Query Model and Similarity-Based Retrieval for Workflow Reuse in the Digital Humanities

Lukas Malburg, Nicolas Münster, Christian Zeyen and Ralph Bergmann

Business Information Systems II
University of Trier
54286 Trier, Germany
[s4lumalb][s4nimuen][zeyen][bergmann]@uni-trier.de,
http://www.wi2.uni-trier.de

Abstract. Scientific Workflows do not seem to be broadly used today in the Digital Humanities to perform text and data analysis. Although they have become established in e-Science, modeling new workflows is usually a demanding task, especially for novice users. *Case-Based Reasoning (CBR)* has been applied in the past to support the development of workflows as an experience-based activity by retrieving past workflows. A query language is needed for this purpose, but current languages do not sufficiently consider different user groups and the information they can provide. To address this issue, we present a query model to support novice as well as experienced users. We identify common expression elements from literature and integrate them in a prototypical CBR application named *Reuse Assistant* to support workflow reuse in the RapidMiner workflow tool. An experimental evaluation with non-expert users indicates the potential of the *Reuse Assistant* to facilitate workflow reuse and thus to simplify workflow development.

Keywords: Case-Based Reasoning · Workflow Reuse · Scientific Workflows · RapidMiner · Digital Humanities

1 Introduction

In recent years, a wide range of research tools have been emerged from various text-oriented *Digital Humanities (DH)* projects, which can be seen in the advent of tool and method collections such as TAPoR and Methodica[1]. The obvious goal of reusing such artifacts is not trivial to achieve since new research questions usually require non-trivial and time-consuming adjustments or combinations of available tools [13]. Thus, modularization and reuse is a topic of interest in DH for further research. In e-Science, *Scientific Workflow Management Systems (SciWFM)* are widely used to create and execute workflows for data analysis. The development of scientific workflows, however, can be a demanding and time-consuming task. This applies in particular for complex data analysis that involve large amounts of data and require complex combinations

[1] See http://tapor.ca and http://methodi.ca

of processing steps [3,15,16,17,21]. Consequently, sharing and reusing scientific workflows is an important topic of research and various approaches have been presented to support workflow reuse by searching available workflows [2,11,22]. Typically, these approaches require the user to specify the properties of a desired workflow in a query. However, many query languages are composed of expression elements that are difficult to understand for unexperienced users, thus restricting their ability to find appropriate workflows [3].

Most recently, the benefits of using scientific workflows have also been recognized in the context of DH [14]. Especially in text-oriented DH projects, scientists can make use of numerous, well-established methods from the fields *Natural Language Processing (NLP)* as well as Data and Text Mining. However, the usage of scientific workflows in DH is still in its infancy. We assume that the more a research domain is related to computer science, the easier it will be for the respective scholar to get familiar with workflow modeling. Thus, a particular challenge to support users in workflow reuse is posed by interdisciplinary DH projects that, for instance, involve computer scientists and humanities scholars. Among other aspects, we investigate this issue in the context of the *eXplore!*[2] project. In particular, we prove the practical application of the RapidMiner[3] workflow tool for text analysis in the DH. A goal of this project is to accompany the workflow creation and to develop a workflow modeling assistance to support researchers in reusing past workflows.

This work is meant to be a first step towards the development of an assistance by adapting and extending our past works on *Process-Oriented Case-Based Reasoning (POCBR)* to this new domain. In a nutshell, POCBR [1,18] integrates CBR with process-oriented information systems and supports the development of workflows as an experience-based activity. Experiential knowledge is represented in form of workflows and can be reused for similar problem situations. POCBR does not require that a given user query matches exactly a workflow in the case base. Instead, the most similar workflows are retrieved assuming they can serve as a basis for creating a new one [19,23]. In this work, we address the question of which expression elements should be available in a query language to support users with different knowledge and experience in retrieving and reusing scientific workflows. In a literature study, we identify expression elements that are integrated in a new query model. The new query model is implemented in a prototype named *Reuse Assistant* to support the reuse of RapidMiner workflows. An experimental evaluation with users indicates the potential of the approach to facilitate the reuse of workflows.

In the following, section 2 introduces different user groups working with SciWFM as well as current assistance tools and their limitations. Section 3 presents our approach for scientific workflow reuse by means of case-based reasoning and section 4 describes the experimental evaluation while section 5 gives a conclusion and discusses future work.

[2] *eXplore!* is a cooperation project launched in 2016 with the Trier Center for Digital Humanities (TCDH) at the University of Trier.

[3] https://www.rapidminer.com/

2 Scientific Workflows in the Digital Humanities

User groups with varying experience in modeling workflows provide different information when they search for desired workflows. Cohen-Boulakia and Leser [3] distinguish between *"true users"* and *"power users"*. The former are domain experts respectively domain scientists who constitute the largest group. Their aim is primarily to analyze scientific data with SciWFM. They usually have no broad experience in developing new workflows or analysis methods. Thus, *true users* search workflows by specifying queries in a lower level of detail using keywords or descriptions of the available input data or the desired output. We assume that most DH scientists without an educational background in informatics rather belong to this group since they are not used to develop new workflows to perform text and data analysis. In contrast, *power users* are workflow developers who know how to create new workflows with SciWFM. Based on their broad experience they provide significantly more information and perhaps describe the desired workflow, e.g. the topology in full or in part. Consequently, there should be a continuum of query languages to express the properties of a desired workflow: On the one side, purely syntactical query languages enable to search for simple keywords in a workflow description. These languages are mainly used by *true users* in workflow repositories such as myExperiment[4]. On the other side, more expressive query languages enable users to specify the topology of a workflow or to search for keywords in descriptions of tasks or within the entire workflow. Such query languages are presumably more suitable for *power users*. Cohen-Boulakia and Leser [3] emphasize that query languages targeting both user groups are largely unexplored.

To the best of our knowledge, scientific workflows and SciWFM are not broadly used in the DH. In contrast, in other domains such as e-Science, numerous SciWFM such as *KEPLER* [16] are used to support analyses. Some of the recent approaches in DH are tailored to certain application areas. For instance, the CLARIN research infrastructure provides WebLicht[5], a web-based tool for the composition of web services for tasks in NLP. WebLicht supports users in creating analysis pipelines by only presenting the user web services that are compatible in terms of input and output data constraints. However, only very few example workflows are available and no reuse assistance is provided.

A few approaches exist for SciWFM that provide an assistance for workflow modeling based on available workflows. For instance, RapidMiner includes the recommender system *Wisdom of Crowds (WoC)* [12] that suggests processing steps and parameter settings. For this purpose, WoC stores and analyzes the workflows created by users worldwide using machine learning techniques. Based on these workflows, context-sensitive recommendations are continuously displayed to users while they are modeling a workflow. WoC works fully automatic and does not provide a query interface for users. Hence, it might be too restrictive for more experienced users. Moreover, it suggests single processing

[4] https://www.myexperiment.org/workflows

[5] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php

steps but no adaptations for the workflow under construction. Unfortunately, the authors did not yet perform an evaluation that differentiates between novice and expert users. The *WINGS* [6] SciWFM supports the generation of workflows with an planning and semantic reasoning approach. Before *WINGS* generates workflows, users create workflow templates by selecting components from catalogs in an editor and specify further properties or constraints using RDF triples. It is also possible to reuse templates from other users. However, the large number of available workflow components for creating workflow templates and in particular the use of RDF triples can be demanding, especially for *true users*. Chinthaka et al. [2] present a generic CBR approach to assist users in reusing scientific workflows. They use a keyword search and information about the workflow structure to retrieve appropriate workflows. However, the keyword search just handles inputs as well as outputs from workflows and does not consider other metadata.

3 Scientific Workflow Reuse by Case-Based Reasoning

We now describe our approach to support workflow reuse by means of POCBR. This work extends our previous works on POCBR [1,19,23] by augmenting the available query language with expression elements derived from literature.

3.1 Similarity-Based Retrieval of Workflows

A case in POCBR is usually a workflow that expresses experiential knowledge. The approach is based on a workflow representation that uses semantically labeled directed graphs named NEST graphs [1]. Figure 1 illustrates the NEST graph of a data mining workflow created with RapidMiner for learning a Naive

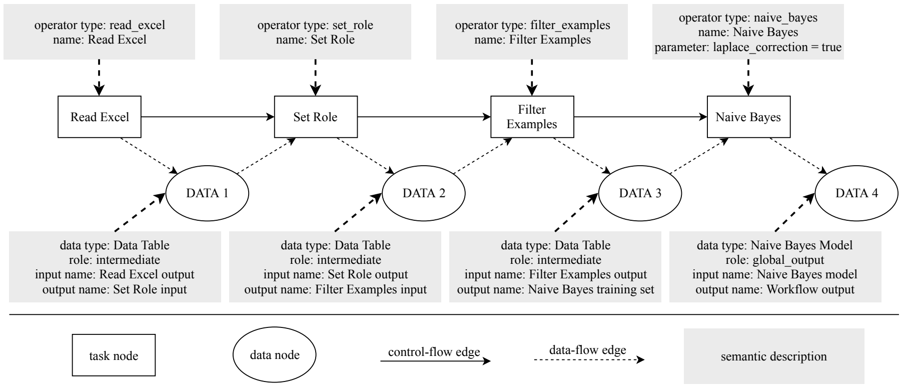
operator type: read_excel
name: Read Excel
operator type: set_role
name: Set Role
operator type: filter_examples
name: Filter Examples
operator type: naive_bayes
name: Naive Bayes
parameter: laplace_correction = true
Read Excel
Set Role
Filter Examples
Naive Bayes
DATA 1
DATA 2
DATA 3
DATA 4
data type: Data Table
role: intermediate
input name: Read Excel output
output name: Set Role input
data type: Data Table
role: intermediate
input name: Set Role output
output name: Filter Examples input
data type: Data Table
role: intermediate
input name: Filter Examples output
output name: Naive Bayes training set
data type: Naive Bayes Model
role: global_output
input name: Naive Bayes model
output name: Workflow output
task node
data node
control-flow edge
data-flow edge
semantic description

Fig. 1. NEST Graph Representation of a Data Mining Workflow

Bayes classifier on a given Excel data file. A NEST graph consists of a set of nodes (N) and edges (E) between nodes. Semantic descriptions (S) are domain-dependent descriptions (key-value pairs) of nodes or edges. Additionally, each node and edge has a specific type (T), e.g. task and data nodes as well as control-flow and data-flow edges.

3.2 Expression Elements for Querying Scientific Workflows

To determine which expression elements are suitable to support the retrieval and reuse of scientific workflows, we first carried out a literature study. Table 1 presents a selection of expression elements and papers from our literature study. According to the categorization by Goderis et al. [9], expression elements are grouped into "workflow structure" that define the topology of a workflow. Elements that relate to the entire workflow and its properties are grouped into "workflow signature". In the following, the term *metadata* refers to the workflow signature. To support both user groups in retrieving and reusing scientific workflows, elements from the workflow structure are presumably more suitable for *power users* while elements from the workflow signature are presumably more suitable for *true users*. The expression elements marked with a star are implemented in the *Reuse Assistant* to support the retrieval of RapidMiner workflows. The papers are organized into *groups (G)* according to the researchers: G_1 includes the authors of Goderis et al. [7,8,9,10], G_2 the researchers of Gil et al. [4,5] and G_3 the authors of Starlinger, Cohen-Boulakia, and Leser [3,21]. Papers that cannot be assigned to G_1, G_2, and G_3 are combined in the group *Others* [16,22]. This classification takes into account that research groups often discuss the same expression elements.

Table 1. A Selection of Derived and Classified Expression Elements

		G_1	G_2	G_3	Others
Workflow Structure	Loops*	[7,9,10]	[5]	[3,21]	[22]
	Conditionals*	[7,9,10]	[5]	[3,21]	[22]
	Data-flow*	[7,9]	[5]	[3,21]	[22]
	Serviceflow / Control-flow*	[7,9]	[5]	[3,21]	[22]
	Operators*	[9]	—	[21]	[22]
	Generalized operators*	—	[4]	[3]	—
	Restrictions of the topology	—	—	[3]	—
Workflow Signature	Input- and Output description	[7,9,10]	[4,5]	[3,21]	[22]
	Workflow description*	[8,10]	—	[3,21]	[22]
	Keywords / Tags*	[8]	—	[3,21]	[22]
	Reliability	[7,8,9,10]	—	—	[16]
	Author	[8,9]	[5]	[21]	—
	Rating by Community	[8]	[5]	[21]	—
	Versioning	[9,10]	—	[3]	—
	Title of the workflow*	[8,10]	—	[21]	—

3.3 Query Model and Case Workflow Representation

Our query model extends the *Query Language for POCBR (POQL)* by Müller and Bergmann [19]. To support both user groups in retrieving and reusing scientific workflows, we have integrated several expression elements from the literature study (see table 1). POQL is an expressive query language that allows defining a *desired workflow (DW)* and one or more *restriction workflows* $(RW_1, ..., RW_n)$ with undesired elements. Therefore, it is possible to express constraints on the graph structure [19]. Current research literature [3] states that existing approaches for retrieving scientific workflows do not allow users to express arbitrary constraints on the graph structure. Therefore, a restriction part for a query language is desirable in order to express undesired workflow properties in addition to the desired properties. The new query model differs from POQL because in addition to the workflow structure metadata is important, e.g. to support users who are maybe unable to express the structure of a scientific workflow. Figure 2 illustrates the structure of a query based on POQL with additional metadata.

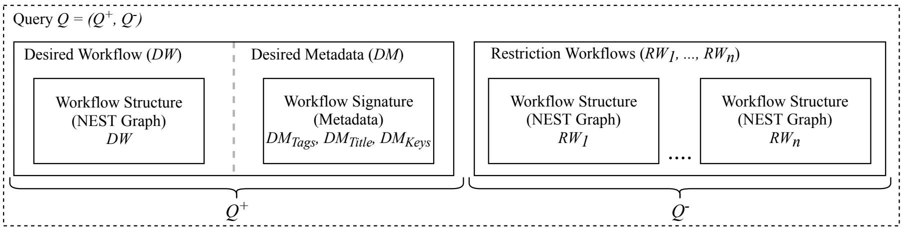
Query $Q = (Q^+, Q^-)$
Desired Workflow (DW)
Desired Metadata (DM)
Workflow Structure (NEST Graph) DW
Workflow Signature (Metadata) $DM_{Tags}, DM_{Title}, DM_{Keys}$
Restriction Workflows ($RW_1, ..., RW_n$)
Workflow Structure (NEST Graph) RW_1
....
Workflow Structure (NEST Graph) RW_n
Q^+
Q^-

Fig. 2. Query Structure for Similarity-Based Retrieval of Scientific Workflows

Based on previous work [19,23], a query $Q = (Q^+, Q^-)$ contains a desired part $Q^+ = (DW, DM)$ and a restriction part $Q^- = (RW_1, ..., RW_n)$. DW represents a desired workflow structure and Q^- represents one or more undesired workflow fragments. The desired metadata properties of a workflow are specified by DM, where DM_{Tags} represents the searched tags of a query, DM_{Title} the desired workflow title, and DM_{Keys} the specified keywords.

The similarity assessment for the desired workflow structure and the restriction workflow structures is similar to POQL in which a graph matching algorithm by Bergmann and Gil [1] is used (see [19]). Furthermore, the literature study has shown that POQL addresses all important structural properties for similarity-based workflow retrieval. For this reason, it is not necessary to add further structural elements and to consider them in the similarity assessment. However, metadata should also be considered in the similarity assessment. Therefore, suitable expression elements should be part of the new query language.

With respect to the extended query model, the case workflow representation must be adapted accordingly. In addition to the workflow structure, the

NEST graph also contains metadata in the semantic description of task and data nodes. For this reason, a strict separation between the workflow structure and the workflow signature is not appropriate for case workflows. However, metadata that characterizes the entire workflow should not be stored in the semantic description of specific nodes. Instead, this metadata should be handled separately in the case workflow. The title of a workflow CW_{Title}, the tags of all operators CW_{Tags}, and the comments for documentation $CW_{Comment}$ have to be taken into account at workflow level. Although these metadata could be stored in the semantic description of the workflow node (see [1]), we have decided to separate the metadata from the workflow structure as much as possible since this fosters the integration of other workflow representations that do not provide semantic enrichments.

3.4 Similarity Assessment and Retrieval Process

The previously extended query model requires an adaptation of the similarity computation presented in previous work (see [19,23]). With regard to the added expression elements, the workflow signature, i.e., the metadata is the new query part to be considered in the similarity assessment. We use a bag-of-words approach to combine all comments in a workflow in a single set. This is also applied for the tags of each operator. The similarity between the workflow structures of the query (consisting of a desired workflow DW and one or more restriction workflows $RW_{1...n}$) and a case workflow graph CW_{Graph} is defined by $sim_{POQL}(DW, RW_{1...n}, CW_{Graph}) \rightarrow [0,1]$ (see [23] for more details). Please note that arbitrary graph similarity measures can be used in this function. For the whole query Q, the similarity to a case workflow CW is calculated as follows:

$$\begin{aligned} sim(Q, CW) = \quad & sim_{POQL}(DW, RW_{1...n}, CW_{Graph}) * w_1 \\ & + sim_{tags}(DM_{Tags}, CW_{Tags}) * w_2 \\ & + sim_{title}(DM_{Title}, CW_{Title}) * w_3 \\ & + sim_{keywords}(DM_{Keys}, CW_{Comment}) * w_4 \end{aligned} \tag{1}$$

$sim(Q, CW) \rightarrow [0,1]$ is composed of several weighted and normalized similarity measures. The weights are individually adjusted according to the application scenario resulting in a sum of 1 (see section 4.1 for the weights chosen).

The tags DM_{Tags} are arranged in a taxonomy. The similarity between a query Q in which DM_{Tags} represents the searched n tags ($n = |DM_{Tags}|$) and a case workflow CW with m tags ($m = |CW_{Tags}|$) is specified as follows:

$$sim_{tags}(DM_{Tags}, CW_{Tags}) = \sum_{i=1}^{n} \left[\max_{1 \leq x \leq m} \{sim(DM_{Tag_i}, CW_{Tag_x})\} * \frac{1}{n} \right] \tag{2}$$

$sim(DM_{Tag_i}, CW_{Tag_x}) \rightarrow [0,1]$ specifies the taxonomic similarity between Tag_i from a query Q and Tag_x from the particular case workflow CW.

DM_{Keys} refers to the keywords specified in a user query that are combined into a bag-of-words. In order to enable an efficient retrieval, an index structure is

used similar to current web search engines. We use Apache Lucene[6] to create the index at the beginning of the retrieval process. Lucene tokenizes the comments of the workflows and subsequently scans the index in main memory. By default, an unnormalized similarity measure $BM25(DM_{Keys}, CW_{Comment}) \rightarrow [0, \infty[$ based on Okapi BM25 (see [20] for more details) for a case workflow CW is returned for a keyword search.
We define the similarity measure $sim_{keywords}(DM_{Keys}, CW_{Comment}) \rightarrow [0, 1]$ for a case base CB and a case workflow CW ($CW \in CB$) as follows:

$$sim_{keywords}(DM_{Keys}, CW_{Comment}) = \begin{cases} \frac{BM25(DM_{Keys}, CW_{Comment})}{\max\limits_{\forall CW \in CB}\{BM25(DM_{Keys}, CW_{Comment})\}} & hit \\ 0 & else \end{cases} \quad (3)$$

Hit applies if Lucene has found one or more case workflows for the specified keywords in the query.

For the similarity calculation of the workflow title, a similarity measure $sim_{title}(DM_{Title}, CW_{Title}) \rightarrow [0, 1]$ based on the Levenshtein distance is used.

4 Evaluation

In this section, we evaluate the *Reuse Assistant* according to two hypotheses:

H1 Using POCBR with the *Reuse Assistant* enables users to retrieve and reuse suitable workflows for their current problem.

H2 Using POCBR with the *Reuse Assistant* supports users to solve a specified problem faster than with the exclusive use of RapidMiner.

4.1 Evaluation Setup

For the evaluation, the *Reuse Assistant* is prototypically implemented based on the CBR component provided by the CAKE framework[7]. The graphical user interface of the prototype is designed as follows: The specification of the title is provided by an input field. To implement the tag search, all the tags available in RapidMiner are extracted and made available to the user in a drop-down menu. The workflow structure and comments created by the user are captured within a XML representation generated by RapidMiner. The XML can be copied and pasted from RapidMiner into the user interface. Once the query has been specified, the user can search for a similar workflow. The similarity assessment aggregates the weighted similarities of the individual expression elements. In a prior experimental test phase, weights of 54 % for the graph structure and 46 % for the metadata[8] has been identified as suitable for the prototype. The most similar workflow is then returned to the user as an image. Within the interface, users can also view the next similar workflow or clear all entries.

[6] https://lucene.apache.org/

[7] See http://cake.wi2.uni-trier.de/

[8] Composed of Title: 11 %, Tags: 25 %, and Keywords: 10 %

In addition to the operators included in RapidMiner, twelve generalized operators are implemented as an extension for RapidMiner. These can be selected by the user in the workflow modeling and thus can be used in the workflow structure of a query for retrieval. For this evaluation, the restriction workflow and the parameter settings presented in section 3.3 have not been implemented within the prototype. For simplicity purposes, the prototype has not been integrated into the RapidMiner user interface.

To evaluate the hypotheses with the prototypical implementation, a case base with 22 workflows is created with a total of 36 distinct operators and an average of 5.8 operators per workflow. Furthermore, we semantically enrich the workflows by adding a workflow documentation to the entire workflow and a descriptive comment to each operator.

For the evaluation, we have invited participants from the "Data- and Web Mining" lecture from the University of Trier for our evaluation. Eleven students could be recruited that stated to have experience in creating RapidMiner workflows. The participants were randomly divided into two groups: six participants in Group A and five in Group B. Group A worked exclusively with RapidMiner and the Recommender WoC. In contrast, Group B worked with RapidMiner and the developed *Reuse Assistant*. The evaluation was carried out separately for each group. The *Reuse Assistant* or the Recommender WoC was introduced to the respective group. During the familiarization phase, we also refreshed the participant's knowledge about workflow modeling with RapidMiner. Afterwards, each student received four Data Mining tasks to solve. We have ensured that the workflows retrieved from the case base do not completely solve the given problems and that adjustments are necessary in Group B. The computer screens were recorded in order to determine in detail, which expression elements were used and to measure the interaction times of each participant. Students were asked to answer a survey after completing the tasks.

4.2 Results

All eleven participants were able to solve the respective tasks. The following results obtained from the answers of Group B confirm hypothesis H1:

- In most cases (75 %), a suitable workflow could be found with the help of the prototypical implementation (see statement a. in figure 3).
- All participants agreed that the prototypical implementation as an integrated plugin in RapidMiner would be useful and that they would use it (see statement b. in figure 3).
- As potential users of the prototypical implementation, the participants stated that it is suitable for both *true users* and *power users*.

In the following, we present the results with respect to hypothesis H2. In order to determine to what extent the participants could solve the tasks with the prototype faster than with the exclusive use of RapidMiner, we measured the time required to solve the problem. We can confirm that the participants

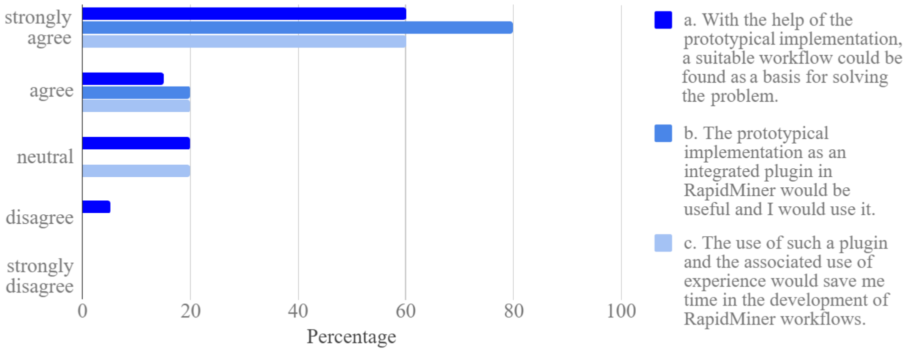

Fig. 3. Results of the Evaluation of Group B

of Group B (Avg. Time: 34:40 min. / SD: 7:02 min.) completed the four tasks faster than those of Group A (Avg. Time: 36:07 min. / SD: 12:29 min.). In addition, it was observed that some participants of Group B had retrieved a suitable workflow with the help of the prototype after a short time. At that moment, however, the participants were not aware that they already had found the solution and were instead searching for further supposedly better workflows. If the participants had analyzed the retrieved workflows more thoroughly, they would have needed less time to solve the given problem. The fact that, unlike WoC, the prototype is not integrated in the RapidMiner user interface and must therefore be run as a second application presumably contributed to an extended period of time until the problem was solved. The self-estimation of the participants from Group B also confirms our hypothesis H2. Four out of five participants mentioned that they would save time in developing RapidMiner workflows, if they would have the possibility to use the prototype as a plugin (see statement c. in figure 3).

In addition, we have examined how often the provided expression elements were used in the evaluation by the five participants of Group B. Since the expression elements could be combined, requests with several expression elements were also possible. It was observed that the workflow structure was used in 96 % of the 49 queries from Group B. Comments with keywords (47 %) and tags (67 %) were also frequently used for searching. In contrast, the participants rarely used the title of the workflow (6 %). However, this could be due to the fact that the participants had no overview of the workflows in the case base and thus the search for the title of a workflow turned out to be difficult. In practice, we assume that the use of the title could be significantly higher.

5 Conclusion and Future Work

In this work, we identified expression elements from current research literature and classified them according to the categorization by Goderis et al. [9]. Based on this literature study, we have developed a first approach for case-based reuse of scientific workflows named *Reuse Assistant*. The *Reuse Assistant* extends POQL to include metadata, which is particularly useful for *true users* when searching for scientific workflows. Furthermore, the similarity assessment is adapted to take the metadata as well as the workflow structure into account. The evaluation indicates that the *Reuse Assistant* enables *true users* as well as *power users* to reuse scientific workflows and thus to facilitate the development process significantly. Therefore, this work can be seen as a first step to support scientists from the DH in using scientific workflows for data analysis. By regularly using the *Reuse Assistant*, they can gain experience and become *power users* over time.

In the future, an extended evaluation particularly involving participants from the DH using their own tasks is desirable to substantiate the indications found in this work. Additionally, such an evaluation may discover further information about the use of expression elements and the process of searching for scientific workflows. An integration of the *Reuse Assistant* into RapidMiner as a plugin is also desirable and could significantly decrease the time needed to develop workflows. As explained previously, some participants have retrieved a suitable workflow, but did not recognize it as a solution immediately. It would thus be desirable to improve the presentation of the retrieval results. Whether the documentation of a case workflow should be enriched with further elements such as parameter descriptions or help texts of operators could also be analyzed in future work. Furthermore, the additional use of semantic technologies such as the use of a thesaurus could make the keyword-based search more powerful. While this project worked with fixed weights for the similarity measures, the importance of the expression elements could be determined by the users themselves. Finally, future research could investigate whether especially *true users* can be better supported in retrieving and reusing scientific workflows by a conversational POCBR approach [23] in combination with our conceptual query model.

Acknowledgments. This work is funded by the German Federal Ministry of Education and Research (BMBF, No. 01UG1606).

References

1. Bergmann, R., Gil, Y.: Similarity Assessment and Efficient Retrieval of Semantic Workflows. Inf. Syst. **40**, 115–127 (2014)
2. Chinthaka, E., Ekanayake, J., Leake, D.B., Plale, B.: CBR Based Workflow Composition Assistant. In: IEEE Congress on Services, Part I, SERVICES I 2009. pp. 352–355. IEEE Computer Society (2009)
3. Cohen-Boulakia, S., Leser, U.: Search, Adapt, and Reuse: The Future of Scientific Workflows. SIGMOD Record **40**(2), 6–16 (2011)
4. Gil, Y., González-Calero, P.A., Kim, J., Moody, J., Ratnakar, V.: A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogues. J. Exp. Theor. Artif. Intell. (2011)
5. Gil, Y., Kim, J., Puga, G.F., Ratnakar, V., González-Calero, P.A.: Workflow Matching Using Semantic Metadata. In: Proceedings of the 5th International Conference on Knowledge Capture (K-CAP). pp. 121–128 (2009)

6. Gil, Y., Ratnakar, V., Kim, J., González-Calero, P.A., Groth, P.T., Moody, J., Deelman, E.: Wings: Intelligent Workflow-Based Design of Computational Experiments. IEEE Intelligent Systems **26**(1), 62–72 (2011)
7. Goderis, A.: Workflow Re-use and Discovery in Bioinformatics. Ph.D. thesis, School of Computer Science, The University of Manchester (2008)
8. Goderis, A., Fisher, P., Gibson, A., Tanoh, F., Wolstencroft, K., Roure, D.D., Goble, C.A.: Benchmarking Workflow Discovery: A Case Study From Bioinformatics. Concurrency and Comp.: Practice and Experience **21**(16), 2052–2069 (2009)
9. Goderis, A., Li, P., Goble, C.A.: Workflow discovery: the problem, a case study from e-Science and a graph-based solution. In: IEEE International Conference on Web Services (ICWS). pp. 312–319 (2006)
10. Goderis, A., Sattler, U., Lord, P.W., Goble, C.A.: Seven Bottlenecks to Workflow Reuse and Repurposing. In: 4th International Semantic Web Conference, Proceedings. pp. 323–337 (2005)
11. Hauder, M., Gil, Y., Sethi, R.J., Liu, Y., Jo, H.: Making Data Analysis Expertise Broadly Accessible through Workflows. In: WORKS'11, Proceedings of the 6th Workshop on Workflows in Support of Large-Scale Science. pp. 77–86 (2011)
12. Jannach, D., Jugovac, M., Lerche, L.: Supporting the Design of Machine Learning Workflows with a Recommendation System. TiiS **6**(1), 8:1–8:35 (2016)
13. Kuhn, J., Reiter, N.: A Plea for a Method-Driven Agenda in the Digital Humanities. In: Book of Abstracts of DH 2015 (2015)
14. Kuras, C., Eckar, T.: Prozessmodellierung mittels BPMN in Forschungsinfrastrukturen der Digital Humanities. In: Eibl, M., Gaedke, M. (eds.) INFORMATIK 2017. pp. 1101–1112. Gesellschaft für Informatik, Bonn (2017)
15. Littauer, R., Ram, K., Ludäscher, B., Michener, W., Koskela, R.: Trends in Use of Scientific Workflows: Insights from a Public Repository and Recommendations for Best Practice. International Journal of Digital Curation **7**(2), 92–100 (2012)
16. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M.B., Lee, E.A., Tao, J., Zhao, Y.: Scientific Workflow Management and the KEPLER System. Concurrency and Computation: Practice and Experience **18**(10), 1039–1065 (2006)
17. Ludäscher, B., Weske, M., McPhillips, T.M., Bowers, S.: Scientific Workflows: Business as Usual? In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) Business Process Management, 7th International Conference, BPM, Proceedings. pp. 31–47. LNCS, Springer (2009)
18. Minor, M., Montani, S., Recio-García, J.A.: Process-oriented Case-based Reasoning. Inf. Syst. **40**, 103–105 (2014)
19. Müller, G., Bergmann, R.: POQL: A New Query Language for Process-Oriented Case-Based Reasoning. In: Bergmann, R., Görg, S., Müller, G. (eds.) Proceedings of the LWA 2015. pp. 247–255. CEUR Workshop Proceedings, CEUR-WS.org (2015)
20. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: Proceedings of The Third Text REtrieval Conference, Gaithersburg, Maryland, USA, November 2-4. pp. 109–126 (1994)
21. Starlinger, J.: Similarity Measures for Scientific Workflows. Ph.D. thesis, Humboldt University of Berlin (2016)
22. Stoyanovich, J., Taskar, B., Davidson, S.: Exploring Repositories of Scientific Workflows. In: Proceedings of the 1st International Workshop on Workflow Approaches to New Data-centric Science. pp. 7:1–7:10. ACM (2010)
23. Zeyen, C., Müller, G., Bergmann, R.: Conversational Process-Oriented Case-Based Reasoning. In: Aha, D.W., Lieber, J. (eds.) Case-Based Reasoning Research and Development - 25th International Conference, ICCBR 2017, Proceedings. pp. 403–419. LNCS, Springer (2017)