Noticeably New: Case Reuse in Originality-Driven Tasks *

Belén Díaz-Agudo[1], Enric Plaza[2],
Juan A. Recio-García[1], and Josep-Lluís Arcos[2]

[1] Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Spain,
jareciog@fdi.ucm.es,belend@sip.ucm.es
[2] IIIA, Artificial Intelligence Research Institute,
CSIC, Spanish Council for Scientific Research,
Campus UAB, Bellaterra, Catalonia, Spain,
{enric,arcos}@iiia.csic.es

Abstract. "Similar problems have similar solutions" is a basic tenet of case-based inference. However this is not satisfied for CBR systems where the task is to achieve *original* solutions — i.e. solutions that, even for "old problems," are required to be noticeably different from previously known solutions. This paper analyzes the role of reuse in CBR systems in *originality driven tasks* (ODT), where a new solution has not only to be correct but noticeably different from the ones known in the case base. We perform an empirical study of transformational and generative reuse applied to an originality driven task, namely tale generation, and we analyze how search in the solution space and consistency maintenance are pivotal for ODT during the reuse process.

1 Introduction

A basic tenet of case-based inference is that similar problems have similar solutions. This is not only a useful way to explain Case Based Reasoning to laypeople but is the central core of so-called similarity-based inference in fuzzy logic. Based on this assumption developing a good CBR system basically has two requirements: (1) acquiring a good sample of cases, and (2) designing a predictive similarity measure (i.e. one that predicts a good solution when the cases are similar). Nevertheless, there are domains where the task is to achieve not only solutions but *new* solutions — i.e. solutions that, even for "old problems," are required to be noticeably different from previously known solutions. Domains like music composition and performance, story plotting and writing, or architecture design, require the solutions to be noticeably dissimilar from previously produced solutions, or at least from previous solutions from other authors. We will call these kind of tasks *originality-driven tasks*.

* Supported by the MID-CBR project (TIN2006-15140-C03-02)

Moreover, several CBR approaches have dealt with originality-driven tasks for innovative design or for "creative" problem solving (as we discuss in Section 6). Focusing on the role of the Reuse process, this paper aims to analyze the issues relevant for CBR systems when dealing with originality-driven tasks in general. We will study how different Reuse techniques effect different search processes in order to elucidate the main issues relevant for the construction of a noteworthy new solution. Specifically, we will consider two existing reuse techniques (a transformational reuse technique and a generative reuse technique), and we will apply them to the domain of folk tale generation to analyze these issues and provide some guidelines for future originality-driven reuse techniques.

The structure of this paper is as follows. In Section 2, we present a search based framework to study Reuse processes and we define novelty (or originality) from the notions of solution space similarity and plagiarism. Section 3 characterizes the two reuse techniques and analyzes them with respect to originality driven tasks. Section 4 describes tale generation as an originality driven task. Section 5 presents the results of some experiments with different reuse approaches. Following a review of the related work in Section 6, Section 7 summarizes the main conclusions and the lines of future work.

2 Search, Reuse and Plagiarism

First, we find it useful to distinguish between analytic and synthetic tasks. In analytical tasks finding a solution is selecting one element from a known and enumerable collection of solutions; examples are classification, identification or single diagnosis. Synthetic tasks, on the other hand, do not not provide in advance with a collection of solutions; synthetic tasks define a collection of *solution elements*, and a solution is *constructed* by a certain combination of some solution elements. In general, a solution can be seen as a graph, where solution elements are nodes and edges are the relationships holding among the solution elements. In some synthetic tasks, like planning, a solution is a special kind of graph, like a sequence or a partial order among actions (the solution elements of the planning task). Clearly, originality-driven tasks are synthetic tasks, and novel solutions can be found by new combinations of the solution elements.

Let us now consider the main differences between the "similar problems have similar solutions" scenario (SPSS, see Figure 1) and the "originality-driven tasks" scenario (ODT, see Figure 2). In the SPSS scenario of Figure 1 a new problem x_0 is compared in the problem space using a similarity measure with other problems in the case base. Moreover, let us view the case base as a repository of the mappings from problem space to solution space given by the known cases $CB = \{(x_i, s_i)\}$. Assuming x_1 is the most similar problem to x_0, case based inference yields s_1 as the solution of case (x_1, s_1). Now, the "similar problems have similar solutions" hypothesis basically states that we expect to find s_0 (the solution for x_0) in the neighborhood of s_1 (depicted as a circle around s_1). The Reuse process, in abstract terms, is the one that moves from solution s_1 to solution s_0 in the solution space; depending on the reuse technique, this

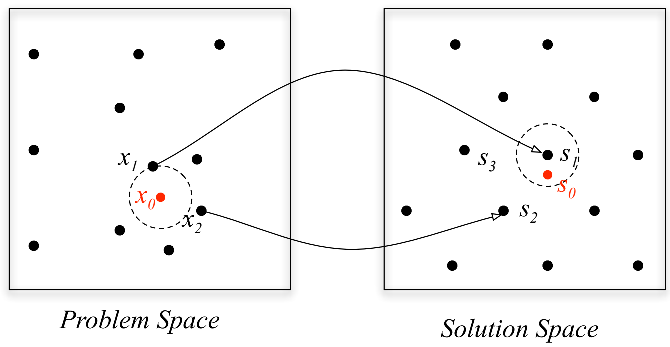


Fig. 1. Scenario 1: Similar problems have similar solutions in CBR.

"trajectory" can be seen in different ways, but we will consider that in general (as argued in [1]) it is some form of search process. However, the bottom line is that CBR systems have been designed with the underlying idea that a *short length trajectory* is desirable or even mandatory.

This assumption can not be satisfied, in general, for ODT using CBR. Figure 2 exemplifies this scenario where a solution to the new problem x_0 cannot be too close to the solutions of similar cases. Consider, for instance, that new problem x_0 is similar to case $C_1 = (x_1, s_1)$; an original solution to problem x_0 cannot be too close to s_1 — they have to be outside the grey circle in Figure 2 centered around s_1. Additionally, an original solution for x_0 must also not be too close to any other existing solutions. The Reuse process in ODT CBR systems has to build a trajectory such as that shown in Figure 2 from s_1 to s_0 — i.e. a trajectory that cannot be ensured to be short and that finds a consistent solution for x_0 in a relatively unpopulated region of the solution space. Therefore, we formulate the following hypothesis:

Hypothesis 1 *ODT CBR Reuse needs a similarity (or a distance) measure on the solution space* $\mathcal{S}$.

Most CBR systems do not require a definition of a similarity measure on the space of solutions. There are exceptions, but we are not claiming any innovation here. We simply state that for the ODT scenario, it makes sense to consider as indispensable the definition of similarity measures on the space of solutions.

There is no problem, in principle, to find solutions in relatively unpopulated region of the solution space: domains where ODT are applicable have large solution spaces since the combination of their solution elements into complex structures is huge. However, there are technical requirements that should addressed by Reuse techniques when abandoning the "short length trajectory" assumption: (1) the Reuse technique needs to search the solution space in a systematic (or even exhaustive) way, and (2) the Reuse technique should ensure the validity and consistency of the solutions

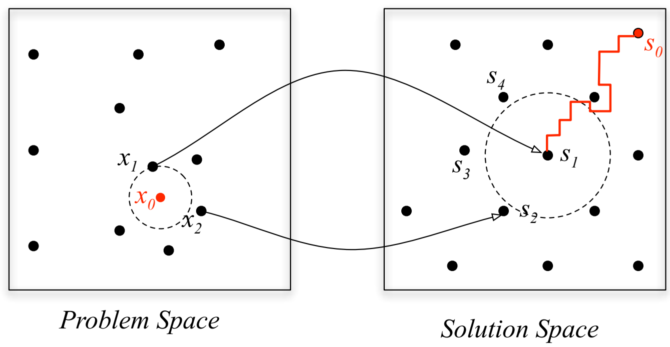


Fig. 2. Scenario 2: originality-driven tasks in CBR.

Assumption (1) is necessary to be able to reach unpopulated regions of the solution space in large Reuse trajectories. Assumption (2) is needed because in the SPSS scenario often the validity and coherence of solutions are not ensured or explicitly tested: the "short length trajectory" assumption implies that, since few changes are made, if the solution of the retrieved case is valid and consistent then the Reuse process most likely will produce a valid and consistent solution. If not, the Revise process is designed to check and/or repair the solution (usually with a human in the loop). Validity and coherence of solutions play a different role in the Reuse process for originality-driven tasks. Since Reuse will perform a large search process it cannot simply present thousands of configurations to be Revised by a human. Moreover, since the solution space to explore is huge, a Reuse process that is able to prune most or all invalid or inconsistent partial solutions will be more efficient in the exploration of the solution space. Therefore, we formulate the following hypothesis for CBR systems in originality-driven tasks:

Hypothesis 2 *ODT CBR Reuse needs knowledge to assess the internal coherence of solutions and partial solutions meaning that (a) either the Reuse process is able to ensure that it will only deal with consistent solutions and partial solutions, or (b) partial solutions (intermediate points in the Reuse trajectory) may have some inconsistencies but they are temporary, detectable, and remediable.*

Later, in Section 3, we will see how generative reuse and transformational reuse employ respectively approaches (a) and (b) to address validity and consistency of solutions for "long length trajectory" reuse.

Indeed, ensuring validity and consistency of solutions requires additional domain knowledge, but it is an empirical question whether "more knowledge" is a large or modest amount. Anyway, domains where originality-driven tasks are usually applied to already have a rather rich ontology, and the *solution elements* and their possible relationships have to be represented in some formalism. Although we do not intend to address this issue in general, we address later in the paper the role of domain knowledge for the domain of folk tale generation, and how it differs in the specific generative and transformational reuse techniques we use.

Finally, we will address the notion of plagiarism in the context of originality-driven tasks. Plagiarism is an argument made against the quality of something being original on the grounds that it is (very) similar to some preexisting body of work. Although definitions of plagiarism in music, literature or architecture may vary in how to measure or assess similarity, or which similarity threshold may legally sustain a plagiarism lawsuit, the core idea of "plagiarism" seems quite stable and transversal. This core idea allows us to define originality or novelty for ODT case-based reasoning:

Definition 1 (Originality). *Given a case base* $CB = \{(x_i, s_i)\}$*, a distance measure* Δ *over the solutions space* $\mathcal{S}$*, and a plagiarism threshold* γ*, a solution* s_0 *is original iff* $\forall (x_i, s_i) \in CB : \Delta(s_0, s_i) > \gamma$*.*

This approach based on the plagiarism/originality dualism offers a pragmatic framework to deal with the issues of novelty and innovation. Instead of proposing some debatable definitions of what is or not "original" (or "novel" or "innovative"), we propose to consider a solution *original* as long as no argument of plagiarism attacks that solution; similarly, if there are plausible plagiarism arguments against some solution, then that solution may be considered of "debatable originality." Another reason for this approach is that we wanted to avoid having "degrees of innovation", i.e. we do not intend to distinguish between something being "very novel" (or "very creative") vs. being not very novel. We think this kind of phrasing mixes together an assessment of quality and an assessment of dissimilarity from an existing body of work. Discussion in this paper of *originality* refers to the definition above and does not imply any assessment about the quality of solutions; for instance, in the domain of folk tale generation presented later we deal with their originality but not with the "tale quality", although a certain consistency of solutions is guaranteed.

3 Reuse Techniques

The purpose of this paper is not to design new Reuse techniques for originality-driven tasks (ODT) in CBR, but rather to analyze existing CBR Reuse techniques inside a ODT framework in order to determine how well adapted they are for these tasks and which possible shortcomings should be addressed to improve CBR in originality-driven tasks. For this purpose we selected two broadly different Reuse techniques, one based on transforming an existing solution into a new solution (Figure 3a) and another based on generating or constructing a new solution (Figure 3b).

Transformational Reuse –or Transformational Adaptation (TA)– is the most widely used approach to case reuse; Figure 3a shows a schema of this approach (where DK means domain knowledge and CK means case knowledge). Although this schema is not intended to cover all existing techniques, it is useful to pinpoint their main features. Typically, a new case is solved by retrieving the most similar case in memory and copying the solution (although some techniques may

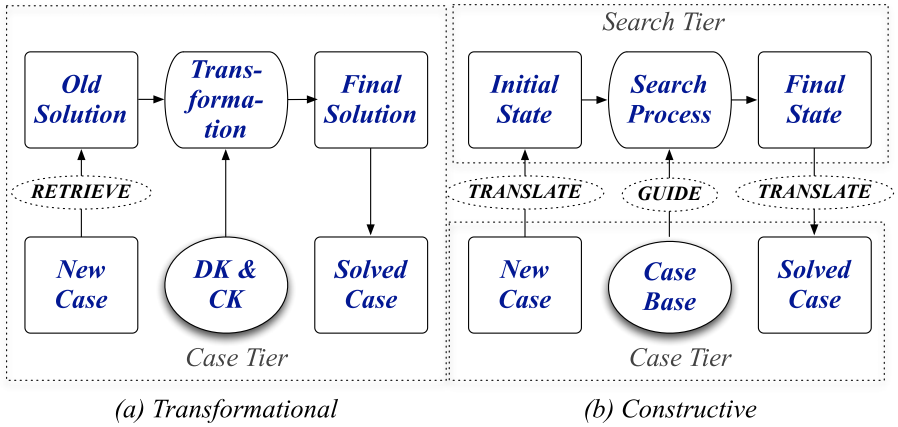


Fig. 3. Schemas of reuse processes based on (a) transforming an existing solution into a new solution, and (b) generating or constructing a new solution.

use solutions from multiple cases); then a transformational process using domain knowledge (DK) and/or case-derived knowledge (CK) modifies that copy (which we consider a form of search) until a final solution adequate for the current problem is found. In the experiments described in Section 5, we used a local search transformational reuse technique; basically, a node in the "working case" is substituted by finding another related node in a taxonomic hierarchy — e.g. a *sword* is a type of *weapon* in the folk tale generation domain, and may be substituted by another weapon like a *crossbow*. Moreover, Transformational Reuse is able to modify more than a single node: *deep substitution* allows to modify a whole subgraph in the solution — e.g. when substituting a character like the *evil wolf* by an *evil wizard* then the constituent aspects of the characters (role, sex, dwelling, physical appearance) are also substituted. Finally, consistency is maintained by the use of explicit *dependencies*; dependencies are used to detect nodes that need to be transformed after some nodes are substituted — e.g. the folk tales domain uses dependencies among actions to assure consistency, like *Release-from-captivity* depends-on *Kidnapping* (see Figure 4).

Generative or Constructive Reuse builds a new solution for the new case while using the case base as a resource for guiding the constructive process. Figure 3a shows the schema of *Constructive Adaptation* [1], a family of methods based on a heuristic search-based process —where the heuristic function guiding search is derived from a similarity measure between the query and the case base. Constructive Adaptation (CA) takes a problem case and translates it into an initial state in the *state space* (Figure 3b); i.e. transform a case representation into a state representation. Then a heuristic search process expands a search tree where each node represents a partial solution, until a final state (with a complete and valid solution) is found. Notice that final but non-valid states can be reached, but this simply means the search process will backtrack to expand other pending states.

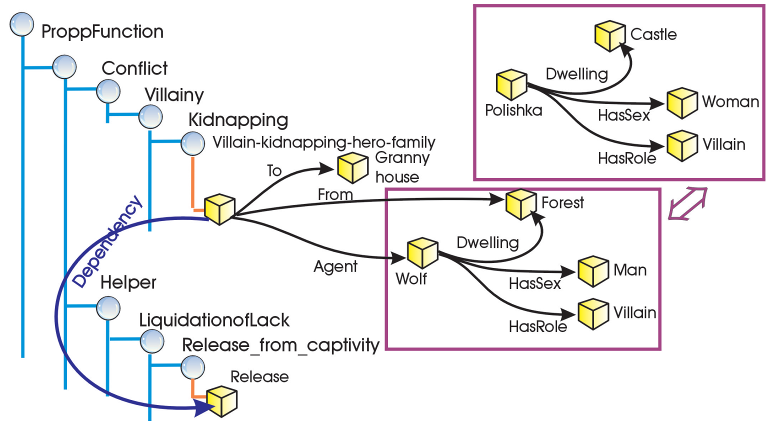


Fig. 4. Deep Substitution and Dependencies.

This process is guided by a heuristic based on comparing the similarity from *states* (represented in the *state space*) to cases (represented in the *space of cases*); the nodes with higher similarity are expanded first during the search process. The result is that CA adds one node to a partial solution as it moves from one state to the next; that is to say, it builds a solution by piecemeal copies of nodes from similar cases. Notice that there is neither retrieval nor "single case adaptation" here since the component nodes are incrementally copied from multiple cases in the case base, depending only on the similarity measure that works on the whole case base. To ensure consistency, however, CA requires that each component is described with *Before-formulae* and *After-formulae* [1]. *Before-formulae* specify what properties are required to be true in order for the component to be validly added to a solution, while *After-formulae* state what properties are true by the incorporation of this component in the solution. A consistent solution is one that satisfies all the *Before-formulae* required by its components, and a valid solution is one that satisfies the current problem.

Thus, the main difference between these techniques is that TA works in the *space of cases* while CA works both in the *state space* and the *space of cases*. Additionally, we are able now to characterize both Reuse techniques in our framework of Reuse as a search process.

Concerning TA, we characterize it as follows: (1) *eager reuse* (copies an old solution as the first step, and later discards parts of it by substituting them); (2) based on *case space search*; and (3) *single-focus reuse* (since all transformations are effected upon a single case solution; this is true even when using substitutes from multiple cases, since parts of these cases are always substituted against the structure of a single "working case" being transformed).

Concerning CA, we characterize it as follows: (1) *lazy reuse* (adds one component at a time to the solution); (2) based on an interplay between *state space search* and similarity on *case space*; (3) *multi-focus reuse* (since components added to a solution come in principle from multiple cases); and (4) an *exhaus-*

tive search approach that can provide solutions even when no similar cases (or no cases at all) are provided.

Finally, consistency is also approached in a different way in both reuse techniques. Transformational Reuse uses explicit dependencies in the space of cases, while Constructive Adaptation uses *Before-formulae* and *After-formulae* that are used in the state space. Both techniques make sense for knowledge-intensive CBR, and as we show in the next sections for folk tale generation, they both use a domain-specific ontology about folk tales. The knowledge required by both techniques for maintaining consistency is not large, and can be derived from an analysis of that ontology.

4 Tale Generation

Automatic construction of tales has always been a longed-for utopian dream in the entertainment industry [2–4]. The automatic generation of stories requires some *formal* representation of the story line (plot), a reasoning process to generate a tale from a given query, and the choices of some (textual) format for presenting the resulting plots. As a case study for the experiments, in this paper we present a CBR approach to the problem of obtaining a structured description of a tale plot from a given query. The problem of transforming the resulting plot into a textual rendition is out of the scope of this paper.

Previous work by the UCM group has shown that Ontologies and Description Logics are a very powerful combination as a resource for generating linguistically correct texts [5, 6]. The UCM group has formalized an ontology including the primitives to represent a plot structure based on Vladimir Propp's theory [7]. Propp's original goal was to derive a morphological method of classifying tales about magic, based on the arrangements of 31 primitive actions or "functions", resulting in the description of folk tales according to their constituent parts, the relationships between those parts, and the relations of those parts with the whole. Propp's work has been used as a basis for a good number of attempts to model computationally the construction of stories [8, 9].

The UCM group approach relies on Propp's main idea that folk tales are made up of components that change from one tale to another, and *actions* or *functions* that act as constants in the morphology of folk tales. What changes are the names and certain attributes of the characters, whereas their actions remain the same. For example, some Propp functions are: *Villainy, Departure, Acquisition of a Magical Agent, Guidance, Testing of the hero*, etc. The ontology (explained in [6]) includes various concepts that are relevant to tale generation and give semantic coherence and structure to the tales. Based on this formalization we previously proposed a CBR approach for storyline representation and adaptation [5]. That work described a process to retrieve one plot based on a user query specifying an initial setting for the story. Then a transformational reuse process modifies the retrieved plot according to the query.

The goal of this paper is studying the role of reuse in CBR systems in *Originality driven tasks*, like tale generation, where the underlying goal is creating a tale that is new and useful at the same time as maintaining narrative coherence. Although in the literature there are different definitions for concepts like *creativity*, *novelty* and *originality*, in this paper we characterize them using an *edit distance* measure[10].

Each case is a story plot that, according to Propp's structure, is formalized by its actions, and each action by its properties, like the participant characters and their roles (Donor, Hero, FalseHero, Prisoner, Villain), the place where the action takes place (City, Country, Dwelling), the involved objects, attributive elements or accessories (a ring, a horse). Each case is composed of a great number of interrelated individuals, i.e instances of concepts, from the ontology.

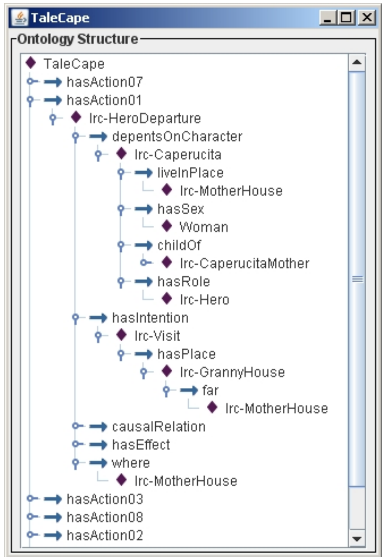
The basic components are the Propp's character functions that act as high level elements that coordinate the structure of discourse. There are some restrictions on the choice of functions that one can use in a given folk tale, given by implicit dependencies between functions: for instance, to be able to apply the *Interdiction Violated* function, the hero must have received an order (*Interdiction* function). There are many other examples, like the dependency between *Release-from-Captivity* and *Kidnapping*, or *Resurrection* and *Dead* functions.

Background domain knowledge required by the system is related with the respective information about characters, places and objects of our world. Domain knowledge is used to measure the semantical distance between similar cases or situations, and for maintaining an independent story plot structure from the simulated world. The domain knowledge of our application is the classic fairy tale world with magicians, witches, princesses, etc. The ontology is formalized in OWL and it includes about 230 concepts, 626 distinct individuals (246 appearing in the initial case base), and 94 properties. Each case representing a complete tale is typically composed of several interrelated actions. Each action refers to a Propp function, and gives answers to the *who* (character), *where* (place) and *what* (object) questions. We distinguish between *temporal* relations (before, after, during, starts-before, ends-before, etc.) and actions with *dependencies* (in which a change in one of them strongly affects the others). There are different types of dependencies like *place-dependency*,*character-dependency*, *object-dependency* and *propagation-dependency*. Dependencies are explicitly represented as relations that link the dependent elements in the ontology.

The initial case base in our system has 6 cases representing story plots for traditional fairy tales like "Fortune Teller", "Little Red Riding Hood", "Cinderella" and "Yakky Doodle". Each one of these cases is a complex structure where many individuals are interrelated. See Figure 5 (right) for a summary of the complexity and number of instances for each tale. The simpler one is "Cinderella" with 36 individuals including actions, characters, places and objects. The more complex is "Goldfish" with 77 individuals. Figure 5 (left) depicts the action structure of the "Little Red Riding Hood" story plot.

5 Experiments

The purpose of our experiments is to take a technique representative of transformational adaptation (TA) and another representative of constructive adaptation



	Title	Complexity level	Nº Instances
T1	Fortune teller	3	22
T2	Little Red Riding Hood	6	28
T3	Cinderella	4	36
T4	Yakky Doodle	4	43
T5	Goldfish	5	77
T6	The fox and the cock	4	40
			246

Fig. 5. Action structure of the Little Red Riding Hood story plot

(CA) and study how they behave in our ODT framework. We have used jCOLIBRI [11] to develop the Tales application and to perform the experiments. We will analyze the results for two specific implementations of TA and CA for case reuse in the tale generation domain. First we describe the query structure and some other decisions taken during the implementation of both approaches, TA and CA. Then, for the same sets of queries we compare the distances between the generated solutions and the solutions in the case base, and the distribution of the generated solutions with respect to those preexisting in the case base.

Queries: The queries use the same vocabulary used to describe the cases in the case base, i.e., the domain ontology. As a query the user provides a set of actions, characters, places, and objects that (s)he would like to include in the tale. Actions in the query are neither ordered nor linked to specific characters, objects or places. For the experimentation we defined four collections of queries named `Q1`, `Q3`, `Q5`, `Q7`. Each collection was populated, respectively, with queries involving 1,3,5, and 7 instances of each first level concept (i.e. actions, characters, places, and objects); 20 queries were randomly generated for each collection.

Originality Measure: In order to assess the novelty of solutions we will measure an edit distance from a new solution to each solution in the case base. The distance between two tale structures will assess the dissimilarity between those solutions. We use the Zhang & Shasha's algorithm [12], where the cost of adding, deleting, or substituting a node in the tree depends on the distances of the elements in the domain ontology. Moreover, the distance between two tales is normalized by the size of the smaller one. We will analyze (1) the distances on

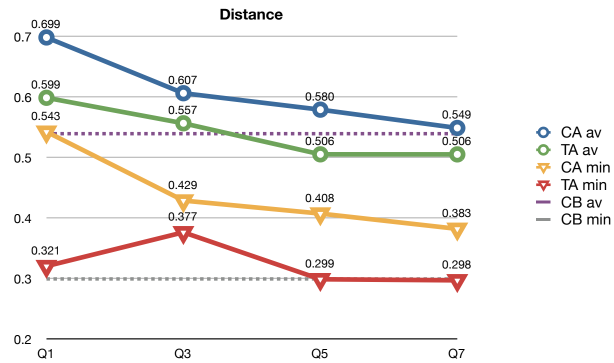


Fig. 6. Average and minimum distance of new solutions w.r.t the case base.

the preexisting tales in the case base, and (2) the distances of the generated tales with respect to the case base for each query in both TA and CA.

We first analyze the distances among the tales preexisting in the case base. Since they are assumed to be original (in the sense that there is no plagiarism among them), the distances among them will give us a qualitative measure of what is desirable for the generated tales to be considered original. The average edit distance over all pairs of the case base solutions is $CB_{av} = 0.54$. Moreover, the two solutions that are more similar have a distance $CB_{min} = 0.3$; thus we can consider this a lower threshold for originality since we assume that the tales in the case base are original. Therefore, if the distance of a generated solution to every solution in the case base is higher than $CB_{min} = 0.3$, we will consider it to be original. According to definition of originality in Section 2 the plagiarism threshold in the example domain would be $\gamma = 0, 3$.

Figure 6 shows the average distances of the solutions for query collections `Q1`, `Q3`, `Q5`, `Q7` generated by TA and CA with respect to the case base. Both TA_{av} and CA_{av} have on average distances higher than the threshold distance $CB_{min} = 0.3$, so they can be considered, on average, to be original with respect to the cases they are built from. Moreover, their average distances TA_{av} and CA_{av} are around $CB_{av} = 0.54$, the average distance among the case base solutions. Therefore, the solutions generated by CBR are as original, on average, as the cases provided by the initial case base.

Another way to visualize this fact is shown in Figure 7, where solutions in the case base and solutions generated by TA and CA are mapped in a two-dimensional space. The original data is a matrix of pairwise distance values among all solutions, while the visualization is built using a force-directed graph-drawing algorithm where the repulsive force between two cases is proportional to their distance. In order to provide original solutions, a CBR system has to look for solutions that are situated in a sparse area of the solution space. We can see in Figure 7 that all solutions (initial and generated) are evenly distributed, without clumps or clusters.

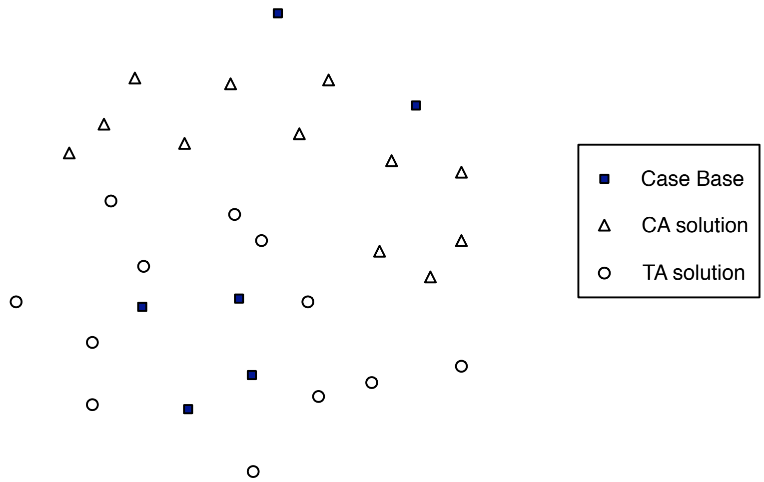


Fig. 7. Distribution of solutions regarding the original case base

Comparing TA and CA, in general CA tends to find solutions in the unpopulated region of the solution space while TA keeps closer to the previously existing cases. This effect was expected by hindsight: since TA works by transforming an existing solution, it seems reasonable to expect that it will change what needs to be changed (following a parsimony principle) while CA builds the solution and opportunistically reuses parts of existing solutions in different cases.

This difference can also be seen in Fig. 6, where CA solutions are more distant on average from the case base than TA solutions. In relation to query complexity, both TA and CA techniques follow the same pattern of decreasing average distance to the case base as the query constraints increase from `Q1` to `Q7`. Our explanation for this effect is that `Q7` constrains much more the set of admissible solutions than `Q1`; e.g. `Q7` specifies 7 actions, 7 characters, 7 places, and 7 objects (and they are generated randomly in our experiments). Nevertheless, `Q7` solutions are around the average $CB_{av} = 0.54$ for the case base, which is good. These results indicate however that very specific queries may cause problems by being over-constraining and reducing admissible solutions to a rather small set; in this circumstance an originality driven task would basically require a lot of search and the usefulness of cases may be reduced. As future work, we suggest later that a conversational CBR approach could be useful in this scenario.

Finally, we have so far analyzed average distance, so we turn to the worst case scenario. Figure 6 also shows the minimal distances TA_{min} and CA_{min} from a solution to the case base for each query collection `Q1, ..., Q7`. Since both TA_{min} and CA_{min} are above or around $CB_{min} = 0.3$, we can safely say that even the generated solutions with lower distances can be safely considered original (with respect to the originality in the content of the case base). As before, CA provides solutions that are more distant from the case base than TA; the explanation is again the parsimony principle of TA, while CA reuses opportunistically parts of different cases in its constructive process.

Since both TA and CA produce solutions without knowing any threshold of "minimal distance" that need be surpassed, it may seem unexpected that all solutions end up being sufficiently original in our experiments. We think the reason is the ontology used in the task of folk tale generation and the handling of solution consistency in both TA and CA (albeit using different mechanisms). Essentially, reuse in TA and CA explore the solution space searching for solutions that satisfy the elements required in the query; this already put further the new solution from the case base. Moreover, the reuse process by either adding a new element (in CA) or transforming an element (in TA) triggers further constraints to be satisfied, which in turn require further additions/transformations. Thus, originality in folk tale generation is obtained by the consistency enforcement during the reuse process in the presence of a large solution space. Clearly, this need not be true for any originality-driven task using CBR; Section 7 we suggest future work where solution space distance is estimated as part of the reuse process for originality-driven tasks.

6 State of the Art

Related to our work are several CBR approaches for the task of innovative design. The FAMING system [13] is an example of the use of case adaptation for supporting innovative design of kinematic pairs; reuse in FAMING combines a structural model with constraint-based techniques for generating solutions different from the ones in the case base. The structural model is akin to our ontology in providing domain knowledge and constraint-based search provides a mechanism for preserving consistency in solutions. The FAMING system thus fits in our ODT framework of CBR systems, in that the originality of the solution is not pursued as such, but is a result of the domain knowledge and the consistency maintenance during reuse. However, the paper [13] is interested in showing that "different solutions" can be found by a CBR system in this way, but it is not intent on developing a framework for originality-driven CBR tasks. Another CBR approach is the IDEAL system [14], that produces innovative solutions by adapting solutions of design cases from one domain to another distant domain by using *structure-behavior-function* models. A survey of CBR approaches to design and innovation can be found in [15].

Regarding tale generation, there have been various attempts in the literature to create a computational model. Many existing systems are somehow related with the CBR paradigm, even if they do not explicitly mention it, because they are based on re-using a collection of plots with the structure of coherent tales [16, 3, 9, 17, 6]. Basically, these story creation systems retrieve a complete plot structure and reuse it by changing secondary elements of the story world, like places or characters. A related approach, that is also based on the Proppian morphology, is that of Fairclough and Cunningham [9]. They implement an interactive multiplayer story engine that operates over a way of describing stories based on Propp's work, and applies case-based planning and constraint satisfaction to control the characters following a coherent plot.

7 Conclusions and Future Work

The purpose of this paper was to analyze CBR in the context of a class of tasks we called *originality-driven tasks* (ODT). We characterized the *originality* of a CBR solution using the pragmatic notion of plagiarism: a solution is original if it cannot be accused of plagiarism with respect to previous solutions (i.e. to the solutions in the case base). Since plagiarism is defined as a measure of similarity between objects, originality of CBR solutions can easily be understood and measured by defining a distance measure (or equivalently a similarity measure) on the *space of solutions*. We then modeled the reuse process in ODTs as a search process that builds solutions that are not only new and valid with respect to the query but also distant enough in the space of solutions from preexisting solutions.

After establishing this conceptual framework, we examined how two different reuse techniques (one transformational and the other constructive) address the issues of originality-driven tasks in CBR; moreover, we designed and performed some experiments in the domain of folk tale generation where originality of solutions could be assessed and analyzed. We saw that the two reuse techniques indeed produced original solutions, even if transformational reuse seemed a priori more likely to produce solutions more similar to preexisting cases. Since existing reuse techniques do not internally use a distance measure in the space of solutions to enforce the originality of the new solution, we had to conclude that this "originality" was a kind of side effect. Solutions are original because of the interplay of two factors: the large solution space and the maintenance of solution consistency that forces the reuse process to search for solutions even more distant in order to build a consistent solutions.

The difference between transformational and constructive reuse was less than a priori expected. We assumed that transformational reuse would find solutions less distant than constructive reuse, as indeed can be observed in Fig. 6. The differences however are not large, and transformational reuse always found solutions that are original. One difference between transformational and constructive reuse is the way in which they maintain solution consistency while searching in the solution space, but this difference is minor compared with the fact that it is this consistency maintenance mechanism that forces changes in the solution and ends up building a solution far away from the initial case base.

Concerning future work we think that both TA and CA reuse for ODT should include a way to measure distances in the solution space to be able to ensure that solutions are original with respect to some appropriate domain threshold. Most CBR systems focus on exploiting similarity on the problem space, but few use similarity on the solution space; we think ODT is a class of problems where new CBR techniques that use similarity on the solution space can be developed. Moreover, the notion of plagiarism can be refined; we were using here a global measure among solutions, but plagiarism accusations can focus on specific parts of solutions (e.g. in music a few notes too similar to another song are grounds for plagiarism claims). This refined notion of plagiarism would require more introspective reuse techniques that estimate and maintain both consistency and originality over partial solutions during the reuse process.

Finally, the effect of over-constrained queries suggests that a conversational CBR approach would be best suited for folk tale generation, and maybe for ODTs in general. A conversational CBR approach could start with a smaller query, allowing the user to augment the query requirements incrementally while the CBR system would assess whether new requirements can be incorporated or compromise the originality of the solution.

References

1. Plaza, E., Arcos, J.L.: Constructive adaptation. In: Advances in Case-Based Reasoning. Volume 2416 of LNAI. Springer Verlag (2002) 306–320
2. Adams, S.: Storytelling and computer games: Past, present and future (2001)
3. Braun, N., Schneider, O., Habinger, G.: Literary analytical discussion of digital storytelling and its relation to automated narration. In: Workshop Understanding User Experience: Literary Analysis meets HCI, London, UK (2002)
4. Bringsjord, S., Ferrucci, D.: Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a StoryTelling Machine. Lawrence Erlbaum Associates, Hillsdale, NJ (1999)
5. Díaz-Agudo, B., Gervás, P., Peinado, F.: A CBR approach to story plot generation. In: ECCBR'04, Springer-Verlag LNCS/LNAI, Madrid, Spain (2004)
6. Peinado, F., Gervás, P., Díaz-Agudo, B.: A description logic ontology for fairy tale generation. In: Language Resources for Linguistic Creativity Workshop, 4th LREC Conference., Lisboa, Portugal (2004)
7. Propp, V.: Morphology of the Folktale. University of Texas Press (1968)
8. Malec, S.A.: Proppian structural analysis and XML modeling. http://clover.slavic.pitt.edu/s̃am/propp/theory/propp.html (February 2004)
9. Fairclough, C., Cunningham, P.: An interactive story engine. In O'Neill, M., Sutcliffe, R., Ryan, C., Eaton, M., N.J.L, G., eds.: 13th Irish International Conference on Artificial Intelligence and Cognitive Science. Volume 2464., Limerick, Ireland, LNCS Springer-Verlag Heidelberg (2002) 171–176
10. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady **10** (1966) 707–710
11. Díaz-Agudo, B., González-Calero, P.A., Recio-García, J., Sanchez-Ruiz, A.: Building CBR systems with jCOLIBRI. Journal of Science of Computer Programming **69 Num:1-3** (2007) 68–75
12. Zhang, K., Shasha, D.: Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**(6) (1989) 1245–1262
13. Faltings, B., Sun, K.: FAMING: Supporting innovative mechanism shape design. The Knowledge Engineering Review **30(3)** (2005) 271–276
14. Bhatta, S., Goel, A.: Learning generic mechanisms for innovative strategies in adaptive design. Journal of Learning Sciences **6(4)** (1997) 367–396
15. Goel, A.K., Craw, S.: Design, innovation and case-based reasoning. The Knowledge Engineering Review **30(3)** (2005) 271–276
16. Turner, S.R.: Minstrel: a computer model of creativity and storytelling. PhD thesis, Los Angeles, CA, USA (1993)
17. Callaway, C.B., Lester, J.C.: Narrative prose generation. Artificial Intelligence **139**(2) (2002) 213–252