A pattern-based ontology for describing publishing workflows

Aldo Gangemi[1,2], Silvio Peroni[1,3],
David Shotton[4], and Fabio Vitali[3]

[1] STLab-ISTC, Consiglio Nazionale delle Ricerche (Italy)
aldo.gangemi@cnr.it
[2] Laboratoire d'Informatique de Paris Nord, Université Paris 13 (France)
[3] Department of Computer Science and Engineering, University of Bologna (Italy)
silvio.peroni@unibo.it, fabio.vitali@unibo.it
[4] Oxford e-Research Centre, University of Oxford (UK)
david.shotton@oerc.ox.ac.uk

Abstract. In this paper we introduce the *Publishing Workflow Ontology* (*PWO*), i.e., an OWL 2 DL ontology for the description of generic workflows that is particularly suitable for formalising typical publishing processes such as the publication of articles in journals. We support the presentation with a discussion of all the ontology design patterns that have been reused for modelling the main characteristics of workflows.

Keywords: PWO, ODP, publishing process, workflow description

1 Introduction

Keeping track of publication processes is a crucial task for publishers. This activity allows them to produce statistics on their goods (e.g., books, authors, editors) and to understand whether and how their production changes over time. Organisers of particular events, such as academic conferences, have similar needs. Tracking the number of submissions in the current edition of a conference, the number of accepted papers, the review process, etc., are important statistics that can be used to improve the review process in future editions of the conference.

Some communities have started to publish data, e.g., the Semantic Web Dog Food[5] and the Semantic Web Journal[6], which describe those scholarly data as RDF statements in the Linked Data, in order to allow software agents and applications to check and reason on them, and to infer new information. However, the description of processes, for instance the peer-review process or the publishing process, is something that is not currently handled – although sources of related raw data exist (e.g., EasyChair metadata). Furthermore, having these types of data publicly available would increase the transparency of the aforementioned processes and allow their use for statistical analysis. Of course, a model

[5] Semantic Web Dog Food: http://data.semanticweb.org.
[6] Semantic Web Journal: http://semantic-web-journal.com.

for describing these data is needed. Moreover, the model should be easy to integrate and adapt according to the needs and constraints of different domains (publishing, academic conferences, research funding, etc.).

In this paper we introduce the *Publishing Workflow Ontology* (*PWO*), that we developed in order to accommodate the aforementioned requirements. This ontology is one of the *Semantic Publishing and Referencing* (*SPAR*) *Ontologies*[7] (which have been created for the description of different aspects of the publishing domain), and allows one to describe the logical steps in a workflow, as for example the process of publication of a document. Each step may involve one or more events that take place at a particular phase of the workflow (e.g., authors are writing the article, the article is under review, a reviewer suggests to revise the article, the article is in printing, the article has been published, etc.). This ontology has been developed in order to allow its use with other SPAR Ontologies as well as other models and existing data.

The rest of the paper is organised as follows. In Section 2 we discuss some related works on workflows within the Semantic Web domain. In Section 3 we provide the definitions of workflow we have used as starting point for modelling our ontology, and discuss the use of some existing ontology design patterns for addressing the modelling issues related to the main characteristics of workflows. In Section 4 we introduce PWO, describing how it extends the aforementioned patterns in order to handle the main components of workflows, and we support the discussion by means of a real example of publication process of an article of the Semantic Web Journal. Finally, in Section 5 we conclude the paper sketching out some future works.

2 Workflows and the Semantic Web

In the last years the Semantic Web community have started on working and proposing models for the formalisation and description of generic workflows, and have shown several applications of these models/theories within the publishing domain. Maybe the first huge-impact project on these topic has been Workflow 4ever (STREP FP7-ICT-2007-6 270192)[8] [8]. This project addresses challenges related to the preservation of scientific experiments through the definition of models and ontologies for describing scientific experiments, to the collection of best practices for the creation and management of *Research Objects*[9] [2], and to the analysis and management of decay in scientific workflows.

As already stated, one of the outcomes of the project has been the proposal for workflow-centric Research Objects [1], i.e., an OWL ontology[10] for linking together scientific workflows, the provenance of their executions, interconnections between workflows and related resources (e.g., datasets, publications, etc.), and social aspects related to such scientific experiments.

[7] SPAR Ontologies website: http://purl.org/spar.
[8] Workflow 4ever project homepage: http://www.wf4ever-project.org.
[9] Research Object website: http://www.researchobject.org.
[10] Research Object OWL ontology: http://purl.org/wf4ever/ro.

Another interesting proposal for describing workflows is the work done by Garijo and Gil [7]. In this work, they describe a framework to publish *computational* workflows, which includes the specification a particular OWL ontology, i.e., the *Open Provenance Model for Workflows* (*OPMW*)[11], for the description of workflow traces and their templates. Along the lines of the aforementioned work, the same authors recently published the *Ontology for Provenance and Plans* (*P-Plan*)[12]. P-Plan is an OWL 2 DL ontology that extends the *Provenance Ontology* [12] in order to represent the plans that guided the execution of scientific processes, describing how such plans are composed and their correspondence to provenance records that describe the execution itself.

Finally, among the other proposals for describing workflows, it worths mentioning the OWL ontology proposed by Sebastian *et al.* [18] for describing generic workflows, which reuses existing ontologies such as the *Change and Annotations Ontology* (*ChAO*) [13], and the *SCUFL2 Core ontology*[13] that has been used to describe workflows in *Taverna*[14], an open source and domain-independent Workflow Management System [19].

3 Foundational material: design patterns

In order to design an ontology for modelling (publishing) workflows, we have to understand what are the minimal characteristics that such ontology should address and if we can reuse some existing modelling solutions. Oxford Dictionaries defines workflow as follows:

> "The sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion."[15]

From this definition it is possible to identify some important characteristics of any workflow, i.e., the fact that it *involves* a *sequence* of processes that allow to initiate and then complete a piece of work during a specifiable *time interval*. The definition of the SearchCIO website is still more specific:

> "Workflow is a term used to describe the tasks, procedural steps, organizations or people involved, required input and output information, and tools needed for each step in a business process."[16]

From this definition we can spot other crucial aspects. First of all, its structural organisation in procedural steps, each of them *describes* tasks performed by organisations and people, and each step *requires* some input information and tools in order to produce an output. Using these two definition as input, we

[11] Open Provenance Model for Workflows: http://www.opmw.org/ontology/.
[12] Ontology for Provenance and Plans: http://purl.org/net/p-plan#.
[13] http://ns.taverna.org.uk/2010/scufl2
[14] http://www.taverna.org.uk
[15] http://www.oxforddictionaries.com/definition/english/workflow
[16] http://searchcio.techtarget.com/definition/workflow

can identify some well-known ontological patterns that already address, from an abstract point of view, some of the aspects related of workflows.

Participation. The *participation pattern*[17] is a simple pattern that allows us to describe processes, events, or states (through the class *Event*), and to specify the various objects (through the class *Object*) that participate in these events.

This pattern seems to be very useful to define workflows as events involving people, organisations, places, and other objects as participants, as well as to link workflows and related activities to the expected steps .

Sequence. The *sequence pattern*[18] is another pattern that can be used between tasks, processes or time intervals, in order to define sequences of such objects through direct (i.e., *directlyFollows* and *directlyPrecedes*) and transitive relations (i.e., *follows* and *precedes*). It is, of course, very useful to describe the logical organisation of the various steps of a workflow.

Control flow and plan execution. The *control flow* pattern[19] is an OWL representation of some of the constructs defined in the Workflow Patterns[20] by Wil van der Alst (cf. [17]). Either action or control (e.g., branching, concurrency, looping) tasks are represented and related by means of the *sequence pattern. Tasks* are distinct from *activities*, which are supposed to be executed based on the task structure. This link is made in the context of the *basic plan description*[21] and the *basic plan execution*[22] patterns, which reuse the foundational *descriptions and situations pattern* to relate task compositions (*plans*) to organised activities (*plan executions*). A comprehensive presentation is provided in [6].

These patterns are of course, very useful to describe the kinds of steps (the term used here for *tasks*) in a workflow and in general in publishing workflows. The action and control tasks from the control flow pattern are not specialised in the publishing workflow pattern, because they are expected to work as they are (by typing the steps according to their workflow semantics) when the need for control flows emerges in a planned workflow.

Time-indexed situation. The *time-indexed situation pattern*[23] allows the description of a situation (i.e., the class *TimeIndexedSituation*) – i.e., a view on a set of entities linked to it through the property *isSettingFor* – that is explicitly indexed at some time specifiable through the property *atTime* linking a time interval (i.e., an instance of the class *TimeInterval*).

This pattern can be used to describe steps from an abstract point of view as kinds of situations representing the settings for all the events and input/output material needed or produced by these steps. Notice that time-indexed situation combines perfectly with plan execution in order to provide a temporal ordering to activities organised into a plan.

[17] http://www.ontologydesignpatterns.org/cp/owl/participation.owl
[18] http://www.ontologydesignpatterns.org/cp/owl/sequence.owl
[19] http://www.ontologydesignpatterns.org/cp/owl/controlflow.owl
[20] The Workflow Patterns page is: http://www.workflowpatterns.com.
[21] http://www.ontologydesignpatterns.org/cp/owl/basicplandescription.owl
[22] http://www.ontologydesignpatterns.org/cp/owl/basicplanexecution.owl
[23] http://www.ontologydesignpatterns.org/cp/owl/timeindexedsituation.owl

Error Ontology. The *Error Ontology*[24] is a unit test that produces an inconsistent model if a particular (and incorrect) situation happens. It works by means of a data property, *error:hasError*, that denies its usage for any resource, as shown as below (in Manchester Syntax [9]):

```
DataProperty: error:hasError
  Domain: error:hasError exactly 0    Range: xsd:string
```

A resource that has an error makes the ontology inconsistent, since its domain is "all those resources that do not have any *error:hasError* assertion".

This model is very useful in our context in order to define constraints on the input/output objects needed by the steps of a workflow. For instance, we could use it to deny the use of a certain object as input of a step if it will be produced only as output of one of the following steps.

4 PWO: the Publishing Workflow Ontology

In order to accommodate workflow requirements, we developed the *Publishing Workflow Ontology*[25] (*PWO*), which is entirely based on the ontology patterns introduced in Section 3. This ontology allows one to describe the logical steps in a workflow, as for example the process of publication of a document. Each step may involve one or more events (or actions) that take place to a particular phase of the workflow (e.g., authors are writing the article, the article is under review, a reviewer suggests to revise the article, the article is in printing, the article has been published, etc.).

As shown in Fig. 1, PWO is based on two main classes, which are:

- class *pwo:Workflow*. It represents a sequence of connected tasks (i.e., steps) undertaken by the agents; it is a subclass of *plan:PlanExecution*[26];
- class *pwo:Step*. It is an atomic unit of a workflow, subclass of *taskrole:Task*; it is characterised by a (required) starting time and an ending time, and it is associated with one or more events (activities) that are executed within the step. A workflow step usually involves some input information, material or energy needed to complete the step, and some output information, material or energy produced by that step. In the case of a publishing workflow, a step typically results in the creation of a publication entity, usually by the modification of another pre-existing publication entity, e.g., the creation of an edited paper from a rough draft, or of an HTML representation from an XML document.

[24] http://www.essepuntato.it/2009/10/error

[25] http://purl.org/spar/pwo

[26] Note that in PWO we are not using explicitly the separation between workflow definition and workflow execution, since PWO has been thought as an ontology to provide a retrospective description of running workflows. Even if this is a simplification of the whole approach described by the imported patterns, we decided to include both patterns for workflow definition and execution in order to handle even workflow definitions in case we may need it (even if we have not yet explored this use of PWO properly).

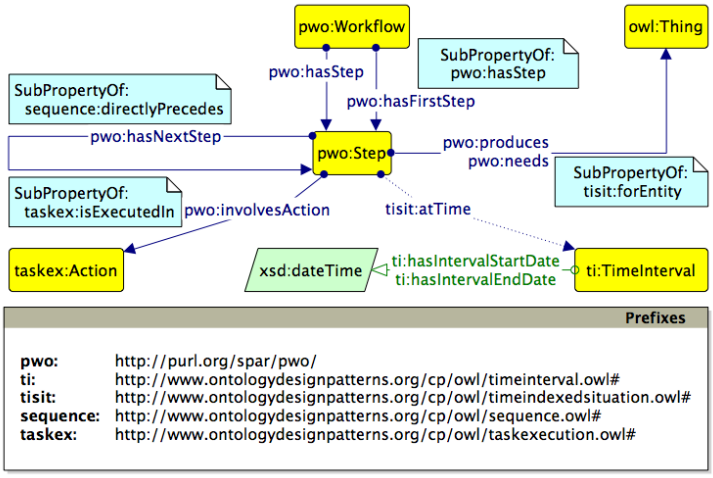

Fig. 1. Graffoo representation [5] of the Publishing Workflow Ontology (PWO).

PWO was implemented according to the aforementioned ontology patterns. As shown in Table 1, such patterns have been used as follows:

- *plan execution* to describe workflows as plans, and their executions;
- *time-indexed situation* to describe workflow steps as entities that involve a duration and that are characterised by events and objects (needed for and produced by the step);
- *sequence* to define the order in which steps appear within a workflow;
- *control flow* to describe the specialization and nature of steps at planning time;
- *participation* to describe events (and eventually agents involved) taking part in the activities carried out according to the steps.

In addition, by means of the Error Ontology, we can generate an inconsistency every time the steps of a workflow are not arranged in a correct temporal order. In particular, an error is raised when a step requires (property *pwo:needs*) to use a particular object that will be produced (property *pwo:produces*) as consequence of another sequent step. The following excerpt shows the implementation of this constraint through a SWRL rule [10]:

```
Step(?step1) , Step(?step2) , needs(?step1,?resource) ,
produces(?step2,?resource) , sequence:precedes(?step1,?step2)
  -> error:hasError(?step1,"A step cannot need a resource that will be
       produced by a following step"^^xsd:string)
```

In the next subsections we show how to describe the process of publication of a journal article step by step. In particular we introduce how PWO can be used in combination with existing data of the *Semantic Web Journal*[27] [11] and other SPAR ontologies, such as PSO [15], C4O [4], FaBiO and CiTO [14].

[27] Semantic Web Journal data: http://semantic-web-journal.com/sejp.

Table 1. A summary of all the entities of PWO and their relations with the original pattern-based entities.

PWO entity	Pattern entity	Description
Workflow	Plan (plan execution)	The class of particular situation types describing a real-life work, composed by a sequence of steps
Step	Task (task role, via control flow)	The class describing specific tasks that form the workflow and that are done within particular time intervals
hasStep	definesTask (basic plan description)	The relation linking a workflow to a component step
hasFirstStep	definesTask (basic plan description)	A sub-property of *hasStep* which identifies the starting step of a workflow
hasNextStep	directlyPrecedes (sequence)	An object property linking a step in a workflow with the step that directly follows it
hasPreviousStep	directlyFollows (sequence)	An object property linking a step in a workflow with the step that directly precedes it
involvesAction	isExecutedIn (task execution)	The object property linking a step in a workflow to an activity done in the context of that step
needs	forEntity (time-indexed situation)	The object property linking a workflow step to anything required to undertake that step
produces	forEntity (time-indexed situation)	The object property linking a workflow step to the thing that the step produces, creates or results in

4.1 A typical publishing workflow of a journal article

From a pure publisher's perspective, the first step of any workflow that brings to a new journal publication starts with a formal submission of a manuscript performed by someone, hereinafter the *author*. This activity expresses, at the same time, interest on the topics of the journal and may acknowledge, indirectly, the quality of the journal itself – since authors (usually) would like to publish articles in a venue that they consider respectful and qualitatively worth for different reasons (e.g., quality of reviews, journal impact factor, definite timing of the publishing process). Then, in the next step, i.e., the reviewing phase, the person (designated by the publisher) in charge of the quality of submitted material, hereinafter the *editor*, invites other people (hereinafter the *reviewers*) for assessing the quality of the submitted manuscript. The opinions returned by the reviewers to the editor are the fundamental input that the editor will use to decide upon the fate of the manuscript during the next step, i.e., the decision phase. Finally, if the manuscript have been considered worth of publication in the present form, the editor will acknowledge the author of the acceptance of his/her work – and the next steps of the workflow will be in charge of the publisher itself. Otherwise, if the article is not ready for being published, the editor either may ask for its rejection, thus finishing the workflow, or (s)he can return a list of issues to be addressed to the author in order to deserve publication. In this latter case, the revision phase will start and the author will revise the paper according to reviewers' comments and editor's suggestions, and thus the workflow will continue with a new submission phase.

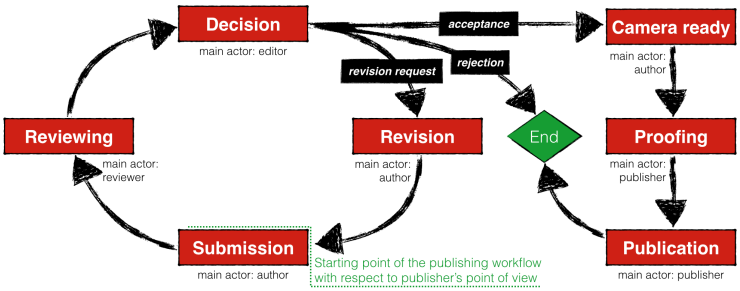

Fig. 2. A diagram describing the typical publishing workflow of a journal article – note that it does not take into account any withdrawing action by the author, nor any comment made by users on publisher's website before/after article publication.

The whole publishing workflow we have described (summarised in Fig. 2) can be formally represented by means of PWO. In the following excerpt (in Turtle [16]) we create an instance of the class *pwo:Workflow* as composed by a definite (but not specified, in this example) number of steps[28]:

```
:workflow a pwo:Workflow ;
  pwo:hasFirstStep :step-one ;
  pwo:hasStep :step-two , :step-three , :step-four, ... .
```

In the next sections we show how to describe the first four steps of such workflow by taking into account real publication data available in the Semantic Web Journal Linked Data repository concerning [3].

4.2 Submission

The first step of the workflow concerned the submission of a manuscript by one of its authors, in this case Paolo Ciccarese. Thus, the manuscript received the status of "submitted" and it was made available to the journal editor and the reviewers for the next step of the workflow. In order to describe all these aspects concerning the first step, we use several entities defined in the ontology patterns imported by PWO, as well as a number of other entities from another SPAR ontology, i.e., the *Publishing Status Ontology* (*PSO*)[29] [15]. This is an ontology for describing the status held by a document or other publication entity at each of the various stages in the publishing process. In addition, existing entities of the Semantic Web Journal Linked Data repository (e.g., people and manuscripts) are reused in order to demonstrate the flexibility of PWO in working with other existing models and data, as shown as follows:

```
:step-one a pwo:Step ; # Submission step
  pwo:involvesAction :submission-action ; tisit:atTime [ a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-01-21T10:08:28"^^xsd:dateTime ;
    ti:hasIntervalEndDate "2013-01-21T10:08:28"^^xsd:dateTime ] ;
```

[28] Prefixes available at http://www.essepuntato.it/2014/wop/prefixes.ttl.
[29] http://purl.org/spar/pso

```
  pwo:needs swj-node:432 ; pwo:produces :submitted-status ;
  pwo:hasNextStep :step-two .
# The event in which one of the authors submits the manuscript
:submission-action a taskex:Action ;
  dcterms:description "Paolo Ciccarese submits the paper" ;
  part:hasParticipant swj:paolo-ciccarese , swj-node:432 .
# The new status 'submitted' associated to the paper after the submission
:submitted-status a pso:StatusInTime ;  pso:isStatusHeldBy swj-node:432 ;
  pso:isAcquiredAsConsequenceOf :submission-action ;
  pso:withStatus pso:submitted ; tvc:atTime [ a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-01-21T10:08:28"^^xsd:dateTime ] .
```

4.3 Reviewing

The step regarding the reviewing phase began with the activity of the editor, Giancarlo Guizzardi, of looking for appropriate reviewers for the paper. Once found, the reviewers were provided with the manuscript, reviewed it, and wrote down their comments that were finally sent back to the editor. In order to describe all the aspects concerning the second step, we use several entities defined in additional SPAR ontologies, i.e., the *Citation Counting and Context Characterisation Ontology* (*C4O*)[30] [4] the *Citation Typing Ontology* (*CiTO*)[31] [14], in order to express the content of reviews and to explicitly link those to the manuscript they reviewed. In the following excerpt we introduce the formalisation in PWO of the second step of the workflow:

```
:step-two a pwo:Step ; pwo:hasNextStep :step-three ; # Reviewing step
  pwo:involvesAction :choosing-reviewers-action ,
    :reviewing-action , :reviews-notification-sending-action ;
  tisit:atTime [ a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-02-18T17:04:32"^^xsd:dateTime ;
    ti:hasIntervalEndDate "2013-04-01T05:53:24"^^xsd:dateTime ] ;
  # The review process can start only when a manuscript has been submitted
  pwo:needs swj-node:432 , :submitted-status ;
  pwo:produces :review-1, :review-2, :under-review-status, :reviewed-status.
:choosing-reviewers-action a taskex:Action ;
  dcterms:description "The editor, Giancarlo Guizzardi, chooses Csaba Veres
      and Fernando Naufel do Amaral as reviewers of the manuscript" ;
  part:hasParticipant swj:csaba-veres , swj:fernando-naufel-do-amaral ,
    swj:giancarlo-guizzardi , swj-node:432 .
:reviewing-action a taskex:Action ;
  dcterms:description "Reviewers review the manuscript" ;
  part:hasParticipant
    swj:csaba-veres , swj:fernando-naufel-do-amaral , swj-node:432 .
:reviews-notification-sending-action a taskex:Action ;
  dcterms:description "The reviews are sent to the editor" ;
  part:hasParticipant swj:csaba-veres , swj:fernando-naufel-do-amaral ,
    :review-1 , :review-2 , swj:giancarlo-guizzardi .
:review-1 a fabio:Comment ; # Review 1 by Csaba Veres
  frbr:realizationOf [ a fabio:Review ] ;
  cito:reviews swj-node:432 ; frbr:realizer swj:csaba-veres ;
  c4o:hasContent "The paper addresses a very practical..." .
:review-2 a fabio:Comment ; # Review 2 by Fernando Naufel do Amaral
  frbr:realizationOf [ a fabio:Review ] ; cito:reviews swj-node:432 ;
  frbr:realizer swj:fernando-naufel-do-amaral ;
  c4o:hasContent "The paper presents the Collection Ontology (CO)..." .
# The paper has been assigned to the under-review status for a while
:under-review-status a pso:StatusInTime ; pso:isStatusHeldBy swj-node:432 ;
```

[30] http://purl.org/spar/c4o
[31] http://purl.org/spar/cito

```
  pso:isAcquiredAsConsequenceOf :reviewing-action ;
  pso:isLostAsConsequenceOf :reviews-notification-sending-action ;
  pso:withStatus pso:under-review ; tvc:atTime [a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-02-26T12:00:07"^^xsd:dateTime ;
    ti:hasIntervalEndDate "2013-04-01T05:53:24"^^xsd:dateTime ] .
# The paper status has changed in 'reviewed' after reviewers' comments
:reviewed-status a pso:StatusInTime ; pso:isStatusHeldBy swj-node:432 ;
  pso:isAcquiredAsConsequenceOf :reviews-notification-sending-action ;
  pso:withStatus pso:reviewed ; tvc:atTime [a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-04-01T05:53:24"^^xsd:dateTime ] .
```

4.4 Decision

During the third step, the editor was responsible for the fate of the paper and provided a decision for it according to reviewers' comments. Once formalised the decision, a decision letter was sent by email to the corresponding author (i.e., Paolo Ciccarese) and the status of the paper changed in then in "minor revision". In the following excerpt we introduce the formalisation in PWO of the third step of the workflow:

```
:step-three a pwo:Step ; pwo:hasNextStep :step-four ; # Notification step
  pwo:involvesAction :decision-action , :notification-action ;
  tisit:atTime [ a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-04-01T05:53:24"^^xsd:dateTime ;
    ti:hasIntervalEndDate "2013-06-10T17:47:53"^^xsd:dateTime ] ;
  pwo:needs swj-node:432 , :review-1 , :review-2 ;
  pwo:produces :minor-revision-status , :decision-letter .
:decision-action a taskex:Action ;
  dcterms:description "The editor decides for acceptance or not" ;
  part:hasParticipant
    swj:giancarlo-guizzardi , :review-1 , :review-2 ,swj-node:432 .
:notification-action a taskex:Action ;
  dcterms:description "The editor notifies his decision to the corresponding
      author (i.e., Paolo Ciccarese)." ;
  part:hasParticipant swj:giancarlo-guizzardi , :decision-letter ,
    :review-1 , :review-2 , swj:paolo-ciccarese , swj-node:432 .
# The decision letter written by the editor
:decision-letter a fabio:Letter , fabio:Email ;
  frbr:realizationOf [ a fabio:Opinion ] cito:citesAsRelated swj-node:432 ;
  frbr:realizer swj:giancarlo-guizzardi ;
  c4o:hasContent "Dear authors, Thank you for your interest in..." .
# The minor revision status assigned to the paper after editor's decision
:minor-revision-status a pso:StatusInTime ; pso:isStatusHeldBy swj-node:432 ;
  pso:isAcquiredAsConsequenceOf :decision-action ;
  pso:withStatus swj:minorRevision ; tvc:atTime [a ti:TimeInterval ;
    ti:hasIntervalStartDate "2013-06-10T17:47:53"^^xsd:dateTime ] .
```

4.5 Revision

During the fourth step, the authors worked in order to revise the content of the previous version of the paper according to reviewers' comments and editor's suggestions. At the end of this step, the main result was the creation of a new version of the paper (i.e., *swj-node:506* in our example) that had to be submitted in the next step. In the following excerpt we introduce the formalisation in PWO of the fourth step of the workflow:

```
:step-four a pwo:Step ; pwo:hasNextStep :step-five ; # Revision step
  pwo:involvesAction :revision-action ; tisit:atTime [ a ti:TimeInterval ;
```

```
    ti:hasIntervalStartDate "2013-06-10T17:47:53"^^xsd:dateTime ;
    ti:hasIntervalEndDate "2013-07-01T05:51:30"^^xsd:dateTime ] ;
  pwo:needs swj-node:432 , :decision-letter , :review-1 , :review-2 ;
  pwo:produces swj-node:506 .
:revision-action a taskex:Action ;
  dcterms:description "The authors revises the paper" ;
  part:hasParticipant swj-node:432 , :decision-letter ,
    :review-1 , :review-2 , swj:silvio-peroni , swj:paolo-ciccarese .
```

5 Conclusion

In this paper we introduced the *Publishing Workflow Ontology* (*PWO*), i.e., an OWL 2 DL ontology part of the Semantic Publishing and Referencing (SPAR) Ontologies, which allows the description of publishing workflows in RDF. The whole ontology is entirely based on existing ontology design patterns that allowed us to model the various aspects of workflows in an appropriate and standardised way. We showed a particular use of PWO for describing the first steps of a real publishing workflow concerning the publication of an article of the Semantic Web Journal, i.e., [3], in which we reused entities and data coming from several models and data, e.g., other SPAR ontologies and existing resources from the Semantic Web Journal Linked Dataset.

Although PWO had been thought in principle to describe publishing-related workflows, it has been developed on purpose as an ontology for the description of generic workflows. In future we plan to align it to other workflow-related models, e.g., PROV-O, the Research Object ontology and the other ontologies described in Section 2. In addition, we are currently studying the applicability of PWO in the legal and scientific domains. In particular, we plan to work on its use for describing workflows that concern the process of codification of the laws of the United States legislation and the series of computational or data manipulation steps in scientific applications.

References

1. Belhajjame, K., Corcho, O., Garijo, D., Zhao, J., Missier, P., Newman, D. R., ... Goble, C. (2012). Workflow-Centric Research Objects: A First Class Citizen in the Scholarly Discourse. In Proceedings of the 2nd Workshop on Semantic Publishing (SePublica 2012). http://ceur-ws.org/Vol-903/paper-01.pdf
2. Belhajjame, K., Zhao, J., Garijo, D., Hettne, K. M., Palma, R., Corcho, O., ... Goble, C. A. (2014). The Research Object Suite of Ontologies: Sharing and Exchanging Research Data and Methods on the Open Web. The Computing Research Repository (CoRR), abs/1401.4307. http://arxiv.org/abs/1401.4307
3. Ciccarese, P., & Peroni, S. (2013). The Collections Ontology: creating and handling collections in OWL 2 DL frameworks. Semantic Web. DOI: 10.3233/SW-130121
4. Di Iorio, A., Nuzzolese, A. G., Peroni, S., Shotton, D., & Vitali, F. (2014). Describing bibliographic references in RDF. In Proceedings of 4th Workshop on Semantic Publishing (SePublica 2014). http://ceur-ws.org/Vol-1155/paper-05.pdf
5. Falco, R., Gangemi, A., Peroni, S., & Vitali, F. (2014). Modelling OWL ontologies with Graffoo. In ESWC 2014 Satellite Events - Revised Selected Papers.

6. Gangemi, A., Borgo, S., Catenacci, C., & Lehmann, J. (2004). Task taxonomies for knowledge content. METOKIS Deliverable D7. http://metokis.salzburgresearch.at/files/deliverables/metokis_d07_task_taxonomies_final.pdf
7. Garijo, D., & Gil, Y. (2011). A new approach for publishing workflows: abstractions, standards, and linked data. In Proceedings of the 6th workshop on Workflows in support of large-scale science (WORKS 2011): 47–56. DOI: 10.1145/2110497.2110504
8. Hettne, K., Soiland-Reyes, S., Klyne, G., Belhajjame, K., Gamble, M., Bechhofer, S., ... Corcho, O. (2012). Workflow forever: semantic web semantic models and tools for preserving and digitally publishing computational experiments. In Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences (SWAT4LS 2011): 36–37. DOI: 10.1145/2166896.2166909
9. Horridge, M., & Patel-Schneider, P. F. (2012). OWL 2 Web Ontology Language: Manchester Syntax (Second Edition). W3C Working Group Note, 11 December 2012. http://www.w3.org/TR/owl2-manchester-syntax/
10. Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, 21 May 2004. http://www.w3.org/Submission/SWRL/
11. Hu, Y., Janowicz, K., McKenzie, G., Sengupta, K., & Hitzler, P. (2013). A Linked-Data-Driven and Semantically-Enabled Journal Portal for Scientometrics. In Proceedings of the 12th International Semantic Web Conference (ISWC 2013): 114–129. DOI: 10.1007/978-3-642-41338-4_8
12. Lebo, T., Sahoo, S., & McGuinness, D. (2013). PROV-O: The PROV Ontology. W3C Recommendation, 30 April 2013. http://www.w3.org/TR/prov-o/
13. Noy, N. F., Chugh, A., Liu, W., & Musen, M. A. (2006). A Framework for Ontology Evolution in Collaborative Environments. In Proceedings of the 5th International Semantic Web Conference (ISWC 2006): 544–558. DOI: 10.1007/11926078_39
14. Peroni, S., & Shotton, D. (2012). FaBiO and CiTO: Ontologies for describing bibliographic resources and citations. Web Semantics,17: 33–43. DOI: 10.1016/j.websem.2012.08.001
15. Peroni, S., Shotton, D., & Vitali, F. (2012). Scholarly publishing and linked data: describing roles, statuses, temporal and contextual extents. In Proceedings of the 8th International Conference on Semantic Systems (i-Semantics 2012): 9–16. DOI: 10.1145/2362499.2362502
16. Prud'hommeaux, E., & Carothers, G. (2014). Turtle - Terse RDF Triple Language. W3C Recommendation, 25 February 2014. http://www.w3.org/TR/turtle/
17. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., & Mulyar, N. (2006). Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22. http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf
18. Sebastian, A., Noy, N. F., Tudorache, T., & Musen, M. A. (2008). A Generic Ontology for Collaborative Ontology-Development Workflows. In Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008): 318–328. DOI: 10.1007/978-3-540-87696-0_28
19. Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., ... Goble, C. (2013). The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Research, 41 (W1): W557–W561. DOI: 10.1093/nar/gkt328