Incorporating Semantics in Pattern-Based Scientific Workflow Recommender Systems

Improving the Accuracy of Recommendations

Abstract—**Recommender systems are used to enable decision-support. Using them to assist users when designing scientific workflows introduces a number of challenges. These include selecting appropriate components and specifying correct parameter values. Pattern-based workflow recommender systems employ historical usage patterns to generate recommendations. Such systems can intelligently adapt with use. Semantics, on the other hand, can enable recommender systems to intelligently infer new relationships between workflow components. Combining both approaches can help to overcome the drawbacks of each approach and improve the accuracy of the suggestions. To this end, a framework for a hybrid workflow design recommender system is presented in this paper along with the accompanying suggestion generation algorithm. An illustrative example is also presented to demonstrate how the system helps in constructing a workflow. The performance of the framework is compared with an existing pattern-based system using a dataset of neuroimaging workflows. The evaluation results demonstrate that the proposed system outperforms the existing system in a number of different scenarios. The improvement in the performance of the proposed system enhances the usability of the system for users and allows them to more efficiently construct workflows.**

***Keywords*—workflow design; recommender systems; workflow execution systems; ontologies**

I. Introduction

Workflows are a way to describe a series of computations on raw e-Science data. These data may be MRI brain scans, data from a high energy physics detector or metric data from an earth observation project. In order to derive meaningful knowledge from the data, it must be processed and analysed. According to [1], workflows have emerged as the principle mechanism for describing and enacting complex e-Science analyses on distributed infrastructures such as grids. Scientific users face a number of challenges when designing workflows. These challenges include selecting appropriate components for their tasks, specifying dependencies between them and selecting appropriate parameter values. These tasks become especially challenging as workflows become increasingly large. For example, the CIVET workflow consists of up to 108 components [2]. Building the workflow by hand and specifying all the links can become quite cumbersome for scientific users.

Traditionally, decision-support systems have been employed to assist users in such time-consuming and tedious tasks. Such systems are also called recommender systems. One of the techniques used by recommender systems has been to predict what the user is trying to do using a variety of techniques. These techniques include using workflow semantics on the one hand and historical usage patterns on the other. Semantics-based systems attempt to infer a user's intentions based on the available semantics. Pattern-based systems attempt to extract usage patterns from previously-constructed workflows and match those patterns to the workflow under construction. The use of historical patterns adds dynamism to the suggestions as the system can learn and adapt with "experience". However, in cases where there are no previous patterns to draw upon, pattern-based systems fail to perform. Semantics-based systems, on the other hand infer from static information, so they always have something to draw upon. However, that information first has to be encoded into the semantic repository for the system to draw upon it, which is a time-consuming and tedious task in itself. Moreover, semantics-based systems do not learn and adapt with experience. Both approaches have distinct, but complementary features and drawbacks. By combining the two approaches, the drawbacks of each approach can be addressed. This paper presents a hybrid recommender system that combines both these sources of knowledge to generate more accurate suggestions. An overview of the proposed framework is presented in this paper along with a comparative evaluation with an existing system.

The paper begins with related work in Section II. The related work gives way to a description of the hybrid framework being presented in this paper in Section III. The suggestion generation algorithm is presented in Section IV. An illustrative example is presented in Section V. Section VI identifies use cases where this framework may be used. In Section VII the proposed framework is compared with an existing system. The results thus obtained are discussed in Section VIII and the paper concludes with a summary and conclusions in Section IX.

II. Related Work

Over the past two decades, there has been a lot of interest in developing systems to suggest relevant items to users. In literature, such systems are termed *recommender systems* [3]. Such systems have found many real-world applications such as recommending books, CDs and other products at Amazon[1], movies by MovieLens and news at VERSIFI Technologies [4], [5], [6]. Due to the large number of items (objects to be recommended) available, recommender systems help users to simplify the often tedious and cumbersome process of sifting through them. Recommender systems are a well-researched

[1] http://www.amazon.com

area in general. However, there has not been much focus on workflow design recommender systems.

In general workflow design recommender systems rely on one of two types of information to generate suggestions; historic usage patterns and workflow component semantics. VisComplete is a pattern-based system developed by Koop et al. that treats workflows as graphs and tries to find the most frequent patterns that occur in the graphs [7]. These patterns represent collections of workflow components that designers frequently use in conjunction with each other. When workflows are represented as graphs, frequent subgraphs within that set represent the frequent patterns in the workflows. Similar to VisComplete, Oliveira et al. have also developed a pattern-based recommendation system that they have integrated into the VisTrails workflow composition tool [8], [9]. It works by parsing the repository of workflows and finding frequent connections. Contrary to VisComplete, however, the system only provides single-step suggestions at every point.

Junaid el at. have integrated a semantics-based system into the ASKALON workflow environment [10], [11]. The semantics considered include *workflow name*, *workflow domain*, *component type*, *component function* etc. Moreover, the system also incorporates some user-defined criteria for filtering suggestions. Users can limit the suggestions to be user-specific, domain-specific or time-specific. In addition to the previous criteria, the system also applies a third filtering step. The filtering at this stage is performed on the basis of two measures; the design-time and the runtime reliability and correctness of the suggestions. Design-time reliability and correctness are determined by the skill level of the users designing the workflows and the frequency of use by expert users. On the other hand runtime reliability and correctness are measured by several factors including the number of successful executions of the component, the degree of correctness of the results, the resources previously used by the component currently available in the grid, and the resources previously used by the component currently reserved in the grid. However, this paper argues that some of the criteria used by this system should have no bearing on the appropriateness of the suggestions generated. These include, for example the resources used by the component when executed last, as often workflows are not designed and executed at the same time. It is conceivable the resources available on the grid might have changed by the time the workflow is executed. Thus, they should have no role in determining if a component is appropriate.

CAT is another example of a semantics-based system [12]. It employs a mixed-initiative approach to workflow composition; the system can generate complete or partial workflows automatically from user-defined descriptions as well suggest actions to users as they compose workflows. At every step, the system uses semantic descriptions coupled with formally defined properties to determine correctness of workflows. The semantics of the components and their input and output ports are described in a knowledge base using two separate hierarchical ontologies; a component ontology and a domain term ontology.

Both semantics-based and pattern-based systems suffer from certain drawbacks. Semantics-based systems require the semantics to have been specified beforehand, which is a time-consuming and tedious task. Pattern-based systems on hand require can learn automatically with time without user intervention. However, there may be certain cases where a user may be constructing a workflow that is not very common. In this situation a pattern-based system would not work. However, a semantics-based system might be useful in this case. This research proposes that by combining both sources of knowledge, the drawbacks of each individual approach may be overcome. Such a hybrid architecture is presented and evaluated in this paper.

III. Architecture for a Hybrid Scientific Workflow Design Recommender System

The proposed suggestion generation framework is shown in Fig. 1. It combines frequent usage patterns and semantics to generate suggestions. The semantics as well as frequent usage patterns are stored in a combined knowledge base (domain ontology), which is then used to generate the suggestions. The semantics are applied in two phases; when mining the patterns and when generating suggestions. The steps involved in applying the semantics and an overview of the framework are described subsequently.

i) **Workflow Composition Tool:** Users use a *Workflow Composition Tool* to compose their workflows.

ii) **Workflow Repository:** The workflows, once composed are stored in the workflow repository.

iii) **Pattern Extraction Engine:** The workflows are then mined for frequent usage patterns by the pattern extraction engine.

iv) **Workflow to Graph Conversion:** Before the workflows can be mined, they must first be converted to graphs using this component.

v) **Component Generalisation:** The converted workflows are then generalised by the generalisation component. This process employs the workflow component semantics to identify their functional type. This allows the system to extract patterns that represent groups of component types that perform a single composite function instead of groups of specific components.

vi) **Usage Pattern Extraction:** Finally the patterns are mined for frequent subgraphs by the usage pattern extraction component. These frequent subgraphs represent frequent usage patterns in the workflows.

vii) **Domain Ontology:** All of the knowledge extracted in this manner is stored in the domain ontology, the central entity in this process. It contains the extracted patterns as well as semantics about the workflow components. The initial classification of components along with a description of their inputs and outputs is specified by domain experts.

viii) **Suggestion Building Engine:** The *Suggestion Building Engine* is responsible for retrieving suggestion candidates from the *Domain Ontology* and sending them to the *Workflow Composition Tool.*

ix) **Semantic Analyser:** This component analyses partial workflows and propagates semantics across components.

When users design workflows, the workflow composition tool contacts the *suggestion building engine* for recommendations. The suggestion building engine sends the partially constructed workflow to the *semantic analyser* for analysis. The semantic analyser employs the relationships between component parameters specified in the domain ontology to semantically-

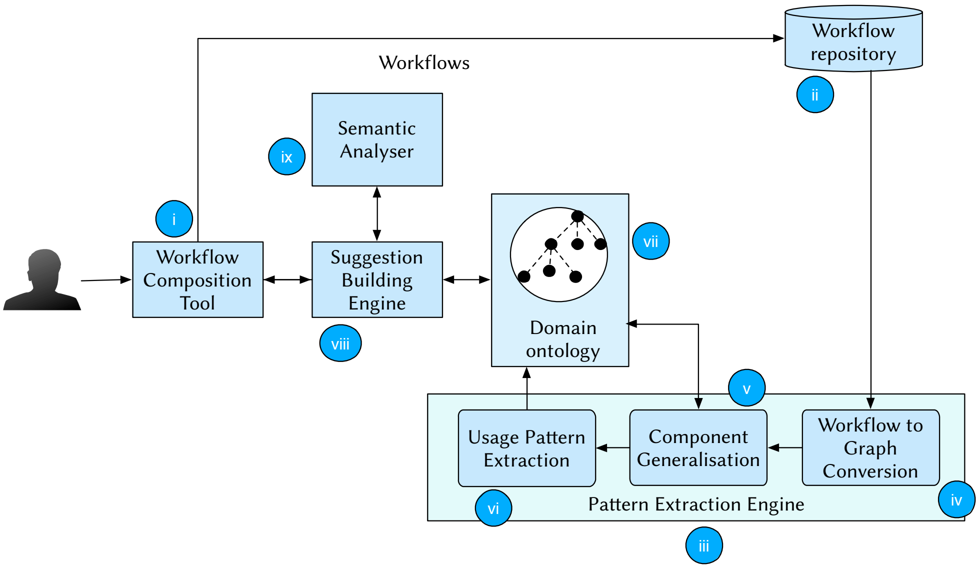


Fig. 1. Hybrid suggestion generation framework architecture.

enrich the workflow. This allows the system to generate more accurate suggestions. The semantically-annotated partial workflow is then sent back to the suggestion building engine. It then queries the domain ontology for suggestions and sends them back to the workflow composition tool. The suggestion generation algorithm is presented, discussed and evaluated in the following sections.

IV. Generating Suggestions

In order to generate suggestions for a partial workflow, it must undergo a series of processing steps. The suggestion generation algorithm is discussed subsequently.

i) The workflow is converted to a graph.
ii) The components in the workflow are generalised.
iii) The repository is searched for patterns that overlap with the partial workflow.
iv) Once matching patterns are found, they are specialised.
v) The partial workflow is then semantically analysed and enriched.
vi) Compatible components are retrieved from the repository based on the enriched workflow semantics.
vii) The results are sorted and returned.

Generalising the components involves replacing specific components with their functional types from the ontology. For example, *Align Linear* is a *RegistrationComponent*. If a workflow contains the former, then generalisation will replace it with the latter. This allows the system to identify composite functions that constitute multiple other functions. For example, when registering one MRI to another, linear alignment is the first step that results in a transformation matrix. This matrix is then used by the *Reslice* component to register the MRI. These two components frequently appear together in workflows since they are closely linked. Moreover, these are not the only two components of their types that perform this function. There are other components that do the same, but are appropriate in different circumstances. Without generalisation, the system would treat every group of components as a different pattern. However, with generalisation, the system would be able to infer that a *RegistrationComponent* and a *ReslicingComponent* together constitute a complete registration process. Specialisation is the opposite process of generalisation.

Semantic analysis involves propagating known semantics of component inputs and outputs across other components. This ensures that when workflow completions are to be suggested based on semantics, as much information as possible is available for the system to reason with. For example, *BrainParser* is a component that labels the different regions of a brain MRI. It requires the MRI to be skull-stripped; i.e. the skull and other extraneous tissue needs to be stripped away so that only the brain is left in the image. Now if a worklow contains the *BrainParser* component, the system can determine that a skull-stripping component such as *SSMA* is required. However, consider the situation where the input of *BrainParser* is connected to another component, e.g. a noise-filtering component such as *Bias Field Corrector*. In this case the system would not be able to infer that a skull-stripping component is required here. However, by propagating the semantics of *BrainParser* back across *Bias Field Corrector*, the system would be able to correctly infer that *SSMA* is required. The use of semantic propagation to ensure more accurate suggestions are generated enriches the workflow semantically.

V. Illustrative Example

This section demonstrates how the system assists a user in constructing a neuroimaging workflow. Fig. 2 shows an example neuroimaging workflow that is just being constructed. This workflow is taken from the LONI repository called “Automated ROI Extraction/Volume Calculation”. As a user adds components to the workflow, the system actively suggests possible completions. In this case, Fig. 2 shows the initial workflow as the user starts building it. Once the user adds the first component, the system presents the user with a list of candidates. For this example, the correct candidate consists of

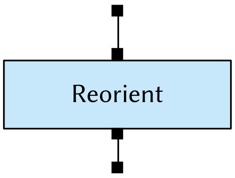


Fig. 2. Example workflow.

a subworkflow. This is because the added component appears in the suggested subworkflow in the repository. Choosing the appropriate suggestions results in the workflow shown Fig. 3.

Adding the subworkflow triggers another suggestion generation request. This time, the correct suggestion is not in the list, so the user has to manually add the correct component. The next batch of suggestions originate from semantic compatibility between components as they do not appear in any frequent patterns. It takes a total of 9 steps to complete this workflow. The completed workflow is shown in Fig. 4.

VI. Use Cases

The performance of the presented framework depends upon several factors. These include the availability of workflow semantics and the existence of overlapping patterns between the workflow being designed and the repository. In the absence of sufficient semantics, the framework would not be able to produce semantics-based suggestions. Therefore, for these use cases it is assumed that sufficient semantics exist in the repository. However, there may be three scenarios with respect to overlapping patterns. These scenarios are described below as different use cases:

Use Case 1 (UC_1): The workflow being constructed may contain overlapping patterns with the workflow repository.

Use Case 2 (UC_2): The workflow may not have any overlapping patterns with the repository before generalisation. However, generalising the workflow may result in the emergence of overlapping patterns.

Use Case 3 (UC_3): The workflow may not have any overlapping patterns with the repository even after generalisation.

VII. Experimental Evaluation

To evaluate the framework, 65 neuroscience workflows from the LONI repository were used [13]. During the pattern extraction phase, the minimum frequency threshold for each pattern was set at 4. It was so chosen because it is approximately equal to 5% of the total workflows, which is the significance level commonly used in statistical significance testing [14]. In addition, Closed Graph mining was enabled to eliminate overlapping patterns and reduce the number of overall patterns mined [15]. These patterns were written into the ontology for use by the Suggestion Building Engine. The descriptions of the various workflow components along with their inputs and outputs were specified beforehand by domain experts.

The Mean Reciprocal Rank (MRR), coupled with the number of steps required to construct the workflow were used to evaluate the suggestions [16]. The MRR is appropriate in this case because for each list of suggestions presented to the user, there will be at most only one correct suggestion. A higher MRR indicates that the relevant suggestion was ranked highly by the system. For the number of steps required to construct the workflow, a lower number is desirable. This would indicate that the system was of greater assistance by minimising user involvement. On the contrary, a high number of steps would indicate that the user needed to be more involved in the design process. The results were compared with those of an existing system for the same dataset. The system chosen for comparison was developed by [8]. It was chosen because it is a system that does not employ any semantics. It only relies on frequent patterns. Since this research attempts to show how combining semantics with patterns can improve the suggestions, Oliveira et al.'s system is a suitable candidate for comparison.

Three sets of workflows were chosen in accordance with the three use cases identified in Section VI. These were workflows that (a) already existed in the repository, (b) did not exist in the repository but had overlapping patterns with it, and (c) did not exist in the repository and had no overlapping patterns with it. For the experiments, three workflows of each category were chosen and are shown in Table I. The table also lists their size in terms of the number of components they comprise and which use case they are meant to evaluate. For UC_2 and UC_3, the workflows were first removed from the original repository in turn, reducing the total count to 62 workflows. The results of the experiments are presented in the next section.

VIII. Results

For each of the workflows provided as input to the framework, the results are shown in Fig. 5. Fig. 5a shows the average MRR for both systems. It was calculated by averaging the MRRs for each individual step required to construct the work-

TABLE I. Workflows used to perform evaluation.

UC	ID	Name	Size
UC_1	Wf_1	BrainParser (Hippocampus)	14
	Wf_2	BrainParser (56 Structures)	14
	Wf_3	Automated ROI Extraction/Volume Calculation	22
UC_2	Wf_4	BrainParser (Hippocampus)	14
	Wf_5	BrainParser (56 Structures)	14
	Wf_6	Automated ROI Extraction/Volume Calculation	22
UC_3	Wf_7	MDT with KLMI (Existing Atlas)	15
	Wf_8	MDT with KLMI (1 Subject Bias)	13
	Wf_9	MDT with KLMI (Multiscale Symmetric)	27

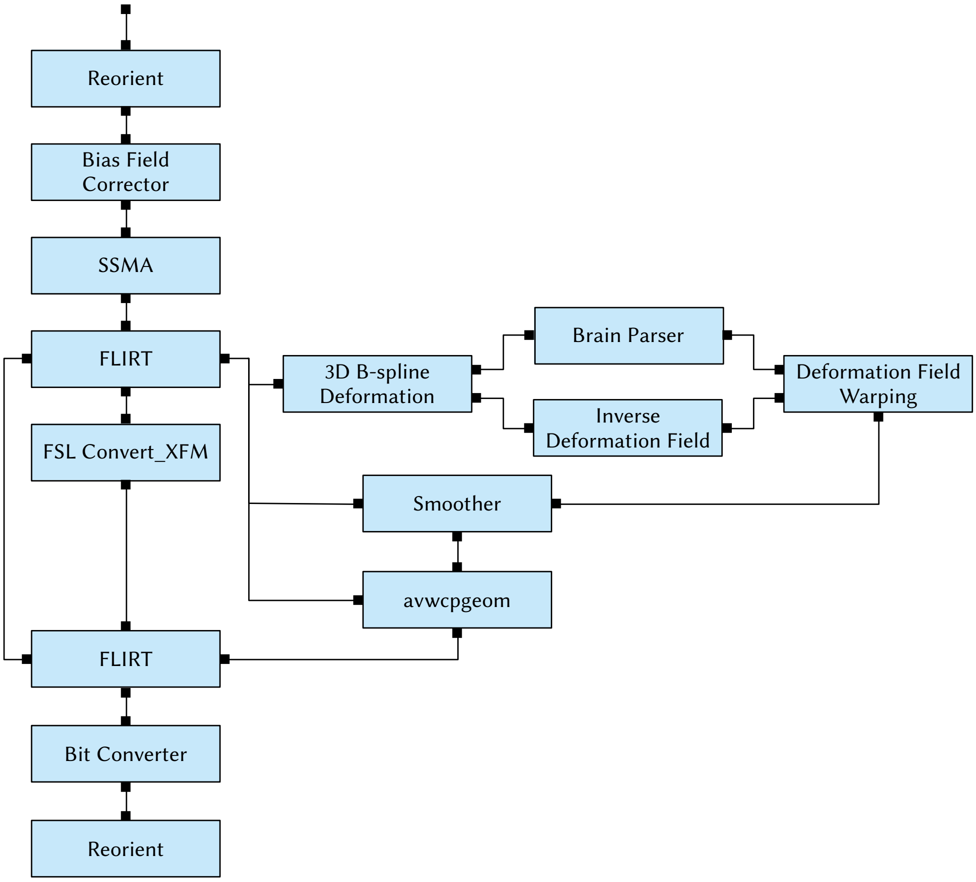


Fig. 3. Example workflow after one step.

flow. Fig. 5b compares the number of steps it took to construct each of the workflows using both systems. For workflows that are already in the repository or have overlapping patterns, the MRR is low as compared to Oliveira et al. because the proposed system suggests more than one component at a time while Oliveira suggests only one. The larger patterns were less frequent than the smaller ones. Since both systems employ frequency-based ranking mechanisms, Oliveira's suggestions occur more frequently, and thus are ranked higher. On the other hand the proposed system's suggestions occur less frequently because they consist of multiple components. Thus they are ranked lower. In the case of workflows with no overlapping patterns, the proposed system clearly outperforms Oliveira et al.'s system. This is because the proposed system also generates suggestions based on semantics, as opposed to Oliveira et al.'s system.

In Fig. 5b, the proposed system clearly outperforms Oliveira et al.'s system since the less steps it takes to construct a workflow, the more efficient is the system. Overall it can be seen that the proposed framework requires few steps to construct workflows that have overlapping patterns with other workflows in the repository. For Wf_1 it took only one step to construct since the entire workflow appears as a sub-workflow in other workflows in the repository. Similarly, for Wf_2 it took only two steps. This is possible since the framework does not suggest only one component at a time; instead it can suggest as many as required. For Wf_3 it took 9 steps to complete the workflow even though it consisted of 22 components. This was again made possible by the existence of overlapping patterns. For workflows Wf_4, Wf_5 and Wf_6, the number of steps required to complete the workflows were still much less than the sizes of the workflows. This was true even though the workflows themselves did not exist in the repository. The results for workflows Wf_7, Wf_8 and Wf_9 show that even though the workflows did not exist in the repository and did not have any overlapping patterns, the number of steps required to construct them were still less. This is possible because even though there were no overlapping patterns directly, the framework was able to find patterns after generalisation.

IX. Conclusions

This paper presents a hybrid workflow design recommender system that combines frequent usage patterns and workflow semantics to generate suggestions and provide decision-support. Doing so, the framework attempts to address the drawbacks of each approach. The framework is described along with the suggestion generation algorithm. The framework is also compared with an existing system that only uses frequent patterns using a dataset of neuroimaging workflows from the LONI repository. Results show that there is a clear improvement in the accuracy of the suggestions when semantics are combined with frequent usage patterns as opposed to only using patterns. The various scenarios that may occur when generating suggestions are also presented and evaluated. These are a) when overlapping

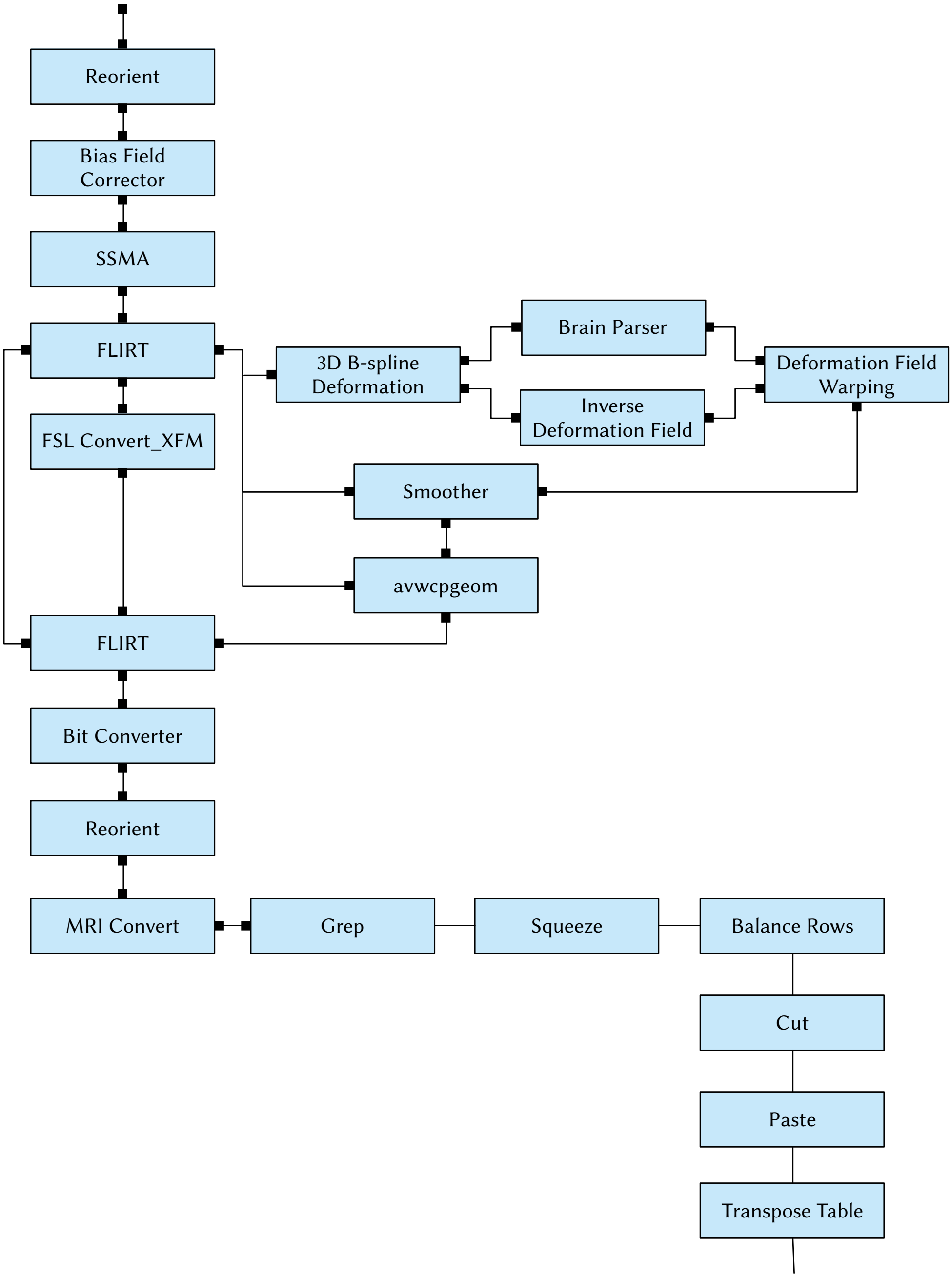


Fig. 4. Example workflow after 9 steps.

patterns between the workflow being constructed and the repository exist, b) when overlapping patterns between the workflow being constructed and the repository do not exist, and c) when no overlapping patterns exist. Results show that semantics do not improve the accuracy of the results in a). However, there is clear improvement in the cases of b) and c).

Future directions for this research may include incorporating parameter value suggestions as well since the correct functioning of workflows also relies on correct parameter values.

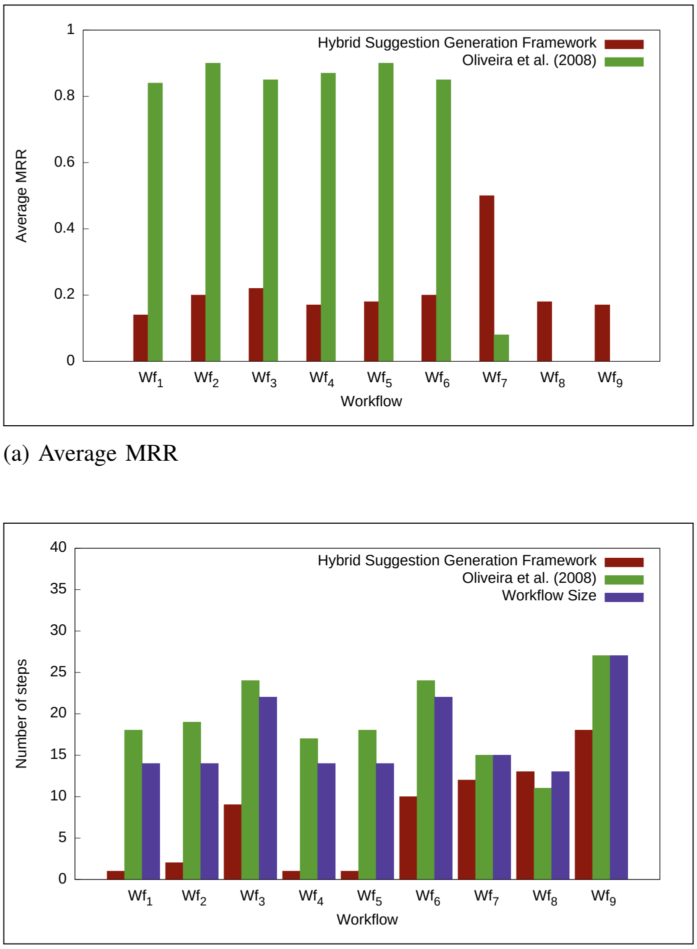
References

[1] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-Science: An overview of workflow system features and capabilities," *Future Generation Computer Systems*, vol. 25, no. 5, pp. 528–540, 2009.

[2] R. McClatchey, I. Habib, A. Anjum, K. Munir, A. Branson, P. Bloodsworth, and S. L. Kiani, "Intelligent grid enabled services for neuroimaging analysis," *Neurocomputing*, vol. 122, pp. 88–99, 2013.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 734–749, 2005.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, pp. 76–80, 2003.



(a) Average MRR

(b) Number of steps

Fig. 5. Comparison of proposed framework and Oliveira et al.

[5] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, “MovieLens unplugged: experiences with an occasionally connected recommender system,” in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ser. IUI ’03. New York, NY, USA: ACM, 2003, pp. 263–266.

[6] D. Billsus, C. A. Brunk, C. Evans, B. Gladish, and M. Pazzani, “Adaptive interfaces for ubiquitous web access,” *Commun. ACM*, vol. 45, pp. 34–38, May 2002.

[7] D. Koop, C. Scheidegger, S. Callahan, J. Freire, and C. Silva, “Viscomplete: Automating suggestions for visualization pipelines,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1691–1698, Nov 2008.

[8] F. T. Oliveira, L. Murta, C. Werner, and M. Mattoso, “Provenance and annotation of data and processes,” ser. Lecture Notes in Computer Science, J. Freire, D. Koop, and L. Moreau, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Using Provenance to Improve Workflow Design, pp. 136–143.

[9] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo, “Vistrails: visualization meets data management,” in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’06. New York, NY, USA: ACM, 2006, pp. 745–747.

[10] M. Junaid, M. Berger, T. Vitvar, K. Plankensteiner, and T. Fahringer, “Workflow composition through design suggestions using design-time provenance information,” *2009 5th IEEE International Conference on E-Science Workshops*, pp. 110–117, 2009.

[11] T. Fahringer, A. Jugravu, S. Pllana, R. Prodan, C. Seragiotto, and H.-L. Truong, “ASKALON: a tool set for cluster and grid computing,” *Concurrency and Computation: Practice and Experience*, vol. 17, no. 2-4, pp. 143–169, 2005.

[12] J. Kim, A. Gil, and M. Spraragen, “A knowledge-based approach to interactive workflow composition,” in *In Proceedings of the 2004 Workshop on Planning and Scheduling for Web and Grid Services, at the 14th International Conference on Automatic Planning and Scheduling (ICAPS 04)*, 2004.

[13] D. E. Rex, J. Q. Ma, and A. W. Toga, “The LONI pipeline processing environment,” *NeuroImage*, vol. 19, no. 3, pp. 1033–1048, 2003.

[14] S. L. Chow, *Statistical significance: Rationale, validity and utility*. SAGE Publications Limited, 1997, vol. 1.

[15] X. Yan and J. Han, “CloseGraph: mining closed frequent graph patterns,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 286–295.

[16] N. Craswell, “Mean reciprocal rank,” in *Encyclopedia of Database Systems*, L. LIU and M. ÖZSU, Eds. Springer US, 2009, pp. 1703–1703. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-39940-9_488