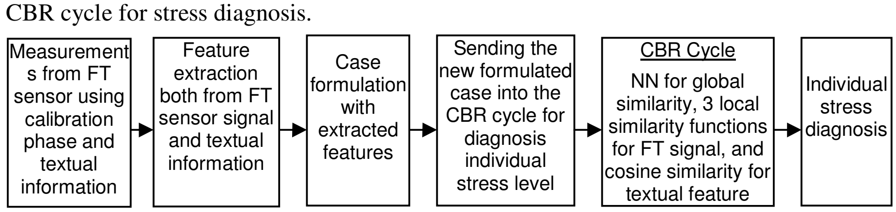


Fig 2. The steps considering both the FT measurement and the textual information**.**

A nearest neighbor (NN) algorithm is applied as a global similarity function to retrieve similar cases and three different local similarity matching functions: 1) modified distance function 2) similarity matrix and 3) fuzzy similarity matching have been implemented into the system. However, only fuzzy similarity is used as default local similarity function to retrieve most similar FT signal cases. For the textual features, the *tf-idf* (term frequency–inverse document frequency) [10] weighting scheme is used in a vector space model [9] together with cosine similarity to determine the similarity between two cases. Additional domain information that often improves results, i.e. a list of words and their synonyms or a dictionary provides comparable words and relationship within the words using classes and subclasses are also included [14]. Note that, the system did not combine textual features and FT features in order to provide a combined solution rather the system represent the most similar textual cases.

Initially, it was difficult to collect reference cases for the clinical deployment of many systems and the limited number of cases can impair the performance of a CBR system. For example, in our case library only 3 cases are VeryRelaxed. So, given a new problem case similar to VeryRelaxed, the system may not find any classification. Here, the artificial cases in the "VeryRelaxed" class could propose a classification. So, our system uses artificial cases along with the reference cases in the initial stage. A fuzzy rule-based reasoning algorithm that automatically classifies new cases is introduced into the CBR system to initiate the case library with artificial cases. The steps of the approach is as follows: a) the FT sensor signal is measured through the calibration phase b) the number of features are extracted from a sensor reading and formulates a generalized feature from the extracted features c) the fuzzy inference system (FIS) is then applied together with the generalized feature to classify a new case d) the output from the FIS provides the new classification along with the feature vector e) finally, this case is saved into the case library as an artificial case. The details information is presented in [20].

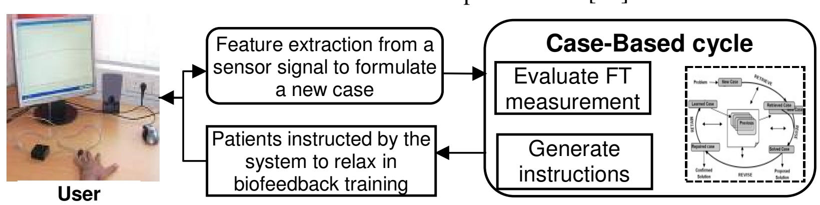


Fig 3. A schematic diagram of the steps in the biofeedback treatment cycle.

In stress treatment i.e. biofeedback treatment, the system use three modules of CBR, they are 1) analyse and classify a patient and make a risk assessment, 2) determine individual levels and parameters, and finally 3) adapt and start the biofeedback training. Here in classify a patient module; the system uses a different protocol which is a 9 min finger temperature (FT) measurement in 4 steps. In this protocol, a person starts with a baseline for 3 min, then 2 min neutral task, then 2 min relaxation and finally, task + relaxation i.e. without any stress [15]. In this cycle shown in Fig. 3, a user connects a sensor to their finger and can see the changes of FT during several instructions in relaxation training. The FT measurements are performed in real time and every 2 minutes the system evaluates the last 2 minutes measurement and if necessary generates instructions for the patient. A CBR cycle is applied for the biofeedback training in stress management; this training time is flexible, which means a patient can choose the duration of their training between 6 minutes (as minimum) to 20 minutes (as maximum)[15].

3 System evaluation and conclusion

The performance of the CDSS for stress management in terms of accuracy has been compared with experts in the domain where the overall goal is to see how close the system works compare to the experts. The reference cases, in their conditional or problem description part, contain a vector of the extracted features and the solution part comprised of expert's defined classification as diagnosis. For the evaluation, 2 senior clinicians and 2 trainee clinicians were involved; one of the clinician who performed the classification task has been working both as a clinician and a researcher in the field of psychophysiology for more than twenty (20) years. The sensitivity and specificity analysis have been done on 68 cases out of 46 subjects and the obtained sensitivity, specificity and overall accuracy compared to an expert are found as 92%, 86% and 88% respectively [13]. Moreover, the system classification performance also compared with 2 trainees and one senior clinician. Goodness-of-fit (R^2) value for both the test groups (setA and setB) are 87% and 86% by the system, whereas the R^2 values by the trainee clinicians are 76% and 72% for the set setA, and for the set setB are 80% and 82%. Similarly, for the senior clinician, the R^2 value is almost same for both the senior clinician and the system i.e. 82% and 88%. The comparison results shows that the system out performs than the clinicians and the details evaluation can be found in [13]. According to the results in [13] and [20], it showed that the system improves its performance with 22% in the criteria of correctly classified case using combined case library (i.e. both artificial cases and reference cases). So, it shows that it is possible to increase accuracy in the classification task by extending the case library with artificial cases if the case library contains a limited number of reference cases.

In this paper, a case study on a CDSS in stress medical domain has been presented in order to assist clinicians in their decision-making tasks such as diagnosis, classification and providing individualised treatment. Here, a hybrid case-based reasoning approach is used where CBR is used as a core technique for the domain and other AI methods are combined in a multimodal approach. We describe the methods and techniques applied and shown their benefits, strengths and weaknesses. With these example systems it can be argued that a hybrid approach shows some promising features and benefits in the medical domain. Problems that have been difficult to solve with a single technique become manageable by combining different methods and techniques bound together by the concept of reasoning with cases. CBR also enable more individualised diagnosis and advice compared with many other approaches, which we see as one of the strengths of the method, and which also leads to a high level of acceptant amongst clinicians.

Acknowledgments. The authors gratefully acknowledge the project IPOS, Integrated Personal Health Optimizing System funded by KKs, NovaMedTech EU objective 2 project and PROEK, Ökad Produktivitet och Livskvalitet funded by Sparbanksstiftelsen for their support of these research projects.

Reference

[1] Aamodt A. and Plaza E., Case-based reasoning: Foundational issues, methodological variations, and system approaches, AICom - Artificial Intelligence Communications, IOS Press, Volume 7, Issue 1, pp 39-59, 1994.

[2] Begum S., Ahmed M.U., Funk P., Xiong N. and Scheele B. V., Using calibration and fuzzification of cases for improved diagnosis and treatment of stress, The Proceedings of the 8th European Workshop on Case-based Reasoning, pp 113-122, 2006.

[3] Begum S., Ahmed M.U., Funk P., Xiong N. and Schéele B.V, A case-based decision support system for individual stress diagnosis using fuzzy similarity matching, Computational Intelligence, Blackwell Publishing, Volume 25, Nr 3, pp 180-195, 2009.

[4] Corchado J.M., Bajo J. and Abraham A., GERAmI: Improving healthcare delivery in geriatric residences, IEEE Intelligent Systems, Volume 23, Issue 2, pp 19-25. ISSN: 1541-1672, 2008.

[5] Demirali J., Reason: An introduction to critical thinking, 1st edition. ISSBN 978-1-4583-7749-4, Analogical Reasoning Institute, 2011.

[6] Marling C. and Whitehouse P., Case-based reasoning in the care of Alzheimer's disease patients, 4th International Conference on Case-Based Reasoning, ICCBR 01, pp 702-715, 2001.

[7] Montani S., Portinale L., Leonardi G. and Bellazzi R., Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Medicine, Volume 37, Issue 1, pp 31-42, 2006.

[8] Nilsson M., Funk P., Olsson E., Von Schéele B. H. C., and Xiong N., Clinical decision-support for diagnosing stress-related disorders by applying psychophysiological medical knowledge to an instance-based learning system, Artificial Intelligence in Medicine, Volume 36, Issue 2, pp 159-176, 2006.

[9] Salton G., Wong A. and Yang C.S., A vector space model for automatic indexing, Communications of the ACM, Volume18, Issue11, pp 613–620, 1975.

[10] Salton G. and Buckley C., Term weighting approaches in automatic text retrieval, Information Processing and Management, Volume 24, Issue 5, p513-23, 1988.

[11] Watson I., Applying Case-Based Reasoning, Techniques for Enterprise Systems, Morgan Kaufmann Publishers Inc, 250 pages, ISBN 1-55860-462-6, 1997.

[12] Begum S., Ahmed M.U., Funk P., Xiong N., Folke M., Case-Based Reasoning Systems in the Health Sciences: A Survey on Recent Trends and Developments, International journal of IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol 41, Issue 4, 2011, pp 421 - 434.

[13] Ahmed M.U., Begum S., Funk P., A Hybrid Case-Based System in Stress Diagnosis and Treatment, Accepted in the IEEEEMBS International Conference on Biomedical and Health Informatics (BHI2012), 2012.

[14] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, Case-based Reasoning for Diagnosis of Stress using Enhanced Cosine and Fuzzy Similarity. International journal of Transactions on Case-Based Reasoning on Multimedia Data, vol 1, Number 1, IBaI Publishing, ISSN: 1864-9734, 2008, pp 3-19.

[15] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, A Multi-Module Case Based Biofeedback System for Stress Treatment. International journal of Artificial Intelligence in Medicine, vol.51, Issue 2, Publisher: Elsevier B.V., 2010, pp 107-115.

[16] Perner P. An Architecture for a CBR Image Segmentation System, In Journal on Engineering Application in Artificial Intelligence, Volume 12, Issue 6, pp 749-759, 1999.

[17] Bichindaritz I, Marling C. Case-based reasoning in the health sciences: What's next? In Artificial Intelligence in Medicine. 36(2), 2006, pp 127-135

[18] Blanchard E.B., Kolb L.C., Pallmeyer T.P. and Gerardi R.J., A psychophysiological study of post-traumatic stress disorder in vietnamveterans, Psychiatric Quarterly, Volume 54, Nr. 4, pp 220-229.

[19] Ahmed M.U., Funk P., A Computer Aided System for Post-operative Pain Treatment Combining Knowledge Discovery and Case-Based Reasoning, Accepted in the 20th International Conference on Case-Based Reasoning, ICCBR'12.

[20] Ahmed MU, Begum S, Funk P, Xiong N. Fuzzy Rule-Based Classification to Build Initial Case Library for Case-Based Stress Diagnosis. In the proceedings of 9th International Conference on Artificial Intelligence and Applications (AIA), 2009, pp 225-230.

A Case-Based Patient Identification System using Pulse Oximeter and a Personalized Health Profile

Mobyen Uddin Ahmed, Asif Moinul Islam and Amy Loutfi

Center for Applied Autonomous Sensor Systems
Örebro University, Sweden
mobyen.ahmed@oru.se; asiish101@studentmail.oru.se;
amy.loutfi@oru.se

Abstract. This paper proposes a case-based system framework in order to identify patient using their health parameters taken with physiological sensors. It combines a personalized health profiling protocol with a Case-Based Reasoning (CBR) approach. The personalized health profiling helps to determine a number of individual parameters which are important inputs for a clinician to make the final diagnosis and treatment plan. The proposed system uses a pulse oximeter that measures pulse rate and blood oxygen saturation. The measurements are taken through an android application in a smart phone which is connected with the pulse oximeter and bluetooth communication. The CBR approach helps clinicians to make a diagnosis, classification and treatment plan by retrieving the most similar previous case. The case may also be used to follow the treatment progress. Here, the cases are formulated with person's contextual information and extracted features from sensor signal measurements. The features are extracted considering three domain analysis: 1) time domain features using statistical measurement, 2) frequency domain features applying Fast Fourier Transform (FFT), and 3) time-frequency domain features applying Discrete Wavelet Transform (DWT). The initial result is acceptable that shows the advancement of the system while combining the personalized health profiling together with CBR.

1 Introduction

The continuous monitoring of physiological measurements using wearable sensors has the potential to provide new opportunities for prevention and early detection of disease. Specifically, with appropriate data analysis, personalization to user specific conditions can be achieved providing detection when relevant to users. Observing parameters such as pulse, blood oxymetry, and blood pressure can be achieved with relatively cheap sensors that are easily integrated on the body in wireless area networks. Typically a three tier architecture is used where first the sensors collect the data, then transmit them to a local server such as a mobile device, and finally data is sent to a remote server which can be accessed via specific clients e.g., physician at a health clinic.

A particular challenge in these applications is to find suitable algorithms for data processing and interpretation, where previously several techniques have been applied such as Neural Network (NN) [16] and Support Vector Machine (SVM) [17]. These methods are often black box classification techniques which make it difficult for experts to gain further insight into the structure present in the data. This point is particularly relevant as medical personal are already challenged by the fact that continuous data e.g. pulse is normally viewed at discrete intervals and seldom over long periods with high sampling frequency (e.g., 1 Hz over 48 hours). Therefore increasing the level of trust and explication is paramount for these new applications which endeavor to provide more information about an individual's health.

In this paper, we investigate the use of Case-Base Reasoning (CBR) techniques which uses a personalized health profiling protocol and applies it to identify patients using based on physiological measurements. To handle the complex data processing, first a feature extraction and selection is performed on the raw measurements. The features are then fed to the CBR system. Patient's data is collected by using a personalized health profiling technique where each patient performs a set of scripted tasks during the first 9 minutes of collection. The objective of the system is to identify a subject based on subsequent measurements. If a subject is wrongly identified, we consider this as an anomalous situation and flag an alert. In this way, the initial profiling period is a form of individualized classifier. This method was chosen as it does not directly depend on the medical expertise which during discussions with the researchers had expressed a difficulty in labeling the continuous data. To better illustrate, if 'Subject A' a young healthy male has a set of measurements classified as 'Subject B' - a elderly female with cardiac arithemia, we consider this to be uncharacteristic of 'Subject A' and flag an alert. Therefore, a measure of success of the system resides in its ability to minimize false negatives when re-identifying the patients outside the 9 minute profile.

2 Overview of the System

The case-based Clinical Decision Support System (CDSS) for patient identification using physiological parameters has been designed and developed using a client-server based approach. That is, all the physiological measurements and cases are saved in a CBR library located on a remote server. Moreover, feature extraction, case formulation, and case-retrieval are also conducted on the server side. In the client side, an android application is deployed in a smart phone which collects subject's measurements using a pulse oximeter. The pulse oximeter is connected to a subject's finger as shown in fig.1, pulse rate and oxygen saturations data are measured and sent to the smart phone using bluetooth communication. The measurements are saved into the smart phone in '.txt' format and finally uploaded to the server. The steps and the data flow of the CDSS are illustrated in fig 1.

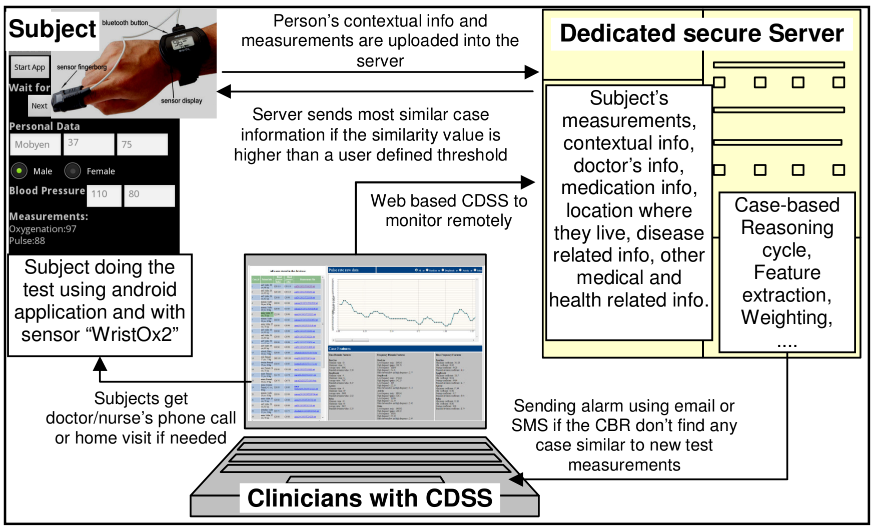


Fig. 1. The steps and the data flow of the CDSS.

The CDSS starts with a data collection step in which the measurements are collected using a personalized health profiling protocol discussed later in this section. The protocol is implemented in a smart phone using android 2.3.3 programming environment where the "WristOx$_2$" sensor is connected through Bluetooth communication. The details of Bluetooth communication and information about "WristOx$_2$" sensor can be found in [20]. After finishing the test protocol, the measurements are uploaded to a server, features are extracted and a new case is formulated. The new case is then entered into CBR cycle and in the retrieval step, the similarity value is calculated among the stored cases as presented in [1]. The server retrieves the most similar case depending on user defined similarity threshold and finally sends the most similar case information to the subject. At the same time, the server also sends a sms and email notification to the clinician and generates a warning alert if the stored cases are not similar to the new case. Here, the system also considers the user defined similarity threshold. Note that the server calculated the similarity using a very basic CBR approach, so only a simple matching algorithm is implemented. However, a web enable case-based CDSS is also provided. Here, clinicians can see details about the contextual information and measurements about a subject; moreover, an advance CBR is functioning in order to monitor the subject remotely. Further details on this implementation are presented later in this paper.

2.1 Establishing a Personalized Health Profile

A personalized health profiling is a test protocol which contains several steps with various tasks in different time durations [21]. Here, the system establishes an individual health profile without considering any clinical details. The steps and the overall procedures are illustrated in table 1.

Table 1. Procedures for establishing a personalized health profile.

Procedure \ Test Phases	Phase 1	Phase 2	Phase 3	Phase 4
Condition	Baseline	Deep Breath	Activity	Relax with Biofeedback
Observation Time	1 minute	2 minutes	4 minutes	2 minutes
Tasks	Subject reads newspaper	Subject takes breathe deeply	Subject is does exercise 	Subject tries to relax using biofeedback training

Phase1 can be seen as an indicator on whether the subject has a stable baseline. Here, the subject is requested to read any neutral text (i.e., daily newspaper) during this step. Using this phase, we can identify an individual's basic pulse rate, and also note fluctuations and other effects. During *phase2* the person breaths deeply which under guidance normally causes relaxation. Also how quickly the changes occur during this step is relevant to health profiling and recorded together with observed fluctuations. In *Phase3*, the subject is asked to walk at 4 to 6 km per hour using a walking exercise equipment or alternatively cycle at 70 to 90 rpm. During this phase, the pulse rate of a subject could be increased and the variation of pulse rate is an important parameter. In *Phase4*, the subject is asked to relax using the system's biofeedback instructions, for example how many times the subject should breathe during the relaxation period. How quickly the pulse rate of a subject is reduced can be observed during this phase.

3 Case-Based Monitoring System

The case-based patient identification system proposed here consists of three modules. The client's side module contains an android application that collects the data from the pulse oximeter following the health profile. The android application also shows the user an acknowledgement obtained from the server when the data is uploaded successfully. When the server side application receives the data file, the file is parsed and the features extracted. It also calculates the similarities with all the other cases in the database and saves the maximum value of similarity into the database. This maximum value could be later used as the priority index of the cases that needs to be solved, since the case containing the lowest similarity value is different from the cases of the case base and might need urgent attention from an expert for review. The third module is a web application that is designed for the expert to analyze the cases.

3.1 Feature Extraction and Case Formulation

The collected measurements from the pulse oximeter are processed to extract features. Features are extracted from time domain, frequency domain and time-frequency domain. In the time domain, statistical features like *maximum*, *minimum*, *arithmetic mean* and *standard deviation* of data were considered. Besides the time domain, frequency domain was also considered for feature extraction as it has been observed in previous works a distinction between healthy and diseased people's data are visible in the frequency domain [2][14].

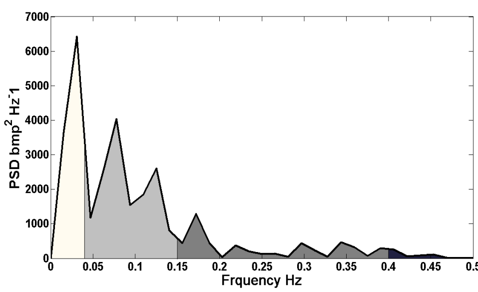


Fig. 2. PSD plot of pulse rate, LF and HF region is marked by gray and dark gray.

In order to obtain the frequency domain features, a Power Spectral Density (PSD) was calculated. The PSD is the squared amplitude of the Discrete Fourier Transform (DFT) which is achieved by using Fast Fourier Transform (FFT) algorithm on the process measurements. Here, the measurements are processed through a scaling procedure to sample the measurements in 1Hz frequency range. From the power spectral density *Low frequency power*, *High frequency power*, *Low frequency power to High frequency power ratio*, *Low frequency peak power spectral density and High frequency peak power spectral density* were calculated [2]. Frequencies between 0.04 Hz and 0.15 were considered as low frequency and frequencies between 0.15 and 0.4 were considered as high frequency as shown in Fig 2 [2]. The power in high and low frequency regions were calculated by numerical integration of the power spectral density of the corresponding frequency range. The unit of the power spectrum density and power for the pulse rate were BPM^2 (beats per minute) Hz^{-1} and BPM^2 respectively. Similarly, the frequency domain features for the oxygen saturation were calculated, with the unit of the power spectrum density and the power of $(\%)^2$ Hz^{-1} and $(\%)^2$ respectively.

A wavelet transformation method is used to extract features from the biomedical signals [15]. Unlike the Fourier transform, the wavelet transform can keep information of both time and frequency, and as a result, features extracted from wavelet transform are considered as time-frequency domain features. Statistical features *maximum*, *minimum*, *arithmetic mean* and *standard deviation*s were calculated from the approximation coefficient of wavelet decomposition of level 1. The function 'Daubechies 2' was used as the mother wavelet. Symmetric padding was used to make the data samples power of two to implement discrete wavelet transform.

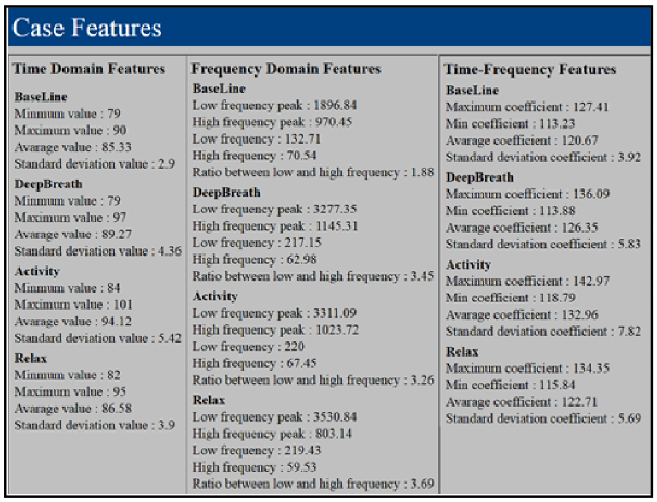


Fig. 3. An example of a case with extracted features from pulse rate.

To formulate a case, all features from the three domains in four sessions plus the subjects' contextual information (age, weight, gender, and blood pressure) were taken. Blood pressure was measured twice, once before and once after the end of taking measurements from pulse oximeter. As a result, each case contains total 59 (4 × 13 + 7) features. Features of the cases are illustrated in the Fig 3. Similarly, features are extracted for oxygen saturation considering three domains as presented above.

3.2 Case Retrieval and Monitoring

Weighting of features is an important task for retrieving similar cases. To determine the degree of importance of each feature, expert knowledge is needed. While feature weighting could be done automatically using some artificial intelligence techniques, t automatic weighting of the features tend not to be preferred in the medical domain. Moreover, in order to perform automatic weighting of the features, the algorithm needs a high volume of cases with their corresponding classification. In this work, the weights of the features are obtained from the domain expert. These weights are then used to retrieve similar cases, illustrated in Fig 4. Similarity of a feature value between two cases was measured using the normalized Euclidean distance between the feature values of two cases. Non numeric features such as gender is converted to numeric value by substituting the contextual value with a numeric one (1 for male, 0 for female).

Weights for general features

IDs	Feature Names	Weights
age	Age of the subject	8
weight	Weight of the subject	7
gender	Gender of the subject	7
bp_before	Blood pressure before the test	9
bp_after	Blood pressure after the test	9
th	threshold	50
nuc	Number of cases	15

Weights for sessions

IDs	Feature Names	Weights
base	BaseLine	6
deepB	DeepBreath	8
activity	Activity	10
relax	Relax	8

Weights for domains

IDs	Feature Names	Weights
tm	time	0
fr	frequency	10
tmfr	time and frequency	0

Weights for time domain features

IDs	Feature Names	Weights
min	Minimum value	7
max	Maximum value	10
mean	Average value	8
std	Standard deviation value	9

Weights for frequency domain features

IDs	Feature Names	Weights
lf_peak	Low frequency peak	5
hf_peak	High frequency peak	5
lf	Low frequency	8
hf	High frequency	8
lf_by_hf	Ratio between low and high frequency	10

Weights for time-frequency domain features

IDs	Feature Names	Weights
max	Maximum coefficient	8
min	Minimum coefficient	7
mean	Average coefficient	8
std	Standard deviation coefficient	8

Fig. 4. Local weights for extracted features defined by domain expert.

$$sim(T_i, S_i) = 1 - \frac{abs(T_i - S_i)}{\max\{T_i, Max(i)\} - \min\{T_i, Min(i)\}} \quad (1)$$

The function to calculate the similarity of a feature between two cases is listed in equation 1, where T_i and S_i are feature value of target and source case respectively; *Max(i)* and *Min(i)* represents the *Maximum* and *Minimum* value of the feature *i* obtained from whole case library; then "max" and "min" function compare the values between the new case feature T_i and *Maximum* and *Minimum* values obtained from the case library. The function returns 1 if the values are the same and returns 0 if the values are dissimilar. This is known as a local similarity function.

$$sim(T, S) = \frac{\sum_{i}^{n} w_i \times sim(T_i, S_i)}{\sum_{i}^{n} w_i} \quad (2)$$

Similarity between two cases is then measured using the weighted average of all the features that are to be considered. The function for calculating similarity between two cases is presented in equation 2, Where W_i is the weight of the feature *i*. Note that, currently the CBR system functions in parallel both for the cases of pulse rate and oxygen saturation in the retrieval since there is no domain knowledge are defined to combined them. In order to achieve a combine similarity value both considering pulse rate and oxygen saturation a further study is needed.

4 Results and Evaluation

The CDSS retrieves the most similar cases considering two categories 1) most similar cases retrieved from the cases within the subject and 2) from the whole case library as illustrated in Fig 5 and Fig 6.

Most similar cases retrieved from within the subject

case_id	Information	Blood pressure	Similarity	Details
1	John Doe: Male; 28 yrs; 63 kg	Before :120/110; after : 120/110	100%	details
2	John Doe: Male; 28 yrs; 63 kg	Before :120/100; after : 120/100	95.67%	details
28	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/80	94.23%	details
9	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	93.45%	details
3	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	92.77%	details
27	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/80	90.89%	details
12	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	90.12%	details
11	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	88.67%	details

Fig. 5. The system retrieves *most similar cases within the subjects*.

Most similar cases retrieved from whole library

case_id	Information	Blood pressure	Similarity	Details
1	John Doe: Male; 28 yrs; 63 kg	Before :120/110; after : 120/110	100%	details
2	John Doe: Male; 28 yrs; 63 kg	Before :120/100; after : 120/100	95.67%	details
28	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/80	94.23%	details
9	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	93.45%	details
3	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	92.77%	details
4	masum: Male; 26 yrs; 62 kg	Before :110/80; after : 110/80	91.45%	details
5	masum: Male; 26 yrs; 62 kg	Before :120/80; after : 110/85	91.13%	details
27	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/80	90.89%	details
12	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	90.12%	details
8	amin: Male; 27 yrs; 70 kg	Before :120/80; after : 120/90	89.56%	details
10	saif: Male; 28 yrs; 85 kg	Before :120/80; after : 120/90	88.89%	details
11	John Doe: Male; 28 yrs; 63 kg	Before :120/80; after : 120/90	88.67%	details
13	subash: Male; 30 yrs; 70 kg	Before :120/80; after : 120/80	86.54%	details
14	ewa: Female; 57 yrs; 75 kg	Before :169/100; after : 169/100	85.67%	details

Fig. 6. The system retrieves *most similar cases from the case library*.

The CDSS using *case-based retrieval* is verified by implementation as a prototype where all the implemented methods are compared according to their outcome, that is, in terms of the technical point of view. According to Watson [22], these trials have been conducted through the following 4 tests: 1) *retrieval accuracy* 2) *retrieval consistency* 3) *case duplication* and 4) *global test*. For the test *retrieval accuracy*, a "leave-one-out" retrieval technique is used i.e. one case is taken from the case library as a query case and then the system retrieves the most similar cases. Among the retrieved cases, the query case is also retrieved as the top similar case with the similarity value 1.0. That is, the similarity value of two same cases is computed as 100% match presented in Fig 5 and Fig 6.

Note that, the solution parts of the cases are not considered for the evaluation purpose, since the hypothesis is "correct identification of a subject". That means, the CBR system can retrieve most of the similar cases of a subject when the query case belongs to that subject. To test the *retrieval consistency*, the same query is used to perform more than one similar search and if it has been found that the same stored cases have been retrieved with the same similarity, then the implemented retrieval function is considered to have consistency. It is also observed that no cases are identical during retrieval except the query case, when it matches itself, thus *case duplication* is also checked. Regarding the *global test*, we are awaiting for expert classification and their involvement which is ongoing in this study.

5 Related Work

Decision Support Systems (DSS) that bear similarities with human reasoning have benefits and are often easily accepted by physicians in the medical domain [3, 4, 5, 6, 7 and 13]. CBR in health sciences that show recent advancement and development can be found in [18][19]. Some of the recent medical DSSs using the CBR approach are presented below: a) ExpressionCBR [8], the system is a decision support system for *cancer diagnosis* and *classification*. b) GerAmi [4] 'Geriatric Ambient Intelligence', is an intelligent system that aims to *support healthcare facilities for the elderly*, *Alzheimer's patients and people with other disabilities*. c) geneCBR [9], is focusing on *classification of cancer*, based on a gene expression profile of microarray data. d) Marling et al. in [10], described a case-based DSS to assist in the daily management of *patients with Type 1 diabetes on insulin pump therapy*. In adjusting patient-specific insulin dosage the system considers a real-time monitor of the patients' blood glucose levels and their lifestyle factors e) the stress management, CBR is used in order to diagnosis stress [3], [11] and thereafter for biofeedback treatment [12]. All the systems mentioned above use CBR, however most of those systems mainly perform classification tasks and only a few of them can be used health monitoring. Some of those systems use similar protocols during collecting data whereas they differ in time duration of the sessions and tasks done by the subjects.

6 Conclusion

This paper presents a case-based patient identification system where the parameters taken through physiological sensors called "$WristOx_2$". Here, the proposed system combines a personalized health profiling protocol with Case-Based Reasoning (CBR) approach. The personalized health profiling helps to established individual health parameters. The overall system was designed and developed in three-tier client server-based approach. Measurements are collected through an android application and a case library is implemented in remote server. A web-enabled CDSS has been provide which can be accessed via specific clients e.g., physician at a health clinic in order to monitor health. Three domain features have been used. The initial result shows that the system can find similar cases for similar subjects.

Reference

[1] Aamodt A. and Plaza E., Case-based reasoning: Foundational issues, methodological variations, and system approaches, AICom - Artificial Intelligence Communications, IOS Press, Volume 7, Issue 1, pp 39-59, 1994.

[2] K. Dingli, T. Assimakopoulos, P.K. Wraith, I. Fietze, C. Witt, N.J. Douglas, Spectral oscillations of RR intervals in sleep apnoea/hypopnoea European Respiratory Journal 2003, pp 4

[3] Begum S., Ahmed M.U., Funk P., Xiong N. and Schéele B.V, A case-based decision support system for individual stress diagnosis using fuzzy similarity matching, Computational Intelligence, Blackwell Publishing, Volume 25, Nr 3, pp 180-195, 2009.

[4] Corchado J.M., Bajo J. and Abraham A., GERAmI: Improving healthcare delivery in geriatric residences, IEEE Intelligent Systems, Volume 23, Issue 2, pp 19-25. ISSN: 1541-1672, 2008.

[5] Marling C. and Whitehouse P., Case-based reasoning in the care of Alzheimer's disease patients, 4th International Conference on Case-Based Reasoning, ICCBR 01, pp 702-715, 2001.

[6] Montani S., Portinale L., Leonardi G. and Bellazzi R., Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Medicine, Volume 37, Issue 1, pp 31-42, 2006.

[7] Bichindaritz I., Prototypical Case Mining from Medical Literature, In Applied Intelligence, vol. 28, Issue 3, 2007, pp. 222-237.

[8] De Paz J.F., Rodriguez S., Bajo J. and Corchao J.M., Case-based reasoning as a decision support system for cancer diagnosis: A case study, International Journal of Hybrid Intelligent Systems (IJHIS), IOS press, Volume 6, Issue 2, pp 97-110, 2009.

[9] Díaz F., Fdez-Riverola F. and Corchado J.M., GENE-CBR: A case-based reasoning tool for cancer diagnosis using microarray data sets, Computational Intelligence, Volume 22, Issue 3-4, pp 254-268, 2006.

[10] Marling C., Wiley M., Cooper T., Bunescu R., Shubrook J. and Schwartz F., The 4 diabetes support system: A case study in CBR research and development, International Conference on Case-Based Reasoning, ICCBR'11, pp 137-150, 2011.
[11] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, Case-based Reasoning for Diagnosis of Stress using Enhanced Cosine and Fuzzy Similarity. International journal of Transactions on Case-Based Reasoning on Multimedia Data, vol 1, Number 1, IBaI Publishing, ISSN: 1864-9734, 2008, pp 3-19.
[12] Ahmed M.U., Begum S., Funk P., Xiong N., von Schéele B, A Multi-Module Case Based Biofeedback System for Stress Treatment. International journal of Artificial Intelligence in Medicine, vol.51, Issue 2, Publisher: Elsevier B.V., 2010, pp 107-115.
[13] Ahmed M.U., Funk P., A Computer Aided System for Post-operative Pain Treatment Combining Knowledge Discovery and Case-Based Reasoning, Accepted in the 20th International Conference on Case-Based Reasoning, ICCBR'12.
[14] Zamarrón C, Gude F, Barcala J, Rodriguez J. R. and Romero P. V., Utility of Oxygen Saturation and Heart Rate Spectral Analysis Obtained From Pulse Oximetric Recordings in the Diagnosis of Sleep Apnea Syndrome; Chest 2003; pp 6-7
[15] Cvetkovic D., Übeyli E. D., Cosic I., Wavelet transform feature extraction from human PPG, ECG, and EEG signal responses to ELF PEMF exposures: A pilot study, Digital Signal Processing, 2008.
[16] Jin Z., Oresko J., Huang S., and Cheng A. C., HeartToGo: A personalized medicien technology for cardiovascular disease prevention and detection, in Proc. IEEE/NIH LiSSA, 2009, pp. 80–83.
[17] Krupa N., Ali M. M., Zahedi E., Ahmed S., Hassan F.M, Antepartum fetal heart rate feature extraction and classification using empirical mode decomposition and support vector machine, BioMedical Engineering OnLine, Vol 10, 2011.
[18] Begum S., Ahmed M.U., Funk P., Xiong N., Folke M., Case-Based Reasoning Systems in the Health Sciences: A Survey on Recent Trends and Developments, International journal of IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol 41, Issue 4, 2011, pp 421 - 434.
[19] Bichindaritz I. and Marling C., Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine, Volume 36, Issue 2, pp 127-135, 2006.
[20] Koshmak G, An android based monitoring and alarm system for patients with chronic obtrusive disease, M.Sc thesis, Department of Technology at Örebro University.
[21] Begum S., Ahmed M.U., Funk P., Xiong N. and Scheele B. V., Using calibration and fuzzification of cases for improved diagnosis and treatment of stress, The Proceedings of the 8th European Workshop on Case-based Reasoning, pp 113-122, 2006.
[22] Watson, I. 1997. Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann Publishers Inc, 340 Pine St, 6th floor, San Fransisco, CA 94104, USA.

Multi-Scale Entropy Analysis and Case-Based Reasoning to Classify Physiological Sensor Signals

Shahina Begum, Mobyen Uddin Ahmed and Shaibal Barua

School of Innovation, Design and Engineering,
Mälardalen University, PO Box 883 SE-721 23, Västerås, Sweden
Shahina.begum@mdh.se, mobyen.ahmed@mdh.se and
sba10001@student.mdh.se

Abstract. Sensor signal fusion is becoming increasingly important in many areas including medical diagnosis and classification. Today, clinicians/experts often do the diagnosis of stress, sleepiness and tiredness based on several physiological sensor signals. Since there are large individual variations analyzing data from a single sensor could mislead the diagnosis result. Therefore, this paper presents a Case-Based Reasoning (CBR) system that classifies 'healthy' and 'stressed' persons based on the fusion of multiple sensor data. In this paper several sensor measurements e.g., Heart Rate (HR), Inter-beat-Interval (IBI), Finger Temperature (FT), Skin Conductance (SC) and Respiration Rate (RR) have been combined using Multivariate Multiscale Entropy Analysis (MMSE) algorithm for the data level fusion. MMSE algorithm supports complexity analysis of multivariate biological recordings. In the proposed system, MMSE has been applied to formulate cases in the case-based classification.

1 Introduction

Today, it is known that psychophysiological parameters i.e., stress, tiredness, sleepiness and fatigue can be diagnosed using several physiological sensor signals such as Electrocardiogram (ECG), Electroencephalogram (EEG), Electrooculogram (EOG), FT etc. However, responses to these parameters are different for different persons and individual variations make it difficult to analyze and use it in a computer-based system. Data that are collected from multiple sensor sources can provide us more reliable and efficient diagnosis of these psychophysiological parameters. As human beings, we have a natural ability to fuse signals that are coming from different sources and supports in reliable and feature-rich judgment. In fact, this is what experts are doing while diagnosing these psychophysiological parameters based on multiple sensor signals.

Sensor signal fusion is a method that gives us the resulting information while using multiple sensors. According to [16], "Sensor fusion is the combining of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually." Multi-sensor information fusion is the process of integrating data or information from multiple sensors to improve quality and accuracy of the information, which cannot be obtained using the sources individually. Since the parameters obtained from physiological sensors could vary because of individual's age, gender, physical conditions etc. analyzing data from a single sensor could mislead the diagnosis result. The main advantage of using data/information from all available sources is that it helps to enhance the diagnostic visibility, increases diagnostic reliability and reduces the number of diagnostic false alarms.

In this paper, we have proposed a CBR system that helps to diagnose 'stressed' and 'healthy' subjects based on several sensor signals. For the study, five sensor measurements (i.e., HR, IBI, FT, SC and RR) have been combined in the data level fusion. Multivariate Multiscale Entropy Analysis (MMSE) [1][2] algorithm has been applied to combine these signals and extract features from the signals. This algorithm supports complexity analysis of multivariate physical and biological recordings.

The rest of the paper is organized as follows: section 2 presents the background of the sensor fusion and entropy analysis. Section 3 discusses the related work on CBR and information fusion, and feature extraction using entropy analysis. Section 4 presents the feature extraction and case formulation using MMSE algorithm. The case retrieval algorithms are explained in section 4 and the result and evaluation are described in section 5. Finally, the paper ends with section 6 where a summary of the paper and its contributions are presented.

2 Background of Sensor Fusion and Entropy Analysis

Information fusion can be achieved usually by combining or integrating information from multiple sources or information from the same source over time. Three fusion levels are defined to combine different sensor signals [10],

- Data level fusion: it combines (unprocessed) sensor data.
- Feature level fusion: it combines the features that are extracted from different sensors.
- Decision level fusion: it combines findings (or detection probabilities) of different sensors.

In the proposed system, for the data level fusion, we have used MMSE analysis which is derived from the multi-scale entropy analysis. The MMSE supports entropy estimation of multivariate/channel data. Previously, traditional entropy methods were applied on physiological or biological sensor signals however all of these methods support entropy estimation only on univariate/channel data. The traditional entropy-based algorithms are used to quantify regularity of time series on a single scale, such algorithms are Shannon entropy, Kolmogorov-Sinai (KS) entropy, Approximate entropy and Sample entropy. Pincus has developed the Approximate Entropy (ApEn) which has widely been used in clinical cardiovascular studies [7]. However, Approximate Entropy has two drawbacks; first, it's intensively dependent on recording length and second, if ApEn is measured on two datasets and for instance if the result of one dataset is higher than the other dataset in that case for all test conditions the expected result should be higher but in such cases the algorithm fails to provide the expected (i.e., higher) result. To overcome the problems Richman et al. [13] have introduced the Sample Entropy (SampEn) Algorithm. They have found that SampEn satisfies entropy theory more accurately than ApEn. The SampEn is the basis of the MMSE algorithm.

The basic idea is that entropy increases with the increase of disorder in a system and for a completely random system the entropy reaches it's maximum possible value. Nevertheless, increase in entropy does not necessarily always associate with the increase in dynamical complexity [8]. Researchers observed that traditional single scale entropy estimate tends to yield lower entropy in time-series data than for their surrogate series data. Shuffling the original data can form the surrogate series data. The shuffled data are more irregular and less predictable than the original data and correlation commonly encompasses several time scales. Thus, the surrogate data generating process destroys the correlation and degrades the information content of the original time series data. Costa et al. [8] have introduced a new method Multiscale Entropy Analysis (MSE) which by discovering the dependence of entropy estimates on multiscale shows that original time series are more complex than their surrogates. In their study they have found that MSE strongly separates healthy and diseased groups. They have applied MSE to analyze heartbeat intervals time-series on three groups of subjects: healthy subjects, patients with severe congestive heart failure and patients with the cardiac arrhythmia and atrial fibrillation. The results show that the system has distinguished all the three groups.

3 Related Work

CBR and knowledge discovery model for the information fusion has been described by Azuaje et al. in [3]. In the paper, they have discussed three data fusion methods based on signal data and database records from the heart disease risk estimation domain. In this paper, they have presented three fusion models where two models fuse information at the information retrieval-outcome level and the other combines data at the discovery input level. They have compared the fusion model with the single source model and from their study it has been shown that fusion of information at the retrieval outcome level outperforms the single source information. Policastro et al. [12] have introduced a hybrid CBR system that applies three Machine Learning (MI) algorithms for the sensor signal fusion. A comparison with the individual machine learning techniques showed that the hybrid CBR improves the accuracy of the system. In addition, application of CBR in the health sciences is discussed in several other studies [5][7][11].

Besides, entropy estimation has been applied in many clinical studies for feature extraction. These researches include biological signal processing, breast cancer diagnosis, EEG signal analysis of sleep stage etc. Costa et al. in [9] have exploited Multiscale Entropy Analysis (MSE) for biological sensor signals for example to investigate the heartbeat intervals and later to the analysis of coding and non-coding of DNA sequences. In the first case for the clinical classification, the slopes for small and large time series are the two features extracted from the MSE curves. The MSE applied on scale factor 20 and the first feature is the slope of the curves identified by values of sample entropy within the scale factor 1 and 5 and the second feature is extracted in the same way between the scale factor 6 and 20. Singh and Singh in their paper [14] have also used entropy measures for texture feature extraction in the digitized mammograms. Wang et al. [15] in an experimental study on rats have shown approximate entropy (ApEn) can be used to extract features from the EEG signals. In the study, approximate entropy combining bispectrum analyses have been applied to extract features from the single sleeping EEG signals of rats. Traditional methods such as time-frequency analysis, frequency domain analysis and wavelet analysis have widely been used for feature extraction and classification of sleeping stages. However, these methods are not easy enough to extract features effectively and to separate/categorize into distinctive sleeping stages in real time using the sleeping EEG signals. The results of the study have provided a new way of feature extraction from non-stationary signals and from the analysis of ApEn it is also showed that the features of bispectrum are so effective to analyze and diagnose brain diseases and for intelligent recognition. The two papers [18][19] have discussed about feature extraction from EEG signal using entropy analysis. Also, it has been found that entropy during the resting state is higher than the entropy during the different cognitive tasks [19]. Winter et al. in [17] have proposed a new algorithm for feature extraction based on selection of a given range of scales. They have used multiscale analysis on aerial images to extract objects that appear within the given scales. The results of the entropic scalar detector of their study illustrate that in the image analysis 'scale' is a relevant detection parameter for some objects.

4 Feature Extraction and Case Formulation using MMSE Algorithm

In the proposed system, MMSE algorithm is applied on five sensor signal measurements (i.e., HR, IBI, FT, SC and RR) to quantify complexity of the sensor signals. A schematic diagram of the feature extraction and case formulation using MMSE is shown in Fig 1.

In the MMSE analysis, the first step is to define temporal scales by averaging p-channel time series using the coarse graining method and then in the second step, evaluate multivariate sample entropy (MSampEn) for each coarse grained multivariate data. To calculate MSampEn the algorithm forms a composite delay vector from the coarse grained data and it's two important embedding parameters are m_k and τ_k. A detailed description of the MMSE algorithm is available in [1][2].

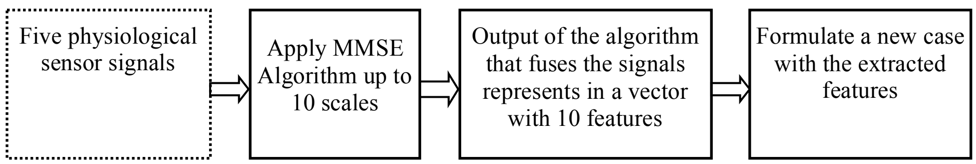


Fig. 1. Steps while features are extracted and a new case is formulated.

In this study for MMSE analysis, we consider the embedding vector parameters, m_k=2 and τ_k=1. The coarse-grained process is obtained by the following equation (1)

$$b_{k,j}^{\varepsilon} = \frac{1}{\varepsilon}\sum_{i=(j-1)\varepsilon+1}^{j\varepsilon} a_{k,i} \ where\ 1 \le j \le \frac{N}{\varepsilon} \quad (1)$$

Where, N is number of data points in every channel, $\{a_{k,i}\}_{i=1}^{N} k = 1,2 \ldots \ldots p$, is a p-varieties time series, $\boldsymbol{\varepsilon}$ is the scale factor, $k = 1, \ldots, p$ is the channel index and $b_{k,j}^{\varepsilon}$ is the coarse-grained data. The MMSE analysis returns a linear vector based on the scale factor. The scale factor is highly dependent on the length of data. However, the MMSE estimates are consistence for data length N ≥ 300 [1]. Here, for the croase-grained process the scale factor is considered 10 because of at the scale 10 the shortest data series has greater than 300 data points. An example of the coarse-grained process up to scale factor 2 is shown in Fig. 2. As we measure MMSE for scale factor up to 10 it returns a vector of length 10 as a result of MMSE estimation for each recording. Thus, these 10 values are considered as the features for each recording.

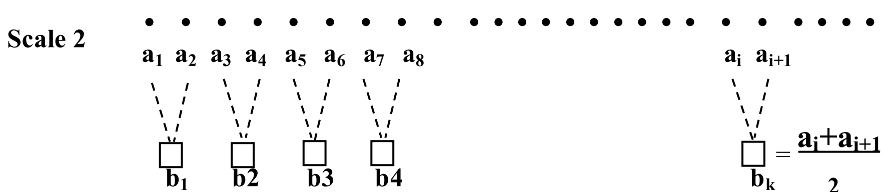


Fig. 2. Illustration of coarse-grained process in MMSE for scale factor 2

Fig. 3 shows the average of MMSE analysis for 12 healthy and 6 stressed cases (classified by an expert). As can be seen from the figure, at a specific point (approximately after the scale factor 7) stressed cases possess a lower complexity value than the healthy cases. Afterwards, this difference (i.e., lower complexity value for stressed cases) remains constant for the rest of the scale factors. However, it is not always true for individual recordings. Fig. 4 and Fig. 5 show the MMSE analysis results for the 12 healthy and 6 stressed cases respectively. From Fig. 4 and Fig. 5 it is shown that MMSE varies a lot depending on individuals.

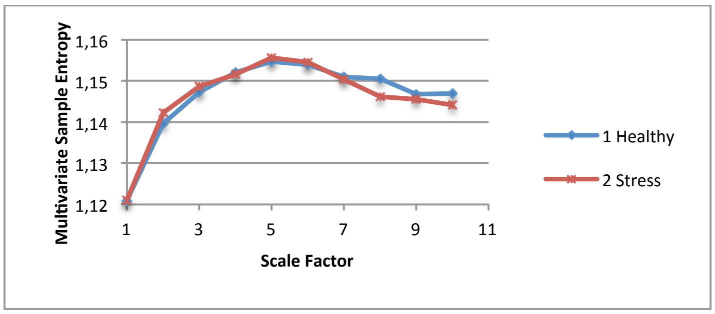


Fig. 3. Average of MMSE analysis for 12 healthy and 6 stressed cases

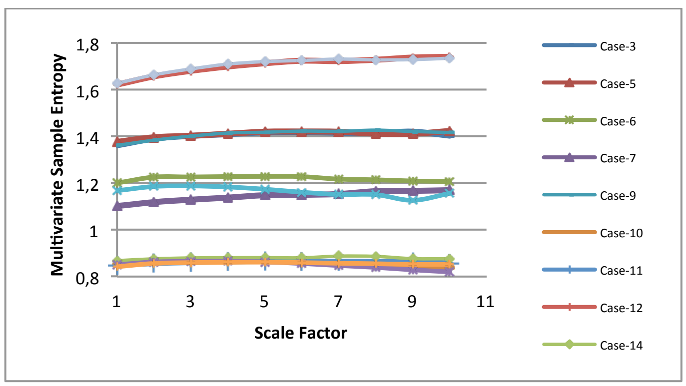


Fig. 4. MMSE analysis for 12 healthy cases

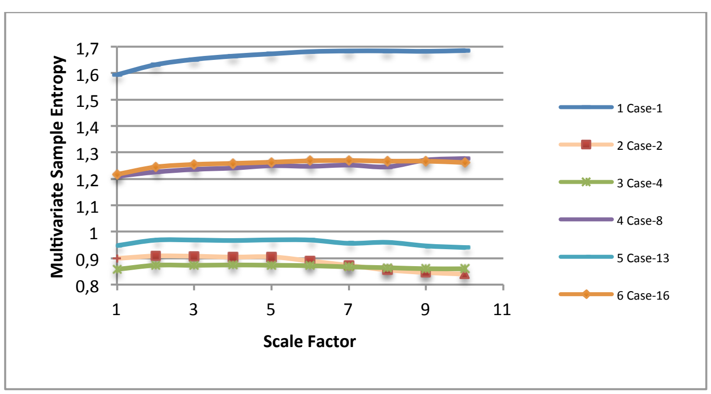


Fig. 5. MMSE analysis for 6 stressed cases

In the CBR system, each case consists of 10 features extracted from the 15 minutes signal recordings. An expert classifies all the reference cases. The 15 minutes recoding contains six sessions data which are baseline, beep breathing, verbal stress, relax, math stress and relax [6]. As the scale factors in MMSE depends on the length of data set (changes become significant for a large datasets) we have considered full 15 minutes recording instead of taking session-by-session data for a reliable estimation.

5 Case Retrieval

In the proposed system, the similarity of a feature between two cases was measured using *modified Euclidean distance* and *fuzzy similarity*. The *modified Euclidean distance* function to calculate similarity of a feature between two cases is shown in equation 2,

$$\mathrm{sim}(T_i, S_i) = 1 - \frac{abs(T_i - S_i)}{\max(i) - \min(i)} \quad (2)$$

Where, T_i and S_i are feature values of the target and source cases respectively. The function returns 1 if the values are same and returns 0 if the values are completely dissimilar.

In *fuzzy similarity*, a triangular membership function (*mf*) replaces a crisp value of the features for new and old cases with a membership grade of 1. In both the cases, the width of the membership function is fuzzified by 50% in each side. Fuzzy intersection is employed between the two fuzzy sets to get a new fuzzy set which represents the overlapping area between them.

$$sim(C_f, S_f) = s_f(m1, m2) = \max(om/m1, om/m2) \quad (3)$$

The similarity between feature values of the old case (S_f) and the new case (C_f) is now calculated using equation 3 where *m1*, *m2* and *om* is the area of each fuzzy set. [4]

The similarity between two cases is measured using the average of all the features that are to be considered. The function for calculating the similarity between two cases is presented in equation 4.

$$\mathrm{sim}(T,S) = \sum_{i}^{n} sim(T_i, S_i) \quad (4)$$

Where T is a current/target case, S is a stored case in the case base, n is the number of the attributes/features in each case, i is the index for an individual attribute/feature and *sim* (T_i, S_i) is the local similarity function for attribute i in cases T and S. It is noted that here for all the extracted features weight values are equal.

6 Evaluation

An experimental work has been carried out with 19 cases where 13 of them are classified as 'healthy' and 6 of them are classified as 'stressed' by an expert. So, the cases are categorized into two classes, -1 as 'Stressed' and 1 as 'Healthy'.

Table1. Sensor signal classification accuracy using *modified Euclidian distance* function

Case id	Expert	System	Accuracy value
1	Stressed	Healthy	0
2	Stressed	Stressed	1
3	Healthy	Healthy	1
4	Stressed	Healthy	0
5	Healthy	Healthy	1
6	Healthy	Stressed	0
7	Healthy	Healthy	1
8	Stressed	Stressed	1
9	Healthy	Healthy	1
10	Healthy	Healthy	1
11	Healthy	Healthy	1
12	Healthy	Stressed	0
13	Stressed	Stressed	1
14	Healthy	Stressed	0
15	Healthy	Healthy	1
16	Stressed	Stressed	1
17	Healthy	Healthy	1
18	Healthy	Healthy	1
19	Healthy	Healthy	1
System's classification accuracy			73.68%

Table2. Sensor signal classification accuracy using *fuzzy similarity*

Case id	Expert	System	Accuracy value
1	Stressed	Healthy	0
2	Stressed	Stressed	1
3	Healthy	Healthy	1
4	Stressed	Healthy	0
5	Healthy	Healthy	1
6	Healthy	Stressed	0
7	Healthy	Healthy	1
8	Stressed	Stressed	1
9	Healthy	Healthy	1
10	Healthy	Healthy	1
11	Healthy	Healthy	1
12	Healthy	Stressed	1
13	Stressed	Stressed	1
14	Healthy	Stressed	0
15	Healthy	Healthy	1
16	Stressed	Stressed	1
17	Healthy	Healthy	1
18	Healthy	Healthy	1
19	Healthy	Healthy	1
System's classification accuracy			78.95%

The weight values for all the extracted features are set to 1. Two similarity functions *Euclidian distance* and *Fuzzy similarity* functions are applied for the case matching. Here, the system's performance in terms of accuracy has been compared with the expert's classification. If the system's classification matches with the expert's classification than it is counted as 1 else 0. Table 1 shows the results based on *Euclidian function* where achieved accuracy is 73.68% and Table 2 illustrates the outcomes considering the *fuzzy similarity* function where the accuracy reaches to 78.95%.

7 Summary

In this paper, a hybrid CBR system is proposed to classify 'healthy' and 'stressed' patients based on the fusion of multiple sensor data. Here, MMSE technique is used for data fusion and CBR for classifying stressed patients. In the proposed system, the features extracted from the entropy estimation are used to formulate cases in the CBR system. An evaluation of the system using the retrieval algorithms i.e., *Euclidian and Fuzzy similarity matching algorithms* is also presented in this paper. The preliminary result shows that the obtained classification accuracy compare to an expert is approximately 79% for the 19 subjects participated into the study. However, in future the system could be evaluated on more subjects to investigate the classification accuracy. In the initial stage, all the features weight values are set to 1 since there is no prior knowledge of these features. Nevertheless, there is a possibility to get increased classification accuracy by updating the weight values of the features for example weight values can be learned from the case library.

References

[1] Ahmed M. U. and Mandic D.P., Multivariate Multiscale Entropy Analysis, IEEE Signal Processing Letters, Volume 19, Nr 2, pp.91-94, 2012

[2] Ahmed M. U. and Mandic D. P., Multivariate multiscale entropy: A tool for complexity analysis of multichannel data, Physical Review E, Volume 84, Nr 6, pp 061918-1 – 061918-10, 2011

[3] Azuaje F, Dubitzky W, Black N, Adamson K. Improving clinical decision support through case-based data fusion, IEEE Transactions on Biomedical Engineering, Volume 46, Nr.10, pp.1181-1185, 1999.

[4] Begum S., Ahmed M.U., Funk P., Xiong N. and Schéele B.V, A case-based decision support system for individual stress diagnosis using fuzzy similarity matching, Computational Intelligence, Blackwell Publishing, Volume 25, Nr 3, pp 180-195, 2009.

[5] Begum S., Ahmed M.U., Funk P., Xiong N. and Folke M., Case-Based Reasoning Systems in the Health Sciences: A Survey on Recent Trends and Developments,

International journal of IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews, Volume 41, Issue 4, 2011, pp 421 - 434.
[6] Begum S., Ahmed M.U., Funk P., Xiong N. and Scheele B. V., Using calibration and fuzzification of cases for improved diagnosis and treatment of stress, The Proceedings of the 8th European Workshop on Case-based Reasoning, pp 113-122, 2006.
[7] Bichindaritz I. and Marling C., Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine, Volume 36, Issue 2, pp 127-135, 2006.
[8] Costa M., Goldberger A.L., Peng C.K., Multiscale entropy analysis of complex physiologic time series. Physical Review Letters, Volume 89, Nr 6, 2002.
[9] Costa M., Goldberger A.L., Peng C.K., Multiscale entropy analysis of biological signals. Physical Review Letters, Volume 71, Nr 2, 2005.
[10] Cremer F., Schutte K., Schavemaker J. G. M., and Breejen E. A Comparison of decision-level sensor-fusion methods for anti-personnel landmine detection, Information Fusion, Volume 2, Issue 3, pp 187-208, 2001.
[11] Montani S., Portinale L., Leonardi G. and Bellazzi R., Case-based retrieval to support the treatment of end stage renal failure patients, Artificial Intelligence in Medicine, Volume 37, Issue 1, pp 31-42, 2006.
[12] Policastro C.A. and Carvalho A.C.P., Applying Case Based Based Reasoning to Sensor Fusion. Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on, pp.419-424, 2007.
[13] Richman, J. S. and J. R. Moorman, Physiological time series analysis using approximate entropy and sample entropy, American Journal of Physiology- Heart and Circulatory Physiology, Volume 278, Nr 6, 2000.
[14] Singh A.P. and Singh B., Shannon and Non-Shannon Measures of Entropy for Statistical Texture Feature Extraction in Digitized Mammograms, Proceedings of the World Congress on Engineering and Computer Science, Volume 2, 2009.
[15] Wang Y. and Wang W. Liu Y., Wang D., Liu B., Shi Y. and Gao P., Feature Extracting of Weak Signal in Real-Time Sleeping EEG with Approximate Entropy and Bispectrum Analysis, Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference on, pp.1-4, 2009.
[16] Wilfried E. An Introduction to Sensor Fusion, Research Report 47/200, http://www.vmars.tuwien.ac.at/documents/intern/805/elmenreich_sensorfusionintro.pdf.
[17] Winter, A., Maitre, H., Cambou, N. and Legrand, E., Entropy and multiscale analysis: a new feature extraction algorithm for aerial images, IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume 4, pp 2765-2768, 1997.
[18] Wu L., and Neskovic P., Feature extraction for EEG classification: representing electrode outputs as a Markov stochastic process, European Symposium on Artificial Neural Networks, pp 567-572, 2007.
[19] Wu L., Neskovic P, Reyes E., Festa E., and Heindel W., Classifying n-backEEG data using entropy and mutual information features, In European Symposium on Artificial Neural Networks, 2007.

System Overview: CARE PARTNER A Case-Based Decision Support System for Evidence-Based Medicine

Isabelle Bichindaritz

State University of New York at Oswego
Oswego, NY 13126, USA
ibichind@oswego.edu

Abstract. CARE PARTNER is a case-based reasoning (CBR) decision support system for the long-term follow-up of patients having undergone stem cell transplantation in oncology. One important goal of the system was to show the clinical effectiveness of providing evidenced-based recommendations at the point of care. For this purpose, this system proposes in particular a representation of clinical guidelines in the form of prototypical cases to ensure that the CBR system would be able to reason both from experiential knowledge (cases) and theoretical evidence-based knowledge in the form of clinical guidelines (prototypical cases). Thus it allows their integration to provide case specific advice. In addition, its extensions have studied how prototypical cases have served various purposes in biomedical CBR systems, among which to organize and structure the memory, to guide the retrieval as well as the reuse of cases, to serve as bootstrapping a CBR system memory, and to maintain knowledge. The evaluation of the system shows the high accuracy of its diagnosis and treatment recommendations and their improvement over time.

1 Introduction

Case-based reasoning (CBR) systems in biomedicine rely on patients' cases to propose diagnosis and treatment recommendations based on clinical experiences. CARE PARTNER system is presented in this article as an example of CBR decision-support system in oncology proposing to foster the clinical application of evidence-based medicine at the point of care. A summary of the system's goals, design, and results is presented in the following sections.

2 CARE-PARTNER System Goals

The CARE-PARTNER system is a computerized decision-support system on the World-Wide Web (WWW). This system is an evolution of the well-known computerized decision-support systems (CDSS) [1] that supports the quality of the knowledge in its knowledge-base, and helps disseminate it to its users. In other terms, CARE-PARTNER assists its users in the performance of their clinical tasks by providing decision-support advice based upon proven and validated practice, and thus helps implement evidence-based medicine [2]. CARE-PARTNER proposes decision-support advice for diagnosis, treatment, and follow-up - using a general framework for reasoning from knowledge sources of varied quality which means that their knowledge is based upon varied evidence, but also that the evidence associated with each piece of knowledge can vary through experience [3].

This system is applied to the *long-term follow-up* (LTFU) of patients having undergone a *stem-cell transplant* (SCT) at the Fred Hutchinson Cancer Research Center (FHCRC) in Seattle, after their return in their home community [4]. It answers online the questions about care of transplant patients that home care providers used to submit by phone to LTFU nurses, who relayed to LTFU clinicians before getting back to the home care providers with clinical answers.

Another aspect of the system is that it provides an electronic contact management system to replace the phone and paper-based previous system, with obvious advantages for research purposes and documentation purposes. This electronic contact management system is an example of knowledge management system in medicine.

CARE-PARTNER decision-support provides the following medical recommendations:

- Interpretation for each laboratory test and procedure result.
- List of differential diagnoses, ranked by likelihood; these diagnoses are often not incompatible, since several diagnoses co-occur to cover all the signs and symptoms exhibited by the patient.
- List of steps of laboratory tests and/or procedures for diagnosis assessment.
- List of steps of planning actions for treatment.
- List of pertinent documents hyperlinked to the previous elements, such as guidelines, or textbook excerpts.

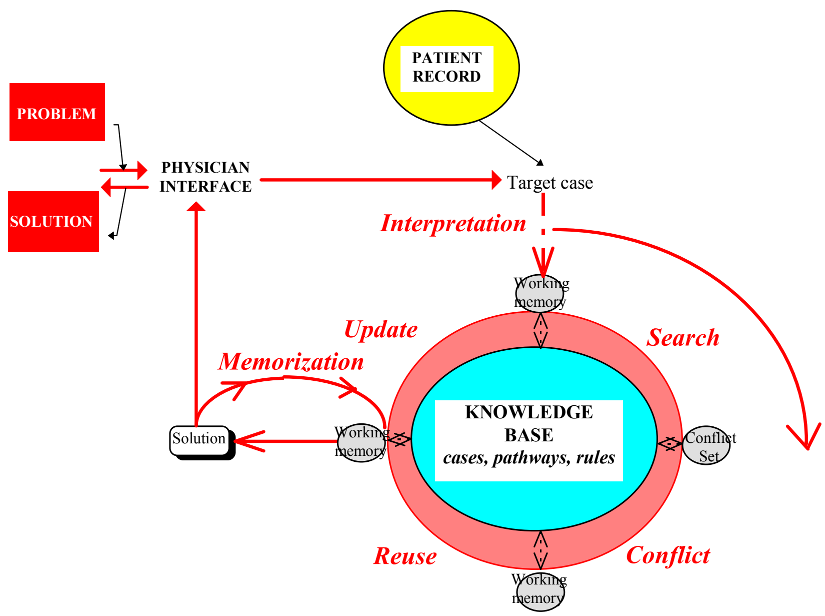
One important requirement of the system is the management of risk. A physician not specialized in a domain may not be in a position to be able to critique or challenge the system advice, and may not notice even severe mistakes. Especially in a domain as stem-cell transplantation, which is a last resort therapy for desperate cases, transplant complications are quite unfamiliar to home care providers. These complications can be rapidly life-threatening, thus imposing very high standards of safety to protect patients. Therefore the reliability of the system, as well as its safety, were of paramount importance.

3 CARE-PARTNER System Design

CARE-PARTNER reasoning cycle (see Fig. 1) is a multimodal reasoning cycle for the cooperation of case-based reasoning, rule-based reasoning, and information retrieval. Its reasoning steps are generalizations of the steps defined in these respective methodologies. The cooperation of the different knowledge sources is driven by the domain, in which, like in most medical domains, knowledge takes several forms:

1. **Practice guidelines:** a *practice guideline* is composed of "systematically developed statements designed for practitioners and patients that will be helpful in making clinical decisions on the prevention, diagnosis, treatment and management of

selected conditions". Guidelines are represented as rules, most of the time embedded in prototypical cases.
2. **Practice pathways:** a *practice pathway* (see Fig. 2) covers the same type of knowledge elements as a guideline, but specialized in the management of diagnosis and treatment related to LTFU. It has been created by a group of LTFU experts exclusively for the CARE-PARTNER system. Pathways correspond to prototypical cases, and are represented as cases in the system.
3. **Practice cases:** a *practice case* is a sample of a problem-solving situation, some being complex, solved by an expert. This problem-solving example may have needed to resort to the expertise and agreement of several experts, but it is essentially a real patient problem-solving situation, and not a prototypical one as a practice guideline or a practice pathway. It is represented as a case in the system.
4. **Medical textbooks:** a collection of documents serves as documentation and explanation during the reasoning process, often in hyperlinked form.



FFig. 1. Multimodal reasoning framework

3.1 Ontology

The system is known for its decision support results, including 94.5% of recommendations judged as clinically acceptable by the experts, figure growing over time [5]. These results come from the achievement of intensive knowledge elicitation efforts to build the case base around a knowledge base of the domain. It was determined early on in the project that cases were not available in electronic format at a level of detail required for CBR. For instance, the patients database did not comprise treatments performed on the patients, nor most of the signs and symptoms, but only main events ‘abstracted’ from the paper charts. The project team had to come up with prototypical cases to bootstrap the system, which took over two years to develop at a level of thoroughness and consistency necessary for achieving the high accuracy in overall recommendations of over 94%. This system was also unique because its proposed recommendations spanned not only diagnosis, but also lab results interpretation, and treatment planning. Table 1 presents the extent of the ontology built for this system.

Table 1. Carepartner’s ontology, showing in the CDSS column the number of objects represented in the system, in comparison with the standard medical nomenclature SNOMED v. 3.4

N	LTFU CDSS	SNOMED v. 3.4
Diseases	1109	35,834
Functions	452	19,221
Labs	1152	30,723
Procedures	547	20,105
Medications	2684	14,846
Sites	460	5,875

N	CDSS
Terms	739,439
Relations	51

N	CDSS
Patient cases	4904

3.2 Prototypical case

The cornerstone of the knowledge acquisition process has been the conception of prototypical cases, called clinical pathways in this system. The clinical pathways, 91 of them having been implemented in the test version of the system, correspond to clinical diagnostic categories for the most part, some of them corresponding also to essential signs and symptoms requiring specific assessment or treatment actions. The clinical pathways are knowledge structures represented from the ontology described above, namely: all diseases, functions (also known as signs and symptoms), labs, procedures, medications, sites, and planning actions. Most of the terms naming these objects are standardized using the Unified Medical Language System (UMLS) semantic network [6]. Only the terms not corresponding to objects in the UMLS have been added to the domain specific ontology. In particular, the planning actions used in the treatment part of a prototypical case did not exist in the UMLS and were all created for the system. Figure 1 shows some of the planning actions in the ontology.

An example of a prototypical case is provided in Figure 2. It shows that a prototypical case, mostly a diagnostic category or disease, such as here chronic graft versus host disease, which is a complication of stem-cell transplantation, comprises three parts:

1. A list of findings, corresponding to signs and symptoms.
2. A diagnosis assessment plan, which is a plan to follow for confirming (or informing) the suspected diagnosis.
3. A treatment/solution plan, which is a plan to follow for treating this disease when confirmed, or a solution when the pathway does not correspond to a disease.

The diagnosis assessment part and the treatment part can also be seen as simplified algorithms, since they use IF THEN ELSE structures, and LOOP structures, as well as SEQUENCE structures of actions in time, which, when instantiated with actual patients' data, provide a diagnosis assessment plan, or a treatment plan, tailored to a specific patient. In this way, this knowledge structure allows for sophisticated adaptation when reusing a prototypical case.

The prototypical case structure presented in this article is at the basis of the prototypical case structure proposed in MEMOIRE (an extension of CARE PARTNER) as a generic prototypical case representation structure. It is also the kind of prototypical case format used for mining from medical literature.

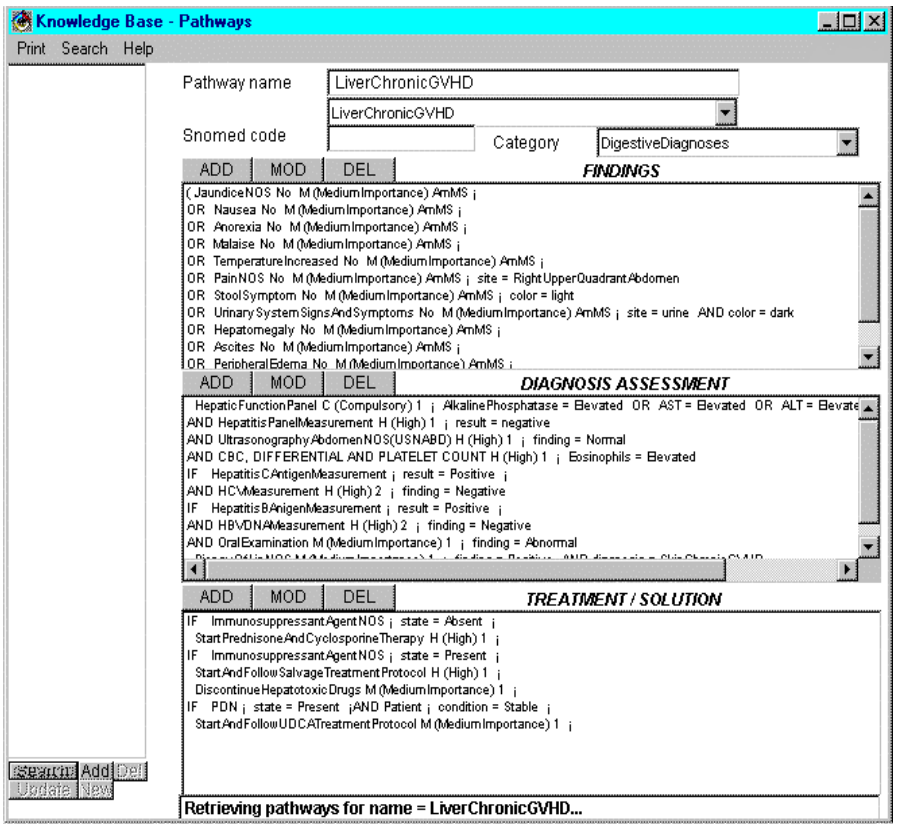


Figure 2. A clinical pathway, corresponding to a prototypical case, for chronic graft versus host disease (CGVHD)

3 CARE-PARTNER System Evaluation

A sample evaluation of CARE-PARTNER decision-support performance is provided in Table 2 [5]. It shows the rating of the system by two independent expert clinicians according to the criteria *Fails to meet standards / Adequate / Meets all standards*. We see that, on 163 different clinical situations or cases, corresponding to contacts between the system and a clinician about three patients, the system was rated 82.2% as *Meets all standards*, and 12.3% as *Adequate*, for a total of 94.5% of results judged clinically acceptable by the medical experts. Table 1 also shows that the advice provided by the system covers most of the clinicians' tasks: labs results interpretation, procedure results interpretation, diagnosis, treatment, and pathways retrieval.

Table 2. CARE-PARTNER CDSS Evaluation Form Inter-Rater Agreement and Summary Ratings for Two Raters over Three Patients

	Applicable Cases			Concordant Cases			
	Number	Percent Agreement Rating	Kappa coefficient of agreement	Number	Fails to meet standards	Adequate	Meets all standards
Labs	57	94.7	.71	54	3.7%	3.7%	92.6%
Procedures	70	95.7	.83	67	8.9%	3.0%	88.1%
Diagnosis	79	86.1	.74	68	16.2%	13.2%	70.6%
Treatment	77	92.2	.81	71	9.9%	11.3%	78.8%
Pathways	53	88.6	.71	47	8.5%	8.5%	83.0%
Overall Appreciation	178	91.6	.77	163	5.5%	12.3%	82.2%

Another part of the evaluation dealt with measuring the progress of the system when solving new contact cases. As noted earlier, case-based reasoning gives the system the ability to learn. This important characteristic of the system has been evaluated on three patients complete charts. The performance of the system has significantly improved between patient 1 and 3 to reach 98.6% for the 54 contacts in the 3rd patient case [5]. Since the episode of the Therac-25 demonstrated that even a 100% reliability was not sufficient to ensure safety [7], the system was further extended to include a safety control system capable of referring the cases requiring particular attention to direct clinician supervision.

4 Conclusion

CARE PARTNER system led to many extensions, in the MEMOIRE project, among which how to maintain the knowledge in clinical guidelines either from new clinical guidelines, from medical literature [8], or from clinical experience.

References

1. van Bemmel, J.H., Musen, M.A.: Handbook of Medical Informatics. Springer-Verlag, Berlin Heidelberg New York (1997)
2. Bichindaritz I, Kansu E, Sullivan KM, Case-Based Reasoning in CARE-PARTNER: Gathering Evidence for Evidence-Based Medical Practice, European Workshop on Case-Based Reasoning, SPRINGER-VERLAG Lectures Notes in Artificial Intelligence n° 1488 (1998) 334-345
3. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodologies Variations, and Systems Approaches. AI Communications. IOS Press, Vol. 7:1 (1994) 39-59
4. Sullivan, K.M., Siadak, M.F.: Stem Cell Transplantation. In: Johnson, F.E., Virgo, K.S., Edge, S.B., Pellegrini, C.A., Poston, G.J., Schantz, S.P., Tsukamoto, N. (eds): Cancer Patient Follow-Up. Mosby Year-Book Publications, St Louis (1997) 490-501
5. Bichindaritz, I.: Mémoire: Case-based Reasoning Meets the Semantic Web in Biology and Medicine. In: Funk, P., Gonzàlez Calero, P. (eds): Proceedings of ECCBR 2004. Lecture Notes in Artificial Intelligence, Vol. 3155. Springer-Verlag, Berlin, Heidelberg, New York (2004) 47-61
6. National Library of Medicine: The Unified Medical Language System. http://umls.nlm.nih.gov [Last access: 2005-04-01] (1995)
7. Leveson, N., Turnwe C.S.: An Invertigation of the Therac-25 accidents. IEEE Computer, 26(7), July 1993 (1993) 18:41
8. Bichindaritz, I.: Prototypical Case Mining from Biomedical Literature. Applied Intelligence 28(3) (2007) 222-237

TRUE:
Traces for Reusing Users' Experiences – Cases, Episodes, and Stories

Workshop at the
Twentieth International Conference on
Case-Based Reasoning
(ICCBR 2012)

Lyon, France
September, 2012

Michael W. Floyd, Béatrice Fuchs, Pedro Antonio González-Calero, David Leake, Santiago Ontañón, Enric Plaza, and Jonathan Rubin (Eds.)

Co-Chairs

Michael W. Floyd
Carleton University, Canada

Béatrice Fuchs
IAE-Université Jean Moulin Lyon 3, France

Pedro Antonio González-Calero
Complutense University of Madrid, Spain

David Leake
Indiana University, USA

Santiago Ontañón
Drexel University, USA

Enric Plaza
IIIA-CSIC, Spain

Jonathan Rubin
University of Auckland, New Zealand

Programme Committee

David W. Aha, Naval Research Laboratory, USA
Amélie Cordier, University Claude Bernard Lyon 1, France
Babak Esfandiari, Carleton University, Canada
Pablo Gervás, Complutense University of Madrid, Spain
Joseph Kendall-Morwick, Depauw University, USA
Luc Lamontagne, Laval University, Canada
Beatriz López, University of Girona, Spain
Alain Mille, University Claude Bernard Lyon 1, France
Mirjam Minor, University of Trier, Germany
Hector Muñoz-Avila, Lehigh University, USA
Pinar Öztürk, Norwegian University of Science and Technology, Norway
Ashwin Ram, Palo Alto Research Center (PARC), USA
Mark Riedl, Georgia Institute of Technology, USA
Thomas Roth-Berghofer, University of West London, UK
Antonio A. Sánchez-Ruiz, Complutense University of Madrid, Spain
Ben Weber, University of California - Santa Cruz, USA

Preface

There has been recent interest in artificial intelligence approaches based on *traces* of various activities. For example, an important part of users' experience interacting with computer systems may be captured in traces of their activities recording their interactions with applications, games, and other systems. Collecting traces and stories enables extracting and reusing relevant parts of past user experiences in case-based reasoning (CBR) processes. Therefore, traces constitute a knowledge container from which a CBR process may build and reuse cases. The goal of the workshop is to gather and contrast approaches to exploiting traces for various tasks and domains through CBR.

This workshop is the sequel to a first workshop on traces, organized at ICCBR 2003 in Trondheim, Norway. Since this first workshop, the theory of exploiting traces has matured, and numerous projects have been carried out. Over the past several years, papers on this topic have appeared in the ICCBR main conference, numerous ICCBR workshops and various other international conferences. User traces remain relevant to many focus areas in CBR (e.g., learning by observation/demonstration, trace-based reasoning, opponent modeling in games). The motivation for this workshop is to encourage the exchange of ideas among CBR researchers working in this area.

The workshop program includes eight papers representing various approaches to reasoning from traces and stories. Mathern, Mille, Cordier, Cram and Zarka propose combining knowledge engineering techniques with a knowledge discovery cycle for building knowledge interactively with the user and benefitting from existing knowledge. Their methodology is illustrated with several complementary tools supporting the knowledge discovery cycle. Zarka, Champin, Cordier, Egyed-Zsigmond, Lamontagne and Mille present the trace-based management system TStore which handles the storage, transformation, and reuse of modeled traces. The approach is applied to Wanaclip, a video clips composer. Fuchs and Belin present an interactive trace-based approach for experience management which allows integrating knowledge in the discovery cycle. The approach is illustrated in two application domains, musical analysis and video annotation. Thevenet, Lefevre, Cordier and Barnachon present a trace-based framework for a gesture-based interface and show how to exploit gestural interaction traces to provide assistance to users. Rubin and Ram describe an architecture for capturing and reusing users traces in the domain of computer role playing games. Traces are first captured in a training phase and a planner adapts them dynamically to reproduce behaviors in another context. The paper from Lamontagne, Rugamba and Mineau proposes acquiring cases by the segmentation of traces in sequential game environments. Traces are broken down into episodes describing states of the game board and actions taken. Cases are built by grouping consecutive correlated episodes with low conditional entropy.

Coman and Muñoz-Avila study how to generate collections of diverse stories using a case-based planning approach in the context of computer games. Finally, Ontañón, Zhu and Plaza present a story generation system that uses amalgamation operations between stories for generating new stories that combine aspects of the reused cases.

Overall, these papers represent a good sample of the recent trends in CBR approaches to reasoning from traces for reusing user experiences, and we expect the workshop discussions to further clarify and advance work in this area. We would like to thank everyone who contributed to the success of this workshop, especially the authors, the program committee members, and the organizers of the ICCBR 2012 conference.

Michael W. Floyd September 2012
Béatrice Fuchs
Pedro Antonio González-Calero
David Leake
Santiago Ontañón
Enric Plaza
Jonathan Rubin

Towards a Knowledge-Intensive and Interactive Knowledge Discovery Cycle

Benoît Mathern[1,2], Alain Mille[1,3], Amélie Cordier[1,3], Damien Cram[1,3,4], and Raafat Zarka[1,5]

[1] Université de Lyon, F-69622, Lyon, France
[2] IFSTTAR, LESCOT, F-69675 Bron, France
[3] Université Lyon 1, CNRS, LIRIS, UMR5205, F-69622, France
[4] myTwatch, F-75002, France
[5] INSA-Lyon, CNRS, LIRIS, UMR5205, F-69622, France

Abstract. Knowledge capture and reuse is a challenging task consisting of many steps. The knowledge discovery cycle presented by Fayyad [7] offers a global overview of how these steps are combined together. By taking a step back and considering data as activity traces, we propose to change this knowledge discovery cycle. We consider that knowledge is built rather than discovered. Therefore human involvement to make sense out of the traces is paramount. We propose to use knowledge engineering techniques to benefit from the user's knowledge as early as possible. The use of modelled traces can solve this issue and change the knowledge discovery process to make it more interactive and knowledge-intensive. The proposed methodology will be illustrated with software applications we have developed which, combined together, support the whole process of knowledge discovery. A discussion of the proposed methodology and of the required tools is given.

Keywords: Interactive Knowledge Discovery, Trace-Based Systems, Human-Computer Interaction

1 General Introduction

Learning from experience requires to capture experience in such a way it can be used for further action. It is difficult to acquire such knowledge (e.g., rules or cases) from experts. In this paper, we propose to combine *knowledge engineering* and *knowledge discovery* (KD) approaches to acquire such knowledge. KD aims at discovering knowledge from data. We propose to adapt KD in the context of activity traces. Our goal is to allow to capture and apply knowledge coming from both traces and human expertise. Therefore, the general goal of this work is what could be called "*man machine interactive knowledge discovery*".

The first issue is to get relevant data among what is available in the environment and to prepare these sources to build a simple model of observed elements. Another issue is to offer interactive tools to support a dynamic knowledge discovery process: thus discovering new knowledge from the sequences of observed elements. A third issue is to take into account what has already been discovered, and what is already known to guide the KD process. Lastly, it is important to be able to provide *actionable knowledge* for humans and machines.

Discovering knowledge by observing processes, their behaviour and their productions is not new. There is a strong tradition in the Human Computer Interactions research [17] to propose concrete methods for discovering explicit knowledge from various observation sources (computer events, video and audio records, data, texts, etc.). We can use the general notion of "interaction traces" to refer to such knowledge sources and we are specifically interested in exploiting traces of events for supporting the Knowledge Discovery of the dynamics of processes. In order to support our claim several points will be discussed in this paper. First we provide a brief overview of the classical Knowledge Discovery Cycle and of its challenges. We consider KD to be a Knowledge Intensive task, and we focus on the specific issue of discovering knowledge from the observation of behaviours. Then, we develop our claim that Trace-Based Man Machine Interactive Knowledge Discovery should be a good approach for dynamic knowledge engineering and would make the reuse of knowledge easier by providing in-depth interaction in the KD process. Examples of softwares are introduced to illustrate a possible general framework which implements the proposed dynamic knowledge engineering approach. The relevance of this approach is then discussed in regards to the feedback obtained from practical knowledge discovery experience. The paper concludes with a discussion about future work and applications.

2 The Classical Knowledge Discovery Cycle

In the early 90's the use of data mining tools in order to discover new knowledge from data led to the development of *knowledge discovery* in databases (KDD). Frawley et al. defined *knowledge discovery* as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data" [9]. This first definition stresses the fact that something new and useful must be discovered.

In 1996, Fayyad et al. proposed a formal description of the knowledge discovery process [7]. The authors presented the KDD process as "interactive and iterative, involving numerous steps with many decisions made by the user" (Fig. 1 shows an overview of this cycle). With this cycle the authors showed that data mining is only one step of the knowledge discovery process. In order to discover knowledge, all the steps of the process are important. More importantly, the knowledge discovery process is guided by a user – generally an analyst. The role of the user is paramount as the user sets the goal of the discovery, defines the parameters at each step, and interprets the discovered patterns.

2.1 Knowledge Discovery Challenges

Data Mining and Knowledge Discovery (KDD) Challenges. The KDD community listed a number of challenges [8] and the most important of them

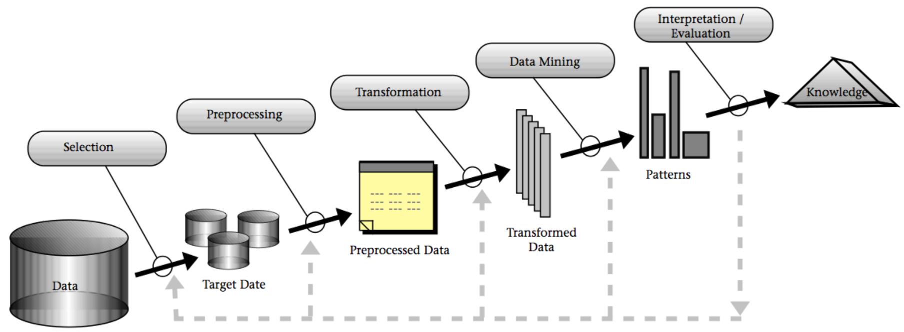


Fig. 1. Overview of the knowledge discovery cycle (from Fayyad et al. [7]). Before any discovery, the user gathers domain knowledge and identifies the goal of the knowledge discovery task. Then, the user creates a data sample (target data) from which to discover knowledge. Then, the user cleans, preprocesses and transforms data to prepare the mining. Then, the user selects a data mining method, and sets its parameters. Afterwards, the user interprets the discovered patterns. Finally, the discovered knowledge can be documented, or used to fullfil the goal initially set by the user. Note that at any step the user can go back to any previous step of the cycle.

regards the way to make the whole KDD cycle efficient for both the experts and the algorithms, as stressed by Uthurusamy [8]:

> The original process centric view of KDD espoused the three “I”s (Integrated, Iterative, and Interactive) as basic for KDD. These are central to the ideas of “Computer Assisted Human Discovery” and “Human Assisted Computer Discovery.” There has been very little work on these in recent years.

Actionability. Researchers have tried to address these challenging issues by using complementary approaches to produce *actionable knowledge*. The notion of actionable knowledge [19,16] has been defined to estimate the usefulness of discovered knowledge, with a measure of its actionability. A knowledge is actionnable if it can actually be used by the end user (i.e. to make decisions). Cao and Zhang [3] proposed a practical methodology to discover actionable knowledge by integrating domain knowledge to guide the data mining process. Yang et al. [20] addressed the issue of providing *actionable knowledge* by learning action models (expressed in PPDL[6]) in order to make the discovered knowledge available for a planner and to express the learned structure in such a way that it is understandable for an expert in planning. Here, actionability actually provides a planner or an expert with what is needed to build new plans.

[6] PPDL: Planning Domain Definition Language.

Knowledge Discovery as a Knowledge Intensive Task. Brachman et al. [2] offer a practical overview of what knowledge discovery actually implies for the user. The authors focus on the importance of the involvement of the user in the discovery process. They propose to redefine knowledge discovery as "a *knowledge-intensive* task consisting of *complex interactions*, *protracted over time*, between a *human* and a (large) database, possibly supported by a *heterogeneous suite of tools*". This definition highlights the main difficulties of knowledge discovery: knowledge discovery is *knowledge-intensive*, requires *interactions*, and the tools often lack *integration*. Our position paper consider this challenge as the central one for guiding our proposals, and explaining our first results.

2.2 Discovering Knowledge from the Observation of Behaviours

Knowledge Discovery in Databases implies the idea that data intrinsically contains information available for an analyst to interpret. In the KDD process, data is an entry point. However, data is a result of the process of data collection. What users actually care about is not the data itself, but what the data represents in the real world. For this reason, we prefer to use the term of *Knowledge Discovery* (KD) as the capture of knowledge has an important role to play in the discovery process.

Data is obtained from a two step process. First, a behaviour or process occurs in the real world. To avoid confusion, we refer to such a process as an *activity*. Second, this activity is observed and data is captured, leading to the creation of databases. The goal of knowledge discovery is to discover knowledge about the observed activity.

In this paper the application domain used to illustrate the KD process is the field of transport, with car driving as the activity of interest. The goal of KD is to understand drivers' behaviour and cognition. Discovered knowledge can be used to develop driving assistance systems and improve road safety.

The record of an activity is a trace of that activity. A trace typically contains information about the activity itself and its context. Traces also typically contain information about several activities, some relevant to the user and some that are considered by the user to be noise.

Often traces are fragmented, in the sense that the activity is recorded from different sources. Thus, the activity is not seen as a holistic process, but as several (fragmented) views of the same process. The goal of KD is to discover interesting knowledge about the activity. Therefore, the user needs to understand and transform the traces in order to, in the end, interpret the information available in the data. It is only when patterns or pieces of information are interpreted by the user that knowledge is created. The general process of knowledge discovery can been seen as follows:

$$\text{Activity} \xrightarrow{\text{activity produces}} \underset{\text{user transforms}}{\underset{\circlearrowleft}{\text{Traces}}} \xrightarrow{\text{user interprets}} \text{Knowledge}$$

Traces are not usually standard digital objects, although they have specific properties including temporal properties and a semantic interpretation of observed elements (events). Considering traces as data for a knowledge discovery process is a non trivial task that does not usually appear in the KD phases. The general task of analysing an activity through observed behaviours has been theorised by Sanderson et al. [17], who proposed the notion of the “transformation” of traces to move from a first simple interpretation (almost raw data coming from sensors) to the actionable knowledge level of abstraction. As analysts better understand what is observed, they can check their interpretation by transforming a source trace into a target trace where candidate knowledge (according to some hypothesis) is used to describe and explain the observed activity. In turn this process is repeated on the target trace, which can be used as a source trace for further interpretation. The analyst is at the heart of the discovery process and has to be assisted in his/her task. The analysis process is a knowledge intensive task and, when using data mining techniques, needs iterative steps, interactivity and integration of the different phases of the data mining task within the discovery process.

3 Trace-Based Man Machine Interactive Knowledge Discovery

We claim that to improve the efficiency of KD from behavioural observations, it is necessary to provide a **knowledge intensive** framework. Such a framework should not only allow for the integration of what is already known in the domain but should also allow the reintroduction of discovered knowledge in the discovery process for future iterations. Throughout the discovery steps, the tools supporting discovery should be *interactive*, allowing the user to select and parameterise mining algorithms, and to build (actionable) knowledge structures, with the support of heuristics and inference mechanisms.

Interactivity is the key issue since the discovering activity is *really* shared between the knowledge discovery framework and the user (expert-analyst, end user, etc.). The three “I”s of the literature (Integrated, Iterative and Interactive) can be summarised by different levels of interactivity. “**Integration**” means that the whole framework provides a unique way to interact at the knowledge level, e.g., tools are able to share data and knowledge representations. “**Iterations**” means that the user can interact with the system to connect the different phases of the discovery process, either to use the knowledge acquired to loop back to a previous phase, or to use it as an input for a next KD phase. “**Interactivity**” means that the user and the framework are “coupled” to achieve a common task corresponding to the particular KD phase, for example, the user shares knowledge with the algorithms (through appropriate knowledge representations, such as models or constraints).

In order to simplify such a process, we propose a general framework unifying knowledge representation with dynamic modelling. In this framework, any discovered knowledge can be used as a new model of the mined sources. Also, this framework keeps track of the transformations applied to the traces, which allows the reuse or adaptation of transformations on new traces, and offers a way to return to the original trace. Moreover, considering the specific case of understanding an activity, we propose to consider automata as a way to represent a discovered behaviour. Such a knowledge representation supports **actionnability** as it allows to use the discovered automata as signatures of behaviours. Automata can be actionable either by detecting matching behaviours in traces or by producing the modeled behaviour, when used in a simulation engine. In the following section we briefly present a way to implement this proposal by using Modelled Trace and Trace-Based Systems. This provides a unified way to manage domain knowledge and discovered knowledge throughout the phases of the KD framework.

3.1 Modelled Trace and Trace-Based System

In order to manipulate traces as formal objects, our research team developed the notions of *modelled traces* and of *trace-based systems* [18]. A modelled trace (*$\mathcal{M}$-Trace*) is the combination of a model and of a temporal sequence of *observed elements (called obsels)*. Obsels are events, each characterised by a date, a type, attributes and relations. The model describes the semantics of the types, attributes and relations, while the temporal sequence of obsels is the record of what has been observed. For instance, for the activity of driving, obsels are time-stamped events (the driver brakes, accelerates, then turns the steering wheel, etc.) and the model is an ontology of the observed behaviours. A *Trace-Based System (TBS)* is a framework for managing $\mathcal{M}$-Traces that provides trace-oriented services including: collecting $\mathcal{M}$-Traces, computing sequences satisfying a pattern in a specific $\mathcal{M}$-Trace, transforming a $\mathcal{M}$-Trace into another one for abstraction purposes (filtering, merging, reformulating obsels), and navigating through the transformation graph of a $\mathcal{M}$-Trace.

3.2 Towards an Interactive Knowledge Discovery Cycle

Fayyad et al. [7] proposed a general process for discovering knowledge from data (see Fig. 1). We propose a slightly different approach where data is collected from the observation of an activity, and a TBS provides support in the preparation and the transformation phases of the KD process. Furthermore, following the general idea defended by Michalski [15], we propose to complete this knowledge discovery process by a computer-supported synthesis step by providing a formal knowledge representation able to be used by a reasoning mechanism. This proposition is outlined in figure 2. These knowledge oriented approaches offer feedback to the analysts at each step with formal knowledge representations. Thus, facilitating the interactions with the analysts so that they understand the results and control the full Knowledge Discovery process. This approach has been introduced and detailed in [5], [18] and [11].

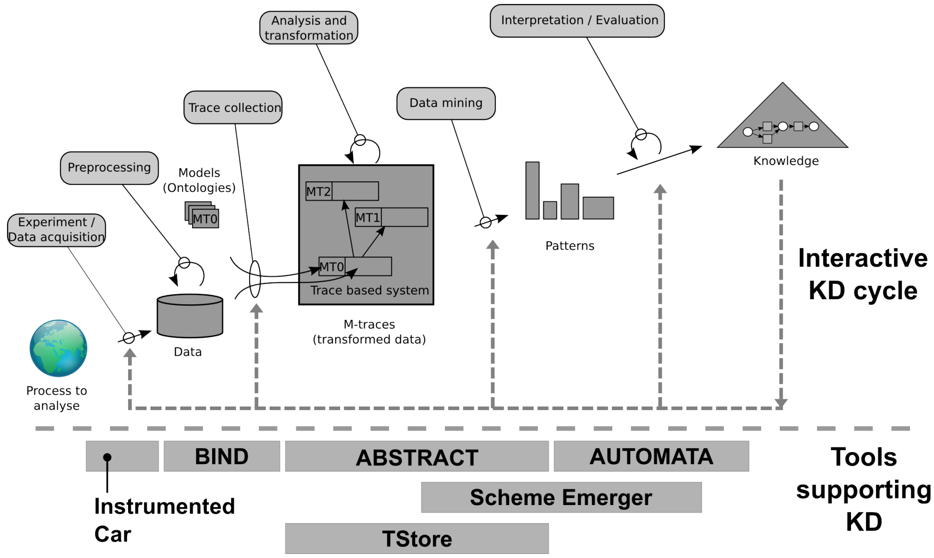


Fig. 2. The top part of the figure describes a new point of view on the (interactive) knowledge discovery cycle. The use of $\mathcal{M}$-Traces allows to explicitly describe the semantics of data and therefore supports the knowledge-intensive process of knowledge discovery. The bottom part of the figure shows examples of tools supporting the interactive knowledge discovery cycle and highlights their role in the process.

4 Examples of Software Systems

In this section software tools that are developed in the context of interactive knowledge discovery are presented. A brief overview of the Bind, Abstract, Automata, Scheme Emerger and TStore softwares is provided. Figure 2 shows how these tools can be integrated to produce a full interactive knowledge discovery cycle.

4.1 Bind – Abstract

Data is first acquired by tools capturing what happens in the environment. This is done by instrumenting the data capture with the means of tracing systems and sensors. For example, an instrumented car (or a driving simulator) is used to collect the driver's actions, the vehicle state and the environment [12, p. 70].

Once collected, the data can be processed using the Bind framework [12, p. 69]. Bind offers high level functionalities to analyse car-driving data, such as signal processing functions or the synchronised visualisation of collected data and videos. Bind also allows the discovery of markers in the numerical data. For instance, it can create event markers when a certain value in the data is reached. These markers can be seen as traces of the activity.

These traces are then imported into the Abstract software [11]. Abstract is a TBS, which means traces are represented as $\mathcal{M}$-Traces. A TBS can be used to *prepare*, *transform* and *visualise* sequences of events as $\mathcal{M}$-Traces at the relevant level of abstraction with explicit semantics (Fig. 3). The Abstract framework has been developed and used to support the Knowledge Discovery process in the context of driver behaviour analysis [10]. In this context, a huge quantity of data is collected during driving sequences, from numerous sensors, video recordings, and observer annotations. Then, the analyst can use the Abstract interface to progressively transform traces in order to discover relevant patterns according to some hypothesis about the driver's behaviour[7].

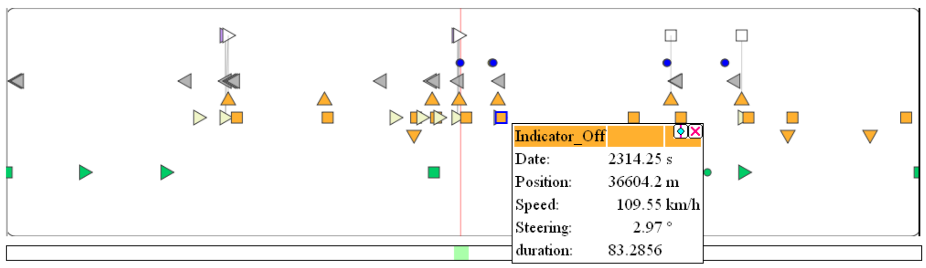


Fig. 3. Graphical interface of the Abstract software. The horizontal axis represents the time and the vertical axis is related to the level of reformulation of the traces. Each symbol represents an obsel (the shape and the colour depends on the type of obsel). The user can click on each obsel to display specific information about this event, such as its type and attributes. The user can display different views of the same trace.

4.2 AUTOMATA as a Support for Interpretation of the Dynamics of Processes

In Abstract new concepts can be defined corresponding to pattern signatures identified in the $\mathcal{M}$-Traces. This is a first step in the general issue of considering "knowledge mining" instead of "data mining" according to Michalski [15], since this process enriches the formal representation of the activity. In order to go a step further, and to actually provide actionability, the Automata software builds a Petri-net of a process being observed in the traces. The produced Petri-net represents knowledge about the dynamics of the process and is actionable, as it can be used to understand the process or to simulate it. Automata therefore supports the interpretation task, with the help of a trace mining process focusing on the available occurrences of patterns satisfying a particular request.

In order to implement a knowledge-intensive process, a mining algorithm has been modified to make it interactive [14]. The user can therefore use his/her

[7] For more details about Abstract, see `http://liris.cnrs.fr/abstract/`

knowledge about the activity to support the mining process. The non-modified algorithm [1] assumes that the traces are complete, which is rarely the case on real data sets. The *interactive* version of this algorithm allows the analyst to incorporate his own knowledge to compensate for the lack of information in the traces.

4.3 Scheme Emerger

The Scheme Emerger platform aims at discovering chronicles occurring in activity traces [4]. A chronicles is a pattern structure describing time constraints between types of events. Scheme Emerger implements an interactive mining algorithm that allows the user to constrain the mining process (e.g., patterns that are required or need to be excluded). The discovered chronicles are displayed to the user in real time, allowing the user to quickly obtain first results and to adapt the mining parameters (Fig. 4). Scheme Emerger can be used to execute queries to check the presence of chronicles defined by the user.

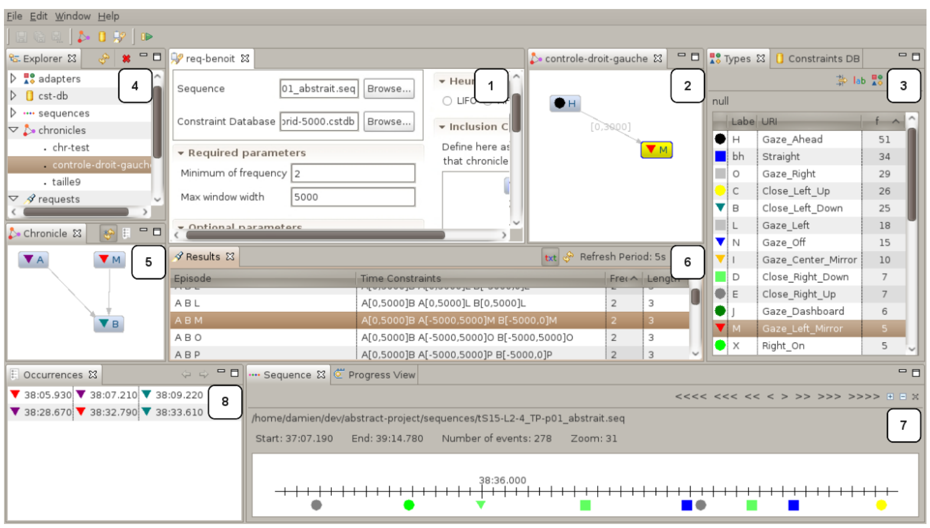


Fig. 4. Graphical interface of the Scheme Emerger platform [4]. View 4 shows the resources of the project that are handled by the analyst: constraint database, chronicles, traces and queries. Tabs 1 and 2 are chronicle and query editors. View 7 displays the trace currently being analysed. The types of events present in the current trace are displayed in view 3. View 6 lists in real time the chronicles discovered by the interactive algorithm. View 5 displays the chronicle selected in view 6, while view 8 lists the occurrences of this chronicle in the current trace. Clicking on one of these occurrences changes the focus of the trace (view 7).

4.4 TStore

TStore is a Trace-Based Management System (TBMS). The role of a TBMS is to store, manage and transform $\mathcal{M}$-Traces. A TBMS realises the $\mathcal{M}$-Traces storage, querying and handling layer of a TBS. The originality of TStore resides in the transformation facilities on $\mathcal{M}$-Traces that it offers. In addition to simple transformations such as $\mathcal{M}$-Trace filtering, $\mathcal{M}$-Traces aggregation or segmentation, TStore can use Finite-State Transducer (FST) to transform traces [21]. Transducers are automata with specific "translation" capabilities. TStore uses FST to transform $\mathcal{M}$-Traces by replacing obsels matching the FST with more abstract obsels.

5 Discussion

Table 1 offers a synthesis of how the different tools discussed in this paper can be used to address the challenging problems of knowledge discovery. Bind, Abstract and Automata offer a complete chain of knowledge discovery which has been used by a research team composed of engineers and psychologists in analysing and modelling car-driving behaviours from data collected on instrumented cars [13]. This experience provides feedback on the relevance of the interactive and knowledge-intensive discovery cycle and on the tools themselves.

Table 1. How the presented tools support the different challenges of KD.

Tool	Know. Int.	Act.	Inter.	Integ.	Iter.
Bind (v.1)	+	-	+	-	-
Abstract (v.0.2)	++	+	++	+	++
Automata (v.0.11)	+	++	++	-	-
Scheme Emerger	+	-	++	-	++
TStore	++	+	-	+	++

Know. Int.: Knowledge Intensive — Inter.: Interaction — Integ.: Integration — Iter.: Iterations. — Act.: Actionability — v.: version

The importance of integration seems fairly obvious. For example, when a specific pattern is discovered, the user wants to be able to visualise the occurrences of that pattern (typically when it is unusual). This is only possible with an integrated suite of tools.

By working with driving experts who use the KD cycle, we discovered that the process of defining the semantics (in Abstract the ontology which constitutes the model of a $\mathcal{M}$-Trace) was an important part of the KD process – from a practical point of view. This is probably due to the fact that experts already have knowledge about driving and part of the process is to make this knowledge explicit, rather than just discovering knowledge from the data. The formalism of $\mathcal{M}$-Traces allows the system to actually be Knowledge Intensive, by combining the data with a model describing their semantics.

Moreover, having explicit semantics describing the data has allowed these experts to more precisely document the data. Having proper documentation of the data is important as it allows users to properly interpret the meaning of data. It is even more important when the KD process involves multidisciplinary teams, i.e., from the electrician who understands the sensors instrumenting the car to the psychologist who understands the behaviour of a driver.

Trace-Based Systems record the transformations that lead from one $\mathcal{M}$-Trace to another. This aspect offers a documentation of the KD process itself which allows the user to easily find how a result has been derived. The documentation of the KD process could be further improved if the activity of the user (of the KD tools) was recorded. The user would then be able to introspect his/her own practice and could also use KD tools to gain a better understanding of how to improve his/her work.

The knowledge-intensiveness is practically tackled with the $\mathcal{M}$-Traces. This knowledge is actionnable as it is explicitly described in models that can be used by algorithms. Using automata to represent knowledge offers even more actionability. Indeed, automata can be used as a model for simulation and for the user to make decisions, or, as has been seen with TStore software, automata can be reintroduced in the KD process to transform $\mathcal{M}$-Traces. Therefore, combining Automata (which allows the discovery of automata from traces) and TStore would improve actionability by providing new types of interaction and globally benefit the KD process.

Overall, the feedback received from users was positive regarding the proposed solutions to support interaction, integration, iterations and actionability. However, a precise methodology needs to be defined in order to precisely measure the relative importance of these concepts in a real KD process. Evaluating the relevance of KD tools itself is a challenging task, as it consists in evaluating generative artifacts [6].

6 Conclusion

In this paper, several challenges of knowledge discovery that remain unsolved have been discussed. In the context of analysing sequential data, we proposed to adapt the classical KD methodology in order to make it knowledge-intensive. Tools supporting interaction, integration, iterations and actionability have been presented. Further work on each of these dimensions is needed to improve the KD process.

Acknowledgements The authors are grateful to the reviewers who gave thorough feedback and raised interesting points of discussion.

References

1. van der Aalst, W.M.P., Weijters, A.J.M.M., Mruter, L.: Workflow mining: Discovering process models from event logs. Knowledge and Data Engineering, IEEE Transactions on 16(9), 1128–1142 (2004)

2. Brachman, R.J., Anand, T.: Advances in knowledge discovery and data mining, chap. The process of knowledge discovery in databases: a human-centered approach, pp. 37–57. The MIT Press (1996)
3. Cao, L., Zhang, C.: Domain-driven data mining: A practical methodology. International Journal of Data Warehousing and Mining (IJDWM) 2(4), 49–65 (2006)
4. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. Expert Systems (2011), wiley Online Library
5. Deransart, P.: Conception de traces et applications (vers une méta-théorie des traces) (2010), document de travail. http://hal.inria.fr/inria-00443648
6. Ellis, G., Dix, A.: An explorative analysis of user evaluation studies in information visualisation. In: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization. pp. 1–7. BELIV '06, ACM, New York, NY, USA (2006)
7. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI magazine 17(3), 37–54 (1996)
8. Fayyad, U.M., Piatetsky-Shapiro, G., Uthurusamy, R.: Summary from the kdd-03 panel: data mining: the next 10 years. SIGKDD Explor. 5(2), 191–196 (Dec 2003)
9. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge discovery in databases: An overview. AI magazine 13(3), 57–70 (1992)
10. Georgeon, O., Henning, M.J., Bellet, T., Mille, A.: Creating cognitive models from activity analysis: A knowledge engineering approach to car driver modeling. In: International Conference on Cognitive Modeling. pp. 43–48. Taylor & Francis (2007)
11. Georgeon, O., Mille, A., Bellet, T., Mathern, B., Ritter, F.E.: Supporting activity modelling from activity traces. Expert Systems (2011), wiley Online Library
12. IFSTTAR: INRETS / LCPC – annual report 2010. Annual report, IFSTTAR (2012)
13. Mathern, B.: Découverte interactive de connaissances à partir de traces d'activité : Synthèse d'automates pour l'analyse et la modélisation de l'activité de conduite automobile. Phd thesis, Université Claude Bernard Lyon 1 (Mar 2012)
14. Mathern, B., Mille, A., Bellet, T.: An interactive method to discover a petri net model of an activity. Scientific report, Université Lyon 1, LIRIS, CNRS, UMR5205 (Dec 2010), http://liris.cnrs.fr/publis/?id=4927
15. Michalski, R.S.: Knowledge mining: A proposed new direction. In: Sanken Symposium on Data Mining and Semantic Web. Osaka University, Japan (mar 2003)
16. Piatetsky-Shapiro, G., Matheus, C.J.: The interestingness of deviations. Tech. rep., AAAI (www.aaai.org) (1994)
17. Sanderson, P.M., Fisher, C.: Exploratory sequential data analysis: foundations. Hum.-Comput. Interact. 9(4), 251–317 (Sep 1994)
18. Settouti, L.S., Prié, Y., Champin, P.A., Marty, J.C., Mille, A.: A trace-based systems framework : Models, languages and semantics. Rapport de recherche, Université Lyon 1 (2009), http://hal.inria.fr/inria-00363260/en/
19. Silberschatz, A., Tuzhilin, A.: What makes patterns interesting in knowledge discovery systems. IEEE Trans. on Knowl. and Data Eng. 8(6), 970–974 (Dec 1996)
20. Yang, Q., Wu, K., Jiang, Y.: Learning action models from plan examples using weighted max-sat. Artificial Intelligence 171(2-3), 107–143 (2007)
21. Zarka, R., Champin, P.A., Cordier, A., Egyed-Zsigmond, E., Lamontagne, L., Mille, A.: Tstore: A web-based system for managing, transforming and reusing traces. In: Proceedings of the ICCBR workshop on Traces for Reusing Users' Experience (2012)

A Trace-Based Approach for Managing Users Experience

Béatrice Fuchs, Amaury Belin

Silex Team, LIRIS CNRS, UMR 5205, Université Lyon 3, Université de Lyon
{beatrice.fuchs},{amaury.belin}@liris.cnrs.fr

Abstract. In this paper we present a framework built upon a trace-based approach for users experience analysis and management. Traces of interactions between a user and the computer system are collected and stored in a trace-based system (TBS). Once collected, the TBS allows transformation of these traces, as a basic operations to manipulate them. This system proved to be helpful in order to carry interactive knowledge discovery process, which consists in analysing traces along different dimensions in order to find relevant patterns. We present two application domains which exemplify this approach. First in musical analysis, the trace based system serves as a container of the analyst's work and various analysis tools assist the analyst in their work. Second in active video reading, computer-mediated users activity are traced and analyzed in order to study the evolution of their practices over time.

1 Introduction

In computer systems, numerous digital traces are left by the users as they are carrying out their multiple activities, either individual or collective. Often these traces are available and constitute knowledge containers testifying users experience. Hence, the exploitation of these containers for various purposes is challenging. The *modeled trace approach*[1] aims at studying knowledge dynamics by relying on traces left during individual or collective activities occurring in a digital environment. It takes advantage of traces as an experience container for different purposes. These may cover from knowledge acquisition to adaptive system, user assistance, human learning or usage analysis, and more generally the study of complex systems.

Based on the modeled trace approach, we propose a framework for interactive knowledge discovery (KD) based on traces. This framework addresses the need encountered by analysts who have to deal with traces, and detection of patterns in particular. It is intended to work in interaction with the user which intervenes in the process. The framework is built upon a *trace-based system* (TBS) which provides basic trace management features such as storing traces associated to a descriptive model, or traces *transformation* in order to manipulate them. Two

[1] studied in the Silex Team of the LIRIS Lab.: http://liris.cnrs.fr/equipes?id=44

application domains exemplify the framework for different issues. The first application is musical analysis where an analyst collects and stores musical scores, and analyzes them to find motives and to highlight the motivic structure of musical pieces. The frameworks acts as a memory and an assistant for the analysts who can store their work. The second application aims at studying how users develop practices with digital tools by analyzing evolutions in their utilization schemes. This study is conducted with cinema researchers who have to appropriate an active video reading application in order to analyze a movie.

The rest of the paper is organized as follows: section 2 presents the main principles of the trace approach and section 3 presents the framework. Finally section 4 illustrates the approach with two applications: musical analysis and active video reading, and is followed by a conclusion.

2 Principles of the modeled trace approach

The *modeled trace* approach relies on the instrumentation of the computer in order to record and collect the interactions between the user and their digital environment. Such instrumentation could be enabled by placing "sensors" in the software environment. A trace is composed of a set of *observed elements* named *obsels*. In a *modeled trace*, the obsels are temporally situated, and they instantiate a model, defined as types in a *trace model*. A trace model describes precisely the different types of obsels, their attributes, and their potential relationships with other obsels. A *trace-based system* (TBS) is a system dedicated to the modeling and the manipulation of digital traces. Its main functionalities are the collecting of traces from computer applications and their *transformation* in a generic manner [1]. A trace which is directly built from data collected by sensors is called the *primary trace*. The primary trace is often difficult to interpret because obsels may have a unintelligible form and a very fine granularity. In [2], it was outlined that traces elements may vary from a quite low abstraction level such as interface events or methods invoked in response, to a higher level such as the components of a task performed by the user. Similarly, in a primary trace, there may be quite important variations of obsels granularity due to the difficulty to automate trace collectors. Consequently, a transformation of the trace is needed in order to make it more "abstract" (figure 1).

In a TBS, *transformations* are the basic manipulation operations on traces. There are several kinds of transformations such as filtering obsels, aggregate obsels, compute some indicators, etc. Among these, the *rewriting* transformation creates a new *transformed trace* based on the primary trace in which occurrences of some pattern are substituted by new obsels named *transformees*. The patterns may result from an analysis of the trace aiming at discovering meaningful schemes in the user's activity. A succession of the several cycles of the KD process may be repeated on the trace leading gradually to more and more abstract and intelligible trace.

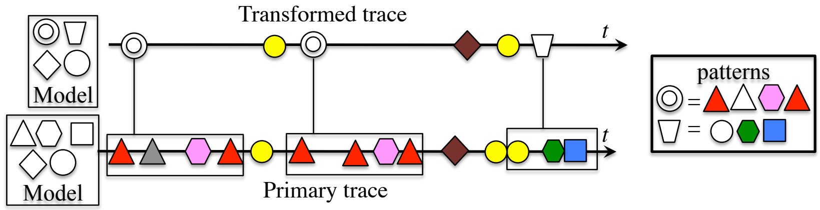


Fig. 1. The principle of trace transformation: obsels shapes symbolize the different kinds of obsels that are described in a trace model. Filled shapes symbolize obsel type instances having specific attribute values.

3 A framework for managing experience

A first version of a trace-based framework has been developed and is still evolving to progressively integrate additional features. It enables an interactive process for discovering knowledge from traces and it more generally serves as an assistant for various tasks related to trace analysis. Currently, the process allows to control a sequence miner algorithm for automatic pattern detection, as well as multiple transformations related to the different steps of the process:

- the pre and post-processing of the mining step, including the preparation of data and assistance to their interpretation,
- the rewriting of traces by replacing selected patterns by new observed elements with a higher abstraction level,
- computing domain dependent or independent indicators in order to filter results or make the interpretation of the extracted patterns easier,
- querying and searching traces or patterns.

Currently, the framework enables *offline* KD where the analysis of the traces is done afterward the activity of the user. A future perspective of the work in progress would be to enable *online* KD, meanwhile the user is performing their activity in real-time. The processing of a new trace should benefit from existing traces and or pre-calculated patterns to help discovering user's schemes. The KD process is made up of six steps (figure 2):

- The first step consists in collecting data from textual sources (trace description files, musical scores, etc.) It builds a trace and computes several additional descriptors.
- The preparation step consists in asking the user to choose an analysis dimension, (e. g. melodic or rhythmic in the musical domain) and to consequently build a sequence in the format required by the sequence miner. The user have then to choose the adequate mining parameters.

– The mining step extracts frequent patterns from one or several sequences. Currently, the Dmt4sp[2] [3] prototype is used, but other programs are also possible, such as Scheme Emerger [4].
– The presentation step is related to the transformation of the miner outputs in an intelligible and interpretable representation for the domain expert by projecting detected patterns into the initial trace (which was used to build the sequence). Other transformations and computations may also be done in order to help the user controlling the - sometimes - huge set of patterns generated by the miner.
– In the fifth step, a human expert interprets and selects the patterns. Additional domain knowledge may be used in order to assist the choice, and also to face the amount of patterns candidates.
– During the last step, the trace is rewritten by substituting occurrences of a pattern by a new symbol. A new model is associated to the new trace.

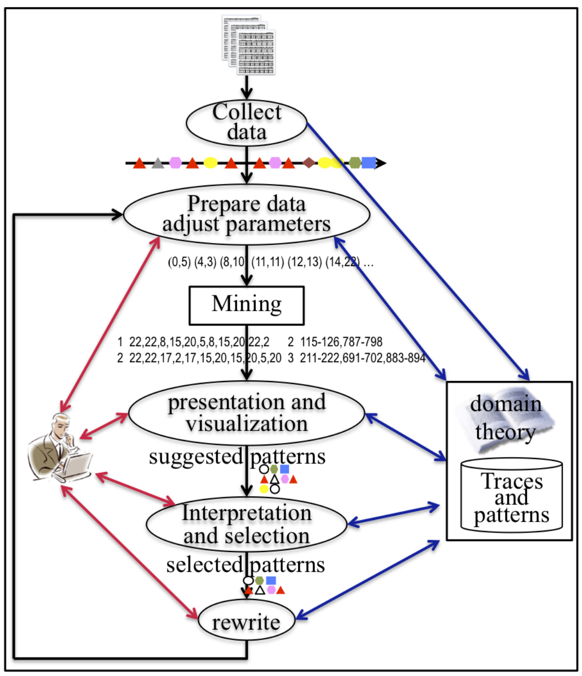


Fig. 2. The different steps of the discovery process. Oval boxes means that the users interacts in these steps. Only the mining step is performed without any human interaction.

The process, iterative and interactive, may be repeated by refining the parameters or by preparing data along another point of view. The different steps are detailed in the next sections through two application domains.

[2] Data Mining Techniques For Sequence Processing

4 Application examples

Although the musical application of the framework is not directly related to the domain of users activity analysis, it was chosen because it is the most advanced implementation and it illustrates well the approach. Moreover, it provides many features to assist an expert in musical analysis: it can serve as a support to the expert by providing a memory of their past work and a toolkit for assisting the analysis of a musical score. By considering the score as a trace of a composer's activity, the issues of this application domain are quite similar to those of the analysis of the computer-mediated user's activity.

4.1 Musical analysis

Motivic analysis consists in characterizing motives of musical pieces along its melodic and rhythmic dimensions and studying how theses motives are interrelated. An approach to motivic analysis is to implement a KD process where a score is analyzed by a sequence mining algorithm. In such a process, multiple complex transformations of data have to be done in order to pre-process data before mining and to post-process the mining results to make them interpretable by a human expert. Moreover, there are additional difficulties such as adjusting mining parameters, coping with the sometimes huge number of results and combinatorial redundancy of the miner outputs.

A musical score is represented as a trace in which patterns are discovered. Patterns are constituents of themes that highlight the articulation of the musical piece. In this domain, musical symbols are mainly musical events, *i.e.* notes and rests of the score associated to a duration, and melodic intervals between notes. An example of the main descriptors of a score is presented in figure 3.

pitch class	R	r	A	C	A	G♯	B	G♯	B	D	B	A	C	A	E	D	C
octave number			5	5	5	5	4	5	5	5	5	5	5	5	6	6	6
diatonic interval				-5	5	-1	-5	5	2	-5	5	-1	-5	5	4	-1	-1
chromatic interval				-9	9	-1	-9	9	3	-9	9	-2	-9	9	7	-2	-2
contour				↘	↗	↘	↘	↗	↗	↘	↗	↘	↘	↗	↗	↘	↘
duration	18	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
duration in beats	6	1	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
timestamp	0	18	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
measure number	1	4	4	4	4	5	5	5	5	5	5	6	6	6	6	6	6
beat number	1	1	2	2	2	1	1	1	2	2	2	1	1	1	2	2	2
position	0	0	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
position in beat	0	0	0	1/3	2/3	0	1/3	2/3	0	1/3	2/3	0	1/3	2/3	0	1/3	2/3

Fig. 3. The main descriptors of a score generated by trace collection and subsequent transformations.

These descriptors are modeled as obsels in the trace model that occur at precise instants over time and have a duration. The simplified trace model corresponding to a score is presented in figure 4.

Collecting the trace is done from textual descriptions of musical scores in the Lilypond format, a LaTeX-like musical score format. A primary trace is build, made of a sequence of musical events associated to several descriptors and additional obsels which are automatically computed, such as melodic intervals.

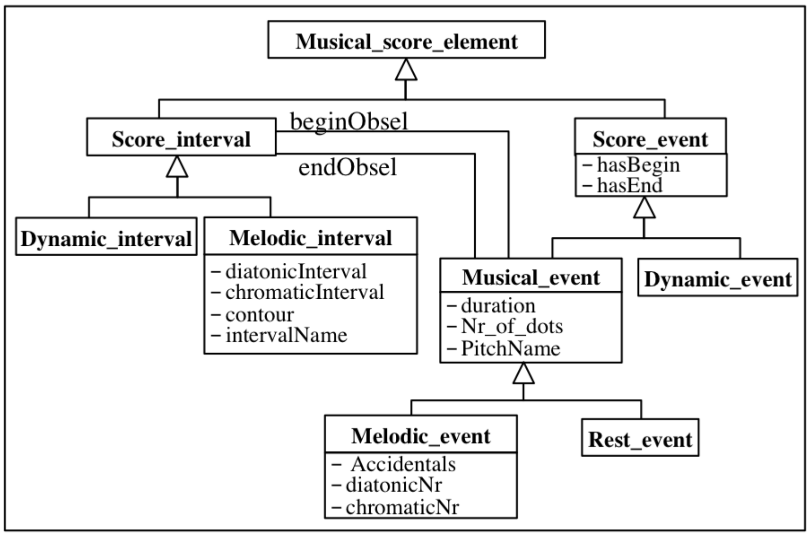


Fig. 4. An excerpt of the hierarchy of the main obsel types of the trace model. Main obsels types are musical events (notes, rests, dynamic or articulation indications) or intervals of events such as melodic intervals.

Based on the primary trace, an event sequence is computed in accordance to the input format of the sequence miner. In Dmt4sp, a sequence is represented by a time stamped event sequence: $S = \{(e_i, t_i)\}$, where $e_i \in \mathbb{N}$ is an event type and $t_i \in \mathbb{N}$ a timestamp. For example, a sequence Dmt4sp corresponding to the previous score will look like:

```
timestamp  0 18 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
event type 1  1 29  4 29 25 34 25 34  9 34 29  4 29 14  9  4 34  4  9  4 34 29 34 25 14 29  4
```

An event type may correspond to a note name or to a duration, or a melodic interval, depending on the dimension chosen by the analyst.

Then, Dmt4sp extracts candidate patterns over a single sequence or several event sequences under the minimal occurrence semantics as defined in [5]. It supports several constraints for limiting the amount of results: minimum frequency, time constraints: windows width, minimum/maximum time span between events, syntactic constraints: minimum/maximum length, beginning or ending events constraint, filtering events over a given frequency.

The patterns outputted by Dmt4sp are presented as a comma separated list of events types followed by the number of occurrences of the pattern, and for each occurrence, the timestamps of the first and last events type of the pattern. These results have to be transformed in order to be intelligible and interpretable by an expert. Patterns are first rewritten by replacing event numbers by note names, and timestamps are converted into measure number and beat number in order to locate the pattern in the score. For example:

the pattern `29,4,29,25,34,25,34,9,34,29,4,29`

is rewritten into : `A,C,A,G♯,B,G♯,B,D,B,A,C,A`.

Users assistance pertains in the transformation process but also the gathering of previous analyses as an experience container that can be mobilized for further analysis or comparison.

This application illustrates an interactive knowledge co-construction process. While the analyst is working, their attention is attracted by motives located at some particular areas corresponding to measure boundaries: the analyst set up mentally their own "heuristic" to focus more directly on potentially meaningful motives. The computer scientist notices this and implemented consequently an heuristic in the form of an indicator in the post-processing of the extracted patterns. Experiments when using the indicator have shown a significant decrease of the number of results to consider.

4.2 Active video reading

Context. With the perspective of assisting users in the organization of their activity, the aim of the project is to study how their practices evolve as they appropriate digital technology [6]. During 3 weeks, we followed the work of 7 students who started to use *Lignes de Temps* (fig. 5), an application which basically allows to annotate and tag segments of movies, see the annotations through customizable timeline representations, and to play only segments of the movie annotated with certain values. Students were interested by the tool and could see its potential value for movie analysis. After a quick overview of the functionalities of the tool, they started to use it on their personal computers. They had 3 weeks to work on a particular movie and prepare an oral presentation.

Preparing trace collection. In order to observe evolutions of students practices, we conducted several interviews, and we also collected interactions traces with Lignes de Temps. Traces were collected by instrumenting the Lignes de Temps applications and placing different sensors in its source code. For each type of event identified as relevant in the context of our study [6], we defined an Obsel Type with different typed properties and relations. We ended up with a trace model of 87 obsel types.

Collected data. We collected the interaction traces of 4 subjects who used Lignes de Temps for 3 weeks. 3 subjects spent between 8 and 12 hours using the application, resulting in interaction traces with size from 15,000 to 34,000 obsels.

The last subject spent more than 25 hours on the application and generated a trace of 70,000 obsels. We also collected 18,000 screenshots associated with these traces.

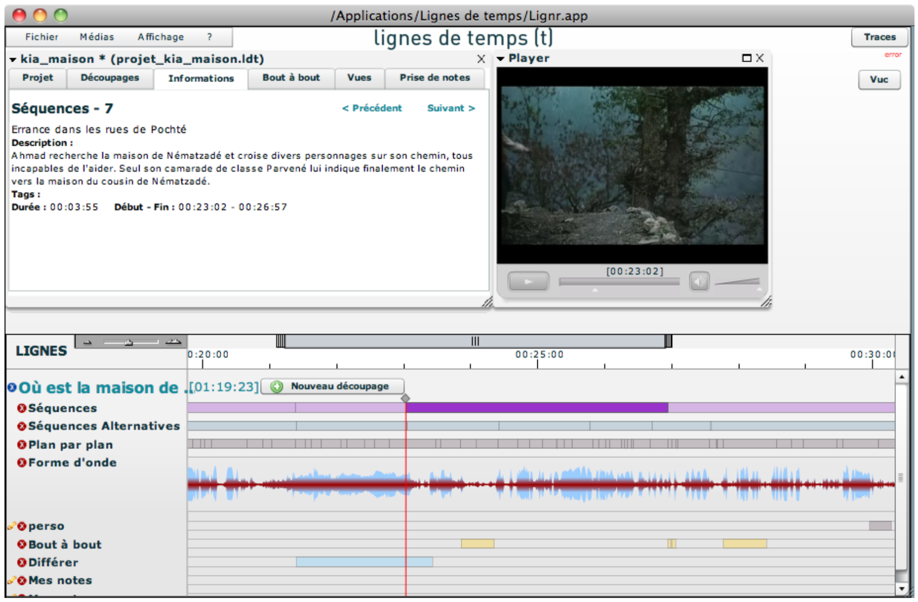


Fig. 5. The video annotation application, Lignes de Temps

Analysis and interpretation of the collected traces. With these interactions traces, our idea was to detect the apparition of patterns of interaction that would be a manifestation of scheme developed by the user. Filtering traces, and navigating through the timelines representation of traces, associated with presentation of screenshot of user interface, helped us to understand the unfolding of students activity, and to identify some high level tasks (such as tagging a group of segments). Our knowledge about users activity and the discussions we had with them certainly helped.

Timeline representation also helped to identify patterns of interaction. Some of them were obvious, like the one presented in figure 6, which was repeated by the user in order to tag each segment of the movie. We were able to detect an evolution of this particular pattern later in the activity, as the user systematically saved their work after the edition of an annotation.

Using pattern detection algorithm to assist usage analysis. Giving the important amount of collected data, automatic mining appears clearly as an interesting solution to facilitate our work. We are currently using pattern detection algorithm in order to refine our analysis of the data. Results provided by the algorithm are often related to evident chain of events related to systematic reaction of the system. Some results are more interesting and lead our attention to interesting patterns in the activity. We observe that it is often necessary to refine the pattern detection process in order to observe existence (or absence) of these patterns more clearly. Also, it is often necessary to visualize the former trace in order to better understand patterns detected by the algorithm.

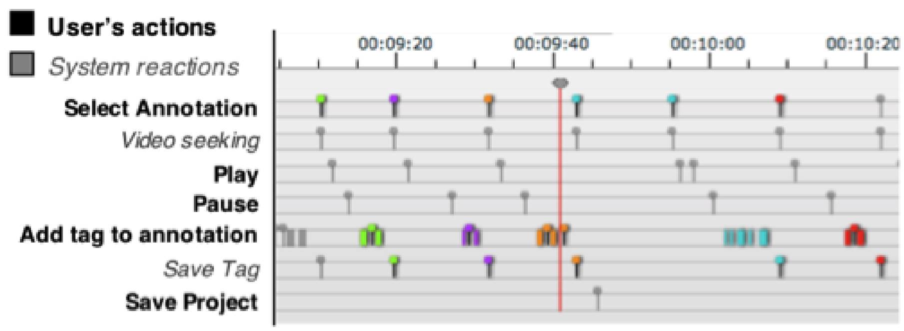


Fig. 6. Timeline visualization of one minute of interaction. The user is annotating a movie. Colors are used to represent the associations of the annotations with each related action. From [6].

4.3 Other targeted applications

Other applications are targeted and two projects have just started. A first project aims at developing generic models and tools allowing the actors of Social Gaming to analyze the level of involvement of their players. A second project concerns diary management in mobile systems in a cloud environment. The project aims at analyzing how time and tasks are managed, study habits and rituals through the detection of use schemes, study the appropriation of the traces by the user, suggest planning and encourage reflexivity on past activity.

5 Conclusion

We presented a trace-based framework for the study of complex phenomena, such as a the analysis of user's cognitive activity mediated by a computer system. The framework provides analysis capabilities thanks to an interactive knowledge discovery process to assist the analyst's work. It is built upon a trace-based system which supports collection and storage of digital traces as well as transformation operations to manipulate them. Beyond specific application domains, the framework may be applied more generally to help managing complex and temporally situated information for analysis, as exemplified by two applications.

Some of the previous works that inspired the presented approach are summarized in [7]. Scheme Emerger [4] is an environment for interactive knowledge discovery from traces providing an algorithm for complete chronicle discovery.

Another approach [8] extended the algorithm α [9] to enable interactivity with the user and taking benefit of their knowledge when building an automata.

The presented work is currently under progress. First prototypes have been implemented without the TBS and current prototypes based upon the TBS are under implementation. They constitute a basis for future experiments, and there are still many issues to be studied. A first main issue is to identify and formalize generic transformation operations related to the different steps of the process. Some basic generic transformations are already provided in the TBS but they are not sufficient to cover the needs of the framework. Another important issue is to provide assistance to the user in the analysis of traces. It may first concern the different steps of the discovery process : choosing features to analyze, tuning mining parameters, filtering the miner's outputted patterns to limit the combinatorial redundancy by the use of specific indicators or additional domain knowledge, providing visualization tools. Many trace management issues have also to be studied such as the organization of traces within a TBS like managing the relations between primary traces and transformed traces.

References

1. Settouti, L.S., Prié, Y., Cram, D., Champin, P.A., Mille, A.: A Trace-Based Framework for supporting Digital Object Memories. In: 1st International Workshop on Digital Object Memories (DOMe'09) in the 5th International Conference on Intelligent Environments (IE 09). (July 2009)
2. Hilbert, D.M., David, Redmiles, D.F.: Extracting usability information from user interface events. ACM Computing Surveys **32** (1999) 384–421
3. Rigotti, C.: dmt4sp : Data mining technique for sequence processing (2006) http://liris.cnrs.fr/ ˜crigotti/dmt4sp.html.
4. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. Expert Systems, The Journal of Knowledge Engineering (may 2011)
5. Mannila, H., Toivonen, H., Verkamo, A.: Discovery of frequent episodes in event sequences. DMKD Journal **1** (1997) 259–289
6. Belin, A., Prié, Y.: DIAM: towards a model for describing appropriation processes through the evolution of digital artifacts. In: Proceedings of the Designing Interactive Systems Conference. DIS '12, Newcastle, UK, ACM (2012) 645654
7. Mathern, B., Mille, A., Cordier, A., Cram, D., Zarka, R.: Towards a Knowledge-Intensive and Interactive Knowledge Discovery Cycle. In Lamontagne, L., Recio-García, J.A., eds.: This issue. (sept. 2012)
8. Mathern, B., Mille, A., Bellet, T.: An Interactive Method to Discover a Petri Net Model of an Activity. Technical Report RR-LIRIS-2010-026, LIRIS UMR 5205 CNRS (December 2010)
9. Van der Aalst, W., Weijters, A., Maruster, L.: Workflow mining: Which processes can be rediscovered. Technical report, BETA Working Paper Series, WP 74, Eindhoven University of Technology, Eindhoven (2002)

TStore: A Web-Based System for Managing, Transforming and Reusing Traces

Raafat Zarka[1,2], Pierre-Antoine Champin[1,3], Amélie Cordier[1,3], Elöd Egyed-Zsigmond[1,2], Luc Lamontagne[4] and Alain Mille[1,3]

[1] Université de Lyon, CNRS
[2] INSA-Lyon, LIRIS, UMR5205, F-69621, France
[3] Université Lyon 1, LIRIS, UMR5205, F-69622, France
[4] Department of Computer Science and Software Engineering,
Université Laval, Québec, Canada, G1K 7P4

{raafat.zarka, pierre-antoine.champin, amelie.cordier, elod.egyed-zsigmond, alain.mille}@liris.cnrs.fr, luc.lamontagne@ift.ulaval.ca

Abstract. This paper presents TStore, a Trace-Based Management System that handles the storage, transformation and reusing of Traces. Traces have been often stored without explicit structure. TStore is a web tool that allows anyone to store and reuse traces with their models from various applications. The transformation of Traces helps to move from a first simple interpretation (almost raw data coming from sensors) to the actionable knowledge level of abstraction. TStore provides predefined transformation functions as well as a customized transformation based on Finite State Transducers. Our experiments demonstrated the efficiency of TStore to handle the storage requests. They show that batch operations are more efficient than sequentially adding traces to the system.

Keywords: Trace-Based Management System, Finite State Transducer, Trace Transformation and Human Computer Interaction.

1 Introduction

Nowadays, many recent applications retain traces of their usage by collecting user information. These traces help to understand users' behaviors. Traces are usually stored in different ways and not formalized. We consider traces as objects satisfying a meta-model implying specific properties and associate methods (*M-Traces*) [1]. M-Traces are organized as sets of observed elements associated with their explicit models. In order to simplify the development of M-Traces Based Systems, it becomes important to develop a new kind of M-trace management system called Trace-Based Management System (TBMS).

In this paper, we describe our TBMS called TStore. It is a web tool that allows different agents (applications and users) to concurrently access, store and reuse M-Traces issuing from various applications. It is composed of four modules: Storage

Manager, Querying System, Transformer and Security Manager. TStore manages M-Traces in a relational database and benefits from its storage and querying facilities. We also describe a customized approach for M-Trace transformation based on the Finite-State Transducer (FST) principle [2]. To reduce network traffic and avoid fire-wall blocking, TStore supports batch operations. Our experiments show that storing multiple observed elements as a chunk is better than storing them separate.

The remaining of the paper is organized as follows. Section 2 describes an illustrating scenario in a video recommendation application. Section 3 introduces the general concepts of trace-based systems. TStore structure and functionalities are presented in Section 4. In Section 5, the implementation details are discussed. We report our experiments and performance study in Section 6. We discuss the related work in Section 7, and conclude our work in Section 8.

2 Illustrating Scenario

Wanaclip[1] is a web application for generating video clips from different media: photos, videos, music and sounds. In Wanaclip, users enter keywords, the system searches video sequences (*rushes*) annotated with these keywords and lets the users drag them into a timeline in order to compose a video clip. Given the large amount of content available, the problem is to quickly find content that truly meets our needs.

In [3] we proposed an approach for contextual video recommendations based on a Trace-Based Reasoning approach. The collection process is the first step that builds the primary M-Traces by observing and storing the interactions between the user and the application. Collected M-Traces need to be stored and managed in a simple way to be retrieved and transformed efficiently. TStore is a TBMS designed for this scenario and other similar scenarios.

3 Background

An interaction trace is a rich record of the actions performed by a user on a system. In other words, a trace is a story of the user's actions, step by step. Here, we focus on a special kind of trace, called *modeled traces* (M-Traces). M-Traces differ from logs in the sense that they come with a model. As shown in Fig. 1, an M-Trace includes both the sequence of temporally situated observed elements (*obsels*) which is the instantiated trace and the model of this trace which gives the semantics of obsels and the relations among them [1].

The obsels are generated from the observation of the interaction between the user and the system. Each obsel has, at least, a type and two timestamps (begin and end). Obsels can also have an arbitrary number of attributes and relations with other obsels. Each *obsel type* has a domain of attributes and indicates the values of its attributes in the range of the attribute type. A *tracebase* is a collection of related M-Traces structured for a specific purpose.

[1] http://www.wanaclip.eu

Specific operations on M-Traces allow producing *transformed M-Traces* for several purposes (filtering, segmentation, reformulation, etc). Transformed M-Traces can be easier to reuse in a given context than the primary M-Trace. A TBMS guarantees the possibility, at any time, to navigate between transformed M-traces. Fig. 1 shows an M-Trace *T1* which is transformed into *T2* and *T3* (level 2). It also shows that *T2*, *T3* are transformed together into *T4* (level 3). Each trace model contains different types of obsels that may have relations between them. For example, in the M-Trace *T1*, *M1* is its trace model that contains four obsel types (*c1*, *c2*, *c3*, *c4*) and one relation type *r1*. *T1* contains seven obsels (*o1*, *o2*... *o7*). Connected M-Traces represent a tracebase ([*M1*, *T1*], [*M2*, *T2*], [*M3*, *T3*], [*M4*, *T4*]).

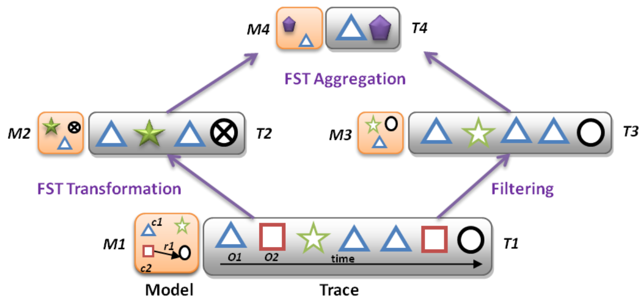


Fig. 1. An example of M-Traces and their transformations

4 TStore Architecture and Specifications

In this section, we describe the structure of TStore and its functionalities (see Fig. 2). TStore is a TBMS for storing M-Traces sent by application specific collecting modules. TStore contains four modules. The Storage Manager receives messages containing M-Traces from the clients and stores them in the database. The Querying System retrieves M-Traces from the database to answer queries of agents. The Transformer contains different functions to perform operations on M-Traces to produce transformed M-Traces. Finally, the Security Manager ensures M-Traces protection and the distribution of roles and privileges.

4.1 Storage Management

The Storage Manager is a module responsible for the communication between the M-Traces stored in the database and the client application connected to TStore. It contains services for creating models, storing M-Traces, obsels and their attribute values. Trace collectors send messages containing the collected M-Traces to the Storage Manager for storing them in the database. The Storage Manager allows concurrent access, so multiple users can interact with the system and store or retrieve their M-Traces simultaneously. Conflicts cannot arise while creating models or storing M-Traces, since each user is responsible for his M-Traces.

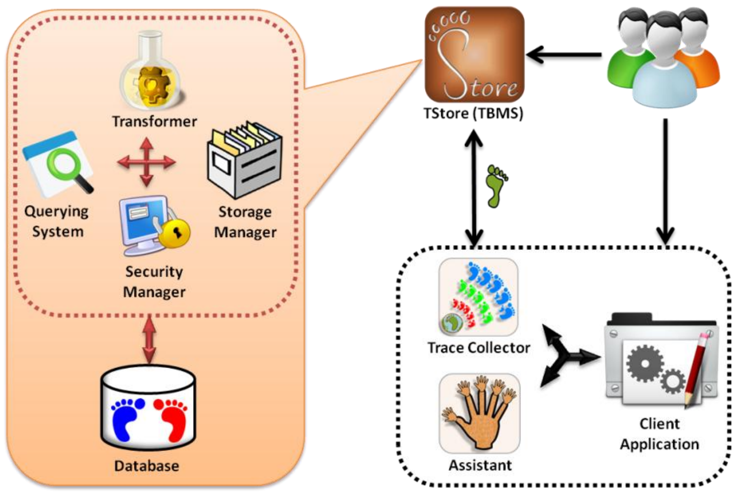


Fig. 2. TStore general structure

As TStore is a web tool, therefore, it is not convenient to make some services await the results of other services. That takes a lot of time for execution and messaging. Hence, TStore makes the storage flexible by scheduling a sequence of executions based on their dependencies. A service can be executed only after the completion of other services. For example, storing an obsel requires finishing the storage of all its hierarchical attributes and their values. TStore benefits from XML structure to store all the components. We will see an example of XML messages in Section 5.

The Storage Manager can also automatically update a trace model if it has some missing items like an obsel type or an attribute type. If some attributes do not have an attribute type in the predefined model, the Storage Manager automatically creates a new attribute type. This feature allows M-Trace collectors to store their M-Traces without defining all the details of their models, like in web logs.

4.2 M-Trace Transformation

The transformation of M-Traces helps to move from a first simple interpretation (almost raw data coming from sensors) to the actionable knowledge level of abstraction. TStore has two different approaches for M-Trace transformations: predefined functions and a customized approach based on the Finite-State Transducer (FST) principle. The Transformer has predefined functions for frequently used transformations such as filtering, aggregation and segmentation. Filtering transformation is only based on obsel types. It takes as input an array of obsel types to generate a transformed M-Trace containing only the input obsel types. Aggregation allows merging several M-Traces in one transformed M-Trace. Obsel types of an aggregated M-Trace contain a union of all obsel types of the original M-Traces. Unlike aggregation, segmentation cuts M-Traces into smaller chunks to be more useful for understanding the behavior of client applications.

In these transformations, we notice that the transformed M-Trace preserves the same obsel types as the original M-Trace and it doesn't produce new obsel types. However, our FST transformation approach allows defining customized transformations using FST task signatures. The task signature concept has been introduced in [4] as a set of event declarations, entity declarations, relations, and temporal constraints. As shown in Definition 1, Transducers are automata that have transitions labeled with two symbols. One of the symbols represents input, the other is output [2]. TStore uses FST to produce new M-Traces. Such a transformation consists in replacing some obsels matching the FST with more abstract obsels. Currently in TStore, it is experts who define the structure of the new obsel in the same way they define a new obsel type. It is possible that two different transducers generate the same transformed M-Trace. However, FST should be deterministic to avoid ambiguity.

Definition 1. A deterministic finite state transducer (DFST) is described as a 7-tuple $(Q, i, F, \Sigma, \Delta, \delta, \sigma)$ where:

- Q is the set of states,
- $i \in Q$ is the initial state,
- $F \subseteq Q$, the set of final states,
- Σ and Δ, finit sets corresponding respectively to the input and output alphabets of the transducer,
- δ, the state transition function which maps $Q \times \Sigma$ to Q,
- σ, the output emission function which maps $Q \times \Sigma$ to Δ.

A transducer is said to be deterministic if both the transition function and the emission function lead to sets containing at most one element. Frequently, the transition function and the emission function are combined into a single function, which may also be called δ, in $Q \times \Sigma \rightarrow Q \times \Delta$, mapping a pair of a state and an input symbol onto a pair of a state and an output symbol [5].

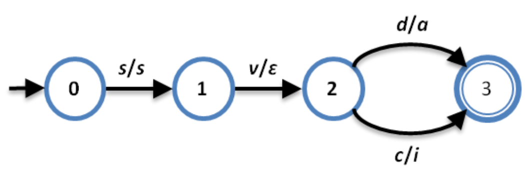


Fig. 3. Example of a FST transformation

By using FST we can apply a large variety of transformations. Fig. 3 shows an example in Wanaclip. A user can search for videos then view them. If he likes the video, he adds it to the selection, otherwise he closes it. This task reflects his satisfaction by accepting or refusing the video depending on the actions. We define a transducer as:

$$T = (\{0,1,2,3\}, 0, \{3\}, \{c, d, s, v\}, \{a, i, s\}, \{(0, s, s, 1), (1, v, \varepsilon, 2), (2, d, a, 3), (2, c, i, 3)\})$$

It represents a task that starts by an obsel of type “search” (*s*) and follows by an obsel of type “view” (*v*). If the next obsel type is “add” (*d*), an obsel of type “accept” (*a*) is generated. Else, if the obsel type is “close” (*c*), an obsel of type “ignore” (*i*) is generated. It transforms (*s v d*) to (*s a*), and (*s v c*) to (*s* i). Where, *state0* is the start state, *state3* is the accept state and ε is the empty character.

4.3 M-Trace Querying

M-Traces are a large source of information. So, we need such a system that enables the extraction of episodes and patterns from M-Traces. TStore allow M-Traces to be retrieved at all levels and to navigate between transformed M-Traces. The Querying System contains some predefined methods allow M-Trace retrieval using different criteria. It can retrieve M-Traces for a specific user, in a specific period, contains a set of obsel types, etc. It provides statistics about M-Traces, obsels, users like their frequencies, relations, reusing, etc. Querying is a matter for our future work. We aim to define a querying language that is able to represent all the types of agents’ queries.

4.4 Security Manager

M-Traces can contain some sensitive data like passwords and credit cards numbers. Agents are responsible of what they send to TStore. However, TStore ensures that only those with sufficient privileges can access the stored M-Traces. It protects the M-Traces by securing the underlying DBMS that stores them. TStore uses a role-based access control (RBAC) approach [6]. The Security Manager allows creating roles for various task functions. The privileges to perform certain operations are assigned to specific roles. Users are assigned particular roles, and through those role assignments acquire the privileges to perform particular functionalities like creating models, adding user, deleting M-Trace, etc. Each user is responsible for his M-Traces and he can also specify their visibility to be public, private or custom. Private M-Traces can be used only for the analysis. Anyone can access and retrieve public M-Traces. While in custom M-Traces, only those with appropriate privileges and those who have the permission from the user himself can access them. It is possible to have privileges on any item such as an M-Trace, obsel and attribute type. For example, salary attribute can be hided from specific users.

5 Implementation

We have implemented TStore as a PHP web service system over a MySql RDBMS. TStore exchanges XML messages that have a predefined structure. For example, the following XML message can be sent to “storeObsel” service to add an obsel of type “SearchMedia” to the M-Trace (id=101) of the model (id=1). This obsel has 4 attributes: tags=”Lyon”, kind=”right”, duration=”30” and rhythm =”0”. If the obsel type and attribute types are not existed, it calls “storeObselType” and “storeAttributeType” services to create them.

TStore is independent of the client environment. Supporting client's APIs facilitate the generation the XML messages, but it is not mandatory. Thus, it is not important to develop APIs for client applications. M-Traces are indexed and partitioned in the database to enhance the retrieval performance. Obsels are sorted according to their traceID so that we can quickly retrieve all the obsels and their attributes of a specific M-Trace. M-Traces are also sorted and indexed according to their timestamps because most of search queries retrieve M-Traces for a specific time interval.

```xml
<obsel id="1" traceID="101" start="2012-05-21 17:31:26"
end="2012-05-21 17:31:26">
  <obselType name="SearchMedia" modelID="1"
    <attributes>
      <attribute id="1" name="tags" type="string">
        lyon
      </attribute>
      <attribute id="2" name="kind" type="string">
        right
      </attribute>
      <attribute id="3" name="duration" type="number">
        30
      </attribute>
      <attribute id="4" name="rhythm" type="number">
        0
      </attribute>
    </attributes>
  </obselType>
</obsel>
```

6 Experiments

In order to test our environment, we have implemented a collection process in Wanaclip. The M-Trace handler captures users' events, models them as obsels and stores them in TStore. We conducted tests to determine how TStore performs in terms of responsiveness, memory usage and stability under a workload. These performance measurements were performed using a laptop with an AMD Phenom II N930 Quad-Core 2.00 GHz processor, 4.00 GB RAM and a Windows7 32bit operating system.

Wanaclip collects about 50 different types of obsels. Each obsel type contains a different number of attributes from 1 to 30. To evaluate the storage performance a mixture of single obsel and multi obsel storage are used. After 1000 random tests to store a single obsel, we found that the average storage time for an obsel is 0.148 seconds (0.054 execution time + 0.094 messaging time). The execution time is the time TStore spends to insert an obsel in the database after receiving the message from the M-Trace collector. The messaging time is the time of exchanging a message between TStore and the M-Trace collector (sending + receiving). The average memory usage to store a single obsel is 26.325 KB.

To examine the storage of multiple obsels, we tried 1000 random tests. At each test, we store a random number of obsels at once as a chunk (1 to 150 obsels in a chunk). As a result, it takes on average 1.368 seconds (1.118 execution time + 0.250 messaging time) to store a chunk. A chunk contains 41.7 obsels on average and each obsel contains 9.6 attributes on average. The average memory usage needed to store a chunk is 50.962 KB. Table 1 shows a comparison between single and multiple obsel storage. These values are calculated based on the average value of all the tests that we performed. Depending on these results, Wanaclip needs about 2.485 seconds to store 100 obsels as a chunk in TStore and it uses 70.792 KB of memory. But, it needs about 14.8 seconds to store 100 obsels separate and 26.325 KB of memory for each process. Therefore, it shows that the multiple obsel mechanism reduces the require time for storage. This is due to the spent time for messaging, parsing command, inserting rows in the database. In multiple obsel, it reduces the number of exchanged messages and the required time for parsing the messages.

Table 1. A comparison table between single and multiple obsel storage

Factor / Storage type	Single obsel	Multiple obsels
Messaging time	0.094 S	0.250 S
Execution time per obsel	0.054 S	0.043 S
Execution time per attribute	0.006 S	0.005 S
Storage time per obsel	0.148 S	0.048 S
Storage time per attribute	0.015 S	0.005 S
Memory usage per obsel	26.325 KB	1.243 KB
Memory usage per attribute	2.742 KB	0.129 KB

As shown in Fig. 4, the execution time and the memory usage for multiple obsels storage have logarithmic growth. The higher the number of obsels stored, it increases the memory usage and the required execution time. This incensement gradually decreases until very small changes in the end. However, messaging time has very little changes, thus it can be considered as a fixed value. The most important factor is the number of attributes in the obsels and their hierarchy. The obsels that have more number of attributes need more time to be stored.

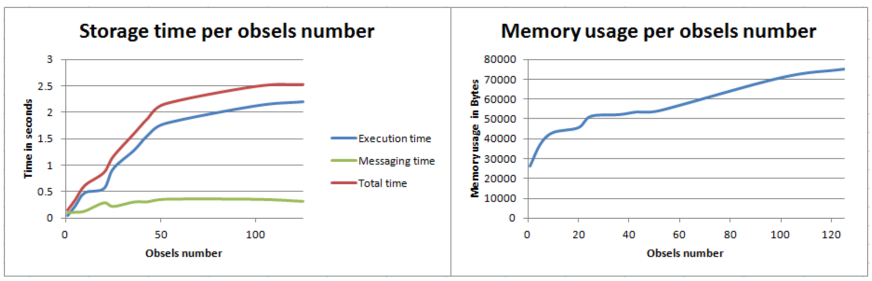


Fig. 4. Storage time and memory usage per obsels number

7 Related work

Many applications write log files to get some information about their usage. Log files typically consist of a long list of events in chronological order and they are usually plain text files. Most often, one line of text corresponds to one log entry. If an entry contains several fields, they are separated by a delimiter, e.g., a semicolon. The problem is that the delimiter may be part of the log information. Many programming environments and networking tools use XML log files to indicate the status of variables, the results of decisions, warnings of potential problems and error messages. For example, WinSCP uses XML logging to find a list of files that were actually uploaded/downloaded and Record operations done during synchronization [7].

In the web, there are two tracing approaches. Tracing systems can be located on the client side (i.e. browser plug-in) or integrated in the traced applications on server side. In both cases, users' activities are usually recorded as web logs that contain mainly the visited links. CoScripter [8] is a Firefox plug-in created by IBM Research. It records user actions and saves them in semi-natural language scripts. The recorded scripts are saved in a central wiki for sharing with other users. WebVCR [9] and WebMacros [10] record web browser actions as a low-level internal representation, which is not editable by the user or displayed in the interface. The UserObservationHub [11] is a small desktop service (daemon) that catches several registered user observation notifications and passes them on to interested listeners. Most of these tracing systems are mainly for collecting users' interactions but not managing and reusing them. In bioinformatics, The International Nucleotide Sequence Databases provide the principle repositories for DNA sequence data. In addition to hosting the text sequence data, they host basic annotation and, in many cases, the raw data from which the text sequences were derived [12].

A Kernel for Trace-Based Systems (kTBS) was the first TBMS developed in Python [13]. Both KTBS and TStore implement the M-Trace concept. KTBS uses RDF files to store M-Traces, while TStore manages M-Traces in a database and benefits from its functionalities. In TStore, it is not important to develop APIs for client application while in KTBS it is important. KTBS currently support APIs for Java, PHP and Flex. KTBS stores obsels separately while TStore handles several obsels together to reduce network traffic. KTBS supports different message formats like, JSON, XML, and Turtle. Currently TStore only supports XML but we hope to support other formats. KTBS and TStore has predefined transformations, in addition, TStore supports FST transformations. KTBS transformations are filtering, fusion and sparql rule.

8 Conclusion and future work

In this paper we described TStore, a web Trace-Based Management System. We presented the modules of TStore that uses the notion of M-Traces. The Storage Manager receives messages containing M-Traces from the clients and stores them in the database. The Querying System retrieves M-Traces from the database to answer queries of client applications. The Transformer contains different functions to produce transformed M-Traces. The Security Manager ensures M-Trace protection and the distribution of roles and privileges. The major contribution is the customized transformation approach based on the Finite-State Transducer (FST).

Our experiments demonstrated the efficiency of TStore to handle the storage requests. The results showed that storing multiple obsels as a chunk is better than storing them separately. We discussed other related work like log files, web logs, DNA repositories and KTBS. The implementation of TStore is still in progress and a lot of services should be added. Future work will involve developing a querying language that allows answering different users' requests. We need to develop our FST Transformation approach and try to define them automatically. A user interface is one of the important things to be provided since it allows users and admin to browse and manage M-Traces according to their privileges. Currently TSore supports XML messages so we want to add new formats. Lastly, we want to develop a visualization module that helps to view and analyze M-Traces.

9 References

1. Settouti, L.S.: M-Trace-Based Systems - Models and languages for exploiting interaction traces, http://liris.cnrs.fr/Documents/Liris-4984.pdf, (2011).
2. Roche, E., Schabes, Y.: Finite-State Language Processing. *MIT Press* (1997).
3. Zarka, R., Cordier, A., Egyed-Zsigmond, E., Mille, A.: Contextual trace-based video recommendations. *21st international conference companion on World Wide Web (WWW-XperienceWeb'12)*. pp. 751–754. , Lyon, France (2012).
4. Champin, P.-A., Prié, Y., Mille, A.: MUSETTE : a framework for Knowledge from Experience. *EGC'04, RNTI-E-2 (article court)*. pp. 129–134. Cepadues Edition (2004).
5. Mohri, M.: Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*. 23, 269–311 (1997).
6. Ferraiolo, D.F., Kuhn, D.R., Chandramouli, R.: Role-Based Access Control. *Artech House* (2003).
7. WinSCP, http://winscp.net/eng/docs/logging_xml, (2012).
8. Leshed, G., Haber, E.M., Matthews, T., Lau, T.: CoScripter: automating & sharing how-to knowledge in the enterprise. *CHI 08 Proceeding of the twentysixth annual SIGCHI conference on Human factors in computing systems*. 1719–1728 (2008).
9. Anupam, V., Freire, J., Kumar, B., Lieuwen, D.: Automating Web navigation with the WebVCR. *Computer Networks*. 33, 503–517 (2000).
10. Safonov, A., Konstan, J.A., Carlis, J.V.: Beyond Hard-to-Reach Pages : Interactive , Parametric Web Macros. *Proc Human Factors and the Web*. 1–14 (2001).
11. Haas, J., Maus, H., Schwarz, S., Dengel, A.: ConTask - Using Context-sensitive Assistance to Improve Task-oriented Knowledge Work. *ICEIS (2)'10*. pp. 30–39 (2010).
12. Batley, J., Edwards, D.: Genome sequence data: management, storage, and visualization. *Biotechniques*. 46, 333–334, 336 (2009).
13. Champin, P.-A., Prie, Y., Aubert, O., Conil, F., Cram, D.: kTBS: Kernel for Trace-Based Systems, http://liris.cnrs.fr/publis/?id=5478, (2011).

Intelligent Interactions: Artificial Intelligence and Motion Capture for Negotiation of Gestural Interactions

Quentin Thevenet, Marie Lefevre, Amélie Cordier, Mathieu Barnachon

Université Lyon 1, LIRIS, UMR5205, F-69622, France

Abstract. Gesture-based interfaces allow instinctive use of applications but are often limited by their arbitrary configuration. To overcome this problem, we propose to design adaptive systems able to negotiate new gestures with users. For that, we want to develop an assistance engine supporting the process of defining new gestures on the fly. The role of the assistance engine is to permit negotiation of gestures between users and the system. To facilitate the negotiation we propose to use Trace-Based Reasoning. In this article, we present a framework to collect and reuse traces in a gesture-based environment.

Keywords: Traces, Trace-Based Reasoning, Gestural interfaces, Motion Capture, Interactions.

1 Introduction

The democratization of motion capture devices (such as the Microsoft Kinect^{TM}) makes gestural interactions increasingly popular. We observe this mode of interaction mainly in video games, but other applicative areas are emerging. Most systems that implement gestural interactions define a predetermined set of actions available and expect users to perform these actions. For end-users, interacting with such applications can be very frustrating. Indeed, if they cannot perform a gesture required by the system or if the set of actions is incomplete with respect to their needs, end-users may not be able to achieve their goals.

To address these problems, we propose to develop a system of gestural interactions capable of learning while it is used. Our initial system is bootstraped with predefined gestures. We combine is with an assistance engine supporting the process of defining new gestures on the fly. The role of the assistance engine is to permit negotiation of gestures between users and the system. As a consequence, our final system is able to adapt itself to its users. To facilitate the negotiation we propose to use Trace-Based Reasoning. Traces help us to infer users' needs and to provide them with a user-friendly and relevant assistance. We perform this work as a part of the project IIBM[1] (Intelligent Interactions Based on Motion [1]), which combines researches in the fields of motion capture and artificial intelligence.

[1] http://liris.cnrs.fr/iibm/

In this paper, we focus on interaction traces. We show how we collect traces of gestural interactions and how we use these traces to assist users in their interactions with the system.

The paper is organized as follows. We first illustrate our motivations with a practical example in section 2. Section 3 presents a brief state-of-the-art regarding traces on the one hand and assistance to users on the other hand. Section 4 presents our approach and our framework to collect and reuse of gestural interaction traces. Section 5 discusses more specifically the use of trace-based reasoning to provide relevant assistance. Section 6 presents our implementation. A discussion is given in section 7.

2 Motivating scenario

To illustrate the context of our work, we present a simple scenario. In this scenario, we assume that a user is interacting with PowerPoint™ by performing specific gestures. We make the assumption that we have a full environment enabling the user to do so. The system is able to recognize a set of predefined gestures and to associate them with specific actions within PowerPoint. Gestures are interpreted by a third-party software and are translated into instructions sent to PowerPoint (such as "Next slide").

When interacting with the system, the user may encounter several problems. These problems occur when movements are badly interpreted by the system. Causes of these problems are manifold. First, the gesture may be badly recognized by the capture system. This cause of failure is out of the scope of this paper. Next, the gesture may be badly interpreted. For example, the user moves his hand to the left, intending to perform a given action (e.g. "Next slide"), but the system performs another action. In this case, negotiation is needed to decide whether the system or the user is wrong. Another cause of failure is when the mapping between a gesture and an action is not available in the system. For example, a user might want to associate a wave gesture to the "Clear Screen" action. In this last case, negotiation is needed to enable the user to define new control gestures.

In this paper, we present a framework for collecting traces of gestural interactions. We show the mechanisms that exploit these traces to support the negotiation process between the user and the system. Our goal is to increase the adaptability of the system by supporting the creation of new gestures or the modification of existing gestures on the fly.

3 State-of-art

This section discusses the role of assistance in the design of adaptive systems. Then, it introduces the theoretical framework that we use to collect and exploit interaction traces.

3.1 Evolutive assistance

In a majority of research on assistance, assistance is defined as the system's ability to provide an answer to a problem given by the user. The role of the user is to provide information needed to find a solution [14]. This design of assistance is criticized because [2]: (i) it does not allow the user to acquire additional knowledge; (ii) it is contrary to the principle of practical assistance in a real situation (indeed, it is more useful to guide the user to find a solution rather than directly provide him with this solution [4]); and (iii) it does not allow the dialogue between human and machine that can guide and improve the search for solutions [10].

To overcome these limitations, an evolutive assistance must be proposed, *e.g.* an assistance able to adapt itself to the changing needs of users, assistance systems must be able to increase their knowledge over time [5]. According to [3], it is possible to use traces to propose an assistance adapted to user needs as the context evolves. Trace-Based Reasoning (TBR) [6] is an artificial intelligence paradigm similar to Case-Based Reasoning. It can solve new problems by reusing past experiences. In [5], the authors present an architecture for assistance TBR. This architecture relies on several knowledge bases that evolve during the use of the system. Reasoning mechanisms use these knowledge bases and thus improve their results over time. In our work, we use this principle to provide an assistance that can adapt itself to the user and evolve over time.

3.2 Interaction traces

Many studies focus on the production and exploitation of interaction traces. According to [11], a trace is as a set of observed elements temporally situated, called *obsels*. Obsels always have a timestamp. A trace model defines the structure and the *obsels* type that are contained in a trace, as well as the relationships between these *obsels*. A modeled trace, or M-Trace, is a trace associated with its trace model. There are two types of modeled traces.

Primary traces: the results of the *obsels* collection process. A primary trace of users actions may contain, for example, the *obsels*: “ctrl key”, “c key”, “ctrl key”, “v key”.

Transformed traces: traces that are produced from one or more source traces on which a transformation method is applied. A transformation method may for example be a temporal filter to keep only the *obsels* located in a given time interval. All traces can be transformed. For example, a transformation may convert the four previous *obsels* in a transformed *obsel* called “cut and paste”.

A Trace-Based Management System (TBMS) is a system managing traces [8]. A TBMS has three main components: the collection module, responsible for storing *obsels* into traces, the transformation module applies transformations on traces, and the query module allows the manipulation of traces.

The assistance engine, presented in the proposed framework (see subsection 6.2) is built upon user's traces, collected during its used of the targeted application.

4 A framework using traces for negotiating gestural interactions

In this section we present our framework for the negotiation of gestural interactions. This framework contains several features: support of interaction between the user and the target application, gestures interpretation module, interactions tracing module, and assistance module. First, we discuss the knowledge models that are used in the framework. Next, we show how the various components are connected.

4.1 Knowledge models

The framework uses several knowledge models. The targeted application model allows us to know actions that the user can do in the application, and the context in which these actions are available. For example, this model indicates that, when one is in "presentation mode" with PowerPoint, the actions "next slide" and "previous slide" are available.

The traces model defines the types of *obsels* that are contained in traces. All obsels are timestamped. Traces record all the interactions (gestures and keyboard events). Our trace model contains the following *obsels*.

- **Gestural event**: informations about the position, direction and speed of movement of each part of the user's body in order to transcribe the gesture made.
- **Keyboard event**: key code and the status of the key (pressed or released).
- **Mouse event**: information about mouse movements and state of the buttons.
- **Targeted application event**: information describing actions performed on the targeted application and parameters of actions.
- **Assistance event**: contains information about the assistance provided to the user and his response.

4.2 General framework

To provide assistance to a system based on gestures users, we propose the framework in Figure 1. In this framework, a user interacts with an **application** (see 1) using gestures captured by a **motion capture** system (see 2) and / or a standard interface such as a keyboard or a mouse.

The different interactions of the user are processed by the **interpretation engine** (see 3). The interpretation engine converts gestures into actions understandable by the target application. To do this, it seeks gestures done in the

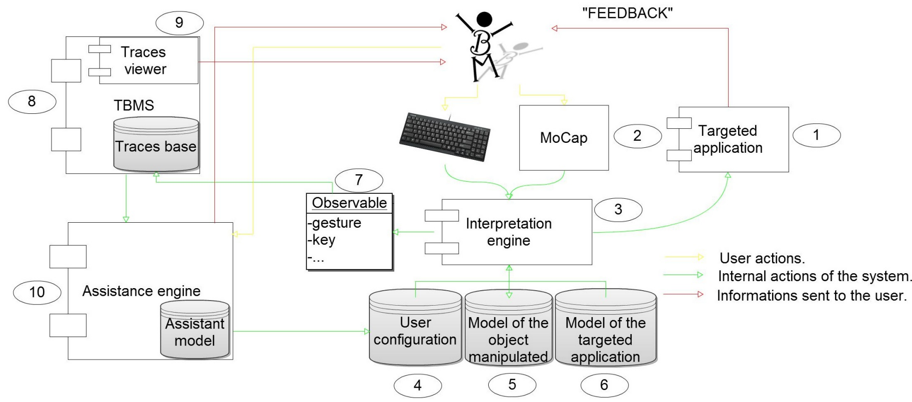


Fig. 1. Framework to use traces for negotiating gestural interactions.

configuration file of the user (see 4) to find the associated action on the target application. This module is also used to collect **observables** (see 7). The **model of the target application** (see 6) describes possible actions of the user on the application. The **model of the manipulated objects** (see 5) describes the current state of the target application. Based on these two models, the interpretation engine creates observables to describe events occurring on the target application. This module is also responsible for updating the model of the manipulated objects.

The **TBMS** (see 8) collects observables, creates *obsels* from these observables and builds primary traces by grouping them. It provides transformation mechanisms on traces. It also allows **visualization** of these traces (see 9) for the user. The **assistance engine** (see 10) searches, in traces, error situations. This module is the core of our assistance system and is described in detail in the next section.

5 Providing assistance based on gestural interaction traces

In this section we show how we can exploit gestural interaction traces to provide assistance to users of an interface based on gestures. It must be noted that the context in which is the user is important to decide if assistance must be provided or not. For example, if the user is doing a presentation in front of the audience, assistance must not interrupt him. It is then shut down. Deciding when to provide assistance is briefly discussed in section 5.2.

5.1 Using pattern recognition to identify assistance needs

First, we need to detect situations where user may need assistance. For that, we seek specific patterns in traces. These patterns detect failures during the use of the system (e.g. something went wrong), that is why we use the term failure pattern. For now, we identified two failure patterns.

The first pattern is called *inconsequential gesture*. It occurs when the users performs several times the same gesture before pressing a keyboard key. For example (see Figure 2), the user moves several times his arm to the right, intending to go to the next slide. But nothing happens. Finally, the user gives up and uses the right arrow on the keyboard to perform his intended action.

Several types of assistance can be provided in this situation. First, the system can show to the user the proper gesture to perform the action. Next, the system can interact with the user to adapt itself to the user's needs. Indeed, if the user cannot remember the gesture associated to an action, or if he does not like it, or if the gesture does not exist yet in the system, he can define a new one.

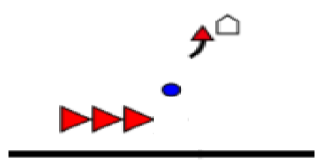


Fig. 2. This trace illustrates the pattern "inconsequential gesture"; red arrows correspond to the "moving arm to the left" gestures, blue round indicates a key pressed. The last arrow indicates that an action has been performed in the application (next slide), and the white pentagon shows that an assistance process has been triggered.

The second pattern we have identified is called *Action/Cancellation*. It corresponds to a succession of actions and cancellations. It occurs when the user performs unconsciously some gestures which cause unwanted actions. Consequently, they immediately cancel the action performed. In order to identify when a user cancels an action, we use the target application model. This model indicates, for each action, which is the reverse action. For example, it indicates that the reverse action of "next slide" is "previous slide".

Again, several types of assistance can be provided in this situation. Depending on the user's needs, we can offer him to change the mapping gesture / action either of the initial "action" gesture, or the "cancellation gesture", or even both.

Table 1 sums up the patterns we are able to identify for the moment and the various assistance possibilities we offer. In the following, we will show how trace-based reasoning will help us to dynamically identify more failure patterns.

5.2 Trace-Based Reasoning to improve assistance

Assistance presented in the previous section can be improved by using Trace-Based Reasoning (TBR) [6] to: identify the most appropriate assistance for a given failure situation, and identify the situations where the user needs assistance.

Failure	Assistance
Inconsequential gesture	show the gesture that enables the action; change the gesture that enables the action;
Action/Cancellation	change the gesture that makes the first action; change gesture that makes the second action; change both gestures;

Table 1. Pattern of failures situations and corresponding assistance.

When the assistance engine detects a pattern in interaction traces, it proposes to the user several assistances on the fly. The user may choose one option among them. For example, the user interacts with the system and the pattern X is detected. The system proposes several options of assistance A1,. . . ,An. Let's assume that the user chooses A2. If this situation is repeated several times, TBR can infer that the pattern X should be associated to assistance A2 by default. To give an actual example, we assume that the gesture for the action "next slide" is "move the right arm to the right", with a low amplitude. If the user moves a lot his arm while he speaks, he can switch to the next slide inadvertently. Therefore, he will cancel this unwanted action. TBR will detect a failure situation. Similar situations will be retrieved and reused to help the user. Here, the system will propose to the user to change the gesture amplitude for this action. The user is free to accept the modification or not.

If the user needs assistance, but is in a context that does not match any known pattern, the system cannot offer any support yet. In this case, TBR could exploit past experiences to provide relevant assistance. For that, the assistance engine will search contexts similar to the current one in traces. Then it will identify, in these contexts, the next actions to perform. The engine will adapt the actions according to the current context and recommend them to the user. For example, a user comes to a slide containing only a video. The reasoning system find in previous traces that usually, in this kind of slides, the users start immediately the video. The assistance system could therefore directly play the video and save the user an action. These two examples can be generalized by using a trace base of multiple users. By using TBR, it is possible to improve the assistance engine. TBR will allow us to discover new mappings between gestures and actions. In addition, it will allow us to discover new patterns of assistance.

6 Implementation

In this section, we present the framework we have implemented. We have used third-party tools for external tasks. The framework is implemented in C++. Up to now, we have implemented a gestural interface for PowerPoint and we have developed an assistance engine able to identify two failure patterns. We have experimented the system with ten different users but the results of this experimentation are not described in this paper.

6.1 Third-party tools

PowerPoint. In order to experiment our tool with all types of users, we chose to instrument a widespread tool. This is the reason why we decided to work with PowerPoint. Controlling PowerPoint with gestures is very intuitive. For example, we can move our hands to the left or right to move to the previous or next slide.

Kinect and FAAST. We decided to build on an existing tool to implement the motion capture component. The use of commonly depth cameras [12] was preferred to marker-based solutions, (like the Vicon system[2]) which need special equipments (expensive infra-red cameras and special suit), and are much more expensive. Our choice was motivated by the fact a device such as Kinect allows an immediate and instinctive interaction with the interface. Furthermore, it can be used in real-time [9], contrary to marker-based solutions. FAAST [13][3] is a free middleware, which allows integration of control gesture-based in the fields of video-games and virtual reality. FAAST emulates a keyboard by binding body posture and simple gestures to keys of a regular keyboard. Customized controls are defined in a configuration file (mapping between gestures and keys). Here, FAAST is configured to associate twelve gestures to keys. In order to avoid confusion with the actual keyboard, we mapped the gestures to keys that do not appear on a classical keyboard (F13 to F24).

Abstract Lite. Abstract Lite [7] is used to propose feedback to the user by showing him his own interaction trace. This graphical visualization enables the user to better understand how the system behaves.

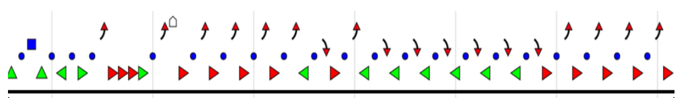


Fig. 3. Trace view with Abstract Lite.

Figure 3 shows an example of traces from our demonstrator in Abstract Lite. The blue square represents the beginning of the presentation. Triangles indicate gestures made by the arms. Arrows indicate transitions to next or previous slides.

6.2 Framework implementation

Interpretation engine. The Interpretation engine uses Windows API to collect keyboard and mouse events. It converts gestures into PowerPoint keyboard shotcuts according to the configuration file of the user. It uses a PowerPoint model and a current presentation model. The PowerPoint model allows to know the possible user's actions in the current state of presentation. The state of the presentation is saved in the model of the current presentation.

[2] http://www.vicon.com
[3] http://projects.ict.usc.edu/mxr/faast/

Trace-based management system. The TBMS transforms observables collected by the interpretation engine into *obsels*. *Obsels* are then stored in an XML trace. The trace-based management system allows us to perform requests on traces to exploit them.

Assistance engine. The assistance engine exploits traces to discover new knowledge. For the moment, it only implements a pattern recognition mechanism to discover failures in traces. Moreover, it offers a list of assistances when one these patterns is detected. We have implemented two patterns: the "Inconsequential Gesture" pattern and the "Action/Cancellation" pattern.

7 Discussion and conclusion

In this paper, we presented a generic system to assist and optimize the use of a gestural interface. To enable this support, all user's actions with the system are traced. Collected traces allow the system to identify the context in which the user is in, and to detect if assistance is needed. We have developed a framework to experiment with gesture-based interfaces. This framework contains a target application, a gesture interpretation module, a trace-base management system, and an assistance module. So far, we have implemented a first level of assistance based on traces. However, this type of assistance remains static. It is limited to the identification of predefined failure situations.

Therefore, it is necessary to develop mechanisms to enhance assistance on the fly. We propose to implement trace-based mechanisms, as suggested in [6]. A first idea is to reason on traces to automatically detect situations in which assistance should be provided. A second idea is to look for patterns in traces and to exploit these patterns to improve the usability of the system. For example, the system can identify that a user has to perform several gestures to perform a single action. In this case, a negotiation process can be triggered in order to map a new gesture (chosen by the user) to this action. More generally, many assistance scenarios, traces-based, can be imagined. Traces can be used to capitalize on the experiences of other users and to reuse these experiences.

When providing assistance to users, the main problem is to ensure that assistance does not disturb the user activity. Therefore it is necessary to find ways to trigger assistance timely, on purpose. Moreover, we need to consider the differences between users. The main perspective of this work is to explore ways of providing assistance to users, in the most pleasant way possible.

References

1. Barnachon, M., Ceccaroli, M., Cordier, A., Guillou, E., Lefevre, M.: Intelligent Interactions Based on Motion. In: Belén Diaz-Agudo, A.C. (ed.) Workshop CBR and Games, ICCBR 2011 (Sep 2011)
2. Cahour, B., Falzon, P.: Assistance à l'opérateur et modélisation de sa compétence. Intellectica 12(2), 159–186 (1991)

3. Champin, P.A.: ARDECO: an assistant for experience reuse in Computer Aided Design. workshop From structured cases to unstructured problem solving episodes - WS 5 of ICCBR'03, Trondheim (NO) pp. 287–294 (2003)
4. Coombs, M., Alty, J.: Expert systems: an alternative paradigm. International Journal of Man-Machine Studies 20, 21–43 (1984)
5. Cordier, A., Lefevre, M., Jean-Daubias, S., Guin, N.: Concevoir des assistants intelligents pour des applications fortement orientees connaissances : problématiques, enjeux et étude de cas. In: Despres, S. (ed.) IC 2010 - 21emes Journées Francophones d'Ingenierie des Connaissances. pp. 119–130. Presses des Mines (Jun 2010)
6. Cordier, A., Mascret, B., Mille, A.: Extending Case-Based Reasoning with Traces. Tech. Rep. RR-LIRIS-2009-005, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université Lumière Lyon 2/École Centrale de Lyon (Mar 2009)
7. Georgeon, O., Mille, A., Bellet, T., Mathern, B., Ritter, F.: Supporting activity modelling from activity traces. Expert Systems (Jun 2011)
8. Laflaquière, J., Settouti, L.S., Prié, Y., Mille, A.: A trace-based System Framework for Experience Management and Engineering. In: Second International Workshop on Experience Management and Engineering (EME 2006) in conjunction with KES2006 (Oct 2006)
9. Raptis, M., Kirovski, D., Hoppes, H.: Real-time classification of dance gestures from skeleton animation. In: Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation (August 2011)
10. Roth, E., Bennett, K., Woods, D.: Human interaction with an intelligent machine. Cognitive engineering in dynamic worlds 20 (1987)
11. Settouti, L.S., Prié, Y., Champin, P.A., Marty, J.C., Mille, A.: A Trace-Based Systems Framework : Models, Languages and Semantics. Research report, LIRIS, SYSCOM (2009), http://hal.inria.fr/inria-00363260
12. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: CVPR (2011)
13. Suma, E., Lange, B., Rizzo, A., Krum, D.M., Bolas, M.: FAAST: the flexible action and articulated skeleton toolkit. In: IEEE Virtual Reality. pp. 245–246. Singapore (Mar 2011)
14. Woods, D., Roth, E.: Aiding human performance ii: From cognitive analysis to support sytems. Le Travail Humain 51(2), 139–172 (1988)

Capturing and Adapting Traces for Character Control in Computer Role Playing Games

Jonathan Rubin and Ashwin Ram

Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
Jonathan.Rubin@parc.com, Ashwin.Ram@parc.com

Abstract. We describe an architecture, in its early stages of development, that processes user traces in the domain of computer role playing games and utilises the resulting traces in order to control the behaviour of characters within the environment. Behaviour execution is handled via an online case-based planner, which dynamically adapts plans given dissimilarities between the learning and testing environments. The overall architecture is presented and we provide an example of applying the architecture to a 2D role playing game environment. We conclude with the future objectives of this work in progress. Our work builds heavily on previous research in the area of learning from demonstration and online case-based planning in real-time strategy games [1].

1 Introduction

In this paper, we detail the efforts of a work in progress in the area of learning from demonstration and case-based planning. We describe an architecture for performing online case-based planning within the domain of modern computer role-playing games. The overall purpose of the architecture we describe is to control game characters based on captured user traces. During demonstration a human expert controls a character within a virtual environment and a trace is recorded to capture their sequence of actions. The user traces gathered are combined with a real-time case-based planner, which results in the generation of similar strategies which can be used to influence the behaviour of autonomous agents within the environment.

Our work builds on previous research efforts that have produced the Darmok [1, 2] and Darmok 2 [3, 4] systems. Darmok describes an architecture for performing online case-based planning based on capturing expert user traces. Darmok has been shown to be successful in producing coherent strategies, especially in the domain of real-time strategy games (RTS). The work we present here differs from Darmok in that it describes an architecture for controlling computer characters in role playing games (RPG). While Darmok 2 was able to be used as a general game player, it was particularly suited for playing RTS type games. The eventual goal of our work is to construct a system that controls one (or more) helpful, non-player characters (NPCs), which are able to aid human players with their goals and objectives in the domain of RPG games.

While our work is heavily influenced by research that has been conducted within the domain of RTS games, there are several important differences that result, given the modified objectives and the differences that exist between RTS and RPG domains.

1. To begin with, RTS environments are adversarial, whereas RPGs may not necessarily be so. While RPG games may contain adversarial scenarios (which are required to be handled by the system) the overall objective typically has more to do with space exploration and the appropriate selection of sequences of actions.
2. RTS games require the coordination of a team of agents, typically with the objective to destroy an enemy team. On the other hand, RPG games place a larger focus on the actions and goals of more well defined individual characters that exist within the environment.
3. Actions within RPG games are typically instantaneous as opposed to durative (as in RTS games). As such, there is less of a focus on the parallel management of durative actions (as in RTS games) and more of a focus on the appropriate selection of sequences of actions and goals to pursue.

We refer to our architecture (and the system it produces) as Komrad[1] and the next section provides a high-level overview of its design.

2 Komrad Overview

Figure 1 displays a high level overview of the current system architecture that consists of a training phase, a real-time planner, as well as adaptation & repair strategies applied within a particular environment. Each of these sections are described in more detail.

2.1 Training

During the initial training phase, a user is able to demonstrate behavior to the system by navigating and controlling a character within the current environment. An environment is composed of a collection of entities, together with a set of actions, which are able to be performed by the user in order to modify the current world state. During this initial training episode traces are captured, which record each action that was chosen by the user.

In addition to the set of possible actions that a user can take within the environment, a collection of goals are also specified that reflect more sophisticated milestones achieved by the user during their interaction with the environment. At present, all goals are required to be pre-specified and known beforehand. However, one of the eventual objectives of our research is to remove this assumption. The series of actions that result from recording a trace episode are processed into cases. Cases have the following representation:

[1] Darmok backwards.

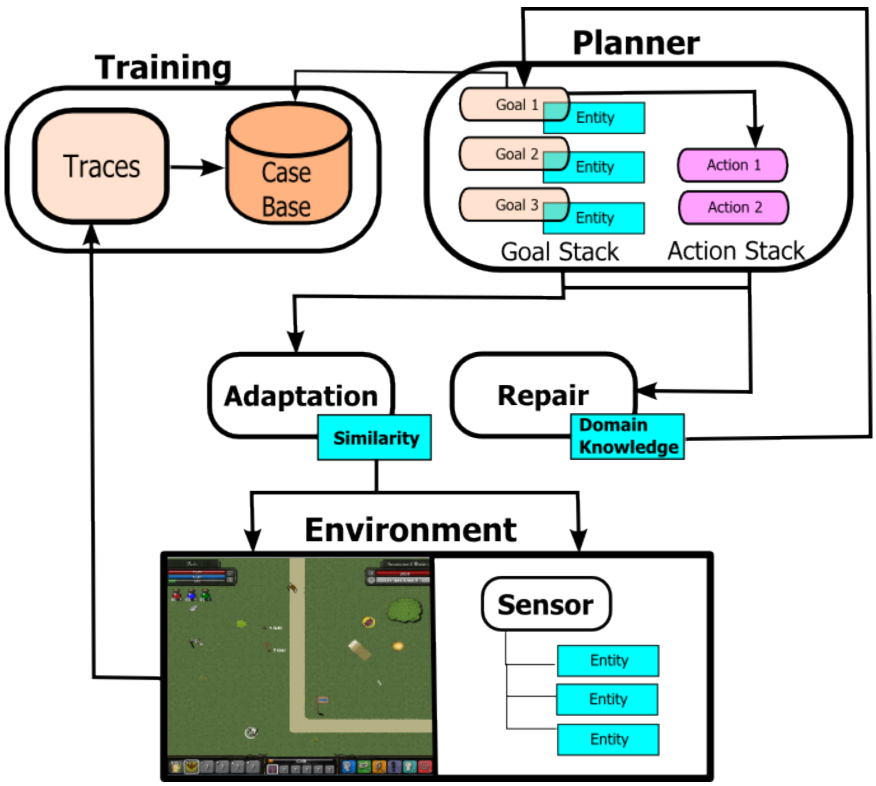


Fig. 1. Highlevel overview of the Komrad architecture.

$$C = (W, G(E), S)$$

Where, W, captures the current world state at the time goal, $G(E)$, was achieved by the user and, S, is the sequence of actions or sub-goals that led to the achievement of the goal. The $G(E)$ notation further highlights that each goal also specifies a single entity within the environment that it acts upon. Entities are described by a collection of attribute value pairs.

The cases produced become the foundation for controlling the behaviour of an autonomous character that will reflect the style of play of the original expert who was used to capture the trace.

2.2 Real-Time Planner

In order to control the behaviour of an autonomous agent the system architecture depicted in Figure 1 includes a real-time planner (top right). The planner within the current architecture functions by maintaining a goal stack and an action stack, where the actions currently present on the stack are required to be performed in order to achieve the goal at the top of the goal stack.

At the start of a planning episode a single goal is placed onto the goal stack. During the episode the planner is continually queried for the next action it recommends. If there are currently no actions on the action stack, the goal at the top of the goal stack is decomposed into the sequence of actions and/or sub-goals that are required to achieve the particular goal (recall that this information was captured within a case in the case-base). In order to decompose the goal at the top of the stack, the case-base is searched for stored cases whose goals ($G(E)$) and world state (W) are similar to the current environment. Once an appropriate case has been found, the sequence of actions or sub-goals recorded by the retrieved case (S) are placed onto their appropriate stack within the planner.

Goal decomposition continues until at least one action is present on the action stack, at which point the action at the top of the stack is returned by the planner. A goal is removed from the goal stack once all the actions required to be performed to achieve the goal have been popped off the action stack. One limitation of the current architecture is that goals can only be decomposed into a sequence of sub-goals or a sequence of actions, but not a mixture of the two, as this could result in obfuscating the order in which actions should be performed, according to the user traces.

2.3 Adaptation & Repair

Once an action is retrieved from the action stack it is ready to be performed in the current environment. However, as the current environment is likely to be different from the environment initially encountered when gathering traces, adaptation needs to take place to ensure that the action that is performed is suitable for the current world state.

Recall that each goal acts upon a particular entity within the environment and that each action on the action stack is associated with the achievement of the goal at the top of the goal stack. The first thing that is required in order to adapt the action is to first adapt the entity of its associated goal. This occurs by sensing the entities in the current environment and determining the similarity between the goal's entity (recorded from the trace) and the entities that currently exist. One of the advantages of this approach is that it allows behaviour adaptation to occur within dissimilar environments by utilising simple similarity metrics associated with the features of particular entities. Once a goal's entity has been adapted to better reflect entities within the current environment it remains to also adapt the sequence of actions required to achieve the updated goal. Each action defined within the environment specifies its own adaptation procedure that takes into account the entity of its corresponding goal. Action adaptation either succeeds or fails in the given environment. If the adaptation succeeds the action can be executed within the environment. On the other hand, if the adaptation fails for any reason a **repair** is required.

Actions can supply optional repair strategies in the event that an adaptation failure occurs. Repair strategies work by specifying goals that should be placed onto the goal stack before the goal at the top of the stack can be achieved. Repair strategies typically require some domain knowledge to be encoded into the system. Figure 2 illustrates the outcome of applying a repair strategy.

The basic idea behind repair strategies is that they allow a dynamic restructuring of the goal stack. The event of an adaptation failure highlights the fact that something observed within the original environment (i.e. observed when gathering traces) is not reflected within the current environment. A repair strategy will attempt to modify the current environment to better align it with the original observed environment. It does so by specifying intermediate goals which need to be achieved before the current active goal (i.e. the goal at the top of the goal stack) can be attempted.

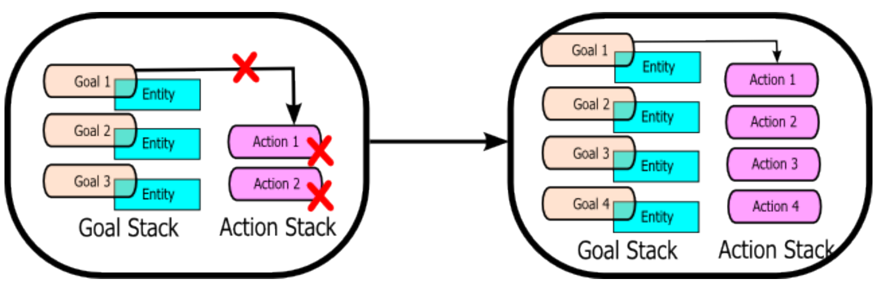


Fig. 2. In the event of an adaptation failure, a repair strategy can directly modify the goal stack to ensure prerequisite goals are achieved before attempting the goal that initially failed.

Like adaptation, repairs can either succeed or fail. In the event of a successful repair, all actions on the action stack will be removed and any intermediate goals required by the repair will be pushed onto the goal stack in order to be decomposed. In the event of a repair failure, the current active goal is considered unachievable and is popped off the goal stack.

3 Application to a 2D RPG

The previous section introduced a high level overview of the Komrad architecture. In this section, we further describe the details of applying the above architecture within the domain of a 2D role playing game. For our initial development and experimentation with the Komrad architecture we have used a 2D tile-based RPG game known as Mystik RPG (http://mystikrpg.com/). One of the reasons for choosing Mystik RPG as the initial experimental domain was due to the fact that it is open source and it offers a simplified, but extensible domain which was useful for early experimentation.

In the game, players control a character within a two dimensional world. Within the world actions can be performed such as moving a character up, down, left and right; opening entrances and teleports; picking up items such as weapons, armour and keys; equipping and dropping items and fighting monsters. Mystik RPG also provides a graphical tile map editor for creating new maps to train and test on.

Figure 3 depicts a snapshot of a currently executing plan within the game of Mystik RPG. At the top of the goal stack is the OPEN goal. Recall that each goal is associated with an entity within the environment. The OPEN goal acts upon an entrance entity within the environment. Given spatial restrictions, the attributes associated with the 'entrance' entity have not been depicted in Figure 3, however this type of entity would be described by attributes such as an (x, y) position within the world and other state information, such as whether the entrance was locked or unlocked. As the OPEN goal is currently on top of the goal stack, it has a sequence of actions, on the action stack, that are required to be performed in order to achieve the OPEN goal. In this case only two actions are required to achieve the goal:

1. Move to a particular position within the world, and
2. Perform a directional open action depending on where the entrance exists in relation to the character.

As the MOVE action is at the top of the action stack, this would be the next action that is attempted by Komrad. However, the planner as depicted in Figure 3 currently represents a plan that was witnessed within the original training environment, which may be dissimilar to the current world state. As such, the current goal and its actions need to be adapted to better reflect the entities that exist within the present environment.

First, the entities within the current environment are sensed. Next, the entity that the current goal acts upon (i.e. the entrance) is adapted to an entity that

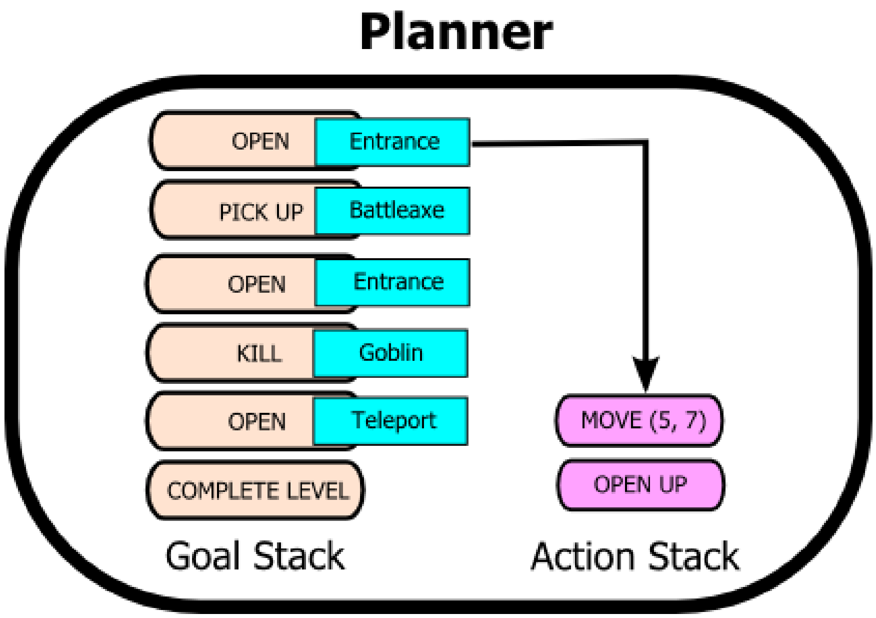


Fig. 3. A snapshot of a real-time case-based planner within the game of Mystik RPG.

exists within the current environment. Within our architecture, entities dictate their own adaptation procedures via similarity assessment. For example, in Figure 3, the entrance entity associated with the OPEN goal, would be updated to the most similar entrance entity which exists in the present world state. This could be the exact same entrance (if the training and testing environment were identical) or it could be the entrance within the current environment that is closest to the location of the original entrance and exhibits the same state (i.e. locked or unlocked). Once the goal's entity has been updated, it remains to update the details of the atomic actions required to achieve the goal. Once again in Figure 3, the (x,y) coordinates of the MOVE action need to be updated to reflect the new goal entity (i.e. the adapted entrance). Also, the OPEN UP action would need to be adapted, in case the new entrance entity is no longer located above the player, but rather to their left or right or below them. In the event that either of the above actions could not be successfully adapted for the current environment e.g. the path was blocked to the entrance or the entrance required a key to open it, this would require a repair to take place. Repairs require domain knowledge; in this example a possible repair strategy could involve pushing a new PICK UP or OPEN goal to the goal stack in an attempt to satisfy the prerequisites of the current OPEN goal.

4 Related Work

The Komrad architecture introduced above is a work in progress and requires further development and testing. As well as building on the work of Ontañón et al. [1–4], the work presented here is related to other research efforts that focus on case-based planning [5], case-based plan adaptation [6] and learning from user traces in the domain of computer games [7–9].

One such work is that of [10] who presents a learning by observation framework (jLOAF) that can be applied within a range of environments. Actions of an expert are observed within several domains, including robotics, simulated soccer and the game of Tetris. The results reported by [10] indicate that the jLOAF framework can successfully re-use the actions of the original expert. jLOAF's focus is on the construction of a reasoning by observation framework that is domain independent. It does not use a case-based planning architecture, as our work does. Instead, the generality of the framework makes it more suited for reactive domains.

Learning from demonstration has recently been investigated in relation to goal-driven autonomy by [9]. Weber et al. [9] describes a system based on the conceptual model of goal-driven autonomy that utilises expert traces in order to reduce the amount of domain knowledge that is typically required by the GDA model. The real-time strategy game of StarCraft was used as the experimental domain. While relevant to our own work due to the utilisation of learning from demonstration, the focus of [9] is on the GDA conceptual model, which is not the case with the work we have described.

5 Conclusion and Future Work

We have presented an early stage architecture for gathering user traces within an interactive environment and utilised those traces within an online case-based planner in order to reproduce observed behaviour. At present, we have applied the architecture to a simple 2D tile-based RPG game. While the architecture described is still in the early stages of development, we have successfully been able to re-use and adapt traces recorded within the environment. The application of the Komrad architecture to this domain was described in Section 3.

One of the future objectives of this work is to construct a system that controls one, or more, helpful, non-player characters, which are able to aid human players with their goals and objectives in role playing games. While the application of the Komrad architecture to the environment we have described in this paper has been useful for initial experimentation and evaluation purposes, it is nonetheless too simplified a domain to fully evaluate our approach. Future work will involve the application of the architecture to a more sophisticated role-playing game domain. We have chosen the game Hands of War 2[2] as our future test-bed for the work, given its larger scope. Within this domain we intend to use traces in order

[2] http://www.axis-games.com/

to identify when a player requires help, determine the type of behaviour required to assist the player and, finally, to execute the appropriate helpful behaviour via auxiliary characters in the environment.

Acknowledgments. The authors wish to acknowledge the efforts of those responsible for the creation of Mystik RPG. We also wish to thank Michael Youngblood, Danny Jugan and Axis Games for kindly providing the source code for the game Hands of War 2.

This material is based upon work supported by the National Science Foundation under Grant No. (IIS-1216253). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: On-line case-based planning. Computational Intelligence **26**(1) (2010) 84–119
2. Ontañón, S., Mishra, K., Sugandh, N., Ram, A.: Case-based planning and execution for real-time strategy games. In: Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007. (2007) 164–178
3. Ontañón, S., Bonnette, K., Mahindrakar, P., Gómez-Martín, M.A., Long, K., Radhakrishnan, J., Shah, R., Ram, A.: Learning from human demonstrations for real-time case-based planning. In: IJCAI-09 Workshop on Learning Structural Knowledge From Observations. (2009)
4. Ontañón, S., Ram, A.: Case-based reasoning and user-generated ai for real-time strategy games. In González-Calero, P.A., Gómez-Martín, M.A., eds.: Artificial Intelligence for Computer Games. Springer-Verlag (2011) 103–124
5. Hammond, K.J.: Case-based planning: A framework for planning from experience. Cognitive Science **14**(3) (1990) 385–443
6. Muñoz-Avila, H., Cox, M.T.: Case-based plan adaptation: An analysis and review. IEEE Intelligent Systems **23**(4) (2008) 75–81
7. Floyd, M.W., Esfandiari, B., Lam, K.: A case-based reasoning approach to imitating robocup players. In: Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference. (2008) 251–256
8. Rubin, J., Watson, I.: On combining decisions from multiple expert imitators for performance. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence. (2011) 344–349
9. Weber, B.G., Mateas, M., Jhala, A.: Learning from demonstration for goal-driven autonomy. In: Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI-12). To Appear. (2012)
10. Floyd, M.W., Esfandiari, B.: A case-based reasoning framework for developing agents using learning by observation. In: IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011. (2011) 531–538

Acquisition of Cases in Sequential Games using Conditional Entropy

Luc Lamontagne, Francis Rugamba & Guy Mineau

Université Laval, Québec, Canada
{luc.lamontagne, guy.mineau }@ift.ulaval.ca
francis.rugamba.1@ulaval.ca

Abstract. In this paper, we present an experiment we conducted on the acquisition of cases in sequential game environments. We describe an approach where demonstration traces are segmented into cases without supervision. Formation of the traces is performed using unsupervised clustering of game states and is guided by the conditional entropy of the sub-sequences. Application of this approach to a real-time reactive game indicate that the performance of the resulting case bases are comparable to a reactive CBR structure while reducing substantially the number of cases obtained from the acquisition phase.

1 Introduction

Developing game characters is a difficult task due to the increasing complexity of the game environments and scenarios. Game developers have to face challenges to create characters with interesting and intelligent behaviours. A promising direction to reduce programming complexity is to develop video game characters using machine learning techniques.

One paradigm gaining popularity for this purpose is learning by demonstration [2][5][7]. In such an approach, the system observes a teacher performing a task and cumulates information from this demonstration. The teacher can either be a human or some external components. The sequences resulting from the demonstration, usually obtained from an execution trace, indicate the various situations that were tackled by the teacher and prescribe the corresponding actions that were taken. Such demonstrations can be used subsequently for constructing a software artefact that can be reused to control some character behaviours.

In this paper, we concentrate on the problem of constructing cases from traces generated in a sequential environment. Our work can be interpreted in two different ways. The first interpretation would be that to build cases from traces where the action recommendations are of varying length. In this setting, each case would cover an arbitrary number of episodes in the traces. In the second interpretation, we aim to build state-transition diagrams (STD) approximated by a set of cases. Here the main issue would be to choose, based on some characteristics of the traces, a compact set of states. But, from a technical point of view, both interpretations are almost equivalent as the cases are constructed from unsupervised segmentation of the traces.

In our experimentations, we evaluate our approach for Pac-Man, a reactive pursuit game. Pac-Man constitutes an interesting test bed as it offers a real time dynamic environment and it involves sequential decision making.

The paper is organized as follows: the next section presents an overview of the proposed approach. In Section 3 we describe how the approach was applied to the game of Pac-Man. Section 4 explains how segmentation is performed using unsupervised clustering and conditional entropy (an information theoretic measure). Finally, experimental results are presented in Section 5 to illustrate how this scheme compares to a reactive approach for case acquisition. The paper ends with a conclusion and some recommendations for future work.

2 Building Cases from Demonstration Traces

As mentioned previously, learning by demonstration was adopted to acquire traces from game plays. The steps required to implement this approach in a dynamic game environment are the following:

(a) To acquire game traces by observing demonstrations from a teacher indicating how some non-player characters (NPC) should behave during a game;
(b) To construct, from the game traces provided by the teacher, an intermediate structure that can be reused to build a decision policy and to activate the NPCs.
(c) To modify the decision structure in order to optimize either behaviour performance or decision time.

As illustrated in **Fig. 1**, the traces acquired in step (a) consist of a sequence of episodes where each episode describes the states of the game board and the action taken by the teacher.

A conversion of the trace into an intermediate representation is required to build a decision policy. We make use of cases to make this representation. CBR offers various techniques to reuses episodes in new situations and to modify the structure of the case base to optimize the performance of the system.

From a CBR point of view, each pair <*game state*, *teacher action*> of an episode can be associated to a single case. In this acquisition strategy, a reactive case corresponds to a single episode.

Another strategy is that a case represents multiple consecutive episodes. A monolithic case could represent a full trace [6]. However, as our game traces are long and can contain a few thousands of episodes, we must provide some techniques to break down the traces into sub-sequences. We propose in Section 4 a technique to break down the sequences into cases containing a variable number of episodes.

As efficient time response is required, video game designers often make use of finite state machines (or state-transition diagrams - STD) [3] to compile the decision policy of non player characters (NPC). With slight modifications to a case representation, it is possible to approximate a state machine using a case base. This is the approach we adopt in this work as we structure cases to represent the elements of a state machine while we exploit the sequence of states observed in the traces to form the case base.

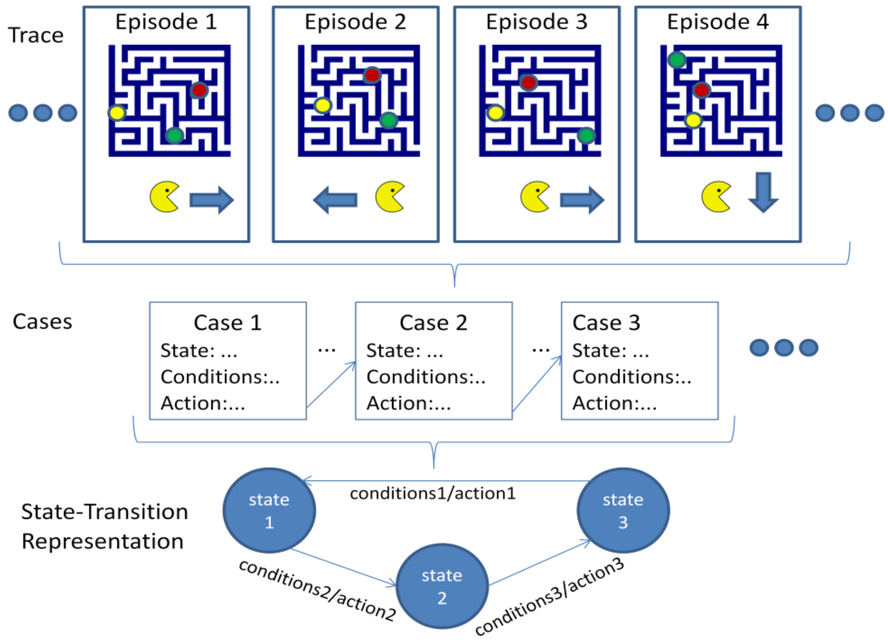


Fig. 1. A trace converted as a sequence of states and actions

A state machine is a directed graph consisting of states related to each other by transitions. We consider Mealy machines where states are associated to symbol(s) observable from the environment. And transitions have two different parts:

a) some conditions that should be satisfied to enable a change of state,

b) an output symbol to be emitted when activating the transition.

To approximate such diagrams with cases, we must map states and transitions to case features. In our experiments, each case is divided into 3 parts as follows:

- We select one or more attributes of the problem description to represent the states of the STD. Hence the set of states correspond to the combination of values for these attributes.
- The remaining <*attributes*, *value*> pairs of the case problem represent the conditions of the transition.
- The action recommended by the teacher (the solution part of the case) is the output symbol of the transition.

Given the current state of the game board, a NPC decision module would consider all the cases associated to this state and would choose the action(s) recommended by the case having conditions most similar to the current game configuration.

To build cases that are not limited a reactive formulation (i.e. having a single action as solution), we propose in Section 4 a scheme where we identify states of the game that are highly correlated. Correlation of the states is evaluated using conditional entropy, an information theoretic measure. Based on these correlations, we merge corresponding cases together. Case merging produces a more compact case base where each case designates a subset of episodes coming from one of the original traces. This scheme could also be considered as a segmentation of the traces into cases of variable length. Before to describe this case acquisition scheme in further detail, we explain in the next section how we applied the general acquisition approach to the game of Pac-Man.

3 Application to a Reactive Game – Pac-Man

To conduct our experiments, we made use of Pac-Man as a test bed. The task for this pursuit game is to move a yellow circled character in a maze to eat dots and fruits while avoiding some ghosts. The structure of the maze does not change during the game (**Fig. 2**). The actions of the Pac-Man correspond to the four possible moves in the maze: up, down, left and right. The motion of the ghosts is pseudorandom and the behavior rules change from one implementation to another. To conduct our experiments, we modified an implementation of this game developed by Benny Chow [1] to extract traces from game demonstrations.

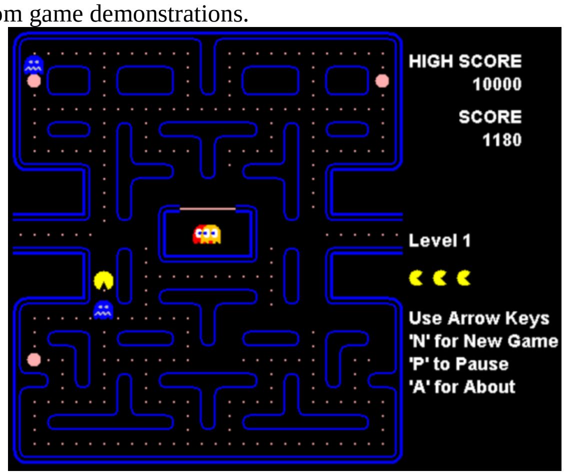


Fig. 2. Configuration of the Pac-Man board

We developed a computer player to make the demonstrations. Given the current location of the Pac-Man, this agent selects a target on the board, i.e. the closest object among the dots, the fruits, the power pellets or the edible ghosts. Then it uses a best-first search algorithm to determine the shortest path to the target object while avoiding the ghosts (if they are in attacking mode). Pac-Man applies the actions to follow this path as long as the target remains unchanged.

3.1 Representation of the state of the game board

To represent traces provided by these demonstrations, we selected some of the attributes found in the literature [4][9] to model the game board:

- *Location*: an identifier designating the current location (the cell) occupied by the Pac-Man.
- *Power*: the state of the power pellet;
- *Wall states*: four (4) attributes indicate if neighbouring cells are accessible (open or closed);
- *Ghosts distance*: some attributes indicating the Manhattan distance between the Pac-Man and the four ghosts. Special variables indicate the distance to the nearest ghost and if this ghost is edible or in attacking mode;
- *Dot distance*`: two variables indicate the distance between the nearest power pellet and fruit.

3.2 Using reactive cases as an approximation of a state machine

From game play traces, we initially form one case for each episode. Doing so, we adopt a reactive approach as each case recommends only one action to cope with a situation. All the episodes of the traces are stored in the case base.

As each case must represent a pair <*state*, *transition*> to approximate a state machine, the cases are structured as follows:

- *State*: the location of the Pac-Man character on the game board.
- *Time*: some index indicating the position of the episode in a trace.
- *Conditions*: some attribute values pairs describing the state of the game board as presented in the previous section
- *Action*: the move applied to Pac-Man.
- *Outcome state*: the location where Pac-Man ended up after applying the action.

Hence the case base represents the various transitions that were observed in the state space during the demonstrations. While this approach seems attractive, the size of the case base represents a limitation from a computational point of view. In our experiments, we generated some case bases containing tens of thousands of cases. This size can represent a serious limitation for applying CBR in real-time games. We investigate in section 4 how to merge cases together to reduce the case base.

3.3 Playing with reactive cases

To perform reasoning with cases structured as state transitions descriptions, we reuse the nearest neighbour from a partition of the case base corresponding to the current state of the game. To do so, the following steps are performed:

- Given the current state of the game board (in our case, the cell containing Pac-Man), fetch all the cases representing transitions originating from this state;

– Compare condition attributes of the cases to their corresponding values on the game board to determine the similarity of each candidate case.
– Apply the action recommended by the case having the most similar conditions.

In our experiments, the similarity of a case corresponds to the proportion of attributes having values corresponding to the game state. Uniform weights were assigned to the attributes.

4 Segmentation of Traces using Conditional Entropy

The scheme presented in the previous section relies on the usage of reactive cases to approximate state machines. This approach results in thousands of cases, each of them recommending a single action. Hence case retrieval must be performed at each frame of the game to determine the actions of the non player character.

We investigated the possibilities to reduce the number of cases in the case base. Analysis of the game traces reveals some interesting aspects that can be exploited. Game traces contain the complete history of states and actions performed during some games. One can try to determine some frequent patterns of states in the traces to determine sub-sequences of episodes that would form cases.

In our application, this would correspond to finding some regularity in the moves of the NPC. Analysis of the sequence of episodes can help to identify positions that are often observed in sequence. We also denote the fact that there is little game play variation in some parts of the game board. This might be explained by the structure of the game board containing long corridors and few intersections. If such-sub-sequences exist, we propose to identify them and to group corresponding episodes as individual cases. We describe in the next paragraph how we used hierarchical clustering and information theory to address this problem.

4.1 Finding correlations in consecutive episodes

To identify sequences of states that are correlated, we evaluate the level of uncertainty associated to specific trajectories. Given that some episodes with state x often lead to episodes with state y, then we can consider that the uncertainty associated to state x is low. In the context of Pac-Man, if one player frequently chose the same direction at one specific intersection, then decision behaviour at this intersection cell is highly predictable (which correspond to low uncertainty for this state). On the other hand, if the sub-sequences of the traces leading to a specific position y are much diversified, then uncertainty on how games can end up at state y is higher. To make these intuitions operational, we use conditional entropy to measure the level of uncertainty related to variations in sequences.

Entropy [8] is an information theoretic measure that estimates the amount of disorder, or uncertainty, related to the distribution of random variables. Given a variable X and a probability distribution indicating how likely the variable can take different values x_i, entropy $H(X)$ is given by

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

where $p(x_i)$ is the frequency of state value x_i in the demonstration traces.

Conditional entropy quantifies the entropy remaining on a variable X given that some random variable Y has been observed. Marginal conditional entropy is the same measure defined for some specific values, and is formulated as follows:

$$H(X = x|Y) = -\sum_{i=1}^{n} p(x|y_i) \log p(x|y_i)$$

By using marginal conditional entropy in our Pac-Man application, we intend to measure if moving from location x to location y during game plays is highly predictable (or unpredictable). Lower conditional entropy values would indicate that players often move from y to x, or equivalently that there is low uncertainty that has x follows y in the demonstration traces. If low entropy values are observed in the traces, we suppose that a case could be formed each time the two states are observed consecutively in the traces without losing too much information or game play performance.

To measure the dependency between two consecutive locations y and x, we measure the residual uncertainty RU which is the reduction in entropy when knowing that location y precedes location x:

$$Residual\ uncertainty(x, y) = RU(x, y) = \frac{H(x|y)}{H(x)}$$

And we merge cells for which the average value of $RU(x,y)$ and $RU(y,x)$ is the lowest.

4.2 Merging Cases using Conditional Entropy

To determine the states, we apply an agglomerative hierarchical clustering algorithm that successively group states together (i.e. merge cells of the game board as new states). Given a threshold value indicating acceptable levels of residual uncertainty, the algorithm determines a set of states as follows:

(a) Form a set containing states for each possible location on the board.
(b) Compute the residual information between each pair of state that can be observed consecutively in the traces.
(c) While some states can still be merged together:
  (i) Find the two states with the lowest residual information;
  (ii) Remove them from the set of states;
  (iii) Form an new state by grouping them together and add it to the set;
  (iv) Evaluate the residual information of the new state with respect to all the neighbouring states;
(d) Once completed, remove from the set all states with residual entropy higher than some threshold values.

Fig. 3 illustrates the first few iterations of the algorithm when applied to Pac-Man where neighboring cells having low entropy values have been merged together to form new states of the game.

Once a set of combined states have been constructed by this algorithm, cases corresponding to the states are merged together. A new combined case contains the following attributes:

- *State*: the locations merged together on the game board;
- *Time*: the time index when the first location in the combined state is first occupied;
- *Conditions*: the state of the game board when the combined state was first entered;
- *Action*: the list of moves applied to the PacMan while staying in the combined state.
- *Outcome state*: the location where PacMan ended up after applying the action.

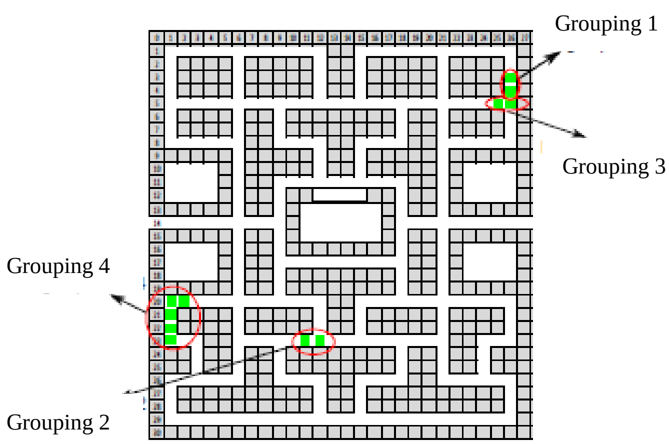


Fig. 3. - Grouping of states in the first few iterations of the algorithm.

To exploit these merged cases as a decision policy, retrieval is performed to select the best case to apply for a specific board configuration. Once a case is selected, all the actions recommended by the case are applied before any further retrieval is executed. In fact, whenever a sequence of actions is triggered, no observation of the game board is made before the complete sequence of actions is executed.

One would expect that execution of combined cases with short list of actions would provide performance similar to reactive cases. However, experiments are needed to determine how cases with longer action lists can influence performance.

5 Experimental results

To conduct our experiments, we built traces acquired during 33 games using the computer player described in Section 3 of this paper. The traces generated from these games contain a total 29 552 episodes. The board configuration in our experiments contains 300 cells that are accessible by Pac-Man. Hence the initial case base used to approximate a state machine contained almost 30 000 reactive cases defined over 300 possible states.

To evaluate the performance that can be obtained using this reactive case base, we played 200 games. An average score of 5684 points was reached over these games.

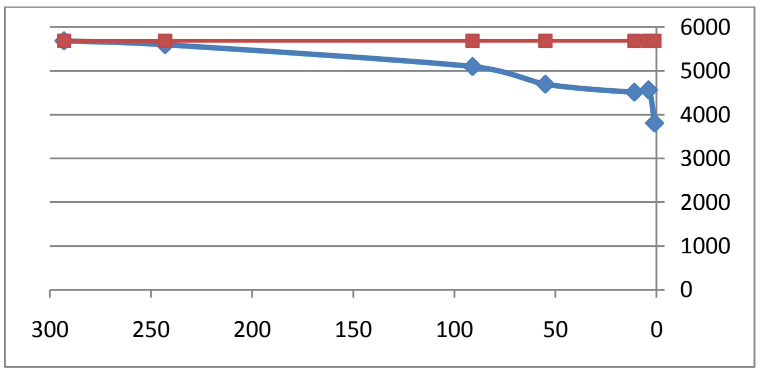


Fig. 4. Average score obtained for different number of states. The square line corresponds to the results of the computer player. And the diamond line corresponds to the case base player.

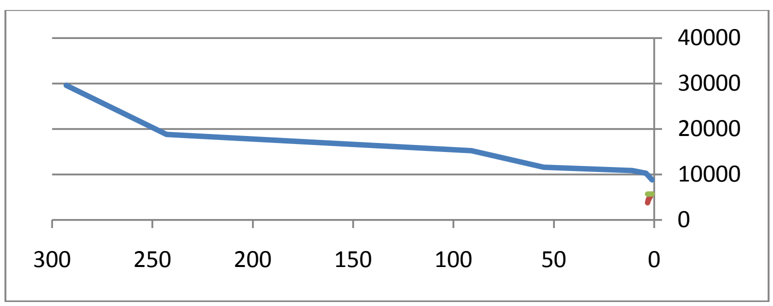


Fig. 5. Average score vs. number of cases resulting from segmentation.

We repeated the same experiments with different case bases generated by merging cases using the scheme described in Section 4. Results presented in **Fig. 4** indicate that a reduction in the number of states (the horizontal axis) degrades the performance (average score). As more states are merged together, lists of actions in the cases get longer which results in less reactivity in the behaviour of the game character.

However, we consider that the loss of performance is minor. By compressing the state space to less than 100 states, which represent less than a third of the original states, we observe a reduction of performance of only 10%. Moreover, a reduction of more than 90% of the states results in a degradation of 17.4% in average game score. Hence applying important reductions of the number of states seem to have limited effects on the performance of the corresponding case base being generated.

Fig. 5 indicates the number of cases (vertical axis) corresponding to different set of states (horizontal axis). It is interesting to note that states that were grouped earlier during the clustering process involve more cases that states during later in the process. Hence the proposed case acquisition scheme seems to substantially reduce the size of the case base while limiting performance degradation.

6 Conclusion

In this work, we conducted a study to determine how cases can be built from sequential traces acquired during game demonstrations. We considered an approach where game episodes are converted into cases representing the states of the game, the conditions describing the game environment and actions that lead to subsequent episodes. We proposed an algorithm to group cases together to represent multiple episodes in the traces. And evaluate for this scheme for the game of Pac-Man indicates that dynamic segmentation of traces into cases seem to be a viable approach. As future work, we would like to explore algorithms to select attributes describing states. And we would like to explore how segmentation algorithms, not relying on unsupervised clustering techniques, could be applied to the acquisition of cases from traces.

7 References

1. Chow, B. Executable open source Pac-Man in java. http://www.bennychow.com (01/06/2010).
2. Floyd, M. W., Esfandiari, B., Lam, K.. A case-based reasoning approach to imitating RoboCup players. FLAIRS, pp. 251-256, 2008.
3. Fu, D., Houlette., H. The ultimate guide to FSMs in games. AI Game Programming Wisdom 2, pp. 283-302, 2004.
4. M. Gallagher and A. Ryan. Learning to play Pac-Man: an evolutionary, rule-based approach. In Evolutionary Computation 2003, IEEE, volume 4, pp. 2462-2469, 2003.
5. M. Mehta, S. Ontanon, T. Amundsen, and A. Ram. Authoring behaviors for games using learning from demonstration. In ICCBR Workshop on CBR for Computer Games, 2009.
6. S. Ontañón, Case Acquisition Strategies for Case-Based Reasoning in Real-Time Strategy Games, FLAIRS 2012.
7. Houcine Romdhane and Luc Lamontagne. Forgetting reinforced cases. In Advances in Case-Based Reasoning, LNCS 5239, Springer Berlin / Heidelberg, pp. 474-486. 2008.
8. R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. Methodology and Computing in Applied Probability, 1(2), pp. 127-190, 1999.
9. I. Szita and A. Lorincz. Learning to play using low-complexity rule-based policies: illustrations through MS. Pac-Man. JAIRS, 30(1), pp. 659-684, 2007.

Creating Diverse Storylines by Reusing Plan-Based Stories

Alexandra Coman, Héctor Muñoz-Avila

Department of Computer Science and Engineering, 19 Memorial Drive West,
Lehigh University, Bethlehem, PA 18015
{alc308, hem4}@lehigh.edu

Abstract. We discuss a case-based planning technique for generating a collection of meaningfully diverse storylines that have a common starting scenario. Our work is flexible, accounting both for situations in which all alternatives need to have the same ending, and for situations in which multiple endings are required/allowed. Our work hinges on the availability of a storyline case base, which can be created through first-principles planning techniques or by recording user gameplay traces.

Keywords: plan diversity, storyline diversity

1 Introduction

The use of planning techniques for the automated generation of stories has been studied, starting with the seminal work by Meehan (1976), by numerous authors (Lebowitz, 1987; Geib, 1994; Young, 1999; Cavazza and Charles, 2005; Riedl and Young, 2004; 2010; Porteous *et al*., 2011). As pointed out by Sack (2012), the main reason for this recurrent interest is that narratives link together events in the form of cause-effect relations, therefore plans (i.e., sequences of actions, where an action causes an event) are natural representations for cohesive narratives.

We propose the use of case-based plan diversity techniques to generate ***diverse*** storylines, i.e., collections of two or more alternative storylines with a common starting situation. We focus on single levels, or what gamers sometimes refer to as "maps", which are self-contained parts of the game. A typical game consists of a series of levels, or maps, that the player must navigate.

Our approach is flexible, accounting both for situations in which all alternatives need to have the same ending, and for situations in which storylines need/can have multiple endings. Having a common ending is desirable when the storyline is part of a larger picture (e.g., a level in a game). In this situation, the designer could state the main events or landmarks that the overall plot should commit to (Porteous *et al*., 2011), and plan diversity techniques could be used to generate alternative storylines between the landmark events, allowing the designer or an automated method, such as random selection, to choose between these alternatives. Not having a common ending requires the ability to coordinate the plot forward towards an overarching storyline (a special case of which is sometimes referred to as the "story arc"). In this situation, it has been shown that techniques such as Hierarchical Task Network (HTN) Planning are effective (Meehan, 1976; Cavazza and Charles, 2005).

The paper is structured as follows. The next section discusses further related work. Section 3 describes how case-based plan diversity can be attained. Section 4 discusses an example of how our approach to case-based plan diversity works for generating multiple storylines. Section 5 discusses how case bases can be generated automatically, and, finally, in Section 6, we make final remarks.

2 Related Work

Using HTN planning techniques to generate overarching storylines has been explored in previous work. The TALE-SPIN (Meehan, 1976) system for narrative generation uses methods to generate storylines. A method decomposes high-level tasks into simpler ones, which are decomposed further, until so-called primitive tasks are generated. These primitive tasks are achieved by executing actions. Hence, a method can be seen as a way to generate a sequence of actions. In the context of generating storylines, this provides the author with the means to more tightly control the storyline by carefully crafting methods that encode narrative elements. Cavazza and Charles (2005) use HTN planning to generate cohesive storylines in the context of interactive storytelling.

Some authors have pointed out that the storylines resulting from TALE-SPIN and other subsequent planning-based programs are simplistic (Sack, 2012). One characteristic of this work is that the output is an actual English text. We do not argue either way on whether the resulting stories, when read, are simplistic, but point out the fact that, in our case (as in many others), the story is experienced by the player while interacting with a gaming environment, hence it could still prove compelling through immediate involvement and immersion. Many games (Duke Nuk'em comes to mind) are well known to have a rather weak storyline, but to be compelling to gamers due to the game mechanics. Others, such as Daikatana, are considered to have an interesting storyline, but were not well received because of poor game mechanics. Finally, games such as the Elder Scrolls series are considered excellent due both to their strong storylines and to their compelling gameplay mechanics.

3 Diverse Storylines through Case-Based Planning

A plan π is a sequence of actions ($a_1 \ldots a_n$). Actions have preconditions and effects. In the context of storylines, each action a_i links the event preceding the action with the event succeeding it (Sack, 2012). Moreover, a sequence of actions will connect events that are important to the storyline. For example, a non-player character (NPC) starting at some location on the map might take several actions that will result in the NPC rescuing a friend.

A case base consists of pairs (p, π), where *p* is a problem description and π is a plan. A problem is a pair (s, g), where *s* is an initial state and *g* is a goal achieved by the plan. In the context of storylines, the state refers to the scenario or level description, while the goal refers to the final event that must take place at the end of the level. If no particular ending is preferred, the goal can be ignored, and only the scenario is considered.

Scenario-Based Retrieval Criterion. Given a storyline case base CB, a scenario *s*', and, potentially, a desired ending *g*', one can use case retrieval techniques to retrieve a suitable case *c* from the case base. This presupposes a user-defined scenario similarity metric, *sim*, such that:

$$\text{sim: } S \times S \rightarrow [0,1] \qquad (1)$$

In Formula 1, *S* is the collection of all possible scenarios. The similarity metric estimates how close two scenarios are to one another.

The retrieval criterion identifies a case $c = ((s, g), \pi)$ in CB, such that: (1) $g = g'$ (although it would be possible to relax this and simply require the goals to be similar to one another, using an appropriate goal similarity metric) and (2) *c* is the case in CB for which *sim*(s, s') has the largest value (i.e., *c* is the nearest neighbor). If there is no required ending, then only condition (2) needs to be satisfied.

Plan Adaptation. After a case *c* has been retrieved, an adaptation procedure is needed to modify π so as to make it executable in scenario *s*'. There are a number of possible adaptation procedures. In the context of plan diversity in games, we have used a domain-specific one: if, in the solution plan from the retrieved case, an action is taken by an NPC of a certain class (e.g., an *archer*), we select, in the new scenario, an NPC of the same class (Coman and Muñoz-Avila, 2011b).

To generate diverse storylines, we need a way of determining if two storylines are meaningfully different. Since our storylines are encoded in the plans, this translates to determining if two plans are significantly different from one another. To this end, we employ user-defined plan distance metrics:

$$D: \Pi \times \Pi \rightarrow [0,1] \qquad (2)$$

where Π is a set of solution plans for a given problem. Carefully crafted domain-specific distance metrics can be used to obtain meaningfully different plans. For example, in the context of combat games, this distance could be defined as greater between a defensive plan and an offensive plan than between two defensive plans. On the other hand, using generic plan distance metrics (e.g., computing the proportion of actions appearing in one of the compared plans, but not the other) might not yield meaningful differences (Coman and Muñoz-Avila, 2011a and 2011b).

Relative Diversity for Storylines. Using the plan distance metric *D*, it is possible to define the relative diversity $RelDiv(\pi,\Pi)$ between a plan π under consideration and a collection of diverse plans Π by using a variant of the formula defined for recommender systems by Smyth and McClave (2001), and adapted for case-based planning by Coman and Muñoz-Avila (2011a; 2011b):

$$RelDiv(\pi, \Pi) = \frac{\sum_{\pi' \in \Pi} D(\pi, \pi')}{|\Pi|} \quad (3)$$

Relative diversity (Formula 3) is the average distance between π and the plans in Π. When D is a plan distance criterion for storylines encoded in the case, then *RelDiv* can be seen as computing the average storyline distance between the current storyline and the diverse storylines generated so far.

Obtaining *k* Diverse Storylines. Retrieving k diverse plans (representing storylines), where $k \geq 2$, is conducted iteratively as follows. First, a case is selected from the storyline case base CB using the scenario-based retrieval criterion discussed before. The plan of the retrieved case is adapted using the plan adaptation procedure, resulting in a plan π. The set of diverse plans is, at this point, $\Pi = \{\pi\}$. Then, the following steps are repeated $k - 1$ times:

1. Retrieve a case from CB using a retrieval criterion (Formula 4) balancing scenario similarity with relative diversity of storylines.

2. Adapt the plan of the retrieved case, resulting in a plan π_i.

3. Add π_i to Π.

Step 1 requires balancing two potentially conflicting criteria. On the one hand, one would like to retrieve a case $c = ((s, g), \pi)$ the scenario *s* of which is as close to the current scenario *s'* as possible. On the other hand, one would like the plan π_i to have a high diversity value (Formula 3) relative to Π. To accomplish this, we use the following retrieval criterion (Coman and Muñoz-Avila, 2011b):

$$\alpha \text{sim}(s', s) + (1- \alpha)\text{RelDiv}(\pi_i, \Pi) \quad (4)$$

where α is a weight parameter which can be used to place more emphasis on either scenario similarity or relative diversity.

An alternative approach is taken by Díaz-Agudo *et al*. (2008), who introduce the dissimilarity criterion at the reuse phase, rather than the retrieval one.

4 Example

Assume the scenario depicted in Fig. 1 (the game is Wargus, previously used in planning research work including that by Coman and Muñoz-Avila, 2011a and 2011b), with the main NPC circled. The NPC's friend (Label 1) is captive at a location surrounded by walls, with 3 entry points: an old tower (Label 2), some trees (Label 3), and an entrance guarded by an Ogre (Label 4). The map also contains treasure (Label 5) and two town halls (Labels 6 and 7).

In our example, computing the distance between two plans π and π' involves identifying the kinds of actions taken by the main character, and computing the degrees of distance between those kinds of actions. An NPC's action is *greedy* if it consists of collecting the treasure or looting the poorly-guarded enemy town hall; *helpful* if the NPC attempts to save the captive friend. Furthermore, a *helpful* NPC might be considered *reckless* if he attacks the powerful enemy ogre, and *intelligent* if he frees his friend by either cutting down trees or demolishing the old guard tower to make a path for the friend to escape by. Any other actions carried out in the plan are considered irrelevant for the purposes of this classification (e.g. it does not matter if our character takes a shortcut or the long way to the treasure). Greed, devotion to friends, and intelligence are, therefore, the traits along which we will create variation. We define a distance metric based on character personality (Formula 5), where *NPCType*(π) is the *type of NPC personality* reflected in plan π, while *d* is a degree of distance between possible NPC personality types. The six showcased NPC personality types are: (1) *greedy* and *indifferent*, (2) *greedy* and *intelligent*, (3) *greedy* and *reckless*, (4) *not greedy* and *indifferent*, (5) *not greedy* and *intelligent*, (6) *not greedy* and *reckless*.

$$D_{NPC}(\pi, \pi') = \begin{cases} 0, if\ NPCType(\pi) = NPCType(\pi') \\ d, 0 < d \leq 1, if\ NPCType(\pi) \neq NPCType(\pi') \end{cases} \quad (5)$$

Fig. 1: An Example Scenario in real-time strategy game Wargus

In this context, a scenario describes the unit type of the NPC, which, in our case, can be: *peasant*, *soldier*, *archer*, or *mage*. Assume all 4 scenario types are represented in the case base CB. In addition to the scenario specifying a unit type, each case contains a plan that the unit type is able to execute (e.g. for a *peasant* unit, the plan might specify moving to a certain location on the map and cutting down trees, then moving to another location and collecting a treasure). There are differences between units in terms of the ways in which they are able to manifest certain character traits (e.g. since no unit type but the *peasant* is able to harvest treasure, *soldiers*, *archers* and *mages* manifest their greed by attacking the enemy town hall).

Assume that the storyline requires the NPC to rescue their friend, and we want to generate two diverse behavior plans for a *peasant* NPC (Fig. 2). First, we select a case from the CB using solely the scenario similarity criterion. Therefore, the acting unit in the case should be a *peasant*. Out of the retrieved *peasant* cases, we pick a case $c1$ randomly, since our retrieval criterion is based on unit type only. Assume that $c1.\pi$, the plan of the randomly picked case, specifies that our character should collect the nearby treasure, then attack the enemy ogre, in an attempt to save their friend (the plan shown by the thin arrows in Fig. 2). According to our description, the plan $c1.\pi$ reflects *greedy* and *reckless* behavior.

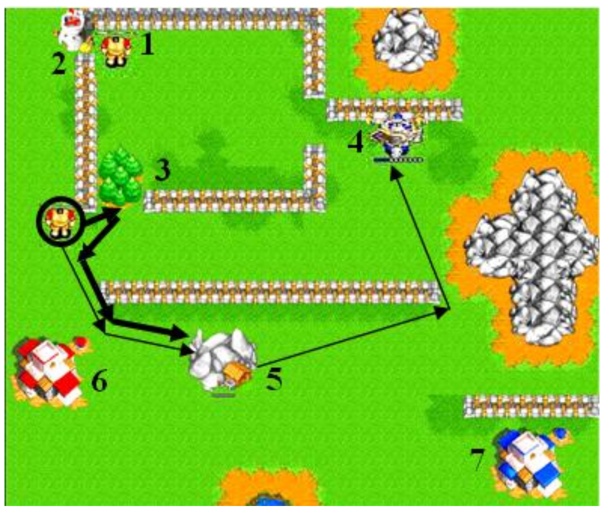


Fig. 2: Sample Storylines: (1) NPC collects treasure, then attacks ogre (thin lines), (2) NPC cuts down trees, then collects treasure (thick lines)

We now select case c_2 based on the composite criterion combining problem similarity and plan diversity with respect to $c_1.\pi$ (Formula 3). Based on the definition of d in Formula 5, the selected case c_2 is another *peasant* case, the plan of which is dissimilar to $c_1.\pi$. Any ties are broken by random selection. The selected plan $c_2.\pi$ might have the character cut down trees to free his friend, then collect the treasure (as depicted in Fig. 2, in bold lines). According to our description, the plan $c_2.\pi$ would be a *greedy* and *intelligent* one.

The storylines resulting from the retrieved plans $c_1.\pi$ and $c_2.\pi$ are meaningfully different: the first tells the story of a *reckless* and *greedy* NPC who, after collecting the treasure, goes on to attack the ogre, in order to save his friend. The second one tells the story of an *intelligent* and *greedy* NPC who saves his friend by clearing the trees blocking the safe escape path, without putting his life at risk by braving the unbeatable ogre. Furthermore, if there is no required ending, the retrieved storylines might vary even more, e.g. the NPC might just collect the treasure and wander around the countryside in search of different adventures, not attempting to assist his friend at all (i.e., *greedy* and *indifferent* behavior).

5 Generating Case Bases

It is clear that the performance of this case-based approach to creating diverse storyline hinges on the availability of a suitable case base, populated with a diverse collection of plans for a large sample of different scenarios. For filling this case base, one can leverage previous work showing how first-principles heuristic planning techniques can be used to generate diverse plans (Coman and Muñoz-Avila, 2011a). Specifically, in heuristic planning, when selecting among potential partial plans $\{\pi_1,\ldots,\pi_n\}$, the planner selects the one (π_i) with the lowest heuristic value $h(\pi_i)$, where h is a function (Formula 6) that, for a given partial plan π, estimates the number of steps $h(\pi)$ necessary in order to complete it.

$$h: \Pi \rightarrow [0,1] \qquad (6)$$

The partial plan with the lowest h-value is selected because it is estimated that it will require the least effort to complete. Coman and Muñoz-Avila (2011a) modify this heuristic criterion to take into account relative diversity (Formula 3):

$$h_{new}(\pi) = \alpha h(\pi) + (1 - \alpha)\ \mathrm{RelDiv}(\pi, \Pi) \qquad (7)$$

On using Formula 7, the selected partial plan π_i is the one that best balances the estimated effort required for its completion with its relative diversity with regard to the set of diverse plans Π computed so far.

6 Discussion and Final Remarks

Given that we have the capability to generate a collection of k plan-based storylines from scratch using first-principles planning, as explained in the previous section, the question arises as to why we would need the case-based planning approach. There are two motivations for this choice.

First, in some situations, action definitions in terms of preconditions and effects (as required for first-principles planning) may not be available, either because certain gaming actions are too complex to be defined this way, or, simply, because no one has undertaken the knowledge engineering effort to define actions in this manner. In both of these situations, the case-based approach outlined above will still work. All we need is a similarity metric, a plan distance metric, and a plan adaptation procedure, which, as exemplified, can be domain-specific and, hence, not require actions to be defined in terms of preconditions and effects. Plans can be captured, for example, from user execution traces.

Second, even when action definitions are known, generating diverse plans from scratch can be an inefficient process (Coman and Muñoz-Avila, 2012). Hence, generating a set of diverse plans on-line might not be feasible. Instead, one could generate the case base, as discussed in the previous section, off-line, and then use it to generate diverse storylines on-line.

Acknowledgements. This work was supported in part by NSF grant 0642882 and NRL. We thank the anonymous reviewers for their comments.

References

Cavazza, M., Charles, F.: Dialogue Generation in Character-based Interactive Storytelling. In: Young, R.M., Laird, J.E. (eds.) Proc. of AIIDE 2005, pp. 21-26. AAAI Press (2005)

Coman, A., Muñoz-Avila, H.: Generating Diverse Plans Using Quantitative and Qualitative Plan Distance Metrics. In: Burgard, W., Roth, D. (eds.) Proc. of AAAI 2011, pp. 946-951. AAAI Press (2011a)

Coman, A., Muñoz-Avila, H.: Qualitative vs. Quantitative Plan Diversity in Case-Based Planning. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 32-46. Springer, Heidelberg (2011b)

Coman, A., Muñoz-Avila, H.: Diverse Plan Generation by Plan Adaptation and by First-Principles Planning: A Comparative Study. To appear in: Díaz-Agudo, B., Watson, I. (eds.) ICCBR 2012. LNCS, vol. 7466. Springer, Heidelberg (2012)

Díaz-Agudo, B., Plaza, E., Recio-García, J.A., Arcos, J.L.: Noticeably New: Case Reuse in Originality-Driven Tasks. In Althoff, K.D. *et al.* (eds) ECCBR 2008. LNCS vol. 5239, pp. 165-179. Springer, Heidelberg (2008)

Geib, C. W.: The Intentional Planning System: It-PlanS. In: Hammond, K.J. (ed.) Proc. of AIPS 1994, pp.55-60. AAAI Press (1994)

Lebowitz, M.: Planning Stories. In: Proc. of the 9th Annual Conference of the Cognitive Science Society, pp. 234-242. Routledge (1987)

Meehan, J.R.: The Metanovel: Writing Stories by Computer. Ph.D. Dissertation. Yale University, New Haven, CT, USA (1976)

Porteous, J., Teutenberg, J., Pizzi, D., and Cavazza, M.: Visual Programming of Plan Dynamics using Constraints and Landmarks. In Bacchus, F. *et al.* (eds.) Proc. of ICAPS 2011, pp. 186-193. AAAI Press (2011)

Riedl, M. O., Young, R. M.: An Intent-Driven Planner for Multi-Agent Story Generation. In Proc. of AAMAS 2004, pp. 186-193 (2004)

Riedl, M.O. Young, R. M.: Narrative Planning: Balancing Plot and Character. JAIR 39, 217-268 (2010)

Sack, W.: Narrative Intelligence. In: The Software Arts. MIT Press. To appear. (2012) URL: http://fdm2.ucsc.edu/~wsack/JavaScript/Spinner/wsack-narrative.pdf

Smyth B., McClave, P.: Similarity vs. Diversity. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNAI, vol. 2080, pp. 347-361. Springer, Heidelberg (2001)

Young, R.: Notes on the Use of Plan Structures in the Creation of Interactive Plot. AAAI Fall Symposium on Narrative Intelligence. AAAI Press (1999)

Case-Based Story Generation through Story Amalgamation

Santiago Ontañón[1], Jichen Zhu[1] and Enric Plaza[3]

[1] Drexel University
Philadelphia, PA, USA 19104
santi@cs.drexel.edu
jichen.zhu@drexel.edu

[2] IIIA, Artificial Intelligence Research Institute
CSIC, Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia (Spain)
enric@iiia.csic.es

Abstract. Computational narrative explores techniques through which computers can analyze, understand and, most importantly, generate stories. This paper explores a CBR approach to story generation based on the idea of *story amalgamation*: given a target partial description of the story we want the system to generate, the system will retrieve a set of full stories (represented as cases), and then reuse them by merging them in such a way that the result satisfies the target partial description. This story merging is performed via a formal operation we call an *amalgam*. We report on a preliminary study showing the potential of the approach.

1 Introduction

Computational narrative explores the age-old creative form of storytelling by algorithmically analyzing, understanding, and most importantly, generating stories. The various AI techniques developed in story generation can be extended to other forms of interactive entertainment and electronic literature, including computer games and interactive narrative. Many different approaches have been studied for the problem of automatic story generation, such as planning or case-based reasoning, each of which providing different aesthetic affordances in the range of stories that can be generated [21].

This paper explores a CBR approach to story generation based on the idea of *story amalgamation*, i.e. generating new stories by selectively merging parts of previous, existing, stories. Specifically, we explore the idea that if stories are represented as *terms* in a generalization space, then story amalgamation can be carried out by performing formal *amalgam* operations over terms [10]. An amalgam is an operation between terms in a generalization space that generates a new term that combines as much information as possible from the initial terms. Formally, an amalgam is related to the idea of partial unification, merging operators [4], and also to the cognitive process called *conceptual blending* [6].

In this paper we present a CBR system that can generate stories using amalgams: each case in the CBR system represents a different story; the input to the system is a target partial description of the story we want to generate; given an input target description, the system retrieves a set of similar cases, and then reuses them by amalgamating them in a way that the amalgam satisfies the target partial description. Moreover, to make this initial study tractable, we have limited ourselves to generate story scenes, rather than full stories.

The remainder of this paper is organized as follows. Sections 2 and 3 provide background on story generation and the amalgam operations respectively. Then Section 4 presents our technical approach. Section 5 shows our preliminary results. The paper closes with related work, conclusions and potential future work.

2 Story Generation

Automatic story generation is an interdisciplinary topic focusing on devising models for algorithmically structuring and producing narrative content and/or discourse. Narrative can be divided into two main parts [3]: *story* and *discourse*, which basically correspond to the story content, and to the way the story is presented respectively. Most story generation systems focus strictly in generating a story, although some are capable of generating a discourse as well [20]. In general, story generation systems can be classified into three main categories [1]: character-centric, author-centric and story-centric:

- Character-centric systems like Tale-spin [9] and the Virtual Storyteller [17] generate stories by simulating characters in a world.
- Author-centric systems, such as the MEXICA system [13], model the author's throught process during the process of story-writing.
- Story-centric systems, such as the Fabulist [15], generate stories by modeling the structural properties of the stories themselves.

The system reported in this paper can be classified as a story-centric system.

Different techniques have been studied in story generation, the most common of which is automated planning. Salient examples of planning-based story generation systems include Tale-spin [9], Universe [8] and Fabulist [15]. However, other techniques, such as CBR and computational analogy have also shown applicability to the problem of story generation. Examples are systems like Minstrel [18], ProtoPropp [7], Riu [12], or the story-translator [16]. In this paper we will explore case-based techniques, and in particular in a technique called *amalgamation*, which can generate solutions to new problems by amalgamating information coming from one or more cases [10].

Story generation is a very challenging task from many points of view. Generating stories means generating coherent plots, believable characters that have common sense, and natural language. Moreover, those stories have to be aesthetically pleasing and creative. In fact, some approaches to story generation, such as Minstrel [18] aim at being general models of computational creativity. Finally, one of the most important open problems in story generation is evaluation [19].

3 Amalgams

In this paper we will make the assumption that cases are terms in some *generalization space*. We define a generalization space as a partially ordered set $\langle \mathcal{L}, \sqsubseteq \rangle$, where $\mathcal{L}$ is a language, and $\sqsubseteq$ is a subsumption between the terms of the language $\mathcal{L}$. We say that a term ψ_1 subsumes another term ψ_2 ($\psi_1 \sqsubseteq \psi_2$) when ψ_1 is more general (or equal) than ψ_2[3]. Additionally, we assume that $\mathcal{L}$ contains the infimum element $\perp$ (or "any"), and the supremum element $\top$ (or "none") with respect to the subsumption order.

Next, for any two terms ψ_1 and ψ_2 we can define their *unification*, ($\psi_1 \sqcup \psi_2$), which is the *most general specialization* of two given terms, and their *anti-unification*, defined as the *least general generalization* of two terms, representing the most specific term that subsumes both. Intuitively, a unifier (if it exists) is a term that has all the information in both the original terms, and an anti-unifier is a term that contains all the information that is shared by the two original terms. Depending on $\mathcal{L}$, the anti-unifier and unifier might be unique or not.

The notion of *amalgam* can be conceived of as a generalization of the notion of unification over terms. The unification of two terms (or descriptions) ψ_a and ψ_b is a new term $\phi = \psi_a \sqcup \psi_b$, called unifier. All that is true for ψ_a or ψ_b is also true for ϕ.; e.g. if ψ_a describes "a red vehicle" and ψ_b describes "a German minivan" then their unification yields the description "a red German minivan." Two terms are not unifiable when they possess contradictory information; for instance "a red French vehicle" is not unifiable with "a red German minivan". The strict definition of unification means that any two descriptions with only one item with contradictory information cannot be unified.

An *amalgam* of two terms is a new term that contains *parts from these two terms*. For instance, an amalgam of "a red French vehicle" and "a German minivan" is "a red German minivan"; clearly there are always multiple possibilities for amalgams, since "a red French minivan" is another example of amalgam. The notion of amalgam, as a form of partial unification, was formally defined in [10], where its relationship with the notion of merging operators [4], is also discussed.

Definition 1. (Amalgam) *The set of* amalgams *of two terms ψ_a and ψ_b is the set of terms such that:*

$$\psi_a \curlyvee \psi_b = \{\phi \in \mathcal{L}^+ | \exists \alpha_a, \alpha_b \in \mathcal{L} : \alpha_a \sqsubseteq \psi_a \;\wedge\; \alpha_b \sqsubseteq \psi_b \;\wedge\; \phi = \alpha_a \sqcup \alpha_b\}$$

where $\mathcal{L}^+ = \mathcal{L} - \{\top\}$

Thus, an amalgam of two terms ψ_a and ψ_b is a term that has been formed by unifying two generalizations α_a and α_b such that $\alpha_a \sqsubseteq \psi_a$ and $\alpha_b \sqsubseteq \psi_b$ —i.e. an amalgam is a term resulting from combining some of the information in ψ_a with some of the information from ψ_b. Formally, $\psi_a \curlyvee \psi_b$ denotes the set of all possible amalgams; however, whenever it does not lead to confusion, we will use $\psi_a \curlyvee \psi_b$ to denote one specific amalgam of ψ_a and ψ_b.

[3] In machine learning, $A \sqsubseteq B$ means "A is more general than B", while in description logics it has the opposite meaning: it is seen as "set inclusion" of their interpretations.

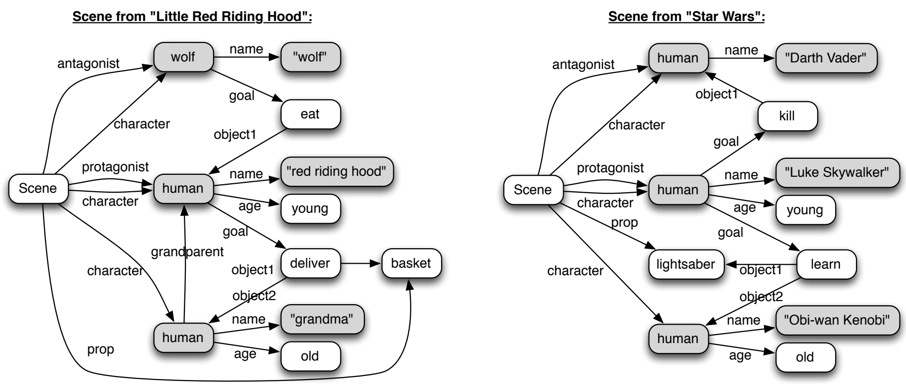


Fig. 1. Two of the scenes used in our experiments.

The terms α_a and α_b are called the *transfers* of an amalgam $\psi_a \curlyvee \psi_b$. α_a represents all the information from ψ_a which is *transferred* to the amalgam, and α_b is all the information from ψ_b which is transferred into the amalgam.

Intuitively, an amalgam is *complete* when all which can be transferred from both terms into the amalgam has been transferred, i.e. if we wanted to transfer more information, α_a and α_b would not have a unifier.

For the purposes of story generation, we introduce the notion of asymmetric amalgam, where one term is fixed while only the other term is generalized in order to compute an amalgam.

Definition 2. (Asymmetric Amalgam) *The* asymmetric amalgams $\psi_s \overrightarrow{\curlyvee} \psi_t$ *of two terms* ψ_s *(*source*) and* ψ_t *(*target*) is the set of terms such that:*

$$\psi_s \overrightarrow{\curlyvee} \psi_t = \{\phi \in \mathcal{L}^+ | \exists \alpha_s \in \mathcal{L} : \alpha_s \sqsubseteq \psi_s \quad \wedge \quad \phi = \alpha_s \sqcup \psi_t\}$$

In an asymmetric amalgam, the target term is transferred completely into the amalgam, while the source term is generalized. The result is a form of partial unification that conserves all the information in ψ_t while relaxing ψ_s by generalization and then unifying one of those more general terms with ψ_t itself. Finally, an asymmetric amalgam is *maximal* when all knowledge in ψ_s that is consistent with ψ_t is transferred to the solution ψ'_t —i.e. $\psi'_t \in \psi_s \overrightarrow{\curlyvee} \psi_t$ is maximal iff $\nexists \psi''_t \in \psi_s \overrightarrow{\curlyvee} \psi_t$ such that $\psi'_t \sqsubset \psi''_t$.

4 Story Generation through Story Amalgamation

In this paper we want to explore the idea of generating stories by amalgamating previously existing stories that correspond to cases in a CBR system. For that purpose, we have designed a system that solves the following task:

Given A case base $\Delta = \{\psi_1, \dots \psi_m\}$, where each case ψ_i is a story, and a partially specified target story ψ_t

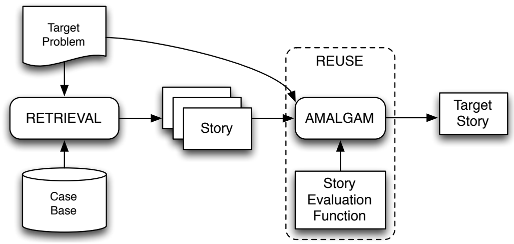


Fig. 2. Overview of the story generation approach studied in this paper.

Generate A story ψ'_t such that $\psi_t \sqsubseteq \psi'_t$ (i.e. the generated story satisfies the partial specification provided as input) by amalgamating a collection of cases from the case base.

From the previous description it can be seen that we are proposing to represent stories as terms in a generalization space. Specifically, we have used the *feature term* formalism [2, 14] to represent stories. Figure 1 shows two sample stories used in our experimentation. The first one represents the opening scene of the famous "Little Red Riding Hood" tale, with three characters: red riding hood, her grandmother and the wolf. As can be seen, the story specifies that red riding hood wants to deliver a basked of food to her grandmother, and the wolf wants to eat red riding hood. For this experimentation, we have avoided representing the notion of time in our stories, and thus each case in the case base represents just a "scene". Full stories could be represented by a collection of scenes composed using time relations (as in the SAM story generation system [12]). However, for our purposes, scenes like the ones shown in Figure 1 are complex enough to test the potential of our approach. As can be seen, for each scene, we represent the set of characters and props, their relations, and the goals of each character in the scene, some scenes also contain actions.

The target given to the system is a partially specified story, which means that it is a term (similar to those in Figure 1), but where only some parts of a scene are represented. For example, we could just specify that we want to have 2 characters, or that we want a story with a "character wanting to kill another character". Basically, the target specifies the constraints that the generated story has to satisfy.

Figure 2 shows an overview of our proposal. When the user provides a new target ψ_t (a partially specified story), the system retrieves a set of k cases from the case-base by using a similarity measure. In our experiments, we used the S_λ similarity metric defined in [11] (which basically measures the amount of information shared between two terms, by computing their anti-unification and measuring its size), to find the cases that are most similar to the provided problem. Let us call R to the set of retrieved cases.

Then, the system generates a new story by combining information from all the retrieved cases in R, using the amalgam operation, to generate a story that satisfies the target. This process is performed in two steps:

1. Retrieved cases amalgamation: in a first step, all the retrieved cases are amalgamated in order to obtain a combination of the parts from the retrieved cases that are consistent (and could later be used to generate a new story). Specifically, given the set of retrieved cases $R = \{\psi_1, ..., \psi_k\}$, this process constructs an amalgam ψ_R in the following way:

$$\psi_R = \psi_1 \curlyvee \psi_2 \curlyvee ... \curlyvee \psi_k$$

   where, $\psi_1 \curlyvee \psi_2 \curlyvee \psi_3 = (\psi_1 \curlyvee \psi_2) \curlyvee \psi_3$, i.e. to perform the amalgam between a set of n terms, we amalgamate the first two, the result is amalgamated with the third, and so on. Notice that ψ_R is not unique, since there are many different possible amalgams that can result from amalgamating a set of given terms. As detailed below, we use an evaluation function that gives a score to each one of them. Then, we select the amalgam that maximizes such evaluation function. To reduce the computational complexity of the approach, we use a greedy approach, and only select the best amalgam at each step, but more complete forms of search could also be employed.
2. Then, given ψ_R, the final story is obtained by performing the asymmetric amalgam of ψ_R with the target ψ_t, obtaining $\psi'_t = \psi_R \overrightarrow{\curlyvee} \psi_t$. As before, ψ'_t might not be unique, and the final story is selected by evaluating all the possible amalgams with an evaluation function and selecting the one that maximizes it. ψ'_t represents an amalgam that is ensured to satisfy the partial description ψ_t, and contains as much information as possible from ψ_R.

As described above, the amalgam operation between two terms $\psi_1 \curlyvee \psi_2$ does not define a single term, but a space of possible amalgams. Each of them obtained by combining different sets of information from the two input terms. That means that the process defined above just defines the space of possible stories that can be generated. In order to determine which of all the possible amalgams is the one the system will produce, we need to introduce some additional criteria that determines which amalgams are better than others. In [10] we introduced the notion of *preservation degree*, which measures how much of the information of the input terms is present in the amalgam. Using the idea of the preservation degree, the system can be programmed to find amalgams that maximize the preservation degree, and thus that generates stories that contain as much information as possible form the retrieved cases. However, we can define other criteria for amalgam selection, which would bias the system towards generating certain types of amalgams that correspond to the stories we are interested in.

Specifically, we have experimented with the following criteria (all of them are defined to provide a numerical score to the resulting amalgam; the system was programmed to output the amalgam that maximizes such score):

– Preservation Degree: as defined in [10], just tries to maximize the information transferred from the input terms into the amalgam. It assigns larger scores to amalgams that transfer more information from the input terms.
– Compactness: when amalgamating stories using the preservation degree, the amalgam operation tends to add all the characters that exist in the input

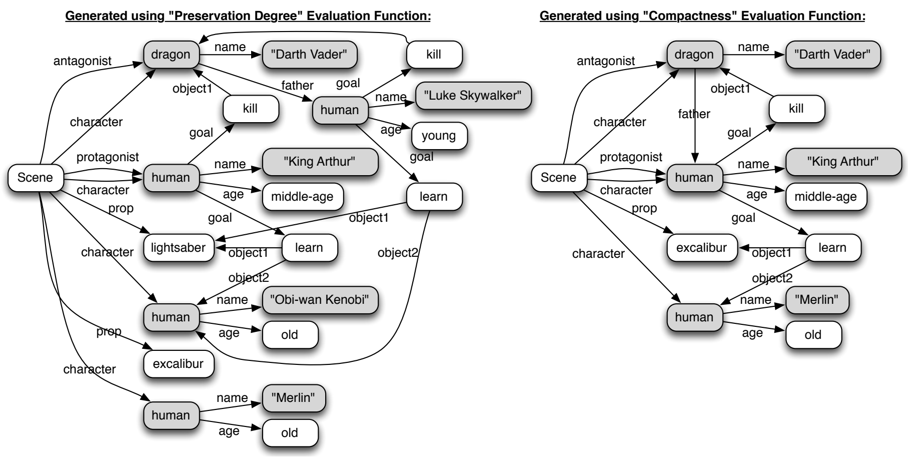


Fig. 3. Two stories generated by retrieving only one case.

stories to the amalgam. However, this does not always result in interesting stories. This measure assigns a score to each story computed as the preservation degree minus the number of variables in the resulting story (each of the nodes in the graphs shown in Figure 1 corresponds to a variable in the term representing the story).

In order to generate amalgams that maximize the previous criteria, we have used a straightforward greedy search method over the amalgam space. This method does not ensure obtaining the amalgam that maximizes the criteria, but is computationally efficient and provides good results (as shown in the next section). The next section shows some example stories generated by our system using the different evaluation functions.

Other evaluation functions that could be explored. For example: *novelty* (as studied in [5]), which would favor stories that differ from those already in the case base. Another interesting measure could be measure the coherency of the character relations in the story, if they are not related in a complex but coherent way to other characters, the story is likely to be uninteresting. Exploring further evaluation functions is part of our future work.

5 Illustration

In this section we show example stories generated by our system using different evaluation functions. To generate the stories described in this section, we used the following partially specified story as the input problem to our system: $\psi_t =$ "the story must have at least three characters, one named King Arthur, the other named Merlin, and the other is a dragon; King Arthur is the protagonist and the dragon is the antagonist; the story also must involve a sword called Excalibur".

Let us start with a simple example, where we set $k = 1$ (i.e. the system retrieves only 1 case). In this scenario, the case being retrieved is the one corresponding to the "star wars" story in Figure 1, which has similarity 0.46. Figure 3 shows the two stories that were generated by using the preservation degree and the compactness evaluation functions. As can be seen in the figure, the story generated using the preservation degree evaluation function is very complex and merely puts together all the characters, props and relations from the retrieved story and from the target without any interesting results. Using the compactness evaluation function however, results in a more interesting story, where the system has amalgamated the two villains of the target and of the retrieved case (the dragon from the query is now called "Darth Vader" and happens to be the father of the main character "King Arthur"). The main character, wants to learn how to use the sword "excalibur" from "Merlin", and plans to kill the dragon with it. This shows that, by determining an adequate evaluation function can have a huge impact in the resulting story, since the space of possible amalgams is quite large and varied. Specifically, the space of amalgams that was explored by the greedy search algorithm using the preservation degree evaluation function contained 1897 amalgams, and was explored in 4.89 seconds (executed in a MacBook Pro with a 2.8GHz intel CPU and 8GB of RAM); using the compactness evaluation function only 109 amalgams were explored, in 0.7 seconds.

We ran an experiment where we set $k = 2$ (the system retrieved 2 cases). In this scenario, the two cases that were retrieved are the ones shown in Figure 1. The space of amalgams being explored using preservation degree for first amalgamating the two cases to find ψ_R contained 1835 amalgams, and the space of amalgams explored when amalgamating ψ_R with the target contained 6681 amalgams and was explored in 30.02 seconds. When using the compactness evaluation function the spaces of amalgams contained 508 and 571 amalgams respectively, and were explored in 5.76 seconds. We don't show the resulting stories due to lack of space, but the story resulting with the compactness evaluation function contained three characters: King Arthur, Merlin and a Dragon called "wolf". Merlin is the grandfather of King Arthur, and King Arthur wants to deliver a basket of food to him. King Arthur also wants to learn how to use Excalibur from Merlin, so he can defeat "wolf" the dragon (who wants to eat King Arthur). As can be seen, this story combines parts from both "Red Riding Hood" and "Star Wars", but using the characters specified in the target story.

In summary, we have seen that by defining a small collection of base stories, a very large number of new stories can be generated by amalgamating them in different ways, and this can be exploited for story generation purposes with interesting results.

6 Related Work

Story generation using CBR approaches has been explored in the past. However, not through amalgams or merging operators, which is the main contribution of this paper. One of the early systems to use CBR for story generation was MIN-

STREL [18]. MINSTREL is a generic model that generates stories by executing TRAMS (Transform Recall Adapt Methods), which are generic operators that encode different problem solving procedures. MINSTREL was designed to be a model of human creativity, and as such TRAMS explore ways in which problems can be solved in creative ways. Other CBR approaches to story generation have focused on other problems, such as incorporating semantic knowledge into the retrieval and adaptation process [7], or on generating stories that are different from those in the case base (in order to show originality) [5].

Similar to CBR, some systems, such as SAM [12] use computational analogy to generate stories. SAM takes as input a partially specified story and a predefined complete story, and completes the partial story by analogy with the complete one. Notice that this is similar to the way the system presented in this paper works when only one case is retrieved. The main different with SAM is that SAM uses structure mapping theory in order to find the best analogy from the source to the target story, while in our work we use an evaluation function that can capture different aspects of the amalgam in order to decide which is the best amalgam to select.

7 Conclusions and Future Work

In this paper we have presented a preliminary study concerning the possibilities of using story amalgamation for story generation. We have presented a CBR system that can generate stories by retrieving and amalgamating stories and showed examples of its execution. One of the most interesting properties of the amalgam operation is that the amalgam of two input stories does not define a single story but a space of possible amalgams of the two stories. Therefore, it is possible to provide an evaluation function to the system that biases the story generation procedure towards specific kind of stories, and allows the inclusion of additional domain knowledge.

As part of our future work, we would like to explore the scalability of the approach to larger stories, and study further evaluation functions that capture the vast amount of existing narratology knowledge. Additionally, we would like to perform formal comparisons of the story generation capabilities of the proposed system with other CBR story generation systems.

Acknowledgements. This research was partially supported by project Next-CBR (TIN2009-13692-C03-01).

References

[1] Bailey, P.: Searching for storiness: Story-generation from a reader's perspective. In: Proceedings of the AAAI Fall Symposium on Narrative Intelligence (1999)
[2] Carpenter, B.: Typed feature structures: an extension of first-order terms. In: Saraswat, V., Ueda, K. (eds.) Proceedings of the International Symposium on Logic Programming. pp. 187–201. San Diego (1991)

[3] Chatman, S.: Story and Discourse: Narrative Structure in Fiction and Film. Cornell University Press (June 1978)
[4] Cojan, J., Lieber, J.: Belief merging-based case combination. In: Case-Based Reasoning Research and Development (ICCBR'09). Lecture Notes in Artificial Intelligence, vol. 5650, pp. 105–119 (2009)
[5] Díaz-Agudo, B., Plaza, E., Recio-García, J.A., Arcos, J.L.: Noticeably new: Case reuse in originality-driven tasks. In: ECCBR 2008. pp. 165–179 (2008)
[6] Fauconnier, G.: Conceptual blending and analogy. In: Gentner, D., Holyoak, K.J., Kokino, B.N. (eds.) The Analogical Mind: Perspectives from Cognitive Science. MIT Press, Cambridge, MA (2001)
[7] Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on cbr. Journal of Knowledge-Based Systems 18(4-5), 235–242 (2005)
[8] Lebowitz, M.: Creating characters in a story-telling universe. Poetics 13, 171–194 (1984)
[9] Meehan, J.: The Metanovel: Writing Stories by Computer. Ph.d., Yale University (1976)
[10] Ontañón, S., Plaza, E.: Amalgams: A formal approach for combining multiple case solutions. In: Case-Based Reasoning. Research and Development, 18th International Conference on Case-Based Reasoning, ICCBR 2010. pp. 257–271 (2010)
[11] Ontañón, S., Plaza, E.: Similarity measuress over refinement graphs. Machine Learning Journal 87(1), 57–92 (2012)
[12] Ontañón, S., Zhu, J.: The sam algorithm for analogy-based story generation. In: AIIDE 2011 (2011)
[13] Pérez y Pérez, R., Sharples, M.: Mexica: A computer model of a cognitive account of creative writing. Journal of Experimental and Theoretical Artificial Intelligence 13(2), 119–139 (2001)
[14] Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases. In: Veloso, M., Aamodt, A. (eds.) Case-Based Reasoning, ICCBR-95, pp. 265–276. No. 1010 in Lecture Notes in Artificial Intelligence, Springer-Verlag (1995)
[15] Riedl, M.: Narrative Generation: Balancing Plot and Character. Ph.D. thesis, North Carolina State University (2004)
[16] Riedl, M., León, C.: Generating story analogues. In: AIIDE 2009. The AAAI Press (2009)
[17] Theune, M., Faas, E., Nijholt, A., Heylen, D.: The virtual storyteller: Story creation by. In: Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference. pp. 204–215 (2003)
[18] Turner, S.R.: Minstrel: a computer model of creativity and storytelling. Ph.D. thesis, University of California at Los Angeles, Los Angeles, CA, USA (1993)
[19] Zhu, J.: Towards a new evaluation approach in computational narrative systems. In: Proceedings of the Third International Conference on Computational Creativity (ICCC). pp. 150 – 154. Dublin, Ireland (2012)
[20] Zhu, J., Harrell, D.F.: Daydreaming with intention: Scalable blending-based imagining and agency in generative interactive narrative. In: AAAI 2008 Spring Symposium on Creative Intelligent Systems. pp. 156–162. AAAI Press, Stanford, CA (2008)
[21] Zhu, J., Ontañón, S.: Towards analogy-based story generation. In: First International Conference on Computational Creativity (ICCC-X). Lisbon, Portugal (2010)

The ICCBR 2012 Doctoral Consortium

Doctoral Consortium at the Twentieth International
Conference on
Case-Based Reasoning
(ICCBR 2012)

Lyon, France
September, 2012

David W. Aha and Thomas Roth-Berghofer (Eds.)

Organizers

David W. Aha
Naval Research Laboratory, USA

Thomas Roth-Berghofer
University of West London, UK

Mentors

Agnar Aamodt, Norwegian University of Science and Technology, Norway
Klaus-Dieter Althoff, University of Hildesheim, Germany
Kevin Ashley, University of Pittsburgh, USA
Ralph Bergmann, University of Trier, Germany
Isabelle Bichindaritz, State University of New York, USA
Sutanu Chakraborti, IIT Madras, India
Béatrice Fuchs, IAE Université Lyon 3, France
Peter Funk, Mälardalen University, Sweden
Pedro González-Calero, University of Complutense, Spain
John Hastings, University of Nebraska, USA
Luc Lamontagne, Université Laval, Canada
David Leake, Indiana University, USA
Jean Lieber, U. Henri Poincaré Nancy 1, France
Cindy Marling, Ohio University, USA
David McSherry, University of Ulster, Coleraine, Northern Ireland
Héctor Muñoz-Ávila, Lehigh University, USA
Enric Plaza, IIIA, Spanish Council for Scientific Research, Spain
Thomas Roth-Berghofer, University of West London, UK
Barry Smyth, University College Dublin, Ireland
Ian Watson, University of Auckland, New Zealand
Rosina Weber, Drexel University, USA

Preface

The objective of a doctoral consortium (DC) is to nurture the interests of students (and others) who recently started studying a specific research field. A DC provides participants with an opportunity to describe and obtain feedback on their research, future work plans, and career objectives from senior researchers and peers. For small groups such as the worldwide case-based reasoning (CBR) community, the DC is very important because it provides a forum for the community to welcome and encourage junior researchers who may become active (and even leading) community members.

The International Conference on CBR (ICCBR) held its first DC at ICCBR-09; this is the 4th ICCBR DC. In the first two years, we naturally had few participants (4 and 3, respectively). We addressed this in 2011 (e.g., with increased publicity), and were fortunate to host 10 participants. We thought that was a good number, yet we have 16 this year, which greatly exceeded our expectations. Its unclear whether too much of a good thing is wise in this case, given ICCBRs scheduling constraints, and we leave this question for future DC chairs to ponder based on feedback from this years event.

We advertised the DC widely to identify prospective participants and asked them to submit: (1) a 3-page Research Summary; (2) a 1-page CV; (3) a 1-page statement on their DC expectations; and (4) a 1-page letter of support from their advisor(s). The summary requires students to describe their objective, progress, and plans using the conferences publishing format, the CV describes the applicants experience, the expectations requires the applicant to consider what they may share or learn at the DC, and the letter ensures that advisors are aware of this event. Our wonderful PC lightly reviewed each application; all were found to be CBR-relevant and were invited to participate. We assigned a mentor per student, matching mentors who could provide valuable feedback from a different perspective (including nationality). Mentors provided iterative guidance/feedback on their presentations prior to the DC.

At the DC, each student gave a 15-minute talk on their Research Summary, followed by a 10-minute Q/A session (on presentation skills and content) led by their mentor. (Each mentor was asked to attend at least 2 students' presentations, thus allowing them to also attend co-timed events.) Also, senior researchers gave presentations to provide the students with insights on community interests and career opportunities. We thank Agnar Aamodt and Santi Ontañón for giving Career Reflection presentations and our panelists for the Dissertation Opportunities panel. Finally, a group lunch and dinner provided students with a relaxed opportunity to chat with other conference participants (the DC was open to all ICCBR registrants).

We thank the PC and mentors for their participation and assistance. We hope that it enhanced each students interest in studying CBR, and strongly encourage them to participate in future ICCBRs and related venues. We wish them well!

David W. Aha September 2012
Thomas Roth-Berghofer

Solution Diversity in Case-Based and Generative Planning

Alexandra Coman

Department of Computer Science and Engineering, 19 Memorial Drive West,
Lehigh University, Bethlehem, PA 18015
alc308@lehigh.edu

1 Introduction

Diverse planning consists of generating multiple significantly different solutions for the same planning problem. Such solutions can represent varied strategies and approaches to achieving the goal, each reflecting different priorities (such as caution versus willingness to explore), thus catering to variation in circumstances, preferences and needs. The practical value of plan diversity has led to it being addressed by several authors, including [1-4] (in generative planning), with cited application domains including route and travel planning (e.g. plans using highways vs. plans using local roads), intrusion detection (considering varied threat plans), and adaptive web-service composition. Solution diversity has been pointed out to be particularly useful in situations in which user preferences are assumed to exist, but not explicitly provided [4]. Outside planning, solution diversity has been addressed in case-based reasoning for analysis tasks, such as product recommendation, by [5,6] and many others.

We explore solution diversity in deterministic as well as nondeterministic planning[1], using various planning techniques: case-based planning [7,8], heuristic planning [9], and strong cyclic planning for nondeterministic planning problems, as described by [10]. Our algorithms produce sets of diverse solutions using a composite criterion for the evaluation of candidate (and, possibly, partial) solutions (similarly to [2,4,5]). This composite criterion balances estimated solution-set diversity with estimated adequacy of the candidate solution, where adequacy is a measure specific to each planning technique (e.g., similarity in case-based planning). The diversity of a set of solutions (which are *plans* in deterministic planning and *policies* in nondeterministic planning) is evaluated based on a solution-distance metric, i.e., a measure of the dissimilarity between two solutions. Distance metrics can be qualitative (domain-specific) or quantitative (domain-independent) [8,9].

Our expected contribution is a set of domain-independent diverse planning algorithms for various planning paradigms, all of them usable with both quantitative and qualitative distance metrics, and with no knowledge-engineering requirements apart from the distance metrics (and, for case-based planning, a case base and adaptation algorithm). Furthermore, we argue for the value of using generative and case-based diverse planning algorithms complementarily.

[1] Nondeterministic planning domains may contain actions with multiple possible outcomes.

2 Completed, Current and Future Work

This section briefly describes completed, current, and planned future work, highlighting ways in which it differs from related work.

Diversity in Deterministic Planning. In [7], we began investigating diversity in case-based planning, comparing diversity based on the initial state of the problem with diversity based on the set of actions in the plan. In [8,9], we presented algorithms for generating quantitatively and qualitatively diverse plan sets in deterministic planning domains, using a forward state-space planner and a case-based planner (hereafter referred to as DivFF and DivCBP), respectively. Unlike [1,2,4], in addition to synthetic domains, we tested our algorithms on a non-synthetic planning domain based on the real-time strategy game Wargus, assessing the diversity of generated plans by running them in the game and analyzing the game-specific results obtained (as opposed to only analyzing the sets of plans themselves). We showed that qualitatively-diverse plan sets (which we generate using only a qualitative distance metric, as opposed to an entire domain metatheory, as previously required by [1]) produce greater in-game variation than quantitatively-diverse plans.

Comparison of DivCBP and DivFF. We conducted a comparison of DivFF and DivCBP, with regard to the diversity of generated plans as well as to planning speed [11], revealing that DivFF may not be as effective as DivCBP at consistently generating highly diverse plan sets, nor as fast. In our experiment, plans obtained with DivCBP using case bases of sufficiently large sizes (with this size limit being low) were consistently maximally diverse (relative to the range of the diversity-evaluation metric), while plan sets generated with DivFF achieved lower diversity scores. We concluded that an effective way of employing the two systems complementarily would be to leverage DivFF for populating the case base up to the point where DivCBP can ensure satisfactory diversity for a sufficient number of problems, then to rely largely on DivCBP.

Plan-Based Character Diversity. We propose game character diversity as an illustrative application of diverse case-based planning. Non-player character diversity, which renders games more engaging and compelling, has been achieved in a knowledge-intensive manner by modeling characters in terms of personalities, emotional states, and social affinities ([12], and many others). We propose character diversity based on plan diversity, where types of character behavior are represented as plans. These plans are generated using qualitative distance metrics encoding indicators of various character traits (e.g. pillaging an enemy village might be indicative of greed; attacking an ogre holding a friend captive shows devotion to friends and/or recklessness). Ours is a bottom-up approach to attaining character diversity: while character modeling endows characters with specific traits which determine their behavior (a top-down approach), we create diversity in terms of behavior and, as different types of behavior reflect different personality traits, this simulates personality variation.

Diversity in Nondeterministic Planning. We have begun exploring diversity in nondeterministic planning, implementing DivNDP (built upon NDP [10]), an algorithm for generating diverse policies. Initial experimental results on synthetic planning domains show that DivNDP produces highly diverse solution sets without unreasonably inflating solution size (an indicator of solution quality, according to [10]). Unlike [3], we address diversity in *non*-probabilistic nondeterministic planning, which makes different assumptions than probabilistic planning regarding the information available for planning: action costs, state rewards, and utility functions are not provided, so planning problems cannot be cast as optimization problems. In the future, we intend to enhance DivNDP, possibly using case-based reasoning techniques, and use it to generate qualitatively-diverse policies (it has so far only been tested with a quantitative distance metric) in non-synthetic domains, such as computer games.

Acknowledgements. This research is being conducted under the supervision of Dr. Héctor Muñoz-Avila. The work was supported in part by NSF grant 0642882 and NRL. I thank Dr. David McSherry and Dr. Barry Smyth for their reviews.

References

1. Myers, K. L., Lee, T. J.: Generating Qualitatively Different Plans through Metatheoretic Biases. In: Hendler, J., Subramanian, D. (eds.) AAAI 1999, pp. 570-576. AAAI Press (1999)
2. Srivastava, B., Kambhampati, S., T., Nguyen, M., Do, B., Gerevini, A.: Domain Independent Approaches for Finding Diverse Plans. In: Veloso, M.M. (ed.) IJCAI 2007, pp. 2016-2022. AAAI Press (2007)
3. Bryce, D., Cushing, W., Kambhampati, S.: Model-Lite Planning: Diverse Multi-Option Plans & Dynamic Objective Functions. ICAPS 2007 Workshop on Planning and Plan Execution for Real-World Systems (2007)
4. Nguyen, T., Do, M., Gerevini, A., Serina, I., Srivastava, B., Kambhampati, S.: Generating Diverse Plans to Handle Unknown and Partially Known User Preferences. *Technical report arxiv: 1101.2279*. To appear in *Artificial Intelligence* (2011)
5. Smyth B., McClave, P.: Similarity vs. Diversity. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNAI, vol. 2080, pp. 347-361. Springer, Heidelberg (2001)
6. McSherry, D.: Diversity-Conscious Retrieval. In: Craw, S., Preece, A. (eds.) ECCBR 2002. LNAI, vol. 2416, pp. 219-233. Springer, Heidelberg (2002)
7. Coman, A., Muñoz-Avila, H.: Case-Based Plan Diversity. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 66-80. Springer, Heidelberg (2010)
8. Coman, A., Muñoz-Avila, H.: Qualitative vs. Quantitative Plan Diversity in Case-Based Planning. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 32-46. Springer, Heidelberg (2011)
9. Coman, A., Muñoz-Avila, H.: Generating Diverse Plans Using Quantitative and Qualitative Plan Distance Metrics. In: Burgard, W., Roth, D. (eds.) AAAI 2011, pp. 946-951. AAAI Press (2011)
10. Kuter, U., Nau, D. S., Reisner, E., Goldman, R. P.: Using Classical Planners to Solve Nondeterministic Planning Problems. In Rintanen, J. *et al.* (eds.) ICAPS 2008, 190-197. AAAI Press (2008)
11. Coman, A., Muñoz-Avila, H.: Diverse Plan Generation by Plan Adaptation and by First-Principles Planning: A Comparative Study. To appear in: Diaz-Agudo, B., Watson, I. (eds.) ICCBR 2012. LNCS, vol. 7466. Springer, Heidelberg (2012)
12. Strong, C.R., Mehta, M., Mishra, K., Jones, A., Ram, A.: Emotionally Driven Natural Language Generation for Personality Rich Characters in Interactive Games. In: Schaeffer, J., Mateas, M. (eds.) AIIDE 2007, pp. 98-100. AAAI Press (2007)

Qualitative Spatial and Temporal Reasoning with Textual Cases

Valmi Dufour-Lussier

Université de Lorraine
LORIA (UMR 7503 — CNRS, Inria)
54506 Vandœuvre-lès-Nancy, France
valmi.dufour@loria.fr

1 Introduction

The goal of my research is to propose and implement a model to reuse textual procedures within a case-based reasoning (CBR) system. Cooking recipes are representative examples of textual procedures. They form the application domain through which I intend to validate this work.

Workflow-based case representation are often used for procedures. I investigate the possibility of replacing or supplementing workflows with qualitative algebras (QAs). This has many benefits: it overcomes certain limits to workflow expressivity, can be extended to different classes of problems (e.g. spatial adaptation), and makes it possible to benefit from existing techniques (e.g. adaptation based on belief revision [1] and the TimeML text annotation standard).

This research is at the meeting point of three fields in artificial intelligence:

- CBR makes it possible to use knowledge from past problems to propose hypothetical solutions to new problems, through retrieval and adaptation.
- Natural language processing (NLP) aims at providing automatic processes to turn text into a machine-processable language. It can nowadays be used reliably to extract specific types of knowledge from text.
- Qualitative algebras define sets of relations representing the possible configurations of pairs of regions, intervals, etc. It could be used to represent and reason on procedural knowledge by associating actions with intervals.

2 Research plan

The roadmap for this research can be seen as a triangle with vertices CBR, NLP and QA. The vertices and edges can be approached in any order, and the implementation work can progress iteratively as the research progresses.

Representing procedures. There exists numerous qualitative algebras, each with a different balance of expressivity and computational efficiency. It is therefore necessary to list the temporal phenomena that must be represented in order to select the proper algebra. Additionally, algebras have certain limits in terms of representing loops or disjunctions. Different approaches to overcome those must be studied, including the possibility of combining algebras and workflows.

Acquiring cases from text and adapting text. While case acquisition from text has received much attention, the NLP-inspired techniques used have often been limited to information retrieval (IR) or information extraction (IE). Full-fledged NLP solutions have been proposed at times (e.g. [2]), but most often have been ruled out as too knowledge-intensive to be practical for CBR [3].

I make the hypothesis that, in order to acquire highly structured case representations, there is a significant advantage in using so called "deep" NLP techniques. I expect those to offer a better recall performance in procedural texts than both IR and IE, and to be more easily adapted to different application domains than IE. An extensive literature exists about spatio-temporal annotation of texts, but the representation formalisms used are not specifically made for inference engines, making further research necessary.

Moreover, it is known that text reuse gives better results than text generation [4]. Therefore the chosen annotation techniques must be such that a mapping is maintained between the text and the case representation: if the formal representation associated to a text is modified by the adaptation process, it will thus be possible to make a comparable change to the text.

Reasoning with qualitative cases. Revision-based adaptation is a CBR framework in which case adaptation is obtained by applying a belief revision operator between the source case and the constraints from the target case, with respect to the domain knowledge [1]. Procedure reuse normally entail substitutions—of ingredients, in the case of recipes. This represents a difficulty, since it is necessary for the belief revision operands to be consistent, making direct substitution impossible.

Qualitative constraint networks are used to represent procedures as constraint satisfaction problems. Recently, a collection of merge operators for those representations have been proposed [5]. Those could be adapted to provide revision-based adaptation of qualitative cases. They provide semantic similarity measures, which could be used to perform retrieval as well.

Implementation and validation. Many algorithms used for NLP and to reason with QAs are known to be intractable. Studying their complexity will be necessary, as might be researching relevant heuristics or approximation algorithms. It is difficult to evaluate the components separately. Once a first implementation of the overall system is running, user evaluation studies will make it easier to identify the weak points.

3 Progress

The following progress has been made so far:

- **Representing recipes.** I have listed the temporal phenomena that may need to be represented in recipes and similar process descriptions. This includes many phenomena that either workflows or QAs cannot represent.

Therefore, I have taken to building bridges between those types of formalisms, by defining a common-ground model-theoretical semantics for workflows and interval algebras [6]. This makes it possible to fully represent all phenomena in recipes. The next step will be to develop efficient algorithm to reason with the hybrid structures, or to create a model for the interoperation of workflow and QA inference engines.

- **Acquiring cases from text and adapting text.** I have developed a framework for case acquisition from procedural texts [7]. It reuses existing NLP tools as much as possible and proposes new solutions for certain problems specific to process descriptions, such as for the resolution of anaphoras (pronouns). Text adaptation is kept in mind but not yet implemented.
- **Reasoning with qualitative cases.** I have implemented revision-based adaptation for qualitative cases as proposed [8]. Substitutions are introduced by dividing them in two steps, each applied to one operand of the belief revision operator.
- **Implementation and validation.** Both case acquisition and revision-based adaptation were the object of a prototype implementation in Perl. The results for case acquisition are promising and the implementation is very efficient. As for revision-based adaptation, the results on the test examples are excellent, but the implementation is extremely slow though on examples larger than 5 or 6 variables.

References

1. Lieber, J.: Application of the revision theory to adaptation in case-based reasoning: The conservative adaptation. In: ICCBR. Volume 4626 of LNCS., Springer (2007) 239–253
2. Gupta, K.M., Aha, D.W.: Towards acquiring case indexing taxonomies from text. In: 17th International FLAIRS Conference. (2004) 307–315
3. Brüninghaus, S., Ashley, K.: Using machine learning for assigning indices to textual cases. In: ICCBR. Volume 1266 of LNCS., Springer (1997) 303–314
4. Gervás, P., Hervás, R., Recio-García, J.: The role of natural language generation during adaptation in textual CBR. In: Beyond Retrieval. (2007)
5. Condotta, J.F., Kaci, S., Marquis, P., Schwind, N.: Merging qualitative constraints networks defined on different qualitative formalisms. In: Conference on Spatial Information Theory. (2009)
6. Dufour-Lussier, V., Lieber, J., Le Ber, F.: Extension du formalisme des flux opérationnels par une algèbre temporelle. In Konieczny, S., Maudet, N., eds.: Sixièmes Journées de l'Intelligence Artificielle Fondamentale. (2012) 133–142
7. Dufour-Lussier, V., Lieber, J., Le Ber, F., Nauer, E.: Automatic case acquisition from texts for process-oriented case-based reasoning. Information Systems: special issue on Process-Oriented Case-Based Reasoning (2012) Submitted.
8. Dufour-Lussier, V., Lieber, J., Le Ber, F., Martin, L.: Adapting spatial and temporal cases. In Díaz Agudo, B., Watson, I., eds.: ICCBR. Volume 7466., Springer (2012) 77–91

Employing Case-Based Reasoning to Improve Symptom Management for End-of-Life Cancer Care

Krista Elvidge

Health Informatics Lab, Dalhousie University
Halifax, NS, Canada
Kelvidge@dal.ca

1 Introduction

In end-of-life (EOL) cancer care, patients commonly experience pain and other symptoms which are ineffectively managed; and subsequently detrimental to patients' physical and mental health [1]. Clinicians working in this arena often have limited training in pain and symptom management, and therefore clinical decision making (CDM) can be challenging, especially with each patient case presenting its own complexities. In this research, we present a case-based reasoning system as a solution to improve pain and symptom management in end-of-life cancer care.

Research literature suggests that care planning of pain and symptom management could be improved if clinicians were to use existing knowledge to inform their clinical decision making process [2]. Clinical decision support systems (CDSS) are computerized tools which use data and models to generate such information to support CDM. Clinical decision support systems consist of a user interface, knowledge repository, and an inference engine. Domain expertise, such as past patient cases is stored within the knowledge base. The inference reasoning engine is designed to process this information using systematic inference steps, similar to the decisional steps used in the human thought process, and uses one or more reasoning methodologies [3].

Case-based reasoning (CBR) is one such methodology that emulates the problem solving methodology commonly used in medical reasoning, and therefore, presents as a natural solution to improve the clinical decision making process in end-of-life cancer care. With case-based reasoning, clinicians rely upon past patient cases and their own clinical expertise to solve new patient cases. CBR is predicated upon the extrapolation of individual knowledge from specific, past cases, and applying their solutions to similar unsolved patient cases. The underlying premise is that similar problems have similar solutions [4]. There are distinct advantages for using case-based reasoning in medicine, especially for knowledge-based systems that are designed to assist health practitioners in diagnosis and in planning patient care [5]. CBR systems rely upon a naturally evolving case base, which is continuously integrated with new cases to reflect the most current workflow, and best practices, used for a certain procedure and clinic setting [6]. Despite the many advantages of adopting case-based reasoning, the complexity of medicine presents some challenges to its implementation [7]. Adaptation is frequently cited as the main challenge to the implementation of CBR in medicine due to the complexity and large number of features of some medical cases [6-7].

2 Research Plan

In this research, we will develop a case-based reasoning system that will enable nurse clinicians to use past patient cases to assist with care planning of pain and symptom management in end-of-life cancer care. As part of this research, we will explore the concept of multi-modal reasoning as a strategy for overcoming the common adaptation problem in medicine.

Personalization Framework. In this research, we will use thematic analysis to develop a conceptual personalization framework for end-of-life cancer care. Our qualitative thematic analysis will identify key patient variables that must be considered when planning pain and symptom management and this model will become the foundation for our CBR system. Specifically, we will integrate these key components into the user interface, case description, and as part of the inference engine.

Case Acquisition. We will extract tacit, experiential knowledge in the form of patient cases from an original dataset that consisted of 276 records of patients who had been admitted to the Victoria Hospice Society, Palliative Care Unit. The selection criteria for the extracted patient cases is the (a) presence of advanced cancer, and the (b) presence of at least one symptom from the Edmonton Symptom Assessment Scale (pain, tiredness, nausea, depression, anxiety, drowsiness, appetite loss, and dyspnea).

Case Representation. Patient cases will be represented by several feature attributes that describe the problem (e.g., mild pain) and the solution to the problem (e.g., 200-800 mg ibuprofen). In this system, the feature attributes will be determined from the thematic analysis conducted with nurse clinicians.

Case Retrieval. In this system, clinicians will be able to interact with the system via a web-based user interface. Users will enter the feature attributes of a new patient case on an electronic web form, and then perform a search query to retrieve a resultant set of cases presented to the user in order of descending similarity. We will need to further explore case retrieval algorithms to determine the best approach. In particular, we will explore multi-modal reasoning strategies.

System Evaluation. We will evaluate the system using common metrics of precision and recall to measure case retrieval accuracy, consistency, case duplication, and case retrieval relevancy.

End-User Evaluation. We will conduct qualitative evaluation with our intended end-users to measure user satisfaction, and the clinical validity of the system output.

3. Research Progress to Date

Prototypical CBR System. As part of my comprehensive requirements, I had developed an initial CBR system to explore the concept in using CBR for end-of-life cancer care. As part of this research, I examined the use of the nearest neighbor algorithm, a combined ID3 nearest-neighbor/decision tree algorithm, and compositional adaptation. This inquiry gave me a basic understanding of these concepts, and clinicians provided valuable feedback about the features of the system.

Thematic Analysis. In March 2012, I obtained ethics approval to conduct qualitative thematic analysis with nurse clinicians who deliver end-of-life cancer care in the home setting. With this inquiry, I explored the concepts of pain and symptom management, whole person care, environmental determinants of health, and personalization. Further to this analysis, I have created a model/framework for the personalization of pain and symptom management for EOL cancer patients. This framework will become the foundation of my proposed CBR system, whereby it (1) informs which patient input variables are required on the user interface; (2) provides structure for both case description and solution variables; and (3) provides information for the query process and the assignment of weights on specific variables.

Case Extraction. We have extracted *tacit, experiential knowledge* in the form of patient cases from an original dataset that consisted of 276 records of patients who had been admitted to the Victoria Hospice Society, Palliative Care Unit. The selection criteria for the extracted patient cases were (a) presence of advanced cancer, and the (b) presence of at least one symptom from the Edmonton Symptom Assessment Scale (pain, tiredness, nausea, depression, anxiety, drowsiness, appetite loss, and dyspnea). We are currently reviewing the treatment given and the associated outcomes and are creating patient case entries in a database.

End-User Evaluation Method. We have developed an evaluation strategy, and tools to measure (1) end-user satisfaction with the system; and (2) clinical validity of the system output with nurse clinicians. This methodology has received ethics approval.

References

1. Dudgeon D, et al. Managing continuity through collaborative care plans: A study of palliative care patients. http://www.chrsf.ca 2007 June.
2. Foley K, Gelband H. Improving Palliative Care for Cancer. Washington, D.C.: National Academy Press (2001)
3. Bichindaritz, I & Marling, C. Case-Based Reasoning in the Health Sciences: Foundations and Research Directions. Computational Intelligence in Healthcare 4, SCI 309, pp. 127–157.
4. Bellazzi R, Montani S, Portinale L, Riva A. Integrating rule-based and case-based decision-making in diabetic patient management. Paper presented at the 3rd ICCBR; 1999 July; Monastery, Germany.
5. Begum et al. Case-based reasoning systems in the health sciences: A survey of recent trends and developments. IEEE Transactions on Systems, Man, and Cybernetics 2011; 41(4); 421-34.
6. Bichindaritz I. Editorial: case-based reasoning in the health sciences. Artificial Intelligence in Medicine 2006; 36: 121-125.
7. Nilsson M, Sollenborn M. Advancements and trends in medical case-based reasoning: An overview of systems and system development. Available at http://www.aaai.org 2006 November.

Enhancing Situation Awareness for Decision Support by Utilizing Data Streams and Past Experiences

Odd Erik Gundersen[1,2]

[1] Verdande Technology, Norway
[2] Department of Computer and Information Science,
Norwegian University of Technology and Science, Norway
odderik@verdandetechnology.com

1 Motivation and Aim

The last decades, data that is recorded digitally has grown substantially. Not only has the sheer amount of it grown, but also the update rate to which it is being shared. Now, with update rates close to real-time, the data can be used as the foundation for making up-to-date decisions. In financial markets, many trading decisions have been fully automated and are performed by algorithmic trading agents that make decisions at a much higher rate than humans to exploit price differences in different markets. In high risk operations such as aviation control and oil well drilling operations where human lives are at stake, the data is used as input to experienced human decision makers. However, in such domains, keeping track of all the information relayed through the data is a complex task if possible at all.

In order to save cost, oil well drilling operators have moved oil well drilling specialists from rigs to centrally located integrated operation centers (IOCs) where they monitor drilling operations continuously. This is typically done by staring at graphs formed by measurements from the rigs that are updated in real-time. From these graphs, the drilling specialists interpret the oil well drilling operation and update their mental models to become aware of the current drilling situation. Based on their understanding of the current situation, the drilling specialits advise the rig crew on what to do in problematic situations. Although this has proven successful to some degree, the mental models of the drilling engineers do not necessarily improve over time. This is partly due to a more complex drilling environment, and because drilling engineers work in shifts that have lead to shorter time on work and longer time off work (now 2 week on and 4 weeks off is not uncommon). Knowledge sharing has also become more difficult: It is impossible to transfer all the information that will be relevant in the future from one shift to the next during the morning meeting. Hence, information is lost and the situation awareness (SA) decreases.

Our aim is to develop a system that will increase the SA of drilling specialits that monitor real-time drilling operations remotely from IOCs by utilizing the real-time data streams of measurements performed on the rigs to support decision making. Such a system should assist in identifying situational elements, as well as improve comprehension and indicate possible future states of the drilling situation. In addition, a system with these capabilities could potentially enable knowledge sharing between less experienced drilling engineers and drilling experts.

2 Problem Definition

Situation awareness is the perception of the elements in the environment, comprehension of the current situation and projecting the future status of the elements. Being aware of the situation is imperative for making the best possible decision in any given situation. Keeping track of all information while comprehending the situation and projecting future states impose a heavy load on the working memory. Experts alleviate the working memory by learning mental models of complex systems, comprehending them through schemata and selecting actions based on scripts, which all are developed through training and experience.

Case-based reasoning (CBR) is a method of reasoning in which past experiences are used to solve current problems and a direct descendant of the research and understanding of episodic memory and scripts. Thus, it appears to be a promising candidate for enhancing the SA of experts that attempt at becoming aware of the situation in the wellbore through observing real-time data streams.

Currently, drilling specialists have little help from decision support software. Their aids include visual and audible thresholds alarms, but they produce a high amount of false positive alarms and are because of this not heavily relied on. Other systems that utilize real-time data for monitoring drilling operations have been developed, but they rely on physically based models and lack explanation capability and experience transfer. So, even if they correctly identify a problem, they do not help the users to make the right decsions.

Furthermore, other CBR systems have been developed that reason on data streams, such as [4], and two workshops have been organized at ICCBR 2003 and ECCBR 2004 on applying CBR to time series prediction. None of these systems focused on enhancing the SA of decision makers. This is however the case of recent research at the Robert Gordon University [5].

The main research goal for the research presented here is to investigate how CBR can be used to enhance the SA of drilling engineers by using real-time data. We have formed several research questions that are to be answered by the research described in this project plan:

- **Q1: Current State of the Art** What is the current state of the art for CBR systems that reason based on data streams?
- **Q2: SA in Oil Well Drilling** How can the SA of a drilling engineer be enhanced? Will the proposed solution alleviate the perception of situational elements, comprehension of the situation and projection of the future?

– **Q3: Case Representation** What encompasses a situation in oil well drilling? How can an oil well drilling situation be represented as a case in a CBR system?
– **Q4: Streaming CBR** How can a CBR system that enhances SA by utilizing real-time data streams be designed?
– **Q5: Performance** Is the performance of the system acceptable for real-time decision support? How will the performance degrade when the complexity of the situation increases?

3 Research Plan and Progress

My research has been performed at Volve and later Verdande Technology, a software company, which has developed a decision support system for the oil well drilling domain called DrillEdge. CBR and case development have been my responsibilities since 2008, and I have lead the research team and later the software development department. In January 2011, I started this research towards a PhD and disseminating our work in Verdande Technology while investigating how our software can enhance the SA of its users. My research has been a part of the research and work of a larger group pulling in the same direction and working closely. Although this makes it harder to track ideas, I firmly believe the co-authors reflect the main contributors.

The plan is to disseminate the research answering the research questions posed in section 2 in papers published in conference proceedings and journals. Seven papers have been planned in total; four of them have been submitted and three accepted [1], [2], [3]. The research questions that are answered by these papers are Q3, Q4 and Q5. The published papers will be the main part and the backbone of the thesis, which I aim at submit in December 2012.

References

1. Gundersen, O. E. Sørmo, F., Aamodt, A., Skalle, P.: A Real-Time Decision Support System for High Cost Oil-Well Drilling Operations. In: Innovative Applications of Artificial Intelligence, IAAI (2012)
2. Gundersen, O.E., Sørmo, F.: An Architecture For Multi Dimensional Temporal Abstraction Supporting Decision Making In Oil-Well Drilling. In: Combinations of Intelligent Methods and Applications, CIMA (2012)
3. Gundersen, O.E.: Toward Measuring the Similarity of Complex Event Sequences in Real-Time. In: Proceedings of the 20th International Conference on Case-Based Reasoning (2012)
4. Montani, S., Bottrighi, A., Leonardi, G., Portinale, L.: A CBR-Based, Closed-Loop Architecture for Temporal Abstractions Configuration. Computational Intelligence 25(3), 235–249 (2009)
5. Nwiabu, N., Allison, I., Holt, P., Lowit, P., Oyeneyin, B.: Case-based situation awareness. In: Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2012 IEEE International Multi-Disciplinary Conference on. pp. 22 –29 (march 2012)

Retrieve and Reuse of Experiential Procedural Knowledge

Mohd. Siblee Islam

University of Trier - Department of Business Information Systems II
D-54286 Trier, Germany
islam@uni-trier.de

1 Introduction

The Internet is a source of experiential knowledge. People post and share their personal experiences through social websites like blogs, forums etc. Procedural knowledge is a kind of knowledge that represents how to achieve a certain goal by means of performing certain activities. Today there is no real support to find and reuse procedural knowledge available in the internet. The main focus of the project EVER(Extraction and Processing of Procedural Experience Knowledge in Workflows), running at University of Trier is to find a feasible solution for it. The PhD topic of mine is a part of the project and mainly focuses on developing methods to find and reuse of the procedural knowledge. Another PhD topic under the same project mainly focuses on the extraction of the procedural knowledge. Procedural knowledge will be extracted and represented in a form of workflows. Typically, workflows are “the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules” [1]. For representing procedural knowledge in a workflow, the representation pattern of workflows will be adopted so that a procedural knowledge will be represented as workflows, which comprises of activities or tasks and flow of control among them like sequence, loop etc. plus data and flow of them.

We consider cooking as our primary application domain for a few reasons. First, the preparation steps of cooking constitute a procedural knowledge to be performed in a sequence which can be represented in a workflow. Second, that is easy domain to understand for common people. Third, a huge number of people are involved in it through the web and contributing their experiential knowledge. Our methods for retrieval and adaptation will also be tested in another experiential domain available in internet communities like installation guidance for software, troubleshooting etc.

Case-based Reasoning (CBR) is considered to find workflows for a query based on similarity matching between query and the workflows stored in the repository. Usually, the workflow retrieved for a particular query is not sufficient to fulfill all requirements of a user. In that case, the workflow should be adapted to fulfill the missing requirements of the user by making necessary changes based on experiential adaptation knowledge. The adaptation can also be required for some unwanted changes in user scenario. As an example, soybean oil is used in a cooking preparation can replace olive oil when it is not available. Adaptation can be done to cope with preference issue. As an example, I want to get a cooking preparation of egg but in the same time I want to avoid an activity like frying or an ingredient like oil.

The research questions can be specified as follows:

1. How to represent workflow-cases for retrieval?

2. How to model similarity measures for workflow retrieval to achieve efficient selection of reusable workflows?

3. How can experience be stored in a workflow and be reused automatically to solve new tasks?

4. How to evaluate new similarity measures with respect to quality and efficiency of the retrieval result and how to evaluate adaptation methods?

5. How to cope with the hierarchical structure of workflows, by referring to a workflow within a workflow?

2 How to approach these research questions

Probable way to solve the research problem are as follows.

1. The workflow can be formulated into cases for CBR by considering the graph representation proposed by Bergmann and Gil [2]. In a graph, activities and data processed or produced by them are represented by task nodes and data nodes respectively. Edges are used to represent the control flow and data flow inside the workflow. Nodes and edges are labeled by semantic information.

2. Similarity function for local and global similarity introduced in [2] can be used as similarity measure between query and case. The domain ontology used for workflow extraction can be adopted to model local similarity measures according to the domain knowledge. *Adaptation Guided Retrieval* (AGR) [3] can also be considered to increase the probability of successful adaptation.

3. Two adaptation approaches, *inductive learning of adaptation rules from the cases* and *case-based adaptation* will be investigated. In the *inductive learning of adaptation rules from the cases*, the rules for adaptation are generalized by means of inductive learning on a set of adapted cases, which can be used for the adaptation later on. In the *case-based adaptation*, similar workflows which have been adapted successfully in the past, will be considered [4] for the adaptation. In this type of adaptation, the similarity measures consider not only the data and tasks used in two workflows but also their structural representations.

4. The evaluation will be done in three ways. First, by using a simulation test bench by considering simulated workflow. Second, by considering real workflows from experts, which can be constructed manually. Third, developing a prototype by implementing the retrieval and adaptation method and use it in CCC (Computer Cooking Contest) competition. It will make a real world evaluation that will be done in international level. The consistency of the workflow after adaptation, the quality of retrieval and its impact on the adaptation if AGR is considered, and sensitivity by considering the missing information in both

retrieved and adapted result, modeling effort required for ontology and similarity measure etcetera will be assessed in evaluation phase. Expert will also be involved in assessment for second type of evaluation.

5. For hierarchical workflows, we have to find out new case representations and necessary adjustments for similarity measures used in retrieval and adaptation. But we will consider this issue at relatively later part of the research. The methods which will be introduced to do it will also be evaluated as discussed before.

3 Current state of research

A prototype for workflow retrieval is being developed inside the CAKE (Collaborative Agent-based Knowledge Engine) framework. CAKE is a framework that is developed in University of Trier. The workflows extracted by an algorithm developed by Pol Schumacher [5] (who is working under same project for his PhD research) are used as a collection of workflow. We can convert the workflows into graph representation proposed by Bergmann and Gil [2] to formulate cases. The similarity model for graphs given in [2] is used. Local similarity is computed by considering the text similarity. We are looking forward to make a model of domain knowledge for an improved similarity measure by considering an ontology for the cooking domain. Currently, the name of task and data are only considered as semantic information. We are looking forward to consider more semantic information for data and tasks, which are available in workflows and transformed into case representation. If the case base grows, the retrieval consumes much time and memory space. We have to find out a possible way to overcome this problem.

References

1. Workflow Management Coalition: Workflow management coalition glossary & terminology. http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf (1999) last access on 05-23-2007.
2. Bergmann, R., Gil, Y.: Retrieval of semantic workfows with knowledge intensive similarity measures. In: Case-Based Reasoning. Research and Development, 19th International Conference on Case-Based Reasoning, ICCBR 2011, Springer (2011) 17–31
3. Smyth, B., Keane, M.T.: Using adaptation knowledge to retrieve and adapt design cases. Knowledge-Based Systems **9**(2) (1996) 127–135
4. Minor, M., Bergmann, R., Görg, S., Walter, K.: Reasoning on business processes to support change reuse. In: 13th IEEE Conference on Commerce and Enterprise Computing, Luxemburg, IEEE Computer Society (2011) 18–25
5. Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: WWW'12 Workshop Proceedings, ACM (2012)

A Case-based Reasoning System for Radiotherapy Treatment Planning for Brain Cancer

Rupa Jagannathan

School of Computer Science, ASAP Research Group, University of Nottingham
Jubilee Campus, NG8 1BB. UK
psxrj@nottingham.ac.uk

1 Introduction

The aim of this research is to investigate the application of case-based reasoning (CBR) to radiotherapy treatment planning in brain cancer patients and to design a decision support system that will help radiotherapists make treatment decisions using CBR. In radiotherapy, tumor cells are destroyed by subjecting them to ionizing radiation. However, excessive radiation adversely affects all cells, including healthy tissue and critical organs. Therefore, the aim of radiotherapy treatment planning is to deliver a tumoricidal dose over the tumor region while minimizing the radiation received by healthy tissue and organs at risk (OAR) in the vicinity of the tumor. To realize this goal, a detailed treatment plan is created for each patient that describes exactly how a patient should be irradiated.

Currently in many hospitals, including our project collaborator, the Nottingham University Hospitals Trust, NHS, City Hospital Campus, treatment planning is done manually using a trial and error approach in which the planning parameters are adjusted iteratively to achieve an acceptable dose of irradiation. Generating a good treatment plan can take from a few hours to a few days in complicated cases and requires the expertise of one or more experienced medical physicists. With advances in technology the complexity of treatment planning has increased. The disadvantage of the mathematical models used in automated treatment planning systems, usually discussed in the literature, are that sometimes the plans generated are not clinically acceptable and also the computation time per plan can be very high [1, 2]. However, the largest drawback of these systems is that they completely disregard the experience gained by expert physicians. After years of practice, many senior oncologists and medical physicists gain a lot of empirical knowledge about what treatment plan configurations are suitable for particular cancer cases even when the exact underlying causes are not entirely understood. [3] In our work, we plan to capture this kind of intuitive and empirical expert knowledge using CBR.

My research work focuses on studying the feasibility of generating treatment plans for brain tumor patients based on previously treated similar patients. In order to accurately calculate the similarity between cases in a way that is meaningful to radiotherapy planning, I have designed a novel two phase retrieval mechanism in which the similarity measured is based on the weighted nearest neighbor technique and fuzzy sets. A large part of the work concerns the identification of attributes and their weights that describe a patient in relevant terms and that can also be utilized by an automated decision support system. A novel approach to attribute weighting that assigns local weights using rules and takes into consideration the reliability of the weight assignment during the learning phase results in a better retrieval accuracy. The retrieval system has been thoroughly tested on real world cases obtained from the City Hospital. The evaluation is carried out using cross validation and the success rate obtained on the test cases is deemed as a good starting point for adaptation.

2 Research Objectives

Identifying and extracting relevant data from brain cancer patient files

Manual treatment planning is usually done based on computed tomography (CT) patient images (in DICOM -Digital Imaging and Communications in Medicine- format [4]), which also form the sole input to the designed CBR system.

Identification of relevant case attributes and their weights

The case attributes have to be carefully selected so that the similarity between two cases accurately represents how applicable the solution of a case is to the target case. [5]. Attribute weighting improves the retrieval accuracy, furthers domain understanding and reduces data storage and attribute measurement requirements [6, 7].

Design of the similarity measure

The retrieval mechanism depends on the design of the similarity measure, which constitutes the main part of the inference engine.

Evaluation of the retrieval mechanism

The designed retrieval mechanism has to be thoroughly tested to ensure its capability of retrieving similar cases whose solution is relevant to the target case.

3 Research Progress

Identifying and extracting relevant data from brain cancer patient files

Information relevant to treatment planning has been identified after consultation with medical physicists at the City Hospital. This includes primarily structural information about the brain tumor and the organs at risk (OAR) found in the DICOM patient files. The DICOM data is first converted to text files and the extracted information is preprocessed into a form that can be used by the CBR system.

Identification of relevant case attributes

Since the aim of radiotherapy treatment planning is to focus the prescribed radiation on the PTV and to avoid the OAR, the treatment plan parameters are determined based on the location and dimensions of the tumor and the spatial relationship between the tumor and the OAR.

Design of the retrieval mechanism and similarity measure

The similarity measure is based on the commonly used weighted nearest neighbor similarity measure and fuzzy set theory. Also, case retrieval is performed in several stages. Each stage retrieves a case using weights optimized with respect to a solution parameter [8].

Attribute selection and weighting

An extensive attribute weights analysis has been carried out and global weights were determined using a wrapper approach. I have also designed a novel weighting scheme that assigns local weights using rules [9].

Evaluation

The design decisions are evaluated mainly in terms of retrieval accuracy using cross validation techniques on real world brain tumor patient cases. The final system will be validated using radiotherapy treatment planning experts.

4 References

1. Kalet, I., Paluszynski, W.: Knowledge-Based Computer Systems for Radiotherapy Planning. Americal Journal of Clinical Oncology 13(4), 344-351 (1990)
2. Meyer, J., Hummel, S.M., Cho, P.S., Austin-Seymour, M.M., Phillips, M.H.: Automatic Selection of Non-Coplanar Beam Directions for Three-Dimensional Conformal Radiotherapy. The British Journal of Radiology 78, 316--327 (2005)
3. Shephard, D.M., Ferris, M.C., Olivera, G.H., Mackie, T.R.: Optimizing the Delivery of Radiation Therapy to Cancer Patients. SIAM Review 41, 721--744 (1999)
4. National Electrical Manufacturers Association: DICOM - Digital Imaging and Communications in Medicine. http://medical.nema.org.
5. Berger, J.: Roentgen: Radiation Therapy and Case-Based Reasoning. Proceedings of the Tenth Conference on Artificial Intelligence for Applications, pp. 171-177 (1994)
6. Li, Y.F., Xie, M., Goh, T.N.: A study of mutual information based feature selection for case based reasoning in software cost estimation. Expert Systems with Applications 36, 5921–-5931 (2009)
7. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: Machine Learning: Proceedings of the Elevent International Conference. (1994)
8. Jagannathan, R., Petrovic, S., McKenna, A., Newton, L.: A Novel Two Phase Retrieval Mechanism For A Clinical Case-Based Reasoning Systems For Radiotherapy Treatment Planning. International Journal on Artificial Intelligence Tools. *To appear* (2012)
9. Jagannathan, R., Petrovic, S.: A Local Rule-based Attribute Weighting Scheme for a Case-based Reasoning System for Radiotherapy Treatment Planning. In: ICCBR 2012. LNCS, Springer, *To appear* (2012).

Integrated Learning for Goal-Driven Autonomy

Ulit Jaidee

Department of Computer Science and Engineering, Lehigh University
19 Memorial Drive West, Bethlehem PA 18015

ulj208@lehigh.edu

1 Introduction

There is a large research effort on AI planning that focuses on automated plan generation in which a planning agent is provided with a set of inputs that include an initial goal or set of goals for the agent to accomplish. In this context, the goal is assumed to be static and the agent is not provided with the ability to reason about whether it should change this goal. For some tasks in complex environments, this constraint is problematic; the agent will not be able to respond to opportunities or plan execution failures that could benefit from focusing on a different goal [1-2].

When the dynamics of the environment suddenly change or keep changing continuously, classic models of planning do not handle the problem well enough or fast enough. AI researchers have repeatedly acknowledged the limiting assumptions of classic algorithms for automated planning [3-4]. For example, these classic algorithms assume static environments (where changes in the environment are due only to the planner's actions), off-line planning (where the planner does not monitor execution), and that all goals are fixed/unchanging. Naturally, these assumptions do not always apply. For example, team shooter games are dynamic environments that are populated by multiple agents resulting in exogenous events. Also, the agents must perform online planning by executing their plans during the game. Finally, the goals of the game change as the game state changes (e.g., if a win is infeasible, then the agent should attempt to gain a draw) [1-2].

Goal driven autonomy (GDA) is a reasoning framework that was introduced to address the limitations mentioned; GDA systems perform anytime reasoning about which goal(s) should be satisfied. Although promising, there are natural roles that case-based reasoning (CBR) can serve in this framework, but no such theory or systems exist [1-2]. In this thesis, I propose a GDA framework and an algorithm that uses CBR to support it. I also propose an empirical study with multi-agent gaming environments such as Domination Game. Later, I propose how goals can be automatically learned and how GDA can apply to some real world situations. Our proposed thesis will be the first comprehensive study of GDA, a field that has gained a lot of attention in recent years.

2 Research Plan

The goal of our research is to build an AI system that has abilities to adapt itself to new environments; the system that can reason about which goals to pursue. Conceptually I have explored so far, algorithms for goal management and for discrepancy[1] detection. For the next step, I would like to explore algorithms for generating policies[2] as opportunities arise to construct learning systems to generate new goals and manage them. I can use learning models such as reinforcement learning to learn a new policy or to update a current policy from the opponent. In this case, I can choose what I should explore next between two directions: (1) learning from the opponent's behavior, or (2) learning to conquer the opponents without assuming prior knowledge about their behavior. For the first direction, when training against very intelligent AI teams to compete with, my agent can learn from them. This is a good idea because the strategy of good teams can help to improve my agent's strategy to beat many other teams. However, because I have the hypothesis of taking actions of a team based on the team's observations, my agent might encounter states never visited before by the observed agents; therefore, my agent will not know what to do in such states.

As per the second direction, learning to conquer the opponents without assuming prior knowledge about their behavior seems like a better idea. However, our reinforcement learning process might require so many iterations before it learns a suitable policy.

I will continue exploring these issues further.

3 Progress

So far I have built several GDA systems; each new system adds new degrees of autonomy by learning new knowledge where my previous system required this knowledge to be given as input. I have done experiments in DOM and Wargus with my GDA systems. DOM is a real-time domination game where each team wins by controlling points called 'domination points' on the map. Wargus is a real-time strategy game where opposing players commands virtual armies in battle against each other or a computer-controlled enemy.

---

[1] A discrepancy occurs when the expected situation after taking an action and the state that is observed do not match.

[2] A mapping from states to probabilities of selecting each possible action

References

1. Munoz-Avila, H., Aha, D.W., Jaidee, U., Carter, E. (2010). Goal-driven autonomy with case-based reasoning. *Proceedings of the 18th International Conference on Case Based Reasoning* (pp. 228-241). Alessandria, Italy: Springer.
2. Munoz-Avila, H., Aha, D.W., Jaidee, U., Klenk, M., & Molineaux, M. (2010). Applying goal directed autonomy to a team shooter game. *Proceedings of the Twenty-Third Florida Artificial Intelligence Research Society Conference*. Daytona Beach, FL: AAAI Press.
3. Ghallab, M., Nau, D.S., & Traverso, P. (2004). *Automated planning: Theory and practice*. San Mateo, CA: Morgan Kaufmann.
4. Nau, D.S. (2007). Current trends in automated planning. *AI Magazine*, **28**(4), 43–58.

Reasoning System for Computer Aided Diagnosis and Explanations Aware Computing for Medical Applications

Abdeldjalil Khelassi

Informatics department, Abou Bakr Belkaid University,
BP 230 Tlemcen, Algeria.
Khelassi.a@gmail.com

1 Introduction

The artificial reasoning systems are now very powerful for resolving much kind of complex problems in all Translational science domains. Also there are many investigations in this area which focus on modeling combining implementing and integrating some intelligent methods for exploring the useful knowledge extracted automatically from a data warehouses or modeling from the human experts expertise's which represent the results of many year of manual observations. These investigations are developed for responding to the challenges of health science applications as accuracy transparency flexibility and other optional needs as adaptability…etc.
The Explanation Aware Computing EXACT, which is hardly and widely developed by the research community, with many goals and many kinds is a suitable trend for all Medical IT users including doctors, developers and patients. For developing a strong relationship based on the trust and believes between the application and users. The EXACT become not an option but indispensable criteria of the newest complex smart applications for medical purpose where the life of the human is the first preoccupation.
Our thesis focus on two trends the first one is to develop a strong Case based reasoning system for medical computer aided diagnosis and explanation aware computing for this we have combined many theoretical aspect in the domain of artificial reasoning as well as case based reasoning, distributed reasoning and reasoning under uncertainty. The second trends is to develop a web centered application which reuses the generated explanations from the CBR system for prevention, home health caring and document recommendation system. This system should ensure an adapted interaction between the reasoning system and all kind of users by using the Web 2.0 (social network, wiki…etc) technologies.

2 Research plan

2.1 CBR classifier for computer aided diagnosis

The case based reasoning is a successful paradigm in the health science applications many realized system was cited in [15] but each one of them is specialized in some diseases and need more accuracy for responding to the uncertainty of medical information. This paradigm has a large use in many domains; also there are many developed variants [13] which give the possibility to solve many kinds of problem as the classification.
The first step on our thesis is to develop an accurate CBR system for medical computer aided diagnosis, by integrating the distributed CBR [17], IK-CBR [16], fuzzy sets [14] and data mining approaches [18]. The developed system should be evaluated and compared with the performance of related works.

2.2 Explanation aware computing

In human to human interaction, the ability to explain its own behavior and course of action is a prerequisite for a meaningful interchange; therefore a truly intelligent system has to provide

comparable capacities. [3] But on the case of human machine interaction where there are a complex recorded knowledge and a mass application users with a different goals and kinds and in sometime a critical kind of application as health science applications an adaptive explanations become a necessity not just an option.
These explanations could be divided into four types [Swartout and Smoliar, 1987; Chandrasekaran et al., 1989; Gregor and Benbasat, 1999]:

• **Reasoning Trace:** Producing an explanation from the trace of the reasoning process used by the system to find the solution. Examples are MYCIN's how and why explanations [Clancey, 1983].

• **Justification:** Providing justification for a reasoning step by referring to deeper background knowledge. This type of explanation was first offered by the XPLAIN system [Swartout, 1983].

• **Strategic:** Explaining the reasoning strategy of the system. The NEOMYCIN system first provided this kind of explanation [Clancey, 1983].

• **Terminological:** Defining and explaining terms and concepts in the domain. This type of explanation was identified in [Swartout and Smoliar, 1987].

Also five goals a user can have with explanations are introduced, namely 1. **Transparency** (explain how the system reached the answer), 2. **Justification** (explain why the answer is a good answer), 3. **Relevance** (explain why a question asked is relevant), 4. **Conceptualization** (clarify the meaning of concepts), and 5. **Learning** (teach the user about the domain). [1, 2]

2.3 Health 2.0

The health 2.0 and homecare systems based on the web 2.0 technologies realize a tie between the health care actors (doctors, patients and the other users) for explanation and prevention. For example the Facebook web site is an interactive social network with a mass uses (880.5 million users in Dec 2012 on the world) launched in February 2004 contains an important API for developers with an ubiquities ability for relevant applications in health sciences. But the problem of online information is that this information can be inaccurate, incomplete, controversial, misleading, and alarming for individuals with health questions [4, 5]. A reasoning system will realize the connection between all these capacities for a newest and original healthcare application. This system will contains a set of cognitive agents each one will be specialized for resolving some sub-problems. The system will be divided on tow parts: 1-a CBR system for online computer aided diagnosis discussed in section 2.1. 2- An explanation agent which reuse the traces on the log files and a sets of documents.
The second step of our thesis will focus on developing an explanation agent which can ensure an adaptive explanations for a mass uses with categorization of these users. Also the agent will enrich the explanation by a document recommendation system centered on the web2.0 technology (social networks, wikis and blogs) for health caring and prevention by avoiding the problems of online information which can be inaccurate, incomplete, controversial, misleading, and alarming for individuals with health questions.

3 Progress

A case based reasoning system is developed for medical diagnosis and classification [1] through some parameter taken from the patient the system can generate the disease of this patient. We have also evaluated this system by a benchmark with two international medical data sets cardiac arrhythmias and breast cancer, the results and the comparison with the related works exist in [8,10,11,12] and [6,7]. Also we have applied and compared between deferent strategies and algorithms in [8,9,10,11,12]. Many problems was discussed and resolved in the published works as transparency, distributed reasoning, case base learning, features selection, intensive-knowledge CBR, data mining with CBR, Uncertainty measures and Fuzzy CBR. A novel similarity measures was developed and evaluated in the classifier for enriching the

retrieving process by merging the fuzzy sets and the traditional global-local similarity measures presented and evaluated in [7,8,9] and an uncertainty measures function is proposed in [6].
As cited above the researchers distinguish four kinds of explanation also five user's goals. In the developed explanation agent just one kind of explanation is published in this moment which is the reasoning trace by visualizing the recorded traces situated in the log files in [8]. An interface for explanation is developed which ensure the justification and strategic explanation. The terminological explanations will be crucial for the document recommendation system. User goals for explanation will be integrated for the categorization of users with the social profile and contacts offered by Facebook API. The design of a health 2.0 application for explanation and prevention is done it lacks just some technical investigations.

References

[1] Thomas R. Roth-Berghofer and Jörg Cassens “Mapping Goals and Kinds of Explanations to the Knowledge Containers of Case-Based Reasoning Systems”Héctor Muñoz-Avila and Francesco Ricci, editors, Case Based Reasoning Research and Development – ICCBR 2005, volume 3630 of LNAI, pages 451– 464, Chicago, 2005. Springer.
[2] Jörg Cassens and Anders Kofod-Petersen, “Explanations and Case-Based Reasoning in Ambient Intelligent Systems” David C. Wilson and Deepak Khemani, editors, ICCBR-07 Workshop Proceedings, pages 167–176, Belfast, Northern Ireland, 2007.
[3] Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, explanation-Aware Computing – Papers from the 2007 AAAI Workshop, number WS-07-06 in Technical Report, pages 20–27, Vancouver, BC, 2007. AAAI Press.
[4] Glynn et al. BMC Cancer 2011, The effect of breast cancer awareness month on internet search activity - a comparison with awareness campaigns for lung and prostate cancer 11:442
[5] Matthew S. Katz, al “The ‘CaP Calculator’: an online decision support tool for clinically localized prostate cancer BJU International” Volume 105, Issue 10, pages 1417–1422, May 2010.
[6] A.khelassi ": Data mining application with case based reasoning classifier for breast cancer decision support" MASAUM International Conference on Information Technology 2012 MICIT’12, 27-29 July, 2012 Liverpool, UK
[7] A.Khelassi, “Fuzzy Case Based Classification applied in the classification of breast cancer pattern” Doctorial STIC11, TEBASSA, Algeria, 2011.
[8] A. Khelassi, MA Chikh “Fuzzy knowledge-intensive case based classification applied in the automatic cardiac arrhythmias diagnosis” Proceeding de la 8ème édition du Colloque sur L’Optimisation et les Systèmes d’Information COSI 2011 P460, 24 -27 Avril, Guelma, Algérie
[9] A.KHELASSI “Fuzzy Similarity Based classification Versus Intensive Knowledge Case Based classification” Colloque International sur les Nouvelles Techniques Immuno-Cognitive dans les Réseaux Informatiques NTICRI ’09, Oran, Algeria
[10] A.KHELASSI, M A CHIKH «Distributed classification using Intensive Knowledge Case Based Reasoning" Conférence Internationale des Technologies de l’Information et de la Communication CITIC’2009. Setif Algeria.
[11] A.KHELASSI, M A CHIKH "Automatic Recognition of Cardiac arrhythmias using a Distributed Case-Based Reasoning Classifier" COSI 08, Tizi ouzou, Algeria.
[12] A.KHELASSI, M A CHIKH " Distributed case based reasoning for cardiac arrhythmias diagnosis" MCSEAI 2008 10th Maghrebian Conference of Software Engineering and Artificial Intelligence, 28- 30 April 2008, Oran, Algérie,
[13] C. Marling, M Sqalli, E Rissland, H Munoz-Avila,D Aha "Case-Based Reasoning Integrations" AI Magazine Volume 23 Number 1 (2002)
[14] R Weber "Fuzzy set theory and uncertainty in case-based reasoning" Eng Int Syst (2006) 3: 121–136 2006 CRL Publishing Ltd
[15] I Bichindaritz, C Marling, Case-based reasoning in the health sciences: What's next? Artificial Intelligence in Medicine 36 (2), 127-135.
[16] Agnar Aamodt “Knowledge-Intensive Case-Based Reasoning in CREEK” ADVANCES IN CASE-BASED REASONING Lecture Notes in Computer Science, 2004, Volume 3155/2004, 793-850.
[17] Plaza Enric; McGinty Lorraine Distributed case-based reasoning. The Knowledge engineering review, Vol. 20, N. 3, pp. 261-265, 2006.
[18] Ruimin Shen and al, Data Mining and Case-Based Reasoning for Distance Learning International Journal of Distance Education Technologies (IJDET) Volume 1, Issue 3. 2003.

Case Selection in Ambient Assisted Living *

Eduardo Lupiani

University of Murcia, Faculty of Computer Science
Murcia, Spain
elupiani@um.es

1 Thesis Hypothesis

In order to build an efficient Case-Based Reasoning system (henceforth CBR Systems), two main issues can be identified: the similarity function between cases and an appropriate case selection to build the case base. On the one hand, a good similarity function may increment the accuracy rate of the CBR System. On the other hand, a disproportionately case base size may cause the *utility problem*, which arises when the number of cases learned, in an attempt to improve a system's performance, degrades performance instead [8]. The consequences are longer problem solving time and worsening of parameters as accuracy and false positive rate, even when the learned cases are correct [7]. This problem appears commonly in Ambient Assisted Living (AAL) systems, where it is needed large quantities of cases to describe the temporal relations between heterogeneous event sequences, for instance, to perform human location tracking. By using Case Selection Techniques (CST) the reduction of the case base size is possible, for instance deleting redundant cases, and to keep the problem domain coverage like the original case base as long as possible [11, 10]. The question that remains is if existing CST are valid for domains where temporal relations exist either among cases or among their features. Therefore, the central hypothesis of our work is that existing static CST can be adapted to develop new CST for cases in which temporal dimension is explicitly represented and, therefore, AAL systems shall not suffer of *utility problem*.

2 Problem Outline

Case-based systems can produce good quality solutions in weak theory domains, and this is the key of their success [1]. The knowledge unit is known as Case, a pair of a description problem and its solution, and the kernel of the CBR systems is its Case Base, a collection of indexed cases. To learn a new case this is added to the Case Base. This learning process could cause the *utility problem*, which arises when the number of cases learned, in an attempt to improve a system's performance, degrades performance instead [8].

* This PhD was partially financed by the Spanish MEC through projects TIN2009-14372-C03-01 and the project 15277/PI/10 funded by Seneca Agency of Science and Technology of the Region of Murcia within the II PCTRM 2007-2010.

In order to perform location tracking of a person on his/her house, we use an AAL system in which person location is acquired by a set of sensors. In this kind of systems, a case describes the activity in the house of a person within a period of time. This description involves temporal relations between the different locations visited by the person. Because the possible combination of events rely on the person behaviour and the house distribution, it makes impossible to build a generic case base for everyone, thus a learning period it is necessary when the track activity starts, and meanwhile the learning process continues, new cases are created to describe the normal activity. That means to create large amount of cases, being some of them noisy because of sensor stochastic behaviour. As we mentioned above, this situation may cause longer problem solving time and worsening of parameters as accuracy and false positive rate.

In order to lessen the utility problem, a Case Selection Technique (CST) can reduce the amount of cases in the Case Base according to a particular selection strategy. However, there is not a best unique CST, mainly because the case distribution and the similarity between cases influence over the cases to be chosen, either positively or negatively.

These CSTs are commonly used by Maintenance Tasks [4], or in the Case Mining process [10], although with different purposes. Case Mining is a global optimization problem, while Maintenance Tasks are local one [10]. To select the most suitable CST, we have to evaluate the results, using either Hold-out or Cross-Validation [10]. However, these methods are aimed to compare the quality of the results only (accuracy, false positive rate, etc.), and they do not take in consideration neither the reduction rate nor the execution time of the CST. This last is a key factor if we are planning to use them in Maintenance Task. To this end, we have proposed a new evaluation technique to include these issues in the evaluation process, furthermore, it uses the same case test set to evaluate the reduced case bases by different CSTs [3].

CSTs are good established in CBR ecosystem. Nevertheless, almost all the literature in CBR neglects the temporal dimension, thus none of them have been studied tirelessly in those domains where temporal relations are in consideration, either inside case features or between cases themselves. There are many application fields with such requisites, in where CBR has been applied, such as process supervision, forecasting, robot control, medical diagnosis, etc. However, some frameworks have been proposed to work with temporal dimension [2, 9]. The question that remains is if it is possible to adapt existing CST to apply them in temporal related domains as our AAL system, and whether the current evaluation techniques can be used in those scenarios or not.

3 Research Outline

Objectives

(A) To propose new CST, considering the huge dimension of case features and reducing the *utility problem* in case bases with temporal cases.
(B) To consolidate a framework for evaluating CSTs.

(C) To validate the use of CSTs in Ambient Assisted Living to test the enhancement of a CBR system which uses them.

Activities (Research plan)

(A).1 To study the most relevant CST [10] and, for those where possible, to review them for their use with temporal cases.
(A).2 To study temporal similarities for being applied in temporal cases.
(A).3 To study existing Temporal CST proposals, and to review them.
(A).4 To propose new Temporal CST methods

(B).1 To build an evaluation framework to choose the best CST for a case base, with temporal or atemporal cases. This software will include the most popular CSTs.

(C).1 To test the evaluation framework in Ambient Assisted Living scenario.

Current Status

1. A study of CSTs, and evaluation of CBR systems [5, 6].
2. CELSEA, a software library with the most known of CSTs in Case Maintenance [3].

References

1. Cunningham, P.: CBR: Strengths and weaknesses. In: Tasks and Methods in Applied Artificial Intelligence. LNCS, vol. 1416, pp. 517–524 (1998)
2. Jaczynski, M.: A framework for the management of past experiences with time-extended situations. In: Proceedings of the sixth international conference on Information and knowledge management. pp. 32–39. CIKM '97 (1997)
3. Juarez, J.M., Lupiani, E., Palma, J.: Case selection evaluation methodology. In: Industrial Conference on Data Mining - Workshops. pp. 17–29 (2011)
4. Leake, D., Wilson, D.: Categorizing case-base maintenance: Dimensions and directions. In: Smyth, B., Cunningham, P. (eds.) Advances in Case-Based Reasoning. LNAI, vol. 1488, pp. 196–207 (1998)
5. Lupiani, E., Jimenez, F., Juarez, J.M., Palma, J.: An evolutionary multiobjective constrained optimisation approach for cae selection: Evaluation in a medical problem. In: CAEPIA. pp. 383–392 (2011)
6. Lupiani, E., Juarez, J., Jimenez, F., Palma, J.: Evaluating case selection algorithms for analogical reasoning systems. In: Foundations on Natural and Artificial Computation, LNCS, vol. 6686, chap. 36, pp. 344–353 (2011)
7. Markovitch, S., Scott, P.: The role of forgetting in learning. In: Proceedings of The Fifth International Conference on Machine Learning. pp. 459–465 (1988)
8. Minton, S.: Quantitative results concerning the utility of explanation-based learning. Artificial Intelligence 42(2-3), 363–391 (1990)
9. Montani, S., Portinale, L.: Case based representation and retrieval with time dependent features. In: ICCBR. pp. 353–367 (2005)
10. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. Artificial Intelligence 171(16-17), 1039–1068 (2007)
11. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning 38(3), 257–286 (2000)

Case-Based Reasoning to Model Miscanthus Allocation: A French Case Study

Laura Martin

INRA - French National Institute of Agricultural Research, ASTER-Mirecourt research unit, 662 avenue Louis Buffet, 88500 Mirecourt, France
laura.martin@mirecourt.inra.fr

1 Introduction

To face the decrease of fossil energy supplies and to reduce the greenhouse gas emissions, new biomass energy[1] resources are of a great interest in Europe [1], like the miscanthus (Miscanthus ex giganteus), which one has interesting caloric values and constitutes a great potential for biofuel and heating plant. Their spatial extension seems then unavoidable. As the production of such biomass crops is perennial (15 to 20 years) and exclusively dedicated to the energy use, it is necessary to anticipate their allocation to prevent a forecasted perennial food / non-food competition.

Modeling can then be a way to anticipate the extension of biomass crop and bring decision-making support for politics. Several land-use change models deal with this specific problem [2]. Most of these models simulate large-scale allocation processes, taking into account numerous biophysical variables but only few human decision-making processes linked to the land system management of farmers, whereas it is a major driving factor of miscanthus allocation process [3]. Indeed, representing and modeling human behavior and decisional processes regarding land use change is difficult and constitutes a main research challenge [4].

The goal of our research is then to model the miscanthus allocation according to farmers' practices and decision-making process. Because the allocation of miscanthus is too recent to use national statistics, we decided to build our model based on a case study in Burgundy (East of France). In the same way, because the allocation of miscanthus is too recent to be fully understood, we decided to rely on a case-based reasoning approach [5, 6] as a pathway to use current practices to predict land use change.

A first prototype is under development, based on jCOLIBRI [7].

---

[1] Biomass energy corresponds to organic matter, essentially from agricultural and forest products (e.g. sugar beet, wood), co-product (e.g. wheat straw) and wastes (e.g. liquid manure).

2 Research plan

Case description

In our work, the problem part of the case corresponds to the features of one land system, i.e. i) to the farmer's attributes (age, education, main activity, etc.), ii) to the cropping plan of the farm (utilized agricultural area, land under permanent grass area, etc.) and iii) to the spatial and biophysical farmland features (distance of plots from the farm-stead, soil classification, etc.). The solution part of the case corresponds to the miscanthus allocation in the farmland (i.e. miscanthus plot allocation and its area). The data used for this case description are based on a survey realized over 111 farmers. The data base is called database 1.

To understand the link between the problem part and its solution, we also did comprehensive interviews. Comprehensive interviews do not include leading questions unlike surveys; we used them in order to catch all driving factors influencing the allocation process (that we didn't know a priori). Data from those interviews are stored in database 2. They are used/useful to build the domain knowledge and to define the adaptation rules.

The case base

To build the prototype, we first realized a statistical approach. We selected 16 representative cases (among the 111 ones) to constitute the case base. The others cases were considered as target cases; we used their problem part to build the retrieval stage and their solution to validate the model.

The selection of representative cases is realized on database 1. We discretized it into two main classes: the producers of miscanthus (called "Misc") and the non-producers of micanthus (called "noMisc"). For both classes we then selected representative cases by the K-medoid method, that selects the most centrally located individuals within a class, using the PAM algorithm (Partitioning Around Medoids).

Choosing a similarity measure

According to the data, we choose an attribute based distance. The main difficulty is the selection of the attributes (and weight) to be used. To select those attributes from database 1, we used the probabilistic approach of the Bayesian network, which was processed over the case base and tested over the target cases. We then refined the attribute selection based on agronomic expertise until we obtained an efficient clustering.

For the CBR application, we used the jCOLIBRI framework. To retrieve cases, we used the nearest neighbor method and the attributes weighting method. The current system enables to select a source case similar to the target one, and to propose a solution as "Misc" or "noMisc". To infer more detailed allocation practices from the source case to the target case, we use adaptation rules.

Adaptation

To adapt the solution of a source case to the specific features of a target case, we rely on two types of knowledge: farmer's decision-making process from database 2 and spatial information on the farmland (e.g. distance calculations). At the present time, the adaptation method is not completed.

3 Progress

In a perspective work, we plan to perform the adaptation stage. We also plan to reduce step-by-step the problem attributes that are difficult to obtain by the final user of the model, in order to facilitate its application for stakeholders.

Used as a decision support tool for land planning, our model is expected to bring various lessons, like evaluating the level of complexity of the decision-making processes thanks to feedbacks with the revision stage and catching correlations between farms and miscanthus allocation. One possible way for the dissemination of those lessons could be the implementation of requests to obtain descriptive statistics (dissemination of the system itself is still under discussion). The criteria to evaluate the success of this CBR model to deal with complex domain and human decisions rely on its ability to predict miscanthus allocation for various kinds of farms, with a minimization of modifications.

As knowledge acquisition and modeling for retrieval and adaptation remain difficult steps in CBR, our work can bring an interesting framework for other domains of application, more particularly for applications dealing with preferences and with complex, ambiguous and uncertain knowledge.

References

1. European Commission. Renewable Energy Directive (Directive 2009/28/EC) of the European Parliament and of the Council, dated of the 23 April 2009.
2. Hellmann, F., Verburg, P. Spatially Explicit Modelling of Biofuels Crops in Europe. Biomass & Bioenergy, Vol. 35, p. 2411-2424, 2008.
3. Martin, L., Wohlfahrt, J., Le Ber, F., Benoît, M. Perennial biomass crops allocation: a French case study regarding miscanthus. L'Espace Géographique, 2012 (in press).
4. Rounsevell. M, Arneth. A. Representing human behavior and decisional processes in land system models as an integral component of the earth system. Global Environmental Change, Online publication date: 1-May-2011.
5. Aamodt, A., Plaza, E. Case-based reasoning - foundational issues, methodological variations, and system approaches. Ai Communications, Vol. 7, p39-59, 1994.
6 Leake, D. Assembling Latent Cases from the Web: A Challenge Problem for Cognitive CBR. ICCBR-2011 Workshop on Human-Centered and Cognitive approaches to CBR, Nineteenth International Conference on Case-Based Reasoing, ICCBR-11, Greenwich, UK, 2011. 7 pages.
7. Recio Garcia, J.A. jCOLIBRI: A multi-level platform for building and generating CBR systems, PhD. Thesis, Universidad Complutense de Madrid, Madrid, 2008.

Development of an Unsupervised Norm Generation Method for Multiagent Systems

Javier Morales Matamoros

Artificial Intelligence Research Institute (IIIA-CSIC)
Campus UAB, E-08193 Bellaterra, Catalonia (Spain)
jmorales@iiia.csic.es

1 Introduction

A Normative System is a Multi-Agent System where a set of norms regulates agents' behavior in order to improve the performance of the system. Usually, Normative Systems have an initial set of norms that have been established by an expert (i.e. a human). Nowadays, the development of an automatic method for generating norms is an open research problem. Key questions of this problem are: When to generate new norms? How to generate them? How to know if the generated sets of norms are correct?.

Research on norms in MAS is quite an active area. Campos et al. [3] have proposed norm adaptation methods to specific network scenarios; Savarimuthu et al. [9], Griffiths and Luck [4] work on norm emergence. Within this area, norm generation has been studied less frequently. Shoham and Tennenholtz [10] focus on norm synthesis by considering a state transition system: they explore the state-space enumeration and state it is NP-complete through a reduction to 3-SAT. Similarly, Hoek et al. [5] synthesize social laws as a model checking problem –again NP-Complete– that requires a complete action-based alternative transition system representation.

The aim of this PhD is to develop an automatic method for norm generation and evaluation in Multi-Agent Systems in an unsupervised manner (without the help of an expert). The system must have different, dependent, and maybe conflicting goals. A regulatory authority will generate norms and experimentally evaluate them in a continuous manner with respect to system goals. The regulatory authority will also activate/deactivate norms in order to maintain a set of norms that improves the performance of the system. The developed approach must accomplish the following requirements:

1. To generate norms in an unsupervised manner at run time.
2. To evaluate and adapt the set of norms with respect to system goals.
3. To discard ineffective and unnecessary norms, maintaining a set of norms that accomplishes system goals and avoids conflicting states.

We approach this regulation generation problem by learning from the experience of MAS on-going activities. As a learning technique to use, we propose a variation of classical Case-Based Reasoning (CBR) [1]. Our proposal is an unsupervised CBR with an initially empty case base that evaluates generated solutions in base of the experience of a simulation. Hence, our variation of CBR does not require and expert to evaluate the performance of solutions. Boella and van der Torre [2] use a similar approach by using a Unsupervised CBR system to solve new situations by learning from experience in a checkers game scenario.

In our proposed method, a regulatory authority is continuously observing the scenario where agents interact. Then, whenever it perceives a new conflicting situation, it sends its description to our Unsupervised CBR system, which generates a new solution by using previous similar cases. Afterwards, generated case solutions are translated into norms that can be understood by agents, that have a local point of view of the system. Whenever norms are applicable, agents decide whether to *apply* or *violate* them. Additionally, the regulatory authority continuously observes the outcome of norm applications and violations. This can be regarded as a feedback that evaluates norms in terms of their *effectiveness* and *necessity* according to system goals. Our approach generates norms at runtime. This has the additional advantage of being able to regulate situations that may not be foreseeable at design-time. CBR allows the application to a wide range of domains, in particular to those where similar social situations require similar regulations (i.e., the continuity solution assumption).

2 Progress

Since November 2010, we have achieved the following results:
Development of a first version of the norm generation method. Our method generates norms as traffic regulatins using an unsupervised approach of Case-Based Reasoning. Generated norms are evaluated by taking into account the results of their applications and violations on the system.
Development of a simulator to test our norm generation approach. The used scenario is a traffic intersection. Agents are cars that cross the intersection, and conflicting situations are collisions between cars and traffic jams. We have obtained promising results in the initial experiments of our method. Performed experiments show that our method is capable to generate sets of norms that eradicate collisions and traffic jams when goals are non-conflicting. With conflicting goals, our approach searches for a trade-off between system goals.
Publication of a paper [7] in the International Joint Conference on Artificial Intelligence (IJCAI2012).
Publication of a demonstration video (Best video award) in the International Joint Conference on Artificial Intelligence (IJCAI2012). Link: `http://ijcai-11.iiia.csic.es/files/videotrack/morales.ogg`
Publication of a paper [8] in the Conference of the Spanish Association for the Artificial Intelligence (CAEPIA2012).

3 Plan

The results we plan to achieve in the next year are:

– Improvement of the *Retrieve* phase. To the present moment, we have defined an initial version of case description as a list of attributes with nominal values. The intention is to improve the definition of cases by (1) including just the most relevant information making cases to be more local, and (2) representing the information of the case description in a more efficient manner. Our similarity function computes the similarity of two case descriptions by counting the number of attributes of each case with the same value. When case descriptions are improved, the similarity function will be improved.
– Improvement of the *Reuse* phase. Once a case has been retrieved, our reuse function adapts its best solution to the current case. In this initial version case descriptions are very simple, so the reuse function is quite simple too.
– Improvement of the *Revise* phase. Our *Revise* phase is performed by evaluating generated solutions (and their associated norms) in base of their outcomes in a scenario along time. Thus, our system uses experiences of a MAS to compute the utility of case solutions and perform the *Revise* phase. In the current version, the evaluation of solutions for one single goal is working well and inefficient solutions are discarded from the system. The intention is to also include the goal of having fluid traffic and make the algorithm to converge with several goals.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1), 39–59 (1994)
2. Boella, G., van der Torre, L.: Regulative and constitutive norms in normative multiagent systems. Proceedings of KR'04 pp. 255–265 (2004)
3. Campos, J., López-Sánchez, M., Esteva, M.: Multi-Agent System adaptation in a Peer-to-Peer scenario. In: ACM Symposium on Applied Computing - Agreement Technologies Track. pp. 735–739 (2009)
4. Griffiths, N., Luck, M.: Norm Emergence in Tag-Based Cooperation. In: 9th International Workshop on Coordination, Organization, Institutions and Norms in Multi-Agent Systems. 79-86 (2010)
5. van der Hoek, W., Roberts, M., Wooldridge, M.: Social laws in alternating time: Effectiveness, feasibility, and synthesis. Synthese 1, 156 (2007)
6. Kota, R., Gibbins, N., Jennings, N.: Decentralised structural adaptation in agent organisations. pp. 54–71. AAMAS Workshop Organised Adaptation in MAS (2008)
7. Morales, J., López-Sánchez, M., Esteva, M.: Using Experience to Generate New Regulations. In: Proceedings of the 22th International Joint Conference on Artificial Intelligence (IJCAI). pp. 307–312. AAAI Press, USA (2011)
8. Morales, J., López-Sánchez, M., Esteva, M.: Evaluation of an automated mechanism for generating new regulations. In: Proceedings of the 14th international conference on Advances in artificial intelligence: spanish association for artificial intelligence. pp. 12–21. CAEPIA'11, Springer-Verlag, Berlin, Heidelberg (2011)
9. Savarimuthu, B., Cranefield, S., Purvis, M., Purvis, M.: Role model based mechanism for norm emergence in artificial agent societies. Lecture Notes in Computer Science 4870, 203–217 (2008)
10. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. Journal of Artificial Intelligence 73(1-2), 231–252 (February 1995)

Investigation of Case-Based Methods to Cross Domain Sentiment Classification

Bruno Ohana

Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland
bruno.ohana@student.dit.ie

1 Introduction

Our research is in the field of sentiment analysis, in particular we explore the task of *document sentiment classification*: for a given piece of opinionated text such as a product review, one can ask whether its overall sentiment can be considered generally positive or negative, favorable or unfavorable. Supervised learning methods for sentiment classification have been extensively studied in the past decade having achieved considerable success [1–3]. One downside of such methods however is the requirement for labelled in-domain data for training, where a domain can be seen as the general area where the document's opinions are concerned with: films, books, holiday destinations, etc. Considering the wide range of potential domains where sentiment classification can be applied, the cost of compiling data sets for each one becomes prohibitively expensive. Moreover, it is known that supervised learning models are strongly associated with the domain used in training, and not easily reusable on a different domain [4]. This has encouraged research on more robust cross domain approaches that minimize the requirement for in-domain training data.

One class of methods that avoid the need for in-domain data amounts to evaluating documents with the assistance of a pre-existing knowledge source. For instance, with the help of a *sentiment lexicon* - a language resource that associates words with numerical values indicating known sentiment information - one could find sentiment contained on words such as *excellent*, *good* or *terrible* and use these to infer overall document sentiment. A sentiment classifier could then scan a document for terms known to the lexicon and produce a prediction based on the overall numerical scores based on terms found to be positive or negative, taking into account factors such as the position of words within the document, part of speech and whether or not the word has been negated by a negating expression [5–7]. With a sentiment lexicon, a classifier can make predictions by using only pre-existing knowledge sources, and thus not requiring any training data.

1.1 Improving Lexicon-Based Methods with CBR

Lexicon-based methods have the advantage of requiring no training data and are good candidates for cross domain sentiment classification. There are however many sentiment lexicons available, with varying degrees of term coverage and accuracy. Before applying them to a classification problem, lexicons need to be determined and fixed. In addition, the technique by which the lexicon will be applied is also subject to a combination of execution parameters that determine its behavior: how are numeric scores assigned to terms from the lexicon, what part-of-speech to use. These need to be determined a-priori, which may lead to sub optimal results where better suited combinations exist for a yet unseen documents.

Instead, if we avail of out-of-domain documents, a more flexible method would be to determine, out of the possible choices available, which ones obtain good predictions for specific *types* of documents, and use this information to build a case base of past predictions for later reuse. Here, type relates to a document's general properties as defined by metrics such as overall size, sentence size, average word length, etc. A case-based approach would also be extensible: it allows for more training documents, new lexicons and new techniques to be incorporated to the case base as they become available.

2 Approach and Methodology

Our main research question is assessing whether we can improve the performance of lexicon-based sentiment classifiers by making use of a case base of out-of-domain documents to determine the most suitable lexicon and combination of classifier parameters to use on an unseen document. With this case base a sentiment classifier can make predictions by retrieving candidate cases according to their similarity to an unseen document. These are then ranked, yielding the preferred choices of lexicon and parameters to be used by the classifier.

2.1 Case-Based Framework for Sentiment Classification

Our approach is to build a framework for cross-domain sentiment classification assisted by a case base, having opinionated text documents as input and composed of the following key components:

1. A *Pre-Processing step* that annotates a plain text document with additional information such as part-of-speech and negated terms, for later use by the sentiment classifier.
2. A *Case Base* built from out-of-domain opinionated documents. For our experiments we will collect user reviews from different domains.
3. *Sentiment Lexicons*: We will use sentiment lexicons available in the literature, with the goal of obtaining a diverse set of lexicons based on different data sources and build methodologies.
4. A *Lexicon-Based Classifier* that applies a sentiment lexicon to a (now annotated) input text and issues a prediction on the document's overall sentiment.

Our research approach will be to collect the necessary data sets and sentiment lexicons and iteratively extend the framework based on experiment results. In particular, for the case base component we wish to investigate: a) good features for encoding types of documents in the case base, b) algorithms for populating the case base from out-of-domain data, c) retrieval and ranking methods for applying past cases to new instances, and d) techniques for growing and renewing the case base when new training data becomes available.

3 Progress Report

Our project has started in 2010, and we have since completed the following milestones:

Case-Based Sentiment Classification Experiment - Using data sets from 6 different domains, we have built an experiment that performs sentiment classification by making use of a case-base of out-of-domain data to retrieve the sentiment lexicons used on similar cases and make a prediction on a unseen document. Our results are being published as a conference paper in the *ICCBR 2012* conference.

Lexicon Performance Across Domains - We have performed an experiment on lexicon-based sentiment classification on multiple domains, which illustrated how the classification performance of a lexicon varies as we change domains. This finding has encouraged us to investigate methods for combining classifier results from different lexicons into a final prediction, yielding better performance. This work has been published as a workshop paper in the *2011 AINA (Advanced Internet and Networking Applications) Conference* [6].

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of 2002 EMNLP, ACL (2002) 79–86
2. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th WWW (2003), ACM (2003) 528
3. Abbasi, A., Chen, H., Salem, A.: Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems (TOIS) **26**(3) (2008) 12
4. Aue, A., Gamon, M.: Customizing sentiment classifiers to new domains: a case study. In: Procs of the RANLP, 2005. (2005)
5. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifters. Computational Intelligence **22**(2) (2006) 110–125
6. Ohana, B., Tierney, B., Delany, S.J.: Domain Independent Sentiment Classification with Many Lexicons. In: Mining and the Web Workshop - AINA 2011, IEEE (2011) 632–637
7. Taboada, M., Voll, K., Brooke, J.: Extracting sentiment as a function of discourse structure and topicality. In: Technical Report TR 2008-20, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada (2008)

Group Recommender Systems Enhanced by Social Elements

Lara Quijano-Sánchez

School of Computing. Complutense University of Madrid, Spain
lara.quijano@fdi.ucm.es

Abstract. My research lines consist of improving current group recommendation techniques by introducing two novel factors: the personality of each individual and trust between group members. In this way I can simulate in a more realistic way the argumentation process followed by groups of people when agreeing on a common activity.

1 Thesis Proposal

In the last few years, the number of recommender systems that focus on making recommendations for groups of people has increased. The main reason is due to the wide range of products like restaurants, movies, etc. that exist. This amount of information makes the task of discovering new things overwhelming and difficult. Besides, it is a fact that when we face a situation where people's interests do not exactly match *conflict situations* appear. Therefore, the task of making group recommendations is quite challenging as it has to present a set of interesting products not only to a single person but to a group of people whose concerns are not always compatible. Most of the previous works in this area consider the preferences of every member of the group with the same degree of importance and try to satisfy the preferences of every individual. However, groups of people can have very different characteristics, like size, relationships among their members and can be made of people with similar or antagonistic personal preferences. My thesis approach determines that the general satisfaction of the group is not always the aggregation of the satisfaction of its members.

My proposal consists of trying to improve current group recommendation techniques by including an analysis of social elements: people's personality and the trust they have in other people's opinions [1]. This strategy is enabled by my architecture ARISE[1]. The main advantage of ARISE is that it allows me to recreate more accurately people's behavior in argumentation processes.

There is an agreement about the need to adapt the recommendation process to groups composition [2, 3]. Recent works have focused their studies on analyzing the effectiveness of group recommendations according to different aspects, such as group size and inner group similarity [4], or on studying different weighting models to combine the preferences of group members according to their role within the group or their activity [5]. Additionally, it is also known that user's

[1] ARISE stands for Architecture for Recommenders Including Social Elements

preferences can be affected by other members of the group [6]. However, most of the aggregation strategies employed in previous works combine user's preferences without taking into account either the relationships among group members or the relevancy of each member's preferences.

I observed that there was a need to modify those existing strategies that consider each user of the group as equal to the others. So I focused my line of work on reflecting the individual aspects of users in groups and how they interact with each other. To do so, I introduce two novel factors [7]. The first one is the personality of each individual, we can easily see the necessity of applying this factor. For example, when a group of 2 or more friends choose a movie, there are some members that are only happy if they impose their opinion, whereas other individuals do not care in letting other people decide. To reflect this behavior I use a personality test to obtain different user profiles that interact in different ways when joining a decision making process.

I introduce a second social factor regarding the structure of the group. It consists on a novel technique of making recommendations for groups of people connected through social networks. Current research has pointed out that people tend to rely more on recommendations from people they trust than on recommendations based on anonymous ratings [8]. I have studied the social factors that affect users in order to calculate the trust between them. Some of the factors that I analyze are: Distance in the social network, number of common friends, intensity of the relationship, duration, status and pictures.

Summing up, my group recommendation method is based on preference aggregation approaches. These approaches [9] aggregate user's individual ratings in order to satisfy the largest number of group members. Based on this knowledge, the basic building block of my group recommender is an individual recommender that computes the preferences for each given user. However, these individual recommendations are modified with the personality and trust factors as my social recommendation method proposes.

The other main contribution of my thesis is the instantiation of my model in the social network Facebook. There are several reasons for this choice. First, Facebook is used by users to create events and invite their friends to join an activity, so my system can help them in the organization of such events. Second, user's activity in social networks can be tracked to obtain information about her trust with other users. And finally, it is a perfect environment to obtain user's personality, the last factor required by my model. The result of this is my application *HappyMovie* [10], which is based on ARISE and provides a group recommendation for a group of people who wish to go together to the cinema.

2 State of Progress

I have evaluated my method in the movie recommendation domain, where I have performed experiments with groups of real users and later repeated them with synthetic data. I use synthetic data in order to explore extreme cases that could, but rarely appear appear in reality and therefore cannot be easily tested. In my experiments, I firstly fulfill the task of obtaining the data required by my method: I ask users to answer to personality and preferences tests. I do not need trust tests as *HappyMovie* gets that information automatically from Facebook profiles (this is one of the main advantages of having the application embebed in a social network, that I can avoid exhausting users with a lot of questionnaires). I have validated my group recommendation proposal by comparing the results obtained by a recommender that follows my approach with the results obtained after asking users to group themselves and simulate that they are going to the cinema. I have also compared the results obtained by my social recommender with the ones provided by a recommender that does not use social factors. The conclusion of my experiments has been that by including social factors I do improve current group recommendation techniques. However it was very difficult to find willing users to answer all the required tests. I hope that by publishing my application in Facebook I would be able to obtain more users, more feedback and more interaction with users which I need in order to continue my Ph.D. investigation. These advantages could allow me to investigate sequences of recommendations over time and consider the evolution of the user's satisfaction or introduce explanations of recommendations and ask users whether they prefer them or not.

References

1. Quijano-Sánchez, L., Recio-García, J., Díaz-Agudo, B., Jiménez-Díaz, G.: Social factors in group recommender systems. In: ACM-TIST, TIST-2011-01-0013, in press (2011)
2. Jameson, A., Smyth, B.: Recommendation to groups. In: The Adaptive Web, Methods and Strategies of Web Personalization. Volume 4321 of Lecture Notes in Computer Science., Springer (2007) 596–627
3. Salamó, M., McCarthy, K., Smyth, B.: Generating recommendations for consensus negotiation in group personalization services. Personal and Ubiquitous Computing **16**(5) (2012) 597–610
4. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the fourth ACM conference on Recommender systems. RecSys '10, New York, NY, USA, ACM (2010) 119–126
5. Berkovsky, S., Freyne, J.: Group-based recipe recommendations: analysis of data aggregation strategies. In: Proceedings of the fourth ACM conference on Recommender systems. RecSys '10, New York, NY, USA, ACM (2010) 111–118
6. Chen, Y.L., Cheng, L.C., Chuang, C.N.: A group recommendation system with consideration of interactions among group members. Expert Syst. Appl. **34**(3) (2008)
7. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Personality and social trust in group recommendations. In: ICTAI'10., IEEE Computing Society (2010)
8. Sinha, R.R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001. (2001)
9. Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. User Modeling and User-Adapted Interaction **16**(3-4) (2006) 281–319
10. Quijano-Sánchez, L., Recio-García, J.A., Díaz-Agudo, B.: Happymovie: A facebook application for recommending movies to groups. In: ICTAI'11. (2011)

Case-Based Reasoning in Hydrometallurgy: Selection of Pretreatment Method for Refractory Gold Ore

Lotta Rintala

Department of Materials Science and Engineering
Aalto University, Finland
lotta.rintala@aalto.fi

1 Introduction

Hydrometallurgy is a field of science, which studies the use of aqueous chemistry for the recovery of metals from ores, concentrates, and recycled or residual materials. Typical hydrometallurgical process chain is illustrated in Figure 1.

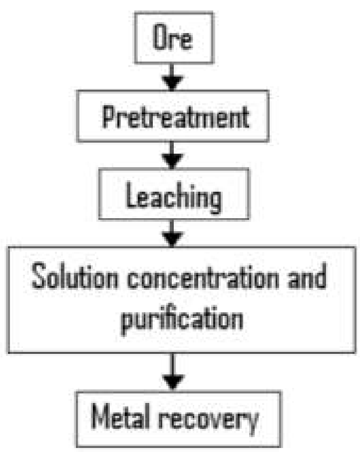


Fig. 1. The stages of the hydrometallurgical process chain.

At least hundred years cyanide leaching has been used as gold extraction technique. When it is a question of refractory gold ores and concentrates simple cyanidation cannot be effectively used [1]. In such ores, gold is encapsulated inside of a host mineral for example, in iron sulfide minerals (pyrite and arsenopyrite). The host mineral must be broken using pretreatment processes to liberate the gold, before cyanide leaching. These processes are for example oxidizing processes, such as roasting, pressure oxidation or bacterial leaching.

The selection of a pretreatment process for the new ore deposit depends on many factors, such as the level of the refractoriness of the gold ore and mineralogical characteristics of the ore. Typically process design relies on laboratory tests, which is time consuming and expensive. Today there is already available a large amount of published experiments about the gold extraction and the amount is increasing continuously. If an expert could manage existing information more efficiently and compare existing information more systematically and objectively, the need for experimental work would reduce.

2 Research Plan and Related Research

The objective of the research is to create a case-based reasoning application for the selection of the pretreatment process for low grade gold ore utilizing published knowledge. The four stages in the case-based reasoning method are: retrieving the closest case from the database, combining the retrieved case with the new case through reuse into a solved case, revising suggested solution and retaining the tested or repaired case (the learned case) into the database [2]. The aim is that planned application retrieves the most similar ores and their pretreatment processes and applies the knowledge of their concept for designing pretreatment for new refractory gold ore (the new case). Source for cases are publications about the hydrometallurgical studies of gold extraction, for example scientific papers and industrial reports.

The plan for research was to:

1. Study what kind of decision support method would help in the selection between hydrometallurgical processes.
2. Study the suitability of the selected decision support method for the problem domain further.
3. Study the quality of published data of the pretreatment of refractory gold ores.

The plan for research is to:

4. Build a CBR application. The following issues needs to be considered:
   - the representation of the cases and appropriate organization and indexing of the case memory for effective retrieval and reuse
   - how to revise suggested solution and retain a tested or repaired case into the database

A somewhat similar attempt, Intelligold, was built up in the 1990´s [3-4]. It was an expert system for project development teams to use at the preliminary evaluation and conceptual project stages. Information and knowledge from geology/mineralogy, processing and economics were organized and recommendations on process options and estimated costs and revenue were given. However, this application seems to have vanished as no publications after 2000 has been found.

3 Progress

1. Six different decision support and optimization methods were compared and based on the method descriptions case-based reasoning was selected to study further [5].

2. CBR method seems to be suitable to assist the selection of hydrometallurgical unit process alternatives [6].

3. The quality of published data of the pretreatment of refractory gold ores was investigated and a preliminary database was built. Published data can be used as source for cases in CBR application. Nevertheless, it was found out that scientific papers and industrial reports lack more data than expected [7]. Especially details about raw material descriptions were often missing.

Acknowledgements

This work has been done in the LOWGRADE project of the ELEMET research program funded by FIMECC Oy. The financial support of TEKES and Outotec Oyj is gratefully acknowledged.

References

1. Marsden, J.O., House, C. I. The Chemistry of Gold Extraction. Society for Mining, Metallurgy and Exploration, Inc. Littleton, Colorado. p. 651. (2006)
2. Aamodt, A., Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Communications. Vol. 7: 1, pp. 39-59. (1994)
3. Torres, V.M., Chaves, A.P, Meech, J.A. IntelliGold – A Fuzzy Expert System for Gold Plant Process Design. 18th International Conference of the North American Fuzzy Information Processing Society – NAFIPS. pp. 899-904. ed. Davé, R.N., Sudkamp, T. New York, U.S.A. (1999)
4. Torres, V.M., Chaves, A.P, Meech, J.A. IntelliGold - An Expert System for Gold Plant Process Design. Second International Conference on Intelligent Processing and Manufacturing of Materials IPMM '99. pp. 309-316. ed. Meech, J.A., Veiga, M.M. Jul 10-15 1999, Honolulu, Hawaii. (1999)
5. Rintala, L., Lillkung, K., Aromaa, J., The use of decision and optimization methods in selection of hydrometallurgical unit process alternatives. Physiochemical Problems of Mineral Processing, vol. 46. pp. 229-242. (2011)
6. Rintala, L., Aromaa, J., Forsén, O. The Use of Decision Methods in the Selection of Leaching Alternatives. European Metallurgical Conference 2011, Vol. 5. pp. 1659-1672. Düsseldorf, Germany. (2011)
7. Rintala, L., Aromaa, J., Forsén, O. Case-based reasoning assisted unit process selection. Kammel's Quo Vadis Hydrometallurgy IV - International Conference. June 4-7 2012. Herlany, Košice, Slovakia. (2012)

Role of Knowledge Extraction for Text Retrieval

Sadiq Sani

School of Computing,
Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
s.a.sani@rgu.ac.uk

1 Introduction

Indexing of text documents is important for many different tasks such as Textual Case-based Reasoning, Information Retrieval, Question Answering and Text Categorisation. Key to all these tasks is the need to represent text in a way that facilitates subsequent retrieval. A commonly used approach is the Bag-Of-Words (BOW) representation, where terms are used as features to represent text documents. This simple approach suffers two major limitations.

Firstly, the BOW approach relies on exact lexical matching of features. However, the same idea can be expressed in many different ways. For example the statements "he is unemployed", "he is out of work", "he doesn't have a job" and "he is jobless" are all expressions of the state of being without employment although the words used in each example are different. Also, the same expression could have multiple interpretations. For example the word "Labour" could mean manual work, the Labour Party or child birth. Hence a purely lexical approach to text representation and retrieval is not sufficient for capturing underlying meaning of text.

Secondly, the unit of representation in the BOW model is a word which is a semantically poor lexical unit. In contrast, certain words and word sequences have special semantics because they represent real-word entities or important domain concepts e.g. location, time/date, monetary value or a medical condition. Thus an indexing approach that is able to reason with such semantic concepts will allow for retrieval of documents based on shared meaning with the query.

2 Research Aim

The aim of this research is to study the role of knowledge in text retrieval and to develop automated knowledge acquisition techniques that address identified limitations in the state-of-the-art. This work will involve an investigation of the contribution of similarity knowledge to retrieval performance and also the development of techniques for combining similarity knowledge from different knowledge sources (e.g. corpus co-occurrence, WordNet [2], Wikipedia and the Web). We also seek to develop IE techniques for automated acquisition of shallow semantics from text. Our particular focus will be on extracting events and semantic entities/concepts with a view to defining a semantic indexing vocabulary for text.

3 Research Plan

Our research plan is presented below:

– Conduct study of the role of similarity knowledge from different sources (e.g. corpus co-occurrence, WordNet, Wikipedia and the Web) by conducting comparative analysis of the different sources on text retrieval tasks.
  The first step of this research is to develop a framework that allows for studying the role of similarity knowledge in text retrieval. Particularly, answers to the following questions will be sought:
  - How useful are different knowledge sources (corpora, WordNet, Wikipedia, Web) for providing background knowledge and what are their limitations?
  - Is there a relationship between domain and source of similarity knowledge that makes some sources better suited in certain situations?
  - How can similarity knowledge from different knowledge sources be combined such that they complement one another?
– Develop algorithms for acquiring advanced indexing vocabulary
  Most work done on utilising Information Extraction (IE) for text indexing rely heavily on domain-specific, manually-built rules and lexicons. In addition, such systems are highly reliant on knowledge from domain experts. Thus, we seek to develop domain independent IE algorithms for the automatic acquisition of indexing vocabulary from text. Our goal is to avoid the knowledge engineering effort associated with building and maintaining domain-specific IE techniques. Instead, we intend to utilise publicly-available, domain-independent knowledge resources such as WordNet, Wikipedia as well as Ontologies like Yago to automatically extract events, entities and concepts for text representation.
– Build a retrieval framework to combine similarity knowledge with advanced indexing vocabularies and apply framework on a number of text retrieval tasks.

4 Progress

Progress made to date is presented below.

– **Term similarity framework:**
  In order to allow for the study of the impact of similarity knowledge for retrieval, we use the following generic retrieval framework that supports the incorporation of term similarity knowledge from any knowledge source. The framework highlights the significance of introducing a measure of term importance in order to reduce the noise from spurious term relationships. The framework is formally presented in equation 1.

$$D' = T \times W \times D \tag{1}$$

  Where D is a standard term-document matrix, T is a term similarity matrix that represents pairwise similarities between all terms in the feature space, and W is a term weights matrix. The separation of the term similarity matrix (T) from the document (D) and term weight (W) representation matrices, allows us to incorporate term similarity knowledge from one or more knowledge sources. The framework was evaluated on five text classification tasks using approaches that extract term similarity knowledge from corpus co-occurrence [3]. We are currently extending this study with term similarity knowledge extracted from WordNet using standard WordNet similarity metrics [1] and also from Wikipedia using the Wikirelate! approach [5]. Our goal is to devise an aggregation strategy to combine evidence of similarity from multiple sources.

- **Event Extraction for Reasoning with Text:**
  This work partially addresses our objective of utilising IE for acquiring an indexing vocabulary. We have developed an algorithm called RUBEE (RUle Based Event Extraction) that identifies automatically extracts atomic events from incident reports [4]. Incident reports typically contain descriptions of circumstances that led to the incident, the incident itself, any injuries that may have resulted from the incident as well as any remedial action taken. Such information is typically centered around events e.g. "gas was **leaking** from a pipe", "this resulted in a **fire**", "the operator was severely **burned**". Correctly identifying these types of events will enable the comparison and retrieval of documents based on shallow semantics rather than syntax alone. Also, a shallow semantic approach does not require huge amounts of domain knowledge and manual effort required as with deep semantic approaches.
- **Ontology-based Information Extraction:**
  While the semantic concepts in some domains are centered around events, in most other domains, semantic concepts are centered around semantic entities e.g. persons, products and diseases. Thus we are currently developing algorithms that utilise Web ontologies e.g. Yago [6] to aid the extraction of these concepts. Our algorithm uses regular expressions to identify a sequence of words that potentially represents a valid domain concept and uses Yago to verify whether or not the sequence represents a valid domain concept. Key to this approach is correctly identifying the nodes (entities) in Yago that represent important domain concepts.

References

1. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (Mar 2006)
2. Miller, G.A.: Wordnet: a lexical database for english. Communications of the Association for Computing Machinery 38(11), 39–41 (Nov 1995)
3. Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Term similarity and weighting framework for text representation. In: Proceedings of the 19th International Conference on Case-Based Reasoning, ICCBR (2011)
4. Sani, S., Wiratunga, N., Massie, S., Lothian, R.: Event extraction for reasoning with text. In: To appear in Proc. of the 20th Int. Conference on Case-Based Reasoning, ICCBR (2012)
5. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: Proceedings of the 21st national conference on Artificial intelligence - Volume 2. pp. 1419–1424. AAAI Press (2006)
6. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proc. of the 16th international conference on World Wide Web. pp. 697–706. WWW '07, ACM (2007)

Creating Human-Level AI in Strategy Games using Hybrid CBR/RL

Stefan Wender

Department of Computer Science, The University of Auckland
Auckland, New Zealand
s.wender@cs.auckland.ac.nz

1 Problem Outline

The research problem to be addressed in this dissertation is the creation of a hybrid RL/CBR technique that use case based generalization to address the problem of combat simulation in a commercial real-time strategy game. The chosen testbed is the video game StarCraft, an imperfect information game with high complexity in combat situations. The game exhibits all the characteristics identified by [1] as interesting AI problem areas in RTS environments. The high complexity results from three different competing factions in the game that have altogether about 50 completely different types of combat units, each with their own strengths and weaknesses and different sets of actions.
As the aim is to work with a non-continuous model, both states and actions in this environment have to be discretized. The extraction of expert knowledge from recorded games will yield a very large case base. The principles of RTS games mean that there is a built-in requirement in terms of execution time, i.e. the algorithm in its eventual optimal form will have to work in near-real time. This is further complicated by the size the case base which will have to be accessed and maintained in possibly novel ways if no existing approach that matches the requirements can be found.
The use of pure reinforcement learning algorithms will not be enough as the problem of the entire combat simulation in a commercial RTS game is vast. The problem of dimensionality means that sufficiently comprehensive state- and action spaces that describe the complexity of the domain would be extremely large. Solutions to this problem such as function approximation only partially solve it as they come with other drawbacks. Therefore, this problem requires new ways for addressing large and complex case bases.

2 Progress to Date

As a first step to create a hybrid RL/CBR agent an evaluation of the suitability of simple reinforcement learning algorithms to perform the task of micro-managing combat units in StarCraft was done [2]. The techniques applied are variations of the common Q-learning and Sarsa algorithms [3]. For both algorithms simple one-step versions as well as more sophisticated versions that use eligibility traces to offset the problem of delayed reward are used. The aim was the design of an agent that addresses the lowest level of combat in the game, involving only one unit that is controlled by the RL agent in a very simplified state- and action-space. The preliminary results show that the agent is capable to learn a near-optimal solution for the selected task. Depending on if the focus lies on maximizing the reward or on the speed of learning, among the evaluated algorithms one-step Q-learning and one-step Sarsa prove best at learning to manage combat units. Though it will have to be greatly extended to cater to the larger scale, this agent will form the base for the CBR/RL algorithm.

3 Planned Research

The eventual aim is to create an AI agent that is able to handle any possible combat situation in StarCraft successfully, independent of the involved numbers and types of units. Using its existing knowledge base, the agent should also be able to adapt quickly to new situations.

The first step to integrating the existing RL agent with a CBR approach [4] is the creation of a knowledge base that will be based on expert knowledge from human players. This expert knowledge will be extracted from the large amount of game traces, so-called "replays", that are available online. Each trace resembles an entire game played by expert players and contains numerous combat situations that will eventually be extracted [5] and classified as case or part of a case. Particular focus will also be on the definition of the model as states could possibly become very complex due to the diversity that is possible for combat situations in the chosen testbed. A gradual increase in complexity is planned, starting off with only a limited amount of attributes and eventually scaling up to the complete set of actions and units.

The next and most crucial step will be the creation of the actual hybrid RL/CBR algorithm that uses the case base. In order to address the complexity of the problem domain I am planning to use state-generalization that leads to a clustering of cases [6]. This will also be used to spread reward to the appropriate cases/state clusters in the case base and address the complexity akin to that in continuous state/action-spaces [7]. Most existing approaches use simple kNN retrieval methods. Unlike this I am also going to look at using other retrieval methods such as induction or knowledge guided induction [8] in order to handle the size of the case base. The definition of the model will play an important part here in deciding the suitability of our algorithm for different case retrieval methods.

Another task involved in the research that will be an ongoing process is the optimization of the parameters used for the underlying RL algorithm, possibly by using genetic algorithms. Should the evaluation lead to kNN being the most suitable method for case retrieval, the best similarity metric will have to be decided.

References

1. Buro, M.: Real-time strategy gaines: a new ai research challenge. In: Proceedings of the 18th international joint conference on Artificial intelligence, Morgan Kaufmann Publishers Inc. (2003) 1534–1535
2. Wender, S., Watson, I.: Applying reinforcement learning to small scale combat in the real-time strategy game starcraft:broodwar. In: IEEE Conference on Computational Intelligence and Games 2012 (to appear). (2012)
3. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
4. Watson, I., Marir, F.: Case-based reasoning: A review. Knowledge Engineering Review **9**(4) (1994) 327–354
5. Synnaeve, G., Bessiere, P.: Special tactics: a bayesian approach to tactical decision-making. In: IEEE Conference on Computational Intelligence and Games 2012 (to appear). (2012)
6. Dilts, M., Muñoz-Avila, H.: Reducing the memory footprint of temporal difference learning over finitely many states by using case-based generalization. Case-Based Reasoning. Research and Development (2010) 81–95
7. Molineaux, M., Aha, D., Moore, P.: Learning continuous action models in a real-time strategy environment. In: Proceedings of the Twenty-First Annual Conference of the Florida Artificial Intelligence Research Society. (2008) 257–262
8. Wess, S., Althoff, K., Derwand, G.: Using k-d trees to improve the retrieval step in case-based reasoning. Topics in Case-Based Reasoning (1994) 167–181