Confidence in Workflow Adaptation

Mirjam Minor, Mohd. Siblee Islam, and Pol Schumacher

University of Trier - Department of Business Information Systems II
D-54286 Trier, Germany
{minor,islam,pol.schumacher}@uni-trier.de

Abstract. This paper is on assessing the quality of adaptation results by a novel confidence measure. The confidence is computed by finding evidence for partial solutions from introspection of a huge case base. We assume that an adaptation result can be decomposed into portions, that the provenance information for the portions is available. The adaptation result is reduced to such portions of the solution that have been affected by the change. Furthermore, we assume that a similarity measure for retrieving the portions from a case base can be specified and that a huge case base is available providing a solution space. The occurrence of each portion of the reduced solution in the case base is investigated during an additional retrieval phase after having adapted the case. Based on this idea of retrieving portions, we introduce a general confidence measure for adaptation results. It is implemented in the area of workflow adaptation. A graph-based representation of cases is used. The adapted workflow is reduced to a set of sub-graphs affected by the change. Similarity measures are specified for a graph matching method that implements the introspection of the case base. Experimental results on workflow adaptations from the cooking domain show the feasibility of the approach. The values of the confidence measure have been evaluated for three case bases with a size of 200, 2,000, and 20,000 cases each by comparing them with an expert assessment.

1 Introduction

Adaptation has achieved significant attention in Case-Based Reasoning (CBR) over the past few years [1–7]. A retrieved solution has to be adapted in order to be reused for a current problem [8]. Several adaptation techniques and frameworks have been introduced, which apply rules or operators [1–3], merge cases [4, 5], or reuse dedicated adaptation cases [6, 7].

In most adaptation approaches, it is difficult to assess the quality of the adaptation result with respect to the given problem a priori. A notable exception is the work on adaptation for configuration tasks [1] where a cost function has been employed as a quality measure for a solution. In the absence of a formal quality measure like a cost function, the quality of the adaptation result can be approximated by quality indicators like the utility, correctness, or completeness of the solution. The utility of the solution might serve as a quality criterion which can be approximated by the value of the similarity function the retrieved solution

B. Díaz Agudo and I. Watson (Eds.): ICCBR 2012, LNCS 7466, pp. 255–268, 2012.
© Springer-Verlag Berlin Heidelberg 2012

has achieved prior to the adaptation [9]. Furthermore, correctness criteria like the syntactical correctness of the adapted solution [7] or the consistency of the solution with a knowledge model [5] can be considered. The completeness of the adaptation describes to what extent the problem is covered by the adaptation result [4].

In this paper, we introduce a confidence measure for an adaptation result, which serves as an additional indicator for the quality of the adapted solution. We focus on a special type of problem cases, namely on workflows that are to be adapted. However, our notion of confidence in adaptation is not restricted to cases that contain procedural knowledge or workflows. It can be applied as well for structural or textual cases. Our ideas have been inspired by confidence measures from data mining where the confidence c of a discovered association rule $X \Rightarrow Y$ for two data items X and Y is predicted by occurrence, namely by the percentage of the data itemsets containing X (i.e. d_X) that also contain Y ($d_{X,Y}$) [10], expressed by $c = \frac{d_{X,Y}}{d_X}$. In CBR, confidence has also been discussed. The confidence in a solution retrieved from a case base has been investigated [11] as well as the confidence in a classification created by a CBR system [12]. Both CBR approaches predict confidence based on similarity measures. In our approach, the confidence in an adapted solution is predicted based on introspection of the case base by determining whether parts of the solution occur somewhere else. The adapted solution is (I) reduced to those portions that have been affected by the adaptation, (II) a retrieval is performed for each portion, and (III) a confidence value is derived from the retrieval results. The approach is limited to positive confidence values from a case base of "good experiences". Furthermore, it is limited to a local perspective since it does not consider dependencies between the portions. Thus, our confidence measure can be regarded one indicator among others for the quality of an adaptation result. Our hypothesis is that such an indicator enables the CBR system to suggest only solutions with adapted portions that have some evidence by occurence in the case base and thus increase the confidence in the entire solutions.

The remainder of the paper is organized as follows. In Section 2, we briefly sketch the adaptation of workflows and explain how the adapted portions can be tracked during this process. In Section 3, we introduce a confidence measure for adaptation results and apply it for adapted workflows. A formative evaluation is described in Section 4. We conclude the paper with a summary and a discussion of future work in Section 5.

2 Workflow Adaptation

In order to assess an adapted solution by a confidence measure, the adaptation process has to be tracked and those portions of the solution have to be identified that have been modified during adaptation. Our approach focusses on workflow adaptation. 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"

[13]. Adaptive workflow systems (also called agile workflow systems) facilitate structural changes of workflows at run time [14–18]. Workflows can be created (for instance based on a template from a repository) and tailored for a particular demand or business case. Workflows can still be adapted after they have been started, for example if some unforeseen events occur. The changes apply to workflow elements, i.e., to atomic parts of the workflow. In our approach, we track workflow elements that have been modified during automated adaptation. The particular adaptation method that has been used to modify the workflow is not of interest for the confidence measure. For the experimental evaluation (compare Section 4), we have chosen a case-based adaptation approach [7]. Any alternative adaptation approach, e.g. a rule-based or plan-based approach, would have been applicable as well.

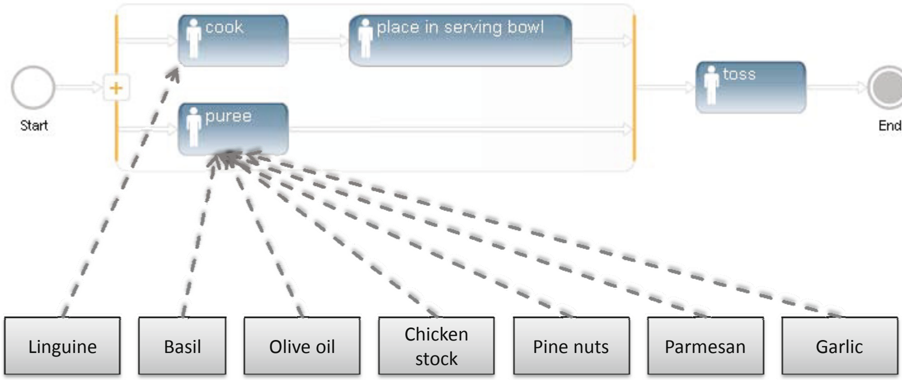


Fig. 1. Sample workflow in CFCN for cooking pasta with basil pesto

Figure 1 depicts a sample workflow from the cooking domain. We use the Cake Flow Cloud Notation (CFCN, compare [19]) to illustrate the work. CFCN has been developed in recent research projects at the University of Trier [19, 20] as a part of the Collaborative Agile Knowledge Engine (Cake) [21, 22]. The cake system provides modeling and enactment support for workflows including adaptation support. CFCN consists of several types of workflow elements like tasks, data objects, data links (from a data object to a task or vice versa), and control flow elements like AND-splits, AND-joins etc. The sample in Figure 1 describes a recipe for basil pesto sauce over pasta. The ingredients are represented by data objects ("Linguine", "Basil", etc.) while the cooking activities form the tasks ("cook", "puree", etc.). The linguine are cooked in parallel to pureeing the other ingredients as indicated by the AND-split symbolized by a '+'. In Figure 2, an adapted workflow is shown, in which the "Pine nuts" have been substituted by "Walnuts". This requires an additional task "crack" that has been inserted before "puree".

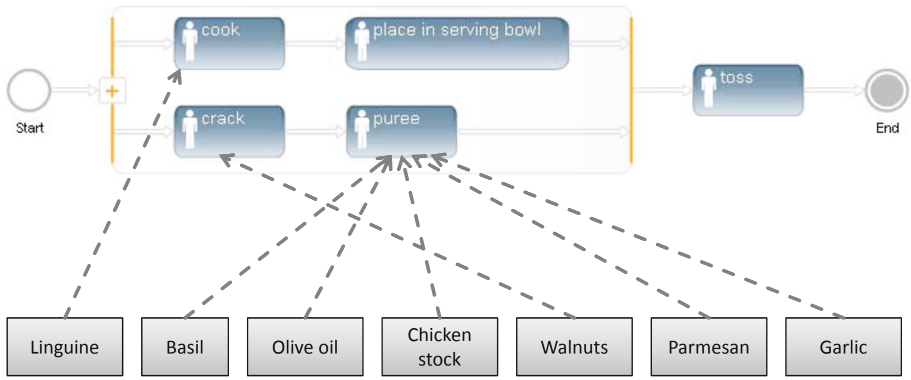


Fig. 2. Adapted sample workflow from Figure 1

In order to track the adaptation process of a workflow, the provenance of each workflow element is used, i.e., whether it stems from the original workflow or whether it has been inserted or modified during the adaptation process, like the task "crack", for instance. As a result, the list of those tasks is stored that have been affected by the adaptation. This includes newly inserted tasks as well as tasks with new or deleted data links. In the above sample, the list of modified tasks consists of the task "crack" which has been newly inserted together and the task "puree" since its former data object "Pine nuts" has been deleted during adaptation.

3 Confidence in Adaptation Quality

The confidence in an adapted solution is predicted by introspection for the changed parts. The approach makes the following three assumptions: The adapted solutions can be decomposed into parts, which are described with the same formalism as the solutions. The provenance information for each of those parts is available. A similarity measure can be defined for the parts with respect to the case base. If these assumptions are fulfilled, a confidence measure can be defined as follows.

Let $\mathcal{S}$ be the (possibly infinite[1]) universe of adapted solutions and $\mathcal{CB}$ the universe of case bases. The confidence c can be predicted by a confidence measure $c : \mathcal{S} \times \mathcal{CB} \rightarrow \mathbb{R}$.

A reduction function $reduce : \mathcal{S} \rightarrow 2^{\mathcal{S}}$ transforms an adapted solution $s \in \mathcal{S}$ into a reduced, decomposed solution $\hat{S} \in 2^{\mathcal{S}}$. The solution $\hat{S} = reduce(s) = \{s_1, s_2, ...s_n\}$ consists only of the portions of s that have been affected by the adaptation according to the provenance information. The atomic units for the decomposition have to be specified properly, such that the s_i's can be compared with the cases from the case base later on. With this, the confidence for a solution

[1] For instance, in case of adaptation rules that can be arbitrarily often repeated.

$s \in S$ with respect to a case base $CB \in \mathcal{CB}$ can be computed by means of an aggregated similarity function sim_Φ:

$$c(s, CB) = sim_\Phi(\{s_1, s_2, ..., s_n\}, CB)$$

where Φ is an aggregation function $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}$ for the similarity values of the particular portions s_i with $i = 1, 2, ..., n$ and the cases c_j from the case base $CB = \{c_1, c_2, ..., c_m\}$. For instance, a maximum function for the best matching case to every portion can be chosen and combined with a minimum function for the portion that has achieved the lowest similarity value:

$$\begin{aligned} sim_\Phi(\hat{S}, CB) = \min_{s_i} \max_{c_j} sim(s_i, c_j) \\ s.t.\ s_i \in \hat{S}, c_j \in CB \end{aligned}$$

The maximum function in Φ is quite natural, as it is implemented by any retrieval, which computes the best matching case. For each portion, the "best" occurrence is chosen. Chosing the minimum value over all portions expresses the pessimistic view that the portion that achieved the smallest maximum similarity value determines the overall confidence. To be more optimistic, the minimum function could be replaced, for instance, by a weighted sum. In the following, the reduce and retrieve steps will be described for adapted workflows.

3.1 Reduce

An adapted workflow is converted into a reduced adapted workflow that comprises only the affected tasks with corresponding data items related to the tasks. The *reduce* function derives the reduced workflow from the list of tasks that have been tracked as inserted or modified during the workflow adaptation (compare Section 2). The tasks are stored with all their related data objects. Figure 3 represents the filtered workflow derived in a first part of the *reduce* function from the main workflow represented in Figure 2. The resulting workflow is a sequence of the two tasks "crack" with the data object "Walnuts" and "puree" with the five remaining data objects "Basil", "Olive oil", "Chicken stock", "Parmesan" and "Garlic".

As the second part of the *reduce* function, the reduced workflow is decomposed into a set of queries $\hat{S} = \{s_1, s_2, ..., s_n\}$ in preparation for the retrieval. For this, we have chosen a graph representation recently introduced by Bergmann and Gil [23]. A workflow can be transformed into a directed graph $W = (N, E, S, T)$ where N is a set of nodes and $E \subset N \times N$ is a set of edges having a type $T: N \cup E \rightarrow \Omega$ and a semantic description $S: N \cup E \rightarrow \Sigma$ where the type and semantic description are associated to each node and edge that are taken from Ω and Σ respectively. Ω consists of the types *workflow node, data node, task node, control flow node, control flow edge, part of edge* and *data flow edge*. W has exactly one *workflow node*. The *task nodes* and *data nodes* represent tasks and data objects respectively. A *control flow node* stands for contol flow elements. The *data flow edge* is used to describe the data links. The *control flow edge* is

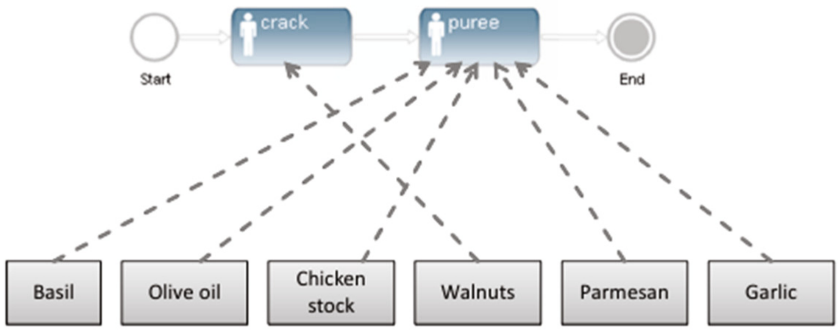


Fig. 3. Filtered workflow

used to represent the control flow of the workflow, e.g., from task to task or from task to control flow element. The *part of edge* shows the relation between *workflow node* and *data node*, *task node*, or *control flow node*. Σ is a semantic meta data language. In our approach, it consists of a universe of names for tasks and data objects. W is then split into sub-graphs for each task that consist of the workflow node and one task node along with the data nodes that are related to it. Thus, $\hat{S}$ consists of a set of sub-graphs s_i. Figure 4a depicts a sample graph W representing the filtered workflow derived from the main workflow in Figure 3. The sub-graphs derived from the above sample graph W are depicted in Figure 4b and 4c.

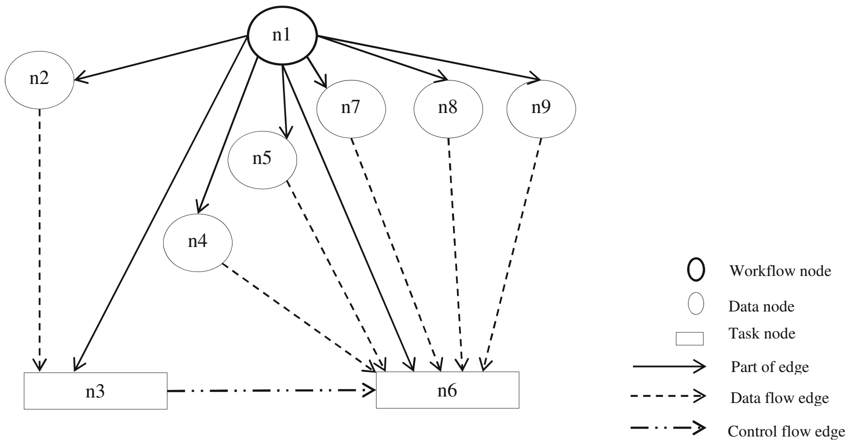
3.2 Retrieve

Each sub-graph s_i of the reduced workflow representation described above is asked as a query to the case base. For each s_i, the retrieval is performed by means of a graph matching method introduced by Bergmann and Gil [23]. Broadly speaking, nodes and edges of the query graph s_i are mapped to the best matching nodes and edges of a graph c_j from the case base. The mapping with the highest similarity is chosen as a retrieval result.

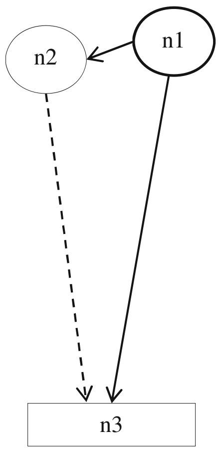
Local similarity functions for graph nodes (sim_N) and edges (sim_E) have to be specified. The similarity of a query node $n_q \epsilon N$ with a case node $n_c \epsilon N$ is described by the similarity function $sim_N(n_q, n_c)$ according to Bergmann and Gil by:

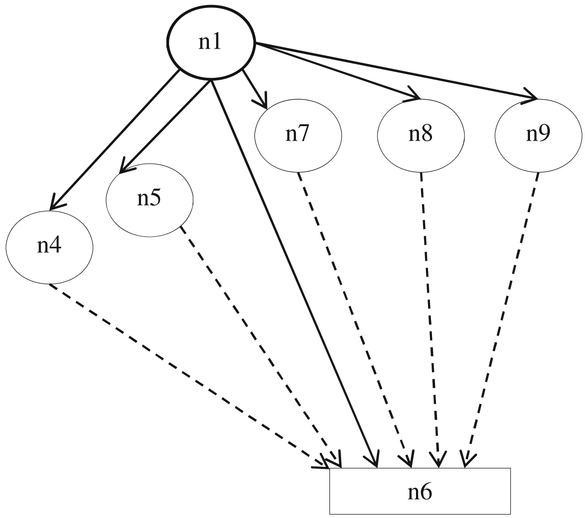
$$sim_N(n_q, n_c) = \begin{cases} sim_\Sigma(S_q(n_q), S_c(n_c)) & \text{if } T_q(n_q) = T_c(n_c) \\ 0 & \text{otherwise} \end{cases}$$

sim_Σ denotes a local similarity function for semantic descriptions. We have chosen a similarity function sim_Σ that is derived from a Levenshtein distance measure. The Levenshtein distance is purely syntactic and measures the minimum number of edit operations to transform one string into another at the character level.



(a) Graph W for the sample workflow from Figure 3





(b) Sub-graph s_1 for task "crack"

(c) Sub-graph s_2 for task "puree"

Fig. 4. Decomposition of a sample workflow graph into two sub-graphs

The similarity of a query edge $e_c \epsilon E$ with a case edge $e_q \epsilon E$ is described by the similarity function $sim_E(e_q, e_c)$ according to Bergmann and Gil by:

$$sim_E(e_q,e_c) = \begin{cases} F_E \begin{pmatrix} sim_\Sigma(S_q(e_q), S_c(e_c)), \\ sim_N((e_q.l),(e_c.l)), \\ sim_N((e_q.r),(e_c.r) \end{pmatrix} & \text{if } T_q(e_q) = T_c(e_c) \\ 0 & \text{otherwise} \end{cases}$$

Where F_E is specified as $F_E(S_e, S_l, S_r) = S_e * 0.5 * (S_l + S_r)$ and

$$sim_\Sigma = \begin{cases} 1 & \text{if } T_q(e_q) = T_c(e_c) \\ 0 & \text{otherwise} \end{cases}$$

The similarity of two graphs s_i and c_j is computed by means of legal mappings, i.e., a node can be mapped by a partial injective mapping function $m : N_q \cup E_q \rightarrow N_c \cup E_c$ if the following five constraints are satisfied:

$$\begin{array}{lll} T_q(n_q) = T_c(m(n_q)) & T_q(e_q) = T_c(m(e_q)) & \\ m_q(e_q.l) = m(e_q.l) & m_q(e_q.r) = m(e_q.r) & \forall_{x,y} m(x) = m(y) \rightarrow x = y \end{array}$$

Two edges can be mapped if the respective nodes that are connected by the edges can also be mapped.

The mapping can be partial. The similarity is computed for all possible mappings as described by the following equation where $Dom(m)$ is domain of m.

$$sim_m(s_i,c_j) = F_w \begin{pmatrix} (sim_N(n,m(n))|n \epsilon N_q \cap Dom(m)), \\ (sim_E(e,m(e))|e \epsilon E_q \cap Dom(m)), \\ |N_q, E_q| \end{pmatrix}$$

Where $F_w((sn_1,.....,sn_i),((se_1,.....,se_j),n_N,n_E) = \frac{sn_1+.....+sn_i+se_1+.....+se_j}{n_N+n_E}$.

The mapping with the maximum similarity value sim_m is chosen as the overall similarity sim between a solution part $s_i \in \mathcal{S}$ and a case $c_j \in CB$. Without loss of generality, we have chosen local similarity functions with values between 0 and 1, i.e., $sim : \mathcal{S} \times CB \rightarrow [0,1]$.

4 Evaluation

We conducted some experiments in the cooking domain in order to evaluate the approach. Cooking instructions have been formalized as workflows as described in Section 2. Three experimental case bases CB_{200}, CB_{2000}, and CB_{20000} with 200, 2,000, and 20,000 cooking workflows each have been created by an workflow extraction approach [24] from recipes of an online cooking community[2]. For 26 sample workflows, change requests have been formulated by hand. 19 of them could be adapted successfully by an automated, case-based adaptation method [7] (compare Table 1). The confidence values for the adapted workflows have

[2] `www.allrecipes.com`

Table 1. Confidence values and quality assessment of successfully adapted workflows

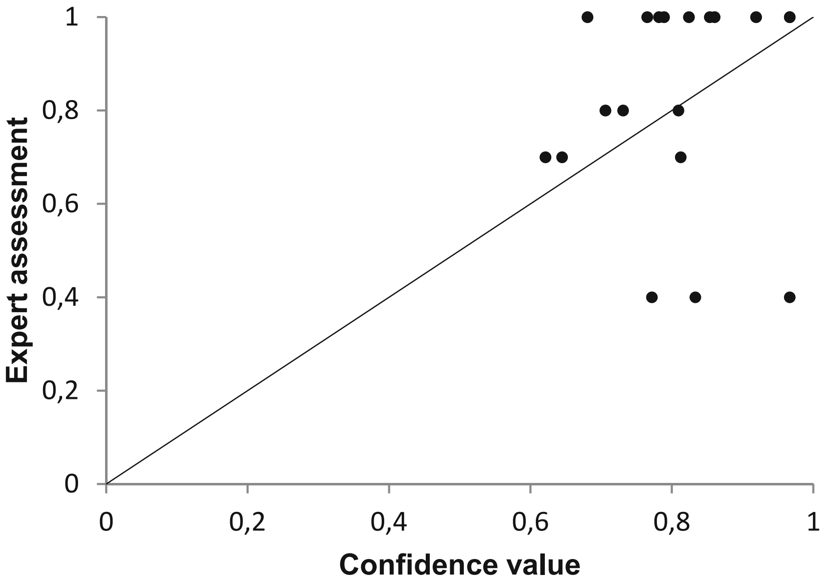
Case no.	Change request	CB_{200}	CB_{2000}	CB_{20000}	e_s
1	Omit onions	0.789	0.846	0.876	1.0
2	Replace commercial soup by sauce hollandaise	0.645	0.694	0.772	0.7
3	Replace garlic by ramsons	0.809	0.809	0.816	0.8
4	Replace butter by olive oil	0.813	0.820	0.887	0.7
5	Replace commercial soup by a mixture of whipping cream and eggs	0.621	0.668	0.717	0.7
6	Omit olives	0.919	0.919	0.967	1.0
7	Replace olive oil by butter	0.967	0.967	0.967	1.0
8	Replace spaghetti by macaroni	0.967	0.967	0.967	1.0
9	Top additionally with cheese	0.967	0.967	0.967	0.4
10	Omit mushrooms	0.731	0.727	0.771	0.8
11	Replace ricotta and cream by cottage cheese	0.833	0.879	0.907	0.4
12	Replace spinach by chard	0.782	0.814	0.814	1.0
13	Replace pine nuts by almonds	0.772	0.872	0.852	0.4
14	Omit pimiento	0.706	0.720	0.794	0.8
15	Omit nutmeg	0.854	0.871	0.898	1.0
16	Replace sundried tomatoes by fresh tomatoes	0.681	0.712	0.808	1.0
17	Refine olive oil with herbs	0.824	0.839	0.887	1.0
18	Omit capers	0.765	0.808	0.876	1.0
19	Omit parsley	0.860	0.882	0.896	1.0

been computed by the introspective confidence measure as described above. An expert was asked to assess the quality of the adaptation results and assign scores from 0 (for "bad") to 10 (for "very well") to each of the adapted workflows. This empirical value reflects the subjective opinion of the expert whether the recipe described by the adapted workflow would produce a tasty dish. We compared the confidence values achieved for CB_{200}, CB_{2000} and CB_{20000} with the quality scores assigned by the expert.

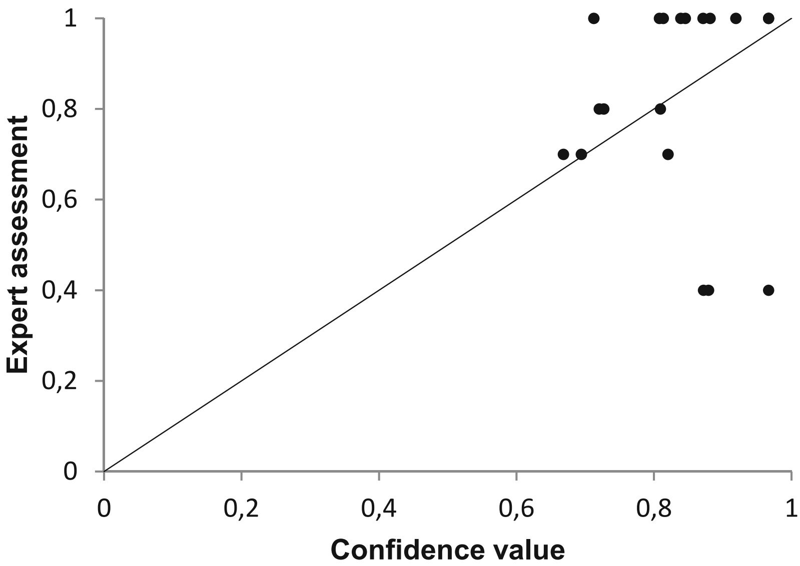
The following hypotheses have been investigated:

H1 The values of the confidence measure increase with the size of the case base.
H2 The values computed by the confidence measure are lower than the values provided by the expert $c(s, CB) \leq e_s$, i.e., the confidence measure provides an underestimation for the empirical quality.
H3 The values computed by the confidence measure approach the expert values with an increasing case base size: $|c(s, CB_i) - e_s| \geq |c(s, CB_j) - e_s|$ if $|CB_i| \leq |CB_j|$.

Hypothesis $H1$ was confirmed by the experiments (see the confidence values in Table 1). Only for change requests 10 and 11, the values of c decrease slightly from case base CB_{200} to CB_{2000}. Figures 5a - c depict the expert values in comparison to the automated values.

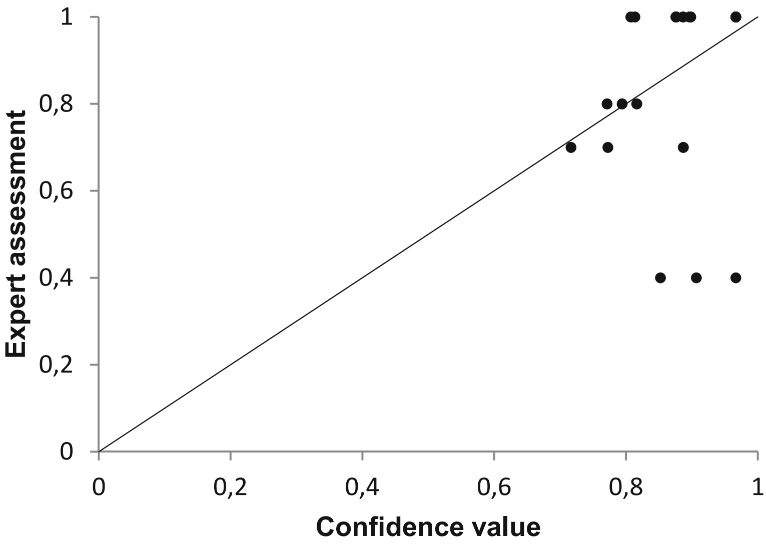


(a) for CB_{200}



(b) for CB_{2000}

Fig. 5. Comparison of the expert assessments with the confidence values



(c) for CB_{20000}

Fig. 5. (*Continued*)

Hypothesis $H2$ could not be confirmed as 17 of the 57 values of c overestimate the quality value assigned by the expert. The values for change requests 9, 11 and 13 clearly contradict $H2$. The reason for the low expert value given for change request 9 is that the adaptation method suggested to cook the additional cheese instead of topping the dish with cheese. In the recipe for change request 11, the cottage cheese has been inserted at the wrong place by the automated adaptation. In the recipe for change request 13, the expert did not like that the almonds have not been cut before using them in the recipe. However, the few other values are only slight overestimations. In Figures 5a - c, the data points below the diagonals illustrate the overestimating values.

Hypothesis $H3$ was confirmed by 30 of 38 values. We repeated the experiment with three groups of 10 case bases of 200, 2,000 and 20,000 cases that have been extracted from the same recipe Web page arbitrarily. The average difference between the expert value and the values of the ten $c(s, CB_{(200,i)})$ is 0.157, 0.137 for $c(s, CB_{(2000,i)})$ and 0.129 for $c(s, CB_{(20000,i)})$. We admit that the number of measured values is too small for a statistically solid statement.

Hypothesis $H1$ and $H3$ have been confirmed by our experiments while $H2$ was contradicted. One reason for the results on $H2$ could be, that the similarity function applied for the sub-graphs was too optimistic since it tolerates incomplete mappings. Furthermore, it seems promising to relax the property of being an underestimation for future evaluations on hypothesis $H2$.

5 Conclusion

In this paper, we introduced a confidence measure for adaptation results based on introspection of the case base. The adaptation results are decomposed into portions, the provenance for each portion is determined, and those portions that stem from the adaptation process are retrieved from the case base. The occurrence of a portion provides some empirical evidence for the feasibility of the adapted solution. Without loss of generality, we restricted the approach to workflow adaptation. We defined a confidence measure based on a graph representation. We have chosen sub-graphs as atomic portions of the adapted solution, which consist of a workflow task with its according data objects. The retrieval has been performed by a sub-graph matching. We conducted some experiments on cooking workflows describing a cooking instruction from a recipe step-by-step. Automatically adapted workflows have been assessed by both, a confidence measure and a human expert. The experiments provided promising results, since the confidence values improved with the size of the case base with respect to baseline values from the expert. The results confirmes the feasibility of the confidence measure.

The approach provides many opportunities for future work. The experiments should be repeated in further domains of workflow adaptation. The measure could be specified also for the adaptation results of structural or textual cases. The confidence measure could be complemented by a measure for potential dependencies between adapted portions or by user scores of the cases containing the retrieved portions. Negative confidence values could be included, e.g. from inverse workflow patterns [25]. Different variants for the similarity measures underlying the confidence measure could be investigated. Our next steps will be to consider more semantic information in the local similarity measures, for instance, from task ontologies, and to conduct experiments with further workflow domains.

Acknowledgements. This work was funded by the German Research Foundation, project number BE 1373/3-1. Furthermore, we would like to thank the student researchers Stephan Zonker and Alexander Stromer for their contributions in implementing and evaluating the approach. In addition, we appreciate the feedback from Ralph Bergmann, which was very helpful to improve the paper.

References

[1] Bergmann, R., Wilke, W.: Towards a new formal model of transformational adaptation in case-based reasoning. In: Proceedings of th German Workshop on CBR, University of Rostock, pp. 43–52 (1998)

[2] De Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., et al.: Retrieval, reuse, revision and retention in case-based reasoning. Knowledge Engineering Review 20(3), 215–240 (2005)

[3] Muñoz-Avila, H., Cox, M.T.: Case-based plan adaptation: An analysis and review. IEEE Intelligent Systems 23(4), 75–81 (2008)

[4] d'Aquin, M., Lieber, J., Napoli, A.: Decentralized Case-Based Reasoning for the Semantic Web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 142–155. Springer, Heidelberg (2005)

[5] Cojan, J., Lieber, J.: Belief Merging-Based Case Combination. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 105–119. Springer, Heidelberg (2009)

[6] Leake, D., Kinley, A., Wilson, D.: Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In: Aamodt, A., Veloso, M.M. (eds.) ICCBR 1995. LNCS, vol. 1010, pp. 229–240. Springer, Heidelberg (1995)

[7] 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, pp. 18–25. IEEE Computer Society, Luxemburg (2011)

[8] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)

[9] Lee-Urban, S., Muñoz-Avila, H.: Adaptation versus Retrieval Trade-Off Revisited: An Analysis of Boundary Conditions. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 180–194. Springer, Heidelberg (2009)

[10] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A., et al.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, vol. 12, pp. 307–328 (1996)

[11] Cheetham, W., Price, J.: Measures of Solution Accuracy in Case-Based Reasoning Systems. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 106–118. Springer, Heidelberg (2004)

[12] Delany, S.J., Cunningham, P., Doyle, D., Zamolotskikh, A.: Generating Estimates of Classification Confidence for a Case-Based Spam Filter. In: Muñoz-Ávila, H., Ricci, F. (eds.) ICCBR 2005. LNCS (LNAI), vol. 3620, pp. 177–190. Springer, Heidelberg (2005)

[13] Workflow Management Coalition: Workflow management coalition glossary & terminology (1999), `http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf` (last access on May 23, 2007)

[14] Reichert, M., Dadam, P.: ADEPT flex–supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems 10(2), 93–129 (1998)

[15] Weber, B., Wild, W.: Towards the Agile Management of Business Processes. In: Althoff, K.-D., Dengel, A.R., Bergmann, R., Nick, M., Roth-Berghofer, T.R. (eds.) WM 2005. LNCS (LNAI), vol. 3782, pp. 409–419. Springer, Heidelberg (2005)

[16] Minor, M., Tartakovski, A., Schmalen, D., Bergmann, R.: Agile workflow technology and Case-Based change reuse for Long-Term processes. International Journal of Intelligent Information Technologies 4(1), 80–98 (2008)

[17] Leake, D.B., Kendall-Morwick, J.: Towards Case-Based Support for e-Science Workflow Generation by Mining Provenance. In: Althoff, K.-D., Bergmann, R., Minor, M., Hanft, A. (eds.) ECCBR 2008. LNCS (LNAI), vol. 5239, pp. 269–283. Springer, Heidelberg (2008)

[18] Montani, S., Leonardi, G.: A Case-Based Approach to Business Process Monitoring. In: Bramer, M. (ed.) IFIP AI 2010. IFIP AICT, vol. 331, pp. 101–110. Springer, Heidelberg (2010)

[19] Minor, M., Bergmann, R., Görg, S.: Adaptive workflow management in the cloud -Towards a novel platform as a service. In: Proceedings of the ICCBR 2011 Workshops, pp. 131–138 (2011)
[20] Görg, S., Bergmann, R., Minor, M., Gessinger, S., Islam, S.: Collecting, reusing and executing private workflows on social network platforms. In: WWW 2012 Workshop Proceedings (2012)
[21] Bergmann, R., Freßmann, A., Maximini, K., Maximini, R., Sauer, T.: Case-Based Support for Collaborative Business. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 519–533. Springer, Heidelberg (2006)
[22] Minor, M., Schmalen, D., Kempin, S.: Demonstration of the agile workflow management system cake II based on Long-Term office workflows. In: CEUR Proceedings of the BPM 2009 Demonstration Track, Business Process Management Conference 2009 (BPM 2009), vol. 489 (September 2009)
[23] Bergmann, R., Gil, Y.: Retrieval of Semantic Workflows with Knowledge Intensive Similarity Measures. In: Ram, A., Wiratunga, N. (eds.) ICCBR 2011. LNCS, vol. 6880, pp. 17–31. Springer, Heidelberg (2011)
[24] Schumacher, P., Minor, M., Walter, K., Bergmann, R.: Extraction of procedural knowledge from the web. In: WWW 2012 Workshop Proceedings. ACM (2012)
[25] Sauer, T., Minor, M., Bergmann, R.: Inverse workflows for supporting agile business process management. In: Maier, R. (ed.) WM 2011: 6th Conference on Professional Knowledge Management. LNI, vol. P-182, pp. 194–203 (2011)