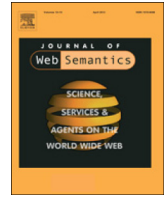




Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

A rule-based agent-oriented approach for supporting weakly-structured scientific workflows

Zhili Zhao^{a,*}, Adrian Paschke^b, Ruisheng Zhang^a^a School of Information Science & Engineering, Lanzhou University, 730000 Lanzhou, China^b Department of Computer Science, Freie Universität Berlin, 14195 Berlin, Germany

ARTICLE INFO

Article history:

Received 27 June 2015

Received in revised form

21 January 2016

Accepted 29 February 2016

Available online 11 March 2016

Keywords:

Scientific workflows

Weakly-structured processes

Multi-agent systems

Logic programming

Semantic Web

ABSTRACT

Weakly-structured Scientific Workflows (WsSWFs) often contain goal-oriented tasks that are logical and complicated, but they are vital for workflow results. They may involve interactions between multiple participants or have complicated logic to express scientific policies and cater to dynamic execution environments. In general, such WsSWFs not only need a rich process and (domain-specific) decision logic specification, but also require a flexible execution and human interaction. In this paper, we propose a Rule-based Agent-oriented Framework (RbAF) to support the WsSWF execution by combining rule-based knowledge representation with agent technology. We describe workflows by messaging reaction rules, which go beyond global Event-Condition-Action (ECA) rules and support performing complex actions locally within certain contexts. We describe (domain-specific) decision logic in workflows by exploiting the benefits of both Logic Programming (LP) and Description Logic (DL). Our evaluation results show that, RbAF well supports the WsSWFs and has higher expressive power than other three considered scientific workflow systems.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

“Scientific workflow has seen massive growth in recent years as science becomes increasingly reliant on the analysis of massive data sets and the use of distributed resources” [1]. They assist scientists to perform data management, analysis and simulation of *in silico experiments* [2]. Compared with business workflows which are already supported by competing specifications and Business Process Management (BPM) standards, scientific workflows have not been widely adopted and supported yet. One significant reason is that scientific workflows have extra requirements over their counterparts in the business domain, such as explicit data/information flow, exact reproducibility, agility to quickly adapt to changed knowledge and human/machine decisions, team cooperation for distributed problem solving and user friendly Graphical User Interface (GUI) tools [3,4]. To address such requirements, existing business workflow technologies need to be thoroughly adapted and extended [5]. Furthermore, existing solutions for business workflows as well as scientific workflows mainly focus on structured compute-intensive and data-oriented tasks, instead of

decision-centric tasks that need the cooperation of scientists or computer agents as a team supported by weakly-structured workflows.

A WsSWF is a process, in which there are complex decision-centric tasks that require agile runtime decisions during their execution; they may involve interactions between multiple participants or have complicated logic to express scientific policies and cater to dynamic execution environments; they could be modeled at a high abstract level with standard graphical workflow representation tools (e.g., Business Process Model and Notation (BPMN)), but the inherent complex and flexible behavior during the task execution cannot be easily implemented. In the current state-of-the-art, there are partial solutions that have been proposed for some of the aforementioned issues, such as increasing the flexibility of service composition [6,7], incorporating knowledge tasks and objects into workflow models [8]. Nevertheless, some core issues of the WsSWFs are still unsolved. Compared with the structured computational scientific workflows, the WsSWFs focus on knowledge-intensive tasks and require:

- *Rich process specification*: the WsSWFs contain complex decision-centric tasks, which require processes to handle new and exceptional situations. Besides simple control-flow descriptions (e.g., a task is enabled after the completion of a preceding task), it is also necessary to describe advanced process

* Corresponding author.

E-mail address: zhaozhil@lzu.edu.cn (Z. Zhao).

logic, which needs dynamic recognition of operational as well as knowledge-based states to implement intelligent routings at runtime.

- **Expressing domain-specific policies:** the WsSWFs often involve complex domain-specific policies, which regulate the behavior of scientific applications. In order to automate the WsSWFs, it is necessary to express such scientific policies and enable machines to deal with them automatically.
- **Flexibility:** the structured processes suffer from limitations with respect to dynamic evolution and adaptation at runtime. In order to provide high flexibility, the WsSWFs should be allowed to be easily modified according to individual situations.
- **Human interaction:** scientific workflow systems are often designed to automate scientific processes and improve their operational efficiency. However, human users still need to perform manual tasks and steer the workflow execution to deal with unforeseen problems at runtime.
- **Exact reproducibility:** provenance plays an important role in verification, explanation, reproduction and informed reuse of data used and produced by scientific workflows, especially by the WsSWFs, which have non-deterministic decision logic (However, provenance is a broad standalone topic and is out of the scope of this work).

This paper mainly focuses on the execution phase of the scientific workflow life cycle and proposes a rule-based, agent-oriented framework, called RbAF, with the purpose of explicitly supporting the WsSWF execution. On one hand, an agent-based framework provides a flexible execution environment. On the other hand, declarative rules provide a declarative programming style to specify the agent behavior. The combination of them offers a promising approach to support the WsSWFs.

The rest of this paper is organized as follows. Section 2 introduces the WsSWFs by means of two real-world use cases. Section 3 presents the state-of-the-art on different solutions with the purpose of improving the flexibility of both business workflows and scientific workflows. Section 4 presents the design of the conceptual workflow framework, RbAF. Section 5 introduces the implementation of RbAF. Section 6 evaluates RbAF based on control-flow and data patterns. Finally, we discuss and conclude the work in Section 7.

2. Use cases

2.1. Treating a newly discovered ant

Fig. 1 presents a fictional but realistic process of identifying a newly discovered *ant* (scientific name: *formicidae*). It is taken from European Distributed Institute of Taxonomy (EDIT), which is a network of excellence gathering 28 major institutions devoted to knowing the living world better with the support of the European Commission. The process involves collaboration of three participants: *fieldworker*, *taxonomist* and *curator*.

The process is organized as follows. First, a *fieldworker* who often works in countryside triggers the identification process. He/She describes a newly discovered ant and then sends the ant description to a *taxonomist*, who has experience and expertise to perform the identification and treat it. Afterwards, a *curator* archives the identification result. Finally, the corresponding treatment schemes are then provided to the *fieldworker*. These participants are often in different locations and collaborate on the ant identification.

It is worth pointing out that the ant identification task itself involves complicated domain-specific logic to distinguish an ant from its homogeneous groups; it is represented as a sub-process (with a “+” mark in the notation) in Fig. 1. The identification details are shown as a process in Fig. 2.

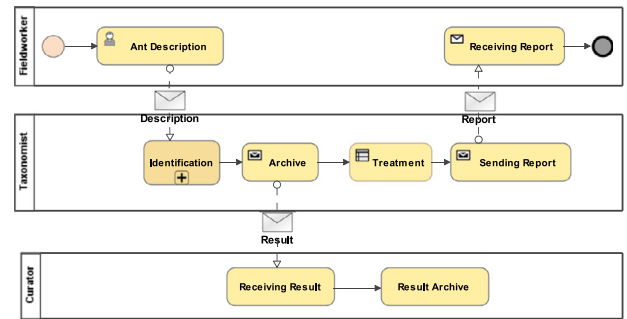


Fig. 1. Process of treating a newly discovered ant.

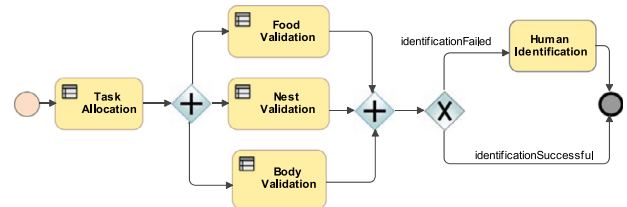


Fig. 2. Process of ant identification.

The identification process starts with allocating the task to an inference service acting on the *taxonomist* behalf in terms of the location, where the ant is discovered. Afterwards, the ant is identified in terms of domain knowledge. Ants can differ widely in their food requirements and behaviors, some pests even can cause a serious impact on crops. According to the Bayer’s ant identification guide [9], the policies used to identify an ant include body features, nest structure and habits (e.g., food preference). Likewise, the task allocation and the ant treatment also need to be encoded with domain knowledge. In Fig. 2, these knowledge-intensive decision-centric tasks are represented as rounded rectangles with small table notations in them. There would be a case that the discovered ant is unusual and the inference service cannot identify it, this happens because ant taxonomists may have different expertises in a certain domain (or area). In this case, the inference service used to identify the ant can forward the request to other services playing the same role for help. Moreover, if no service is available or completes the identification, it might involve domain experts to identify it manually. From a technical perspective, it is difficult to implement this kind of knowledge-intensive decision-centric process by traditional Workflow Management System (WfMS)s.

2.2. Protein prediction result analysis

Fig. 3 shows a process used to analyze the precision of protein prediction algorithms. Nair et al. declared that, “proteins perform most important tasks in organisms, such as catalysis of biochemical reactions, transport of nutrients, recognition and transmission of signals” [10]. In general, protein function can be thought of as, “anything that happens to or through a protein” [11]. For the purpose of describing protein functions, the Gene Ontology Consortium [12] provides an ontology of protein functions based on a dictionary of well-defined terms, also known as Gene Ontology (GO) terms. Each GO term defines gene product properties as well as the relationships with other terms. The protein prediction is often conducted by computational algorithms that generate one or more GO terms indicating the functions that a protein may have. The prediction is considered correct if the protein has some true annotations (i.e., GO terms) that lie on a path in the gene ontology tree from the root to a leaf that visits the predicted annotation (i.e., GO term) [13].

Download English Version:

<https://daneshyari.com/en/article/557701>

Download Persian Version:

<https://daneshyari.com/article/557701>

[Daneshyari.com](https://daneshyari.com)