**Responses to Comments from Reviewer 1**

**NOTE:** The following comments are responded in the revised manuscript using **Red** coloured text and the same are copied here.

**Reviewer’s Comments and Suggestions for Authors:**

The manuscript proposes a model to capture the provenance about the control flow of scientific workflows. The authors argue that the proposed model captures accurate and sufficient provenance to reproduce scientific workflows.

The proposed model is an interesting and important extension to the standard W3C PROV model. The paper explains the model well and provides examples that show the missing components and their importance in existing provenance models. I have some minor comments, as listed below:

**Response:**

We are thankful to the reviewer for the suggestions and the recommendation to improve the quality of this work. We have responded to the suggestions in the following way:

***Comment 1.1:***

i) The authors claim that the model is extendable for other patterns. It is not clear how the model is extendable for other patterns that are not listed in Table 1 or are not supported by current SWfMSs. Are the defined concepts enough? Are there any special control operators that are required to support the other patterns?

***Response:*** We have mentioned in our paper (see line 174-177 on Page 5):

* “It is worth noticing, we are aiming for a model which can specify most of the listed patterns in Table 1 and is extendable for other patterns. However, in this paper, we discuss and provide provenance description for the aforementioned eleven requirements only.”

Our focus is on patterns that Scientific Workflow Management Systems (SWfMSs) support for the scientific workflows. However, we provide sample provenance description of two patterns that are not supported by current SWfMSs at <https://github.com/anilabutt/control-flow-model/blob/master/swcf-aligned-prov/control-flow-patterns/or-operators.pdf>. These provenance descriptions are a proof of concept that the proposed model can describe patterns other than the core requirements, set for our work presented in this paper, just by using another control operator. However, to consider other patterns ( i.e., business workflow patterns or the ones that are not supported by the current major SWfMSs) we plan to extend and evaluate Scientific workflow Control-flow (SWCf) model in future and same is mentioned in the paper (see line 629-631 on Page 21).

* “In future work, we will consider specifying all identified control-flow patterns using or extending our model to make SWCf model equally effective provenance model for the scientific as well as business workflows.”

***Comment-1.2:*** Do the authors mean that the workflow execution patterns can also be supported by the proposed model?

**Response:** The model proposed in this article is aimed for capturing workflow specification patterns only. We argue that the missing control constructs semantics affect only prospective provenance and our model can capture workflow specification patterns only. However, to capture the execution provenance, the proposed model can be integrated with the existing provenance models e.g., ProvOne, Triana, Triverna. The following text is included in the paper to make understandings clearer (see line 92-97 and line 105-109 on Page 3) .

* “The retrospective provenance of a control-flow workflow does not include workflow control constructs as the execution trace itself typically specifies the activity and the discrete-time points at which the activity has happened. Therefore, overlooking control constructs in existing provenance models does not affect their ability to capture the retrospective provenance (i.e., the information about a workflow execution). However, it results in underspecified workflows (i.e., incomplete prospective provenance) that raise several problems”
* “To address these problems, we identified control-flow patterns that a model must be able to describe to be an acceptable model for scientific workflows prospective provenance. Further, we present a provenance model to capture prospective provenance for scientific workflows that can specify control-flow patterns and provide the means to integrate this model with existing provenance models.”

***Comment-2:*** ii) (Page 6) It is not clear why ProvOne cannot specify some control patterns.

**Response:** The text has been changed in the revised manuscript to highlight ProvONE’s limitations in context of specifying some control patterns. These changes are included in the paper (see line 200-205 on Page 7) as:

* “The parallel split and synchronisation patterns, which represent an unconditional dataflow ‘to’ and ‘from’ multiple tasks, can be modelled using the ‘Controller’ class in accordance with the cardinality constraints defined with ‘Program’ in ProvONE. However, ProvONE is unable to define other control patterns (i.e., R4, ..., R11) because it lacks a class or property to model the conditional constructs like if/else or case structure.”

***Comment-3:*** iii) How is the definition of swcf:Task different from prov:Activity? What uniqueness does the task concept bring to the model that prov already does not provide?

**Response:** The changes to address this comment can be seen on line 244-249 on Page 8. These changes are as follows:

* “A Task in SWCf represents a set of actions or steps that consume and(or) produce data to achieve some goals. In our proposed model, swcf:Task is a sub class of prov:Plan and prov:Entity rather a prov:Activity (i.e., swcf:Task prov:Plan prov:Entity). It represents a plan within a workflow, which always processes data and generates the output. An actual execution or processing of the task that occurs over a period of time and acts upon the steps defined as Task would be a prov:Activity.”

***Comment-4.1:*** iv) The evaluation section explains some operators (and-join, xor-split et.) in the evaluation section. Though the operators have been introduced in Section 3.5., they are not explained in detail there. It will be better to explain all possible operators in that section.

**Response:** We have added the following text in Section 3.5 to explain control operators (see line 379-388 on Page 11) .

* “A ControlOperator class is created to define the behaviour of the controllers within a ControllerCollection. Each ControllerCollection has a ControlOperator. In SWCf, we model the behaviour of controllers in-terms of logical operators. In this current version, we have defined six operators i.e., AND-JOIN, AND-SPLIT, XOR-JOIN, XOR-SPLIT, OR-JOIN, and OR-SPLIT. AND-JOIN defines that all Controllers in ControllerCollection should execute before the execution of a subsequent task. AND-SPLIT defines that all Controllers in ControllerCollection should execute and the execution may occur in parallel. XOR-JOIN defines that a subsequent task should enable right after one of the member Controller of the preceding ControllerCollection enables. XOR-SPLIT defines that only one Controller from ControllerCollection should enable on the basis of the output of ControlTask as a part of the workflow execution. OR-JOIN defines that the subsequent task should enable right after the completion of all enabled member Controller of the preceding ControllerCollection. OR-SPLIT defines that one or more Controller from ControllerCollection may enable on the basis of the output of ControlTasks as a part of the workflow execution.”

***Comment-4.2:*** It will also be worth mentioning the fan-in and fan-out of these operators.

**Response:** Thanks for bringing this aspect into attention. Fan-in and fan-out of these operators is implicit and depends mostly on the tasks involved in a pattern. Apply explicit restrictions on fan-in and fan-out would limit the usability and extendibility of the proposed model. Therefore, we leave it to the users to define such restrictions according to the application of this model.

***Comment-4.3:*** Also, can the authors prove that the six operators are sufficient to specify all the control patterns mentioned in the paper?

**Response:** We provide SWCf aligned provenance description for control-flow patterns mentioned in the paper. The provenance description proves that the operators defined in this article are enough to specify all the control patterns mentioned in Section 2.1.

Please refer to Section 5.1 for a detailed evaluation. The relevant text is highlighted red in the paper (see line 447- 485 on Page 13).