

Sequential Decision Making with Medical Interpretation Algorithms in the Semantic Web

Patrick Philipp^(✉)

Institutes AIFB, KSRI, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
patrick.philipp@kit.edu

Abstract. Supporting physicians in their daily work with state-of-the art technology is an important ongoing undertaking. If a radiologist wants to see the tumour region of a headscan of a new patient, a system needs to build a workflow of several interpretation algorithms all processing the image in one or the other way. If a lot of such interpretation algorithms are available, the system needs to select viable candidates, choose the optimal interpretation algorithms for the current patient and finally execute them correctly on the right data. We work towards developing such a system by using RDF and OWL to annotate interpretation algorithms and data, executing interpretation algorithms on a data-driven and declarative basis and integrating so-called meta components. These let us flexibly decide which interpretation algorithms to execute in order to optimally solve the current task.

Keywords: Sequential decision making · Linked APIs · Data-driven and declarative framework

1 Introduction

Supporting physicians in their daily work with state-of-the art technology is an important ongoing undertaking. Technical experts are, therefore, developing interpretation algorithms to, for instance, automatically process medical images. To help radiologists assess the development of tumour patients, a tumour progression mapping (TPM) is beneficial. The brain has to be stripped out of a patient's headscan, registered with respect to prior headscans of the patient and normalized until a final interpretation algorithm can generate a TPM. These interpretation algorithms need to be fed with the correct data and executed in correct order. In addition, there might be several interpretation algorithms available for one subtasks (e.g. for segmenting the brain) which might not all perform 'optimally'. Above all, as the state-of-the-art in image processing evolves, new interpretation algorithms might need to be taken into account for this task.

Besides TPM generation, there are numerous other complex tasks to support physicians. We divide them into pre-surgical, intra-surgical and post-surgical tasks and give an abstract and incomplete overview in Table 1.

These tasks are complex because they either need rich function classes to solve them, comprise numerous subtasks or both. Interpretation algorithms, such

Table 1. Complex medical task classification.

Pre-surgical	Intra-surgical	Post-surgical
Diagnosis	Interpreting sensor outputs	Health stability estimation
Treatment proposition	Risk analysis	Further treatment proposition

as image processors, might be important in all three phases. We work towards a system able to solve numerous different complex tasks by choosing among a large pool of interpretation algorithms. We, therefore, propose a semantic framework for sequential decision making. Interpretation algorithms are annotated with semantic concepts formalized in RDF and OWL, wrapped as Linked APIs and integrated into a data-driven, declarative workflow. We use so-called *meta components* to choose among interpretation algorithms for a given task. A central open problem deals with how and to what degree we can leverage semantic descriptions for optimally solving complex tasks.

2 State of the Art

In this section, we give an overview of research related to our setting. The section is divided into two parts. First, we depict research about (semantic-) workflow systems, as we need to enable workflows of interpretation algorithms to solve complex tasks. The second part deals with decision making within these workflows. We have to find eligible interpretation algorithms to reach a goal and then choose the optimal candidate.

2.1 Workflow Systems

The work centered around semantic workflows [5] aims to enable the automatic composition of components in large-scale distributed environments. Generic semantic descriptions support combining algorithms and enable formalizing ensembles of learners. Therefore, conditions and constraints need to be specified. The framework also automatically matches components and data sources based on user requests.

Taverna [8] is another scientific workflow system supporting process prototyping by creating generic service interfaces and thus easing the integration of new components. Semantic descriptions are being used to better capture the view of the scientists. Taverna is able to integrate data from distributed sources and automate the workflow creation process for users.

Wood et al. [16] create abstract workflows as domain models which are formalized using OWL and enable dynamic instantiation of real processes. These models can be automatically converted into more specific workflows resulting in OWL individuals. The components can be reused in another context or process, and one can share abstract representations across the Web through OWL classes.

All of the above approaches develop abstractions of interfaces between workflow components in terms of meta data. Ontologies and taxonomies are, therefore, used to represent central structures for workflows. While some approaches use OWL to generate meta-data, we try to model a low amount of axioms and keep the approach flexible. We use this flexibility and incorporate decision making strategies, as will be discussed in the subsequent section.

2.2 Decision Making for Workflows

In our setting, we need to decide among $2, \dots, n$ interpretation algorithms for a subtask and build a workflow of $1, \dots, m$ interpretation algorithms to solve a complex task. We, therefore, distinguish between *meta learning* and *planning* approaches from the literature and will further classify our setting in Sect. 3.

Planning. Automatic orchestration of analytical workflows has been studied by Beygelzimer et al. [2]. The system essentially uses a planner, a leaner and a large (structured-) knowledge base to solve complex tasks. A large amount of potential workflows are taken into account to answer a user specified query with the optimal choice. The decision process comprises complex learning and planning approaches, and entails exploring large possible feature spaces. Lastly, atomic actions are lifted with semantic annotations to better adapt to user queries. Although our goal equals automatically orchestrating workflows, we also want to enable have multiple possibly situation-dependent learners, as there might not be a generic solution.

Markov Decision Processes (MDP) are often employed to learn workflows. Applications to the health care sector comprise the work of Sahba et al. [11]. They use MDPs to model segmentation algorithms for transrectal ultrasound images and to optimize the prevalent parameters. Besides, Gao et al. [3] used MDPs to enable the composition of web services described with Web Service Definition Language (WSDL). The goal is to optimize decisions in terms of web service availability and runtime. We, in contrast, deal with possibly heterogeneous interpretation algorithms in one single workflow. We need to capture important features to optimally choose them in correct situations. This also distinguishes our work from optimizing for availability or runtime.

Meta Learning. Besides optimizing workflows, one can use ensemble learning strategies to choose between two competing candidate interpretation algorithms. A prominent strategy is the multiplicative weights method [1] (e.g. used in boosting). Here, candidates are combined based on their performance on training sets. The method was already applied to decision trees with patient factors (e.g. by Moon et al. [7]) to combine predictions. We, however, focus on the interplay of such strategies with available semantics for interpretation algorithms and data. We, thus, want to enable to use such sophisticated ensemble learners within our framework if they are well-suited for the current (sub-) task.

3 Problem Statement and Contributions

Let X be the set of all tasks, Y the set of all abstract tasks and A the set of all available interpretation algorithms. Let further S be the set of *abstract states* defined by a subset of objects O , literals L and relations R . We denote, for simplicity, F_{s_k} as the set of features of a state s_k (i.e. a subset of $O \times R \times O$ and $O \times R \times L$). A grounded state $g(s_k)$ depicts an instance of s_k in nature. The set A_{s_k} defines the subset of applicable interpretation algorithms in s_k which is known to some degree. We, thus, assume that an interpretation algorithm $a_i \in A$ can be defined by a subset of features of F in a similar way as states $s_k \in S$. Knowing A_{s_k} depends on how we define features $f \in F$ for s_k and a_i . Let $T(s, a, s')$ be the transition function for some state s and interpretation algorithm a ending in s' . Our knowledge of $T(s, a, s')$, again, depends on the available features for s , a and s' . $T(g(s), a, g(s'))$ is not known and requires further knowledge to be approximated. A task $x(g(s_k), (s_K))$ is a function defined on a grounded start state $g(s_k)$ and an abstract goal state s_K . Reaching an unknown grounded goal state $g(s_K)$ takes 1 to n state transitions $(g(s), a, g(s'))$. To solve $x(g(s_k), (s_K))$, we need to find a sequence of interpretation algorithms a_i ending in the unknown grounded goal state $g(s_K)$ with high probability. An abstract task $y(g(s_k), (s_K))$ is defined similarly and we need to find any sequence a_1, \dots, a_n to get from $g(s_k)$ to s_K . Our setting is much related to a Markov Decision Process (MDP) (S, A, T, R, γ) with R , in addition, being the reward function for state, interpretation algorithm pairs (s, a) and γ the discount factor. The latter regulates the influence of future interpretation algorithms a_i taken in future steps s_k on the value estimations of current states and actions. Defining $R(s, a)$ for $x(g(s_k), (s_K))$ is not straightforward as $g(s_K)$ is unknown. An absorbing state with $R(s_k, a_i) = 0$ can be artificially modelled to denote the goal s_K .

We define **abstract planning** as trying to solve an *abstract task* $y(g(s_k), (s_K))$. Here, we ignore that multiple interpretation algorithms a_i might be available for s_k . **Meta learning** considers $|A_{s_k}| > 1$ and tries to solve a subtask $x_i(g(s_k), s_K)$ to find the optimal a_i for $g(s_k)$. **Planning** deals with solving $x(g(s_k), s_K)$ with known T and R , and planning-related **learning** considers T, R unknown and tries to approximate them (as, for instance, is done in model-based reinforcement learning).

We disclosed the following **challenges** for sequential decision making with medical interpretation algorithms:

- (a) Interpretation algorithm might be developed by different researchers from different institutions. We need a common (meta-) representation to integrate the interpretation algorithms and the data they consume.
- (b) Interpretation algorithms need to be quickly and concurrently accessible if numerous complex tasks have to be solved for different endusers.
- (c) We need to reduce the effort to manually define procedures to solve complex tasks in order to quickly integrate new interpretation algorithms and use them if they perform better.
- (d) To handle heterogeneous and competing interpretation algorithms, we need meta components with potentially different (degrees of-) specialisation

- (e.g. some might only deal with continuous outcomes, others with discrete ones; some might leverage groundings, others abstract levels).
- (e) It is unclear which information we need to incorporate into our decision making. Investigating the connection between (abstract-) planning and (meta-) learning problems and (meta-) representations of interpretation algorithms is, thus, important.
- (f) For physicians to use the system, they have to trust the proposed solutions.

Based on these challenges, we see the following **contributions** of our work:

- (i) We formalize the problem setting and introduce a framework for interpretation algorithms and meta components to automatically solve complex medical tasks.
- (ii) We develop meta components to conduct abstract planning and meta learning, and disclose further challenges for planning and planning-related learning.
- (iii) We are analysing the interplay between semantics (for meta components, interpretation algorithms and data) and added value for solving complex medical tasks.
- (iv) We started to investigate the issue of ‘trust’ and try to give accurate confidence estimates for solutions.

4 Research Methodology and Approach

An overview of our approach is illustrated in Fig. 1. Based on this abstract representation of our ideas, we will focus on three different parts in this section. We, first, describe our research methodology for developing data-driven, declarative workflows to enable sequential decision making for complex medical tasks by explaining the illustrated components. We, then, dwell on meta component scenarios in part two and three - (abstract-) planning and (meta-) learning - and explain their interfaces within the framework.

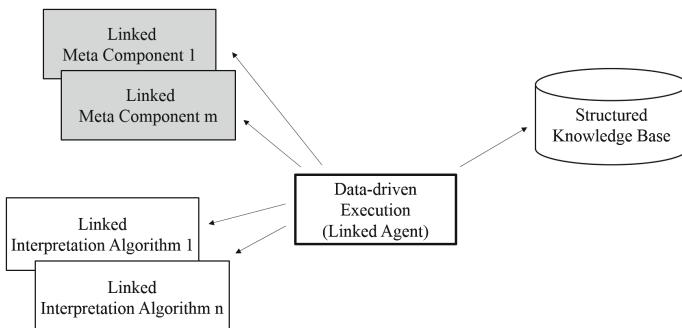


Fig. 1. A schematic overview of our approach.

4.1 Data-Driven, Declarative Workflows

To integrate interpretation algorithms and make them easily accessible, we build on Linked APIs [12] and Linked Data-Fu [13]. These concepts help us to tackle challenges (a) and (b), and build a foundation for challenge (c).

Linked APIs describe a class of RESTful services which are extended with descriptions inspired by the Semantic Web. In our approach, we define important information about interpretation algorithms in terms of a description based on RDF and OWL, and publish RDF wrappers of interpretation algorithms persistently on the Web. We will refer to them as *Linked interpretation algorithms*. To have a central and controlled vocabulary, we developed ontologies for both data types and interpretation algorithms and are working on formalizing evaluation metrics. Besides non-functional requirements (e.g. contributors, textual descriptions or example requests and responses), we model functional aspects of the interpretation algorithms. Here, it remains open how to model the inputs and outputs, and their respective pre- and postconditions to optimally leverage interpretation algorithms for (abstract-) planning and (meta-) learning (challenge (d), see Sect. 6 for a short discussion).

To execute Linked interpretation algorithms, we use the Linked Data-Fu Engine [14]. It enables virtual data integration of distributed data sources to properly execute Linked interpretation algorithms. This is crucial as we have different kinds of information/knowledge relevant for solving complex tasks. **Knowledge from experts** helps estimating the interpretation algorithms' performances on data sets. It highly influences the initial belief about the applicability of interpretation algorithms. As the system gathers more evidence, beliefs might be challenged. **Evidence-based knowledge from other sources** (i.e. other researchers and published papers) essentially comprises domain knowledge as well, but might be subjective and hard to validate. **Statistics-based knowledge** is gathered by testing the interpretation algorithms on training samples and keeping track of their performances on new data.

We automatically generate rules for Linked Data-Fu based on preconditions of interpretation algorithms. The term *Linked agent* is used to denote a Linked Data-Fu instance with access to Linked interpretation algorithms, to a structured (and distributed-) knowledge base and to *meta components*. The latter extend the data-driven, declarative workflows with strategies (of arbitrary complexity) to choose Linked interpretation algorithms for solving a complex task. The Linked agent can, then, easily execute the proper workflows based on a subset of chosen Linked interpretation algorithms. Figure 2 visualizes the interactions of the Linked agent for using abstract planning and meta learning components. The meta components are only called if the current state s_k matches their preconditions, which is one step towards solving challenge (d). The next section deals with our endeavours to develop such meta components and to extend them to the pure planning and learning scenarios.

4.2 (Abstract-) Planning and Planning-Related Learning

As defined in Sect. 3, we are trying to solve an abstract task $y(g(s_k), (s_K))$ with abstract planning. We, therefore, use the MDP formulation and only evaluate

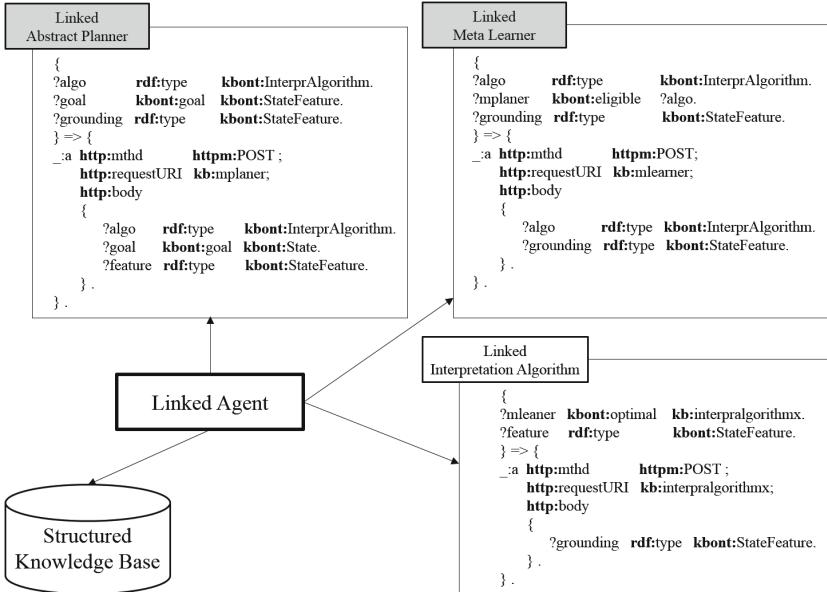


Fig. 2. Components of the framework with general rules.

Linked interpretation algorithms on a concept level based on the grounded start state $g(s_k)$. We model a finite MDP by using the pre- and postconditions of Linked interpretation algorithms as states s_k with local scopes, i.e. we only consider preconditions of states s_k for transitions, and assume to know the state features F_{s_k} . The transition probabilities T are defined in Eq. 1 and make up a $S \times (A+1) \times S$ matrix by adding a dummy interpretation algorithm pointing to the goal state, when the latter was reached. The reward function R is a $S \times (A+1)$ matrix and shifts all rewards to the dummy interpretation algorithm (see Eq. 2). By using any strategy to solve the MDP (e.g. value iteration), we find eligible Linked interpretation algorithms to solve the task.

$$T(s, a, s') = \begin{cases} t_{sas'} = \frac{1}{|A_{s_k}|} & \exists(s, a, s') \text{ based on } F_s \text{ and } F_{s'} \\ t_{sas'} = 0 & \text{otherwise} \end{cases} \quad (1)$$

$$R(s, a) = \begin{cases} r_{sa} = 1 & \text{if } a \text{ equals dummy algorithm and } s \text{ equals goal} \\ r_{sa} = 0 & \text{otherwise} \end{cases} \quad (2)$$

Based on this first simple case, we can model and solve more complex workflows with unsure or stochastic transitions between states and might be able to tackle the case where pre- and postconditions do not exactly match. However, besides abstract tasks, we need to solve $x(g(s_k), (s_K))$ based on the grounded states $g(s_k)$ resulting from executing Linked interpretation algorithms in a workflow. This is a different scenario, as we have to assess the quality of results in terms of different evaluation criteria. Moreover, T and R might be unknown

for numerous states and Linked interpretation algorithms, and we need to learn them with planning-related learning approaches. See Sect. 6 for our next steps towards solving $x_i(g(s_k), (s_K))$.

4.3 Meta Learning

In the meta learning setting, we try to solve $x_i(g(s_k), s_K)$. We approach this problem setting by investigating ensemble learning strategies and extend them by incorporating semantics. A first meta learner might assess the expected performance of a Linked interpretation algorithm for classification based on training samples close to the grounded state $g(s_k)$. We train the Linked interpretation algorithms (if possible in terms of their preconditions) on a subset of samples and predict on the remaining ones (i.e. we cross-validate). The heuristic repeats the process until all training samples have been assessed and derives the probability for a new grounded state $g(s_k)$ based on its performance on similar instances. We can use any similarity function to derive these similar instances (i.e. nearest neighbours). The heuristic is summarized in Algorithm 1.

Algorithm 1. Majority Heuristic given Linked interpretation algorithms L , number of neighbours to consider k , state s_k , cut t

```

1:  $N \leftarrow \text{nearestNeighbours}(g(s_k), k)$ 
2:  $T \leftarrow \text{set of training samples cut into } t \text{ subsets}$ 
3: for all  $t \in T$  do
4:   for all  $l \in L$  do
5:      $\text{train}(l, T \text{ without } t)$  //if possible
6:      $\text{updatePerformanceTable}(l, t)$ 
7: for all  $l \in L$  do
8:    $w_l \leftarrow \text{estimatePerformance}(l, N)$ 
9: for all  $p \in P$  do
10:   $H(p) \leftarrow \sum_{l \in L} w_l \mathbf{1}[h_l(g(s_k)) = p]$ 
11:  $h^* \leftarrow \arg \max_{p \in P} H(p)$ 
12: return  $h^*$ 

```

5 Preliminary Results

We applied the approach to two medical scenarios - image processing for tumour progression mappings (TPM) and sensor interpretation to recognize surgical phases - and evaluated them in terms of correctness, time consumption and effect of meta components. Figure 3 illustrates the shared architecture comprising a structured knowledge base integrated via a Semantic MediaWiki (SMW), several Linked interpretation algorithms, the Linked agent and two Linked meta components.

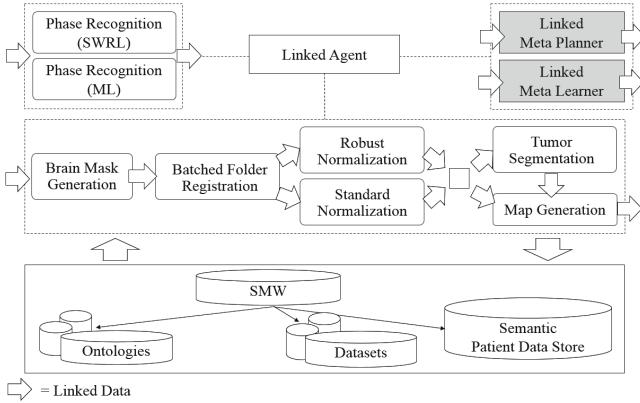


Fig. 3. The semantic framework for medical sequential decision making (extended based on [9]).

5.1 Tumour Progression Mapping and Abstract Planning

As discussed in Sect. 1, a TPM supports radiologists in assessing the development of a patient when treated for brain tumours. The available Linked interpretation algorithm are listed in Fig. 3. Listing 1.1 comprises the preconditions to generate a brain mask, which takes as input a headscan and two reference images. The image ontology is available in the knowledge base (**kbont** namespace). We evaluated the correct functioning and time consumption in [4, 10], and modelled the scenario as finite MDP, as explained in Sect. 4.2. We, therefore, added Linked interpretation algorithms for sensor interpretation to test if only goal-oriented Linked interpretation algorithms were chosen. The MDP consisted of 9 states according to possible transitions we derived based on modelled pre- and postconditions of the interpretation algorithms. When applying value iteration to solve the MDP with T, R following Eqs. 1 and 2, we derived $V = <0.32805, 0.3645, 0.405, 0.405, 0.45, 0.45, 1.00, 0, 0>$ after 6 iterations with discount factor 0.9. The starting state was a grounding for ‘Brain Mask Generation’ and the goal state was the generated TPM. V gives us the estimated values of states s_k and assigns 0 to states related to sensor interpreters. We, thus, do not have to execute them to reach the goal state ‘TPM’.

?headscan	rdf:type dc:format	kbont:Headscan; "image/nrrd".
?brainAtlasImage	rdf:type dc:format	kbont:BrainAtlasImage; "image/mha".
?brainAtlasMask	rdf:type dc:format	kbont:BrainAtlasMask; "image/mha".

Listing 1.1. Preconditions of the brain mask generation step (namespaces are omitted)

5.2 Surgical Phase Recognition and Meta Learning

In surgical phase recognition, one tries to predict the phase of the ongoing surgery based on sensor outputs. Based on this phase, one could visualize risk structures and, thereby, support the surgeon. We wrapped two phase recognizers as Linked interpretation algorithms. ‘SWRL’ uses rules formalized with the Semantic Web Rule Language to predict the phase, and ‘ML’ uses machine-learning and can be trained with annotated surgeries. We evaluated the correct functioning and time consumption in [9] and integrated the meta learning heuristic (see Algorithm 1) into the architecture. Table 2 summarizes the results. The meta learner was able to provide stable and sometimes better results than the single Linked interpretation algorithms.

Table 2. Performance evaluation of phase recognition algorithms and meta learner in 5 different surgeries.

Algorithm	Surgery 1	Surgery 2	Surgery 3	Surgery 4	Surgery 5
ML-based	0.91	0.66	0.90	0.45	0.64
SWRL	0.93	0.78	0.89	0.81	0.72
Linked meta learner	0.93	0.78	0.92	0.78	0.72

6 Evaluation Plan

Our research approach and preliminary results showed that our framework is able to solve first complex tasks. The next step to systematically approach challenge (e) is to explore the effect of feature selection on the meta learning case and the planning-related learning case. We, here, study the interplay between semantics and meta components to be able to assess how fine-grained the pre- and postconditions for Linked interpretation algorithms need to be. Hence, we need to find out how much semantics we need to confidently learn transition probabilities $T(s, a, s')$ and the rewards $R(s, a)$ associated with taking interpretation algorithm a in state s .

We, also, want to better leverage semantics in the (abstract-) planning case. We only modelled a flat image structure for the TPM scenario which restricts generalization and flexibility. We will, therefore, investigate relational MDPs [15] and try to better capture the semantics of Linked interpretation algorithms when modelling the MDP. This might enable to better generalize to new unknown states s_k if their features F_{s_k} have been sufficiently explored before. Generalization, in turn, helps to solve the pure planning case $x(g(s_k), g(s_K))$.

We will focus on giving confidence estimates for the performance of meta components on new states s_k . We aim to extend our abstract planning (and pure planning-) approach with learners of the ‘Knows what it knows’ (KWIK) framework [6] and give theoretical justifications for the framework’s performance (challenge (f)).

7 Conclusions

Solving complex tasks with heterogeneous Linked interpretation algorithms depicts a diverse problem setting and imposes interesting challenges. The medical domain exhibits sufficient complexity for investigating this problem and provides scenarios for sequential decision making under uncertainty. We, first, formalized the setting and disclosed the necessity of semantics, a data-driven and declarative execution and meta components. We, then, proposed a framework which enables to easily incorporate Linked meta components as well as new Linked interpretation algorithms, and to automatically solve complex tasks (contribution (i)). Lastly, we presented first conceptual and practical results for the meta learning and abstract planning cases (contribution (ii)). We work towards extending the capabilities of the meta components and especially investigate their interplay with semantics (longterm contribution (iii)). As trust is an important issue for endusers of the framework, we aim to give exact and transparent confidence estimates for the generated solutions (longterm contribution (iv)).

References

1. Arora, S., Hazan, E., Kale, S.: The multiplicative weights update method: a meta-algorithm and applications. *Theor. Comput.* **8**(1), 121–164 (2012)
2. Beygelzimer, A., Riabov, A., Sow, D., Turaga, D., Udrea, O.: Big data exploration via automated orchestration of analytic workflows. In: ICAC, pp. 153–158. USENIX, San Jose (2013)
3. Gao, A., Yang, D., Tang, S., Zhang, M.: Web service composition using markov decision processes. In: Fan, W., Wu, Z., Yang, J. (eds.) WAIM 2005. LNCS, vol. 3739, pp. 308–319. Springer, Heidelberg (2005)
4. Gemmeke, P., Maleshkova, M., Philipp, P., Götz, M., Weber, C., Kämpgen, B.: Using linked data and web apis for automating the pre-processing of medical images. In: COLD (ISWC) (2014)
5. Gil, Y., Gonzalez-Calero, P., Kim, J., Moody, J., Ratnakar, V.: A semantic framework for automatic generation of computational workflows using distributed data and component catalogues. *Exp. Theor. AI* **23**(4), 389–467 (2011)
6. Li, L., Littman, M., Walsh, T., Strehl, A.: Knows what it knows: a framework for self-aware learning. *ML* **82**(3), 399–443 (2011)
7. Moon, H., Ahn, H., Kodell, R., Baek, S., Lin, C., Chen, J.: Ensemble methods for classification of patients for personalized medicine with high-dimensional data. *AI Med.* **41**(3), 197–207 (2007)
8. Oinn, T., Greenwood, M., Addis, M., Alpdemir, M., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M., Senger, M., Stevens, R., Wipat, A., Wroe, C.: Taverna: Lessons in creating a workflow environment for the life sciences: research articles. *Concurr. Comput. Pract. Exper.* **18**(10), 1067–1100 (2006)
9. Philipp, P., Katic, D., Maleshkova, M., Rettinger, A., Speidel, S., Wekerle, A., Kämpgen, B., Kenngott, H., Studer, R., Dillmann, R., Müller, B.: Towards cognitive pipelines of medical assistance algorithms. In: CARS (2015)

10. Philipp, P., Maleshkova, M., Götz, M., Weber, C., Kämpgen, B., Zelzer, S., Maier-Hein, K., Klauß, M., Rettinger, A.: Automatisierung von vorverarbeitungsschritten für medizinische bilddaten mit semantischen technologien. In: BVM, pp. 263–268. Springer (2015)
11. Sahba, F., Tizhoosh, H., Salama, M.: Application of reinforcement learning for segmentation of transrectal ultrasound images. *BMC Med. Imaging* **8**(1), 8 (2008)
12. Speiser, S., Harth, A.: Integrating linked data and services with linked data services. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) *ESWC 2011, Part I. LNCS*, vol. 6643, pp. 170–184. Springer, Heidelberg (2011)
13. Stadtmüller, S., Norton, B.: Scalable discovery of linked apis. *Metadata Semant. Ontol.* **8**(2), 95–105 (2013)
14. Stadtmüller, S., Speiser, S., Harth, A., Studer, R.: Data-fu: a language and an interpreter for interaction with read/write linked data. In: *WWW*, pp. 1225–1236 (2013)
15. van Otterlo, M.: The logic of adaptive behavior: knowledge representation and algorithms for adaptive sequential decision making under uncertainty in first-order and relational domains, vol. 192. Ios Press (2009)
16. Wood, I., Vandervalk, B., McCarthy, L., Wilkinson, M.D.: OWL-DL domain-models as abstract workflows. In: Margaria, T., Steffen, B. (eds.) *ISoLA 2012, Part II. LNCS*, vol. 7610, pp. 56–66. Springer, Heidelberg (2012)