

## Alternative approaches for workflow similarity

Andreas Wombacher  
 Database Department  
 University of Twente  
 Enschede, The Netherlands  
 a.wombacher@utwente.nl

Chen Li  
 Information Systems Group  
 University of Twente  
 Enschede, The Netherlands  
 lic@ewi.utwente.nl

**Abstract**—Service discovery of state dependent services has to take workflow aspects into account. To increase the usability of a service discovery, the result list of services should be ordered with regard to the relevance of the services. Means of ordering a list of workflows is a similarity measure of the workflow and a query. In this paper different similarity measures facilitating structured workflows and higher level change operations are presented and evaluated based on a pilot of an empirical study. In particular the different measures are compared with the study results. It turns out that the quality of the different measures differ significantly. The best results can be achieved by facilitating n-gram multisets as a workflow as a basis for the similarity measure calculations.

**Keywords**—workflow, service discovery, similarity

### I. INTRODUCTION

A service oriented architecture is based on services maintained by independent service providers and invoked by service requesters. A service invocation of a stateless service consists of a single request-response sequence. In case services are statefull further interaction may be necessary. The set of allowed interaction sequences is also known as a choreography. The challenge is to find services which guarantee a successful interaction of service requester and service provider which is also known as service discovery.

Service discovery is not only applied at run-time of a system to realize late binding of services, but can also be applied during the design process to support the re-use of services and to reduce the maintenance costs of services. In particular, at design time service discovery can be used to determine services which can be composed using e.g. a programming in the large approach [1]–[3]. Further, service discovery may be used to check whether a certain service has already been implemented before starting your own implementation. In addition, it can be applied to reduce maintenance costs of a collection of services by clustering services based on their functionality. The explication of shared functionality and its re-use in services of the cluster ensures that the functionality must be maintained only once. Due to the increasing number of services in enterprises these techniques get more and more important.

Service discovery with a focus on choreographies has been addressed in different approaches like e.g. [4]–[10].

However, the usability of the service discovery depends on the correctness and the applicability of the derived results. Let's consider a choreography based service search engine like e.g. [11] which may result in the following two scenarios:

- the service discovery provides an extensive list of hundred or more services: in this case a user of such a system would expect the result list to be ordered such that the most significant results are at the top and the least significant results are at the bottom of the list;
- the service discovery provides no result at all since the query is too specific: in this case a user would expect that the services being most similar to the query - up to a certain threshold - are provided in the result list ordered based on their significance.

In either case a metric is needed to express the equivalence / bisimulation of the query stated in the service discovery with the service descriptions contained in the repository. Such a metric is the similarity of services, or to be more focused the similarity of choreographies.

From a conceptual point of view choreographies represent allowed sequences of interactions, which can be represented as a workflow model. As a consequence, similarity of choreographies can be represented in a more general way as similarity of workflow models. In previous work, different workflow similarity measures [12]–[14] have been investigated and evaluated using the preliminary results of a pilot study [15]. In particular, workflow mining measures and measures comparing collections of workflow states have been applied as similarity measures. From this previous work trace based approaches have been beneficial, specifically n-gram based approaches showed a good performance. In the last years we looked more into workflow models with an explicit structure like e.g. BPEL [16], [17] and the effect of change operations on these structures. From the previous experiments we learned that several approaches (including the n-gram approach) had difficulties dealing with loops in workflows. The idea of this paper is to investigate whether explication of structure in a workflow and the introduction of higher level change operations improves the similarity measures with regard to the results of the pilot study [15].

The contribution of this paper is to evaluate three new and innovative approaches to workflow similarity by applying them on a questionnaire [18] used in the pilot study [15]. The pilot study aimed to check whether the used hypotheses and the constructed questionnaire are usable, thus, the empirical study performed on it will provide reasonable results. However, the already obtained results generated by workflow experts during the pilot study give some general indication on the importance of different aspects of a workflow similarity measure which is now used to evaluate the technical workflow similarity measures. The outcome of this comparison is that explicating workflow structure is beneficial for workflows containing many cycles, but is hindering in simpler cases. The best results can be achieved by extending the n-gram based approach by reflecting higher level change operations on n-grams.

The paper continues with a discussion of related work. Next, the pilot study and its results are summarized in Section III and the best approach discussed in [12] is presented in Section IV as a base line. Then the different measures are introduced and their evaluation based on the pilot study results are described in Section V, VI, and VII respectively. Section VIII summarizes the findings and discusses future work.

## II. RELATED WORK

Similarity in general is a measurement indicating “closeness” between two entities, that is, in our case a measure indicating the equivalence or bisimulation of workflows. Similarity is a symmetric function, which is normalized (values are between 0 and 1) and fulfills the triangular inequation, that is, the sum of the similarities of workflow A and B, and B and C is bigger than the similarity of workflow A and C.

The approaches mentioned below are categorized into language and structure based approaches <sup>1</sup>.

**Language based approaches** often make use of distance measures of strings as a basis to calculate the similarity measure. There exist different distance measures for strings like for example the Hamming distance used in information theory [19] or the edit distance usually applied on strings in the context of text. The edit distance (or Levenshtein distance) [20] between two strings is the smallest number of substitutions, insertions, and deletions of symbols that can be used to transform one string into another. This definition based on a single string can be extended quite easily to a set of strings, i.e. languages. However, this extension does not work in case at least one language is infinite. In [21] an approach where costs are assigned to each change operation is proposed, which can also be applied to infinite languages. Actually, the approach calculates the minimal distance of a

string accepted by the first automaton with a string accepted by the second automaton. The issue with this approach is that the similarity drills down to the difference of two strings, which is quite unspecific in case the languages contain a lot of strings.

**Structural based approaches** are based on relating or transforming a structural representation of a workflow. For instance, a workflow can be interpreted as a directed graph and therefore graph similarity measures can be applied. An example similarity measure based on edit distance has been proposed in [22] addressing graph isomorphism, while [23] addresses subgraph isomorphism.

Another structural approach for reconciliation of processes is presented in [24] also providing a similarity measure. The approach focuses on the common alphabet of two workflows and removes the exclusively used messages of the alphabets. Further, workflow transformation rules, as specified in [25] for Workflow Nets, are used to transform both workflows to the same automaton using only the shared alphabet.

An extended class of structural similarity measures also considers the probability of the occurrence of certain interaction sequences for calculating the distance measure. This is, e.g., the case for the mining based measure (see [12] for more details) and distance definitions based on labeled Markov processes [26], [27]. Since the setting in this paper is per se without concrete interaction sequences, there is no knowledge about probabilities of interaction sequences. In this paper an equal distribution of the interaction sequences is assumed and the focus is on the mining measure in a first glance.

In either case, the language aspects are neglected and only structural aspects are considered, which is not sufficient since e.g. language equivalent workflow models with different graph representations are not considered equivalent.

An approach called causal footprint has been proposed in [28]. The approach is based on activities combined with a set of loop-back links and a set of look-ahead links. Different similarity measures are defined on this abstract representation of a workflow and empirically evaluated.

In [29] several different approaches are investigated with a set of 100 workflows and 10 queries. The authors calculate the mean average precision according to a ground truth. Unfortunately the data set is not available and therefore the presented approaches can not be related to the ones mentioned in the paper.

## III. PILOT STUDY

The evaluation of the technical measures introduced in Section VI, V IV and VII aims to indicate to which extend a measure is meaningful to a human user. Therefore, the evaluation has to be based on an empirical study with the aim to get a good understanding of the human intuition of workflow similarity. Potentially, each individual will have a

<sup>1</sup>A detailed discussion of most of the approaches mentioned below is available in [13].

different intuition of what is important for the similarity of workflows. Therefore, the best way to conduct an empirical study is to ask multiple persons, thus, the results will be more reliable. This can be achieved in an efficient manner using a questionnaire. To determine whether the designed questionnaire will be suitable a pilot study has been conducted. For this evaluation of the similarity measures, the preliminary findings of the pilot study [15] are used as a first indication on the quality of the technical measures. While in [15] the design, conduction and analysis of the study is described, in this paper the focus is on the comparison of some technical measures with the results of the study. In the following, the pilot study is described. In particular, a brief description of the design of the questionnaire is provided, followed by a description of the data collection phase and a summary of the preliminary finding of the pilot study.

#### A. Formal Workflow Model

Finite State Automata (FSA) [30] are the simplest possible model to represent workflows offering mainly sequence, choice and iteration of tasks. More complex models provide additional expressiveness like parallel execution or recursion. However, higher expressiveness requires to investigate more scenarios for determining/evaluating a similarity measure and the aim is to keep it simple first. Thus, Finite State Automata are used as a formal model. With regard to practical applicability of the presented results, we can refer to prior work on transforming a BPEL subset into FSA [31]. Furthermore, the results derived from this study are applicable to service discovery by applying it to the service discovery engine [11] which is based on service matchmaking [32].

A Finite State Automaton is based on a set of states represented as circles, a start state, a set of finite or accepting states represented by circles with thick lines, and labeled transitions represented as directed arcs. In particular, a labeled transition means that a state is changed when a certain message is either sent or received. Example Finite State Automata (FSA) are depicted in Fig 1. An automaton describes the potential execution sequences of a workflow which is also called the language of an automaton. The example automata can be classified in acyclic automata (like e.g. Fig 1 A and B) providing finite languages and cyclic automata (like e.g. Fig 1 C and D) representing infinite languages.

#### B. Questionnaire

It is expected that the workflow similarity is influenced by several aspects like the language of an automaton, its structure and its semantics. With language the possible execution sequences of workflow represented as an automaton is meant. Structure means the structural representation of an automaton comparable to a directed graph. The semantics

Table I  
EXAMPLES OF USED PIPS

Code	Label	Name
PIP3A4	p and p'	Request purchase order (PO)
PIP3A9	c and c'	Request PO cancellation
PIP3B2	n	Notify of advance shipment
PIP3C3	i	Notify of invoice
PIP3C6	r	Notify of remittance advice

of the used transition labels determines the semantics of the complete workflow.

Since there is no clear understanding on how the different aspects depend on each other a set of hypotheses has been set up. Semantics is considered implicitly in all questions of the questionnaire by using semantically meaningful workflows. In particular, RosettaNet Partner Interface Processes (or PIPs) [33] are used as transition labels. Examples of the used labels and brief descriptions of their semantics are given in Table I. Since some of the PIPs are covering two messages which are usually request and response messages these messages are labeled without and with prime respectively.

The example automaton depicted on the left hand side of Fig 1 uses the labels described in Table I. This workflow starts with a request for a purchase order (transition labeled  $p$ ), followed by an acceptance of the purchase order ( $p'$ ). Then, an invoice for this specific order ( $i$ ) is sent. The customer can now choose to pay the order ( $r$ ), after which the order is shipped ( $n$ ), or to send a cancellation request ( $c$ ) followed by a cancellation confirmation ( $c'$ ).

Based on the hypotheses (see [15] for details) the questions of the questionnaire are constructed in such a way that the intended decision criteria on ordering the results gives some indication on the validity of the hypothesis. Each question contains a reference automaton and a set of either three or four solution automata ( $A, B, C, D$ ). A respondent has to order the solution automata by similarity with respect to the reference automaton. If a respondent finds multiple solution automata equally similar to the reference automaton, she can assign several automata to the same position of the order. Respondents are also asked to state their reason on how they derived the provided order. An example question is depicted in Fig 1. The questionnaire is available at [18].

#### C. Results

The pilot study has been based on a group of 27 international technical workflow specialists from which 12 responded from seven different countries. The respondents have different backgrounds and different areas of expertise, like e.g. inter-organizational workflows, workflow matchmaking, or semantic service composition.

The questions are analyzed by having a look at the number of supporters of a hypothesis and the maximum number of equal answers. Due to the small number of respondents in

Question 3				
Reference	A	B	C	D
Order	1	2	3	4
Reason				

Figure 1. Question 3

the pilot study, only those questions with a strong support can be considered for the evaluation of the similarity measures. There are hypotheses with a strong support stating that the language is more important than the structure on different levels of granularity (the corresponding questions are Q1, Q3, Q5, Q6, Q13, and Q17). The question Q14 indicates that super-automata are considered more similar than automata with extra transitions before or within the paths of the reference automaton. The questions Q16 and Q23 have quite some supporters. The underlying hypothesis states that an automaton having a transition as a loop is more similar than a comparable automaton not having the transition at all.

In case of the remaining hypotheses and questions the number of supporters and opponents is quite high and therefore they will not be considered for the evaluation of the measures introduced in this paper. The number of respondents supporting each question are depicted in result figures of all approaches as reference.

The methods presented in this paper differ from the previous measures by considering structure of the workflows, thus, are not purely language based. In Section V a measure based on edit distance of workflow structures is presented, while Section VI introduces hierarchical Finite State Automata called Nested Word Automata. As a reference the best unstructured approach [12] based on workflow state sets (Sec IV) is repeated from [12]. The lessons learned from these three approaches result in an extension of workflow state sets considering structure (Sec VII). This evaluation is preliminary due to the limited number of participants and focus on aspects relevant to a similarity measure for service discovery, where e.g. additional information on occurrences of execution sequences are usually not available. The different measures can be evaluated in other scenarios with different results.

#### IV. WORKFLOW STATE SET BASED SIMILARITY MEASURE

##### A. Approach

This is a recap of the measure proposed in [12], which has been the best approach from the previous study and which

is extended in Sec VII. The approach is based on the Levenshtein or edit distance [20], which specifies the smallest number of substitutions, insertions, and deletions of symbols to transform one string into another. For example, the edit distance of the strings *abab* and *aab* is one by removing the first occurrence of *b*. Based on this distance value  $d$  the similarity value *sim* can be calculated by subtracting the distance value  $d$  from the maximum difference  $m$ , and dividing the difference by the maximum difference  $m$ , that is,  $sim := \frac{m-d}{m}$ .

The distance of automata can be calculated based on their language representation as long as the languages and words are finite. Since infinite strings are constructed from a finite automaton, in [34] it has been proposed to represent an automaton based on its finite set of states. States are represented as  $n$ -grams<sup>2</sup>, i.e., a sequence of  $n$  transition labels ending in the state. Start and terminal state require special representations using special characters  $\$$  and  $\#$ . For example in the reference automaton in Fig 1 the target state of transition with label  $r$  is represented as a 2-gram by  $ir$ , and as a 4-gram by  $pp'ir$ . Details on how to construct  $n$ -grams can be found in [34].

An automaton is represented by a set of  $n$ -grams. A combination of  $n$ -grams can be used to construct all possible execution sequences of a single automaton and thus has a strong relation to the language accepted by the automaton.

Using this automaton representation it is quite obvious that there exist a lot of different automata resulting in the same representation, thus, the ambiguity of the representation depends on the value of  $n$  in the  $n$ -gram.

##### B. Evaluation

The evaluation of  $n$ -gram sets is based on the similarity derived from the edit distance between the  $n$ -gram sets of the reference and the solution automaton respectively. The edit distance of two  $n$ -gram sets is calculated by summing up the minimum edit distance of each  $n$ -gram within the first set and an  $n$ -gram in the second set, added to the sum of the minimum edit distance of each  $n$ -gram within the second set and an  $n$ -gram in the first set. This can be formally described for two  $n$ -gram sets  $A := \{a_1, \dots, a_l\}$  and  $B := \{b_1, \dots, b_k\}$  as

$$d(A, B) := \sum_{i=1}^l \left( \min_{j=1..k} d(a_i, b_j) \right) + \sum_{j=1}^k \left( \min_{i=1..l} d(a_i, b_j) \right)$$

where  $d(a_i, b_j)$  is the edit distance of the two  $n$ -grams  $a_i$  and  $b_j$  which are considered as strings. Be aware that an  $n$ -gram is a sequence of transition labels, where each label is treated as a unique token, that is, a character in terms of a string. The maximum distance between the sets of  $n$ -grams is twice the product of the maximum number of  $n$ -grams

<sup>2</sup> $N$ -grams are known from substring matching in full-text indexes [35].

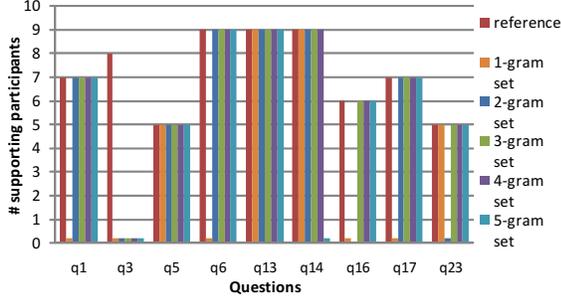


Figure 2. Pilot results and n-gram set results

contained in one of the sets and the length  $n$  of the n-gram, that is,  $m = 2 * \text{Max}(|A|, |B|) * n$ , where  $|A|$  specifies the size of the set of n-grams  $A$ .

This distance definition is applied to each solution automata and the reference automaton. Further, the maximum distance is calculated per reference and solution automaton pair since it is needed to calculate the similarity measure. The derived values are collected and the corresponding order of the solution automata is derived. The resulting order is compared with the results from the pilot study and the results are depicted in Fig 2. In particular, values from one to five for  $n$  have been used.

The results are promising. It turns out that question 3 has no support independent of the amount of context information taken into account. The reason for this is that the n-gram set approach generates quite high distance values in case an automaton contains cycles. In particular, question 3 (see Fig 1) has two solution automata containing a lot of cycles, which result in a wrong order of the solution automata. However, the other questions are supported for at least one specific  $n$ . It can be observed that e.g. for question 1 the context information for  $n$  equals one is not sufficient to generate the correct order, while the context information bigger than one is sufficient. Further, it can be observed at question 14 that the correct order is determined as long as not too much context information is considered. In particular, in case of  $n$  equals five the correct order can no longer be derived. Again this is due to the non-proportional increase of distance values due to cycles in solution automata. A curiosity can be observed at question 23. Here the correct order is determined for all n-gram sets except for  $n$  equals two. The explanation for this is that the differences between the similarity values of the corresponding solution automata are quite small. In particular, for  $n$  equals two the similarity values for two solution automata get equal under the considered precision which results in the wrong order of solution automata. For this particular data set, the best result is achieved for  $n$  equals four and the worst result for  $n$  equals one. However, the usage of  $n$  equals one or two in general is really unlikely due to its high ambiguity. The results for  $n$  equals three and

five provide eight and seven supporting results respectively. In future work we will investigate how to determine a good estimate for the  $n$  and in Sec VII we propose an approach to decrease the dominance of cycles on the similarity value. As a reference in the following approaches we add 4-gram set results to the graphs to indicate the base line for the evaluation.

## V. STRUCTURE BASED SIMILARITY MEASURE

### A. Approach

In the last years we studied change operations in block structured workflows to mine optimal variants of workflows from a set of alternative workflows. The number of change operations is called the distance between two workflows. The distance defines the similarity by the maximum distance minus the actual distance over the maximum distance.

The approaches we investigated are based on block structured workflows like for example BPEL workflows [17]. A block structured workflow consists of activities and control structures connected with directed edges. Sequences, branchings, and loops are represented as blocks with well-defined start and end nodes. Transitions in an automaton represent message exchanges, which correspond to activities. Control structures are e.g. XOR split and joins, and loops, which are implicitly represented in automata and explicated in block structured workflows. The creation of block structured workflows from automata has been done manually with the premise that the block structured process model shows the same behavior as the original automaton. In the context of this research, we consider two process models to have the same behavior if they are trace equivalent, i.e., their language of valid and complete execution sequences are identical [36]. For example consider Fig 3, where the automaton is depicted on the left hand side and the block structured workflow model on the right hand side. The loop structure shown by transitions  $i$  and  $j$  in automaton  $S$  are represented by constructs like sequence (control flow), loop and XOR in its corresponding process model  $S'$ <sup>3</sup>. Though they are structurally different, their behavior are completely the same.

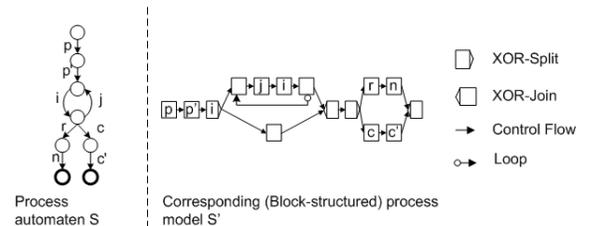


Figure 3. Transformation FSA to Block Structure

<sup>3</sup>Such transformation can also be seen when transforming a BPMN model into BPEL.

The distance between two block structured workflow models is defined as the minimal number of move, insert, and delete operations on activities to transform one workflow into another workflow. To calculate the distance [17], we first compute the order relations (e.g., successor, predecessor, XOR or AND) between each pair of activities and then identify the activity pairs which have different order relations in the two models. These differences can be represented as a logic expression so that we can optimize it to find the minimal number of change operations to transform one workflow model into another. For example consider two workflow models  $A \rightarrow B \rightarrow C$  and  $B \rightarrow C \rightarrow A$ . The order relations between activity A and B and activity A and C are different in the two models (A is a predecessor of activities B and C in the first model while it is a successor of them in the second model). We can then represent the differences using logic expression  $\overline{AB} + \overline{AC}$ . After optimize this logic expression (e.g., by Quine-McCluskey algorithm [37]), we obtain  $\overline{A}$ . This means that we only need to perform one change operation (i.e., to move activity A from the beginning to the end) to transform the first model into the second one. Consequently, the distance between the two models is one.

The similarity between two workflow models is calculated based on their activity sets. The number of disjoint activities specifies the maximum distance between two workflows. Thus the similarity is the difference of maximum distance and actual distance over the maximum distance. Details on how to compute distances and similarities can be found in [17].

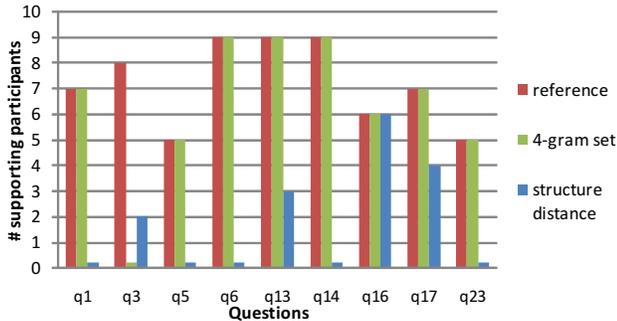


Figure 4. Pilot results, 4-gram set, and structure based similarity results

### B. Evaluation

The results of applying the approach on the survey questions as well as the survey reference and the 4-gram set results are depicted in Fig 4. The results are not good. Classical language based approaches perform quite well on the first questions and are often bad when loops are involved. It turns out that the structure based similarity approach is working quite well with loops (questions q16 and q23) compared to language based approaches. However,

the overall performance of the approach does not provide many benefits. This observations go along with the results in [13] indicating that graph based matching approaches are too limited. The limitations comes from the fact that language equivalence can not be determined by comparing workflow structures. However, the fact that loops are dealt better with indicates that considering structure can help to compensate the effect of loops on language based similarity.

## VI. LANGUAGE BASED HIGH LEVEL CHANGES

### A. Approach

Another approach of adding structure to an automaton is structured languages resulting in Nested Word Automata [38]. Nested Word Automata (NWA) are hierarchical Finite State Automata where each hierarchy level describes a language on its own. As a consequence, BPEL can be represented as NWA [16]. The advantage of this formalism is that the available structure is explicated, while the language concepts is still maintained. The explication of structure allows to define higher level change operations on language elements. In particular, change operations like copy, move, or delete operations defined on subtrees in the hierarchical structure. Further, the explication of structure semantics (XOR splits and loops) allows to introduce a higher level change operation on the semantics of these structure elements. The aim is to reduce the overhead of these change operations on the language level.

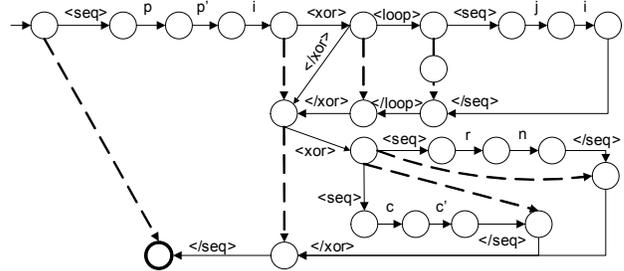


Figure 5. Transformation FSA to Nested Word Automaton

Similar to Sec V the NWA are manually constructed to be trace equivalent, i.e., language equivalent, to the automata used in the survey. An example NWA for the automaton used on the left hand side of Fig 3 is depicted in Fig 5. The symbols have the same semantics as automata (see Sec III-A). Changes on the hierarchy level are indicated by dotted arrows, where the source of the arrow is the call/opening tag and the target of the dotted arrow is return/closing tag of the structure element.

The applied approach can best be explained on the generic structure of the example NWA in Fig 5 depicted in Fig 6. The round nodes are structure nodes while the square nodes are word fragments. The empty square node represents an empty fragment. Structure nodes are associated with a

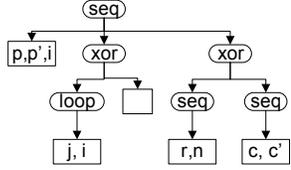


Figure 6. Generic Structure of Nested Word Automaton in Fig 5

word derived from a deep traversal of the child nodes and a concatenation of the fragments in the child nodes. E.g. the word associated to the right XOR node in Fig 6 is  $r,n,c,c'$ , which is considered a fragment again.

The similarity of two NWA is calculated in two steps: first all nodes are associated with the best matching, i.e., minimal edit distance, fragment of the other NWA. Then the structure is evaluated: for each structure node  $n$  it is checked whether the sum of the child node distances  $d(c)$  is greater the edit distance of the associated fragment derived from node  $n'$  ( $d(n.getWord(), n'.getWord())$ ). In case the type of associated nodes differ a *penalty* of one additional change operation is added for changing the node. For all  $n \in NWA_1$

$$d(n) = \min\left(\sum_{c \in n.getChild()} d(c) + \text{penalty}, \min_{n' \in NWA_2} d(n.getWord(), n'.getWord())\right)$$

Based on this algorithm, the optimization propagates bottom up to the root node, providing the final distance. The similarity is then calculated by the difference of the maximum distance and the actual distance over the maximum distance. The maximum distance is the sum of the product of number of nodes and the maximum length of a fragment per NWA.

### B. Evaluation

The results of the approach with regard to the survey are depicted in Fig 7. Further, the reference of the survey and the base line of the 4-gram set has been added. The results are better than for the structure based similarity, but they are still quite off from the reference and the 4-gram set. It turns out that the language related questions q1, q5, and q6 are performing really bad. The questions containing automata with loops perform better (q3 and q16). The conclusion is that high level change operations have a benefit when dealing with more complex structures. This approach benefits in particular from the notion of re-use of fragments.

## VII. WORKFLOW STATE MULTISSET BASED SIMILARITY MEASURE

### A. Approach

As a conclusion from all previous experiments, it turns out that language similarity is very important for most questions. Further, more complex structures like loops are overly represented in n-grams and are better dealt with in structure based approaches. Based on these two observations we propose to reduce the influence of loops on n-grams by

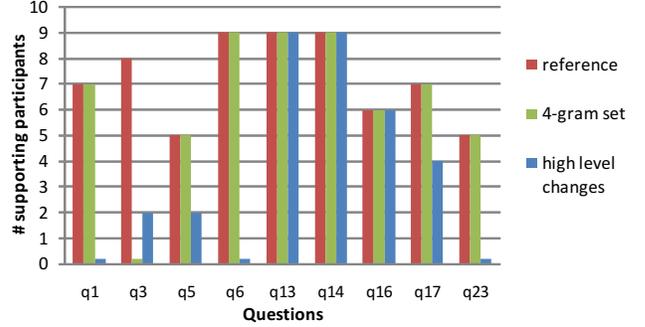


Figure 7. Pilot results, 4-gram set and high level changes similarity results

strengthening the influence of non loop related n-grams. In particular, we propose to change the set of n-grams into a multiset of n-grams. A multiset is a set where each element is associated with a number indicating the coefficient of an element, i.e., how often the element has been added to the set. Multisets are formally defined in [39] and are e.g. used in the formal definition of Petri Nets. Coefficients can be used in the distance calculation of n-gram multisets as a factor on each distance value of an n-gram contained in the set, thus for a coefficient  $m(a)$  for an element  $a \in A$  the following distance can be defined

$$d(A, B) := \sum_{i=1}^l \left( m(a_i) * \min_{j=1..k} d(a_i, b_j) \right) + \sum_{j=1}^k \left( m(b_j) * \min_{i=1..l} d(a_i, b_j) \right)$$

Based on the distance the similarity is calculated by approximating the maximum distance as a product of the maximum number of n-grams, the maximum length of an n-gram and the maximum coefficient of an n-gram in both multisets. Similarity is then the difference of the maximum distance and the actual distance over the maximum distance.

### B. Evaluation

The results of this approach are depicted in Fig 8. Again, the reference and the 4-gram set results are included too. The n-gram multiset approach works better on loops than the 4-gram set approach as indicated by question 3. The n-gram multiset results differ in question 3 since small differences in similarity values change the order and therefore produce a different supporting number of respondents. The difference is about 5% of the similarity value. Similar to n-gram sets the result of the approach depends on the selection of a proper value for  $n$ . If the value is too small then specific effects are not observable like e.g. for 2-gram multisets and question 23, while a too big  $n$  introduces artifacts obstructing the result like e.g. for 5-gram multisets and question 14. In future work we will investigate how to determine a good estimate for the value or  $n$ .

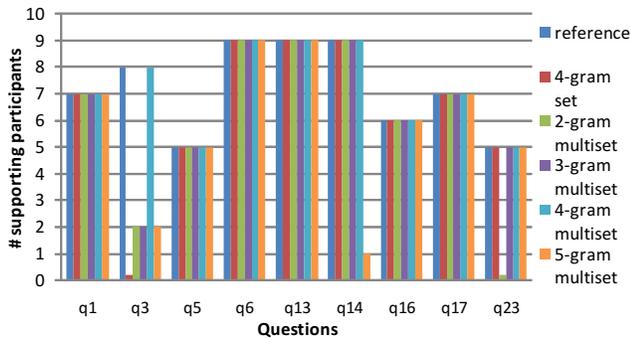


Figure 8. Pilot results, 4-gram set and n-gram multiset results

## VIII. CONCLUSION AND FUTURE WORK

Service discovery of statefull services requires to search for services based on their choreography, that is, their workflow, and present the most significant results in an ordered list to the human user. A measure of significance is the similarity of a workflow describing a service and the query represented by a workflow again. Therefore different similarity measures have been summarized and have been evaluated with regard to a pilot study on the human understanding of workflow similarity.

As a conclusion of this specific service discovery evaluation it turns out that the very simple approach based on n-gram sets delivers better results than the more complex structure based similarity measure or the language based high level changes. The best results however, can be accomplished by compensating the over representation of complex structures like loops by introducing n-gram multisets. This is supported by the results of the pilot study. The variability of n-gram multisets and as a consequence the determination of a good value for  $n$  is future work. Further, the empirical study has to be conducted with a bigger number of participants.

## REFERENCES

- [1] F. DeRemer and H. H. Kron, "Programming-in-the-large versus programming-in-the-small," *IEEE Transactions on Software Engineering*, vol. 2, pp. 80–86, 1976.
- [2] W. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey," in *Proceedings International Conference Business Process Management (BPM)*, ser. Lecture Notes in Computer Science, W. van der Aalst, A. H. M. ter Hofstede, and M. Weske, Eds., vol. 2678. Springer, 2003, pp. 1–12. [Online]. Available: <http://link.springer.de/link/service/series/0558/bibs/2678/26780001.htm>
- [3] M. P. Singh, A. K. Chopra, N. Desai, and A. U. Mallya, "Protocols for processes: programming in the large for open systems," *SIGPLAN Notices*, vol. 39, no. 12, pp. 73–83, 2004. [Online]. Available: <http://doi.acm.org/10.1145/1052883.1052893>
- [4] C. Molina-Jimenez, S. Shrivastava, E. Solaiman, and J. Warne, "Contract representation for run-time monitoring and enforcement," in *Proceedings of Conference on Electronic Commerce (CEC)*. IEEE, 2003, pp. 103–110.
- [5] M. Mecella, B. Pernici, and P. Craca, "Compatibility of e-services in a cooperative multi-platform environment," in *Proceedings of 2nd International Workshop on Technologies for E-Services (TES)*, F. Casati, D. Georgakopoulos, and M. Shan, Eds. Springer LNCS 2193, 2001, pp. 44–57.
- [6] E. Folmer and D. Krukkert, "openXchange as ebXML implementation and validation; the first results," in *Proceeding of XML Europe 2003 Conference & Exposition*, May 2003.
- [7] D. Krukkert, "Matchmaking of ebXML business processes," openXchange Project, Tech. Rep. IST-28584-OX\_D2.3\_v.2.0, Oct 2003.
- [8] W. van der Aalst, "Interorganizational workflows: An approach based on message sequence charts and petri nets," *Systems Analysis - Modelling - Simulation*, vol. 34, no. 3, pp. 335–367, 1999.
- [9] E. Kindler, A. Martens, and W. Reisig, "Inter-operability of workflow applications: Local criteria for global soundness," in *Business Process Management, Models, Techniques, and Empirical Studies*. Springer-Verlag, 2000, pp. 235–253.
- [10] X. Fu, T. Bultan, and J. Su, "Realizability of conversation protocols with message contents," in *Proceedings IEEE International Conference on Web Services (ICWS)*. IEEE Computer Society, 2004, pp. 96–103.
- [11] A. Wombacher, B. Mahleko, and E. Neuhold, "IPSI-PF: a business process matchmaking engine based on annotated finite state automata," *Journal on Information Systems and E-Business Management*, vol. 3, no. 2, pp. 127–150, 2005.
- [12] A. Wombacher, "Evaluation of technical measures for workflow similarity based on a pilot study," in *OTM Conferences (1)*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 4275. Springer, 2006, pp. 255–272. [Online]. Available: [http://dx.doi.org/10.1007/11914853\\_16](http://dx.doi.org/10.1007/11914853_16)
- [13] A. Wombacher and M. Rozie, "Evaluation of workflow similarity measures in service discovery," in *Tagungsband der Multikonferenz Wirtschaftsinformatik - Service Oriented E-Commerce Track (MKWI)*, ser. Lecture Notes in Informatics (LNI), vol. P-80. Gesellschaft fuer Informatik, 2006, pp. 57–72.
- [14] A. Rozinat and W. van der Aalst, "Conformance testing: Measuring the fit and appropriateness of event logs and process models," in *Business Process Management Workshops*, 2005, pp. 163–176. [Online]. Available: [http://dx.doi.org/10.1007/11678564\\_15](http://dx.doi.org/10.1007/11678564_15)
- [15] A. Wombacher and M. Rozie, "Piloting an empirical study on measures for workflow similarity," in *accepted at IEEE International Conference on Services Computing (SCC)*, 2006.
- [16] A. Wombacher, "Alignment of choreography changes in BPEL processes," in *IEEE Intl Conf on Services Computing, SCC*, sept. 2009, pp. 1–8.

- [17] C. Li, M. Reichert, and A. Wombacher, "On measuring process model similarity based on high-level change operations," in *ER*, ser. Lecture Notes in Computer Science, Q. Li, S. Spaccapietra, E. S. K. Yu, and A. Olivé, Eds., vol. 5231. Springer, 2008, pp. 248–264. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-87877-3\\_19](http://dx.doi.org/10.1007/978-3-540-87877-3_19)
- [18] M. Rozie and A. Wombacher, "Questionnaire of the empirical workflow similarity study," [http://www.cs.utwente.nl/~wombachera/papers/questionnaire\\_v1.0.zip](http://www.cs.utwente.nl/~wombachera/papers/questionnaire_v1.0.zip), 2005.
- [19] H. Tzschach and G. Hasslinger, *Codes fuer den stoerungssicheren Datentransfer*. Oldenburg Verlag, 1993.
- [20] L. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
- [21] M. Mohri, "Edit-distance of weighted automata: General definitions and algorithms," *International Journal of Foundations of Computer Science*, vol. 14, no. 6, pp. 957–982, 2003.
- [22] G. Chartrand, G. Kubicki, and M. Schultz, "Graph similarity and distance in graphs," *Aequationes Mathematicae*, vol. 55, pp. 129–145, 1998.
- [23] B. T. Messmer and H. Bunke, "A new algorithm for error-tolerant subgraph isomorphism detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 493–504, 1998.
- [24] Z. Du, J. Huai, Y. Liu, C. Hu, and L. Lei, "IPR: Automated interaction process reconciliation," in *Proceedings of IEEE/ACM International Conference on Web Intelligence (WI)*, 2005, accepted for publication.
- [25] W. van der Aalst and T. Basten, "Inheritance of workflows: an approach to tackling problems related to change," *Theor. Comput. Sci.*, vol. 270, no. 1-2, pp. 125–203, uary.
- [26] F. van Breugel, "A behavioural pseudometric for metric labelled transition systems," in *Proceedings 16th International Conference on Concurrency Theory (CONCUR)*, ser. Lecture Notes in Computer Science, M. Abadi and L. de Alfaro, Eds., vol. 3653. Springer, 2005, pp. 141–155. [Online]. Available: [http://dx.doi.org/10.1007/11539452\\_14](http://dx.doi.org/10.1007/11539452_14)
- [27] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled markov processes," *Theoretical Computer Science*, vol. 318, 2004.
- [28] B. F. van Dongen, R. M. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *CAiSE*, ser. Lecture Notes in Computer Science, Z. Bellahsene and M. Léonard, Eds., vol. 5074. Springer, 2008, pp. 450–464. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-69534-9\\_34](http://dx.doi.org/10.1007/978-3-540-69534-9_34)
- [29] M. Dumas, L. García-Bañuelos, and R. M. Dijkman, "Similarity search of business process models," *IEEE Data Eng. Bull.*, vol. 32, no. 3, pp. 23–28, 2009. [Online]. Available: <http://sites.computer.org/debull/A09sept/marlon.pdf>
- [30] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.
- [31] A. Wombacher, P. Fankhauser, and E. Neuhold, "Transforming BPEL into annotated deterministic finite state automata enabling process annotated service discovery," in *Proceedings of International Conference on Web Services (ICWS)*, 2004, pp. 316–323.
- [32] A. Wombacher, P. Fankhauser, B. Mahleko, and E. Neuhold, "Matchmaking for business processes based on choreographies," in *Proceedings of International Conference on e-Technology, e-Commerce and e-Service (EEE-04)*. IEEE Computer Society, 2004.
- [33] RosettaNet, "RosettaNet home page," <http://www.rosettanet.org>, 2004.
- [34] B. Mahleko, A. Wombacher, and P. Fankhauser, "A grammar-based index for matching business processes," in *Proceedings of IEEE International Conference on Web Services (ICWS)*. IEEE Computer Society, 2005, pp. 21–30.
- [35] R. A. Baeza-Yates, "Text retrieval: theory and practice," in *Proceedings of the 12th IFIP World Computer Congress*, J. van Leeuwen, Ed. Madrid, Spain: North-Holland, 1992, pp. 465–476.
- [36] J. Hidders, M. Dumas, W. van der Aalst, A. ter Hofstede, and J. Verelst, "When are two workflows the same?" in *CATS '05*, 2005, pp. 3–11.
- [37] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*. McGraw-Hill, 2003.
- [38] R. Alur and P. Madhusudan, "Adding nesting structure to words," *Journal of the ACM*, p. 45, 2009.
- [39] K. Jensen, *Coloured Petri Nets*. Heidelberg: Springer Verlag, 1992.