Collaborative provenance for workflow-driven science and engineering

Altıntaş, İ.

Publication date
2011

5

Collaborative Provenance: A Definition

"Great discoveries and improvements invariably involve the cooperation of many minds. I may be given credit for having blazed the trail, but when I look at the subsequent developments I feel the credit is due to others rather than myself."

– Alexander Graham Bell

The lifecycle of scientific workflows—which includes the design, execution, sharing, and management of data and provenance products—depends not only on the workflow itself, but also the overall scientific research infrastructure and scientific collaborations within which scientists use these workflows. Without this thinking, we cannot expect the real users, i.e., *scientists*, to adopt scientific workflows as a solution for problem solving. To completely benefit from scientific workflows, environments that support workflow-based research must be in tune with the way researchers work and collaborate.

5.1 Collaborative Provenance

Within their daily process, scientists who work with workflows run multiple workflows using datasets from various resources. They often download workflows in their private space online or offline, experiment with it, change a workflow or its parameters, and publish their results when they are ready to share the workflow or its outputs. When running workflows, researchers check the status of their workflows, resubmit workflows to try new data and parameters, and create reports of past workflow runs they have conducted. They validate and share the results by linking to online or local archives. In reality, sharing occurs between closed groups of collaborators until the results of the study are mature enough to be published for public access. Once a workflow product is published, more workflows can use it as input data. This creates a chain of custody for scientific data that is beyond the execution of a

[0]This chapter is based on (Altintas *et al.* 2010*e*) co-authored by Altintas.

single workflow that can be linked using the information within provenance information collected through a collaborative platform. In this chapter, we define a *collaborative provenance* notion which is based on three of the above-mentioned user actions, namely, users publishing data, users publishing workflows, and users running workflows. A goal of this approach is to extend the current single-workflow and single-user targeted provenance approach to a number of workflow runs within a controlled environment such as a website community portal for sharing data and workflows. This new approach puts user actions and collaborations in the center of the conducted research independent of computational technologies used to generate results.

Underlying assumptions in such a system for the goals of this thesis are:

- All information, i.e., data, workflows and information related to workflow runs, is public. Information privacy is out of the scope of this thesis.
- Data and workflows within the system are globally identified by a common namespace and identifier throughout the system or through a network of cooperating repositories.
- Model of provenance is shared between different workflow systems and conforms to a global repository of data artifact identifiers, i.e., an artifact produced by one workflow system and consumed by another can be uniquely identified through the provenance repository.

Run Interoperability. A scientific discovery can be the result of executions of many workflows which might be created and executed in different workflow systems. In such cases, the provenance information related to workflows executing in different systems might be recorded in different provenance models, which might not conform to each other. Also, if the execution of a workflow depends on the data produced by the execution of previous workflow runs executed in different system, then we need a mechanism where the provenance of one system can talk to the provenance of the other system. To promote and facilitate exchange of information between multiple workflow systems supporting provenance, the Open Provenance Model (OPM) has been proposed by the scientific workflow community. Further, a query mechanism to access provenance information related to multiple inter-related workflow executions is also required for addressing interoperability. In (Altintas *et al.* 2010*d*), we presented an architecture that seeks to overcome this issue through a querying engine. We will revisit the interoperability issue in Chapter 9.

5.2 Collaborative Scenario

Figure 5.1 shows different observables of shared data, workflows and workflow executions (runs) in a typical scientific research project, where all (or part of) the output of workflow runs can be used as input to subsequent runs. Figure 5.1(a) shows that data $\{d_1, d_2, d_3, d_7\}$ are published by users $\{u_3, u_5, u_6\}$ in a shared data store, where u_3 published d_2, u_6 published $\{d_1, d_3\}$ and u_5 published d_7. Figure 5.1(b) shows workflows $\{wf_1 \ldots wf_5\}$ being published

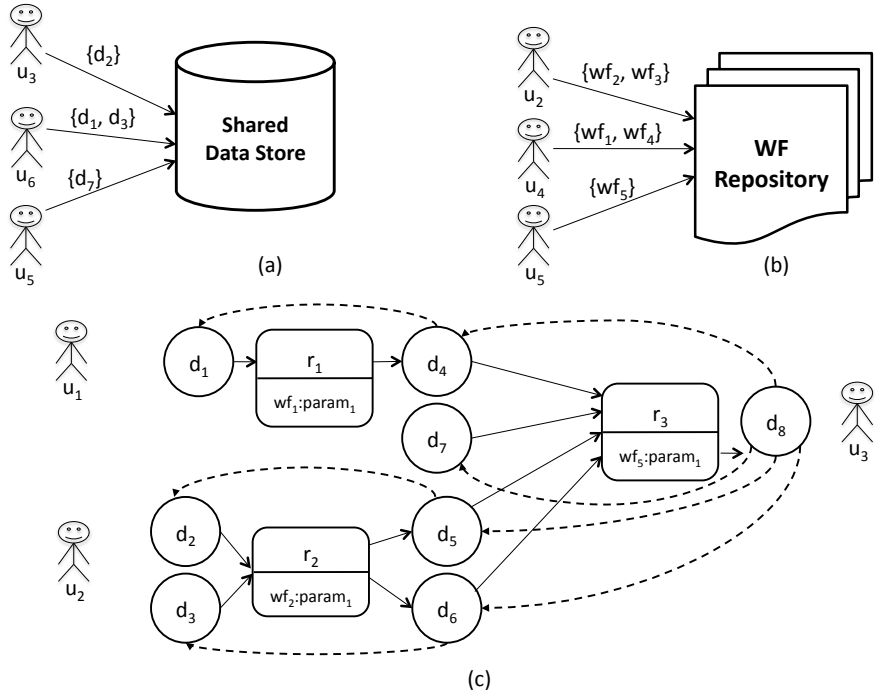


Figure 5.1: Different observables of shared data, workflows and workflow executions (runs) in a typical scientific research project, where all (or part of) the output of workflow runs can be used as input to subsequent runs: (a) data $\{d_1, d_2, d_3, d_7\}$ published by users $\{u_3, u_5, u_6\}$; (b) workflows $\{wf_1 \ldots wf_5\}$ published by users $\{u_2, u_4, u_5\}$; and (c) flow graph for workflow runs (customized through their parameters) and related provenance data $\{d_1 \ldots d_8\}$ in user space $\{u_1, u_2, u_3\}$.

by users $\{u_2, u_4, u_5\}$ in a workflow repository. Figure 5.1(c) shows a flow graph for workflow runs (customized through their parameters) and related provenance data $\{d_1 \ldots d_8\}$ in user space $\{u_1, u_2, u_3\}$. Here, user u_1 performs a run r_1 of workflow wf_1 with parameter settings $param_1$ and input dataset d_1. Run r_1 produces data d_4 as its output, and dependencies between the outputs and inputs of run r_1 is shown with dashed link. Similarly, user u_2 performs a run r_2 of workflow wf_2 with parameter settings $param_1$ and input dataset d_2 and d_3. Run r_2 produces data d_5 and d_6 as its outputs, and dependencies between the outputs and inputs of run r_2 is shown with dashed link. Here, d_5 depends on d_2, and d_6 depends on d_3. User u_3 performs a run r_3 of workflow wf_5 with parameter settings $param_1$. Run r_3 uses data from previous runs d_4 from r_1 and d_5, and d_6 from run r_2, along with other published data d_7. Run r_3 produces d_8 as its output, where d_8 depends on d_4, d_5, d_6, d_7.

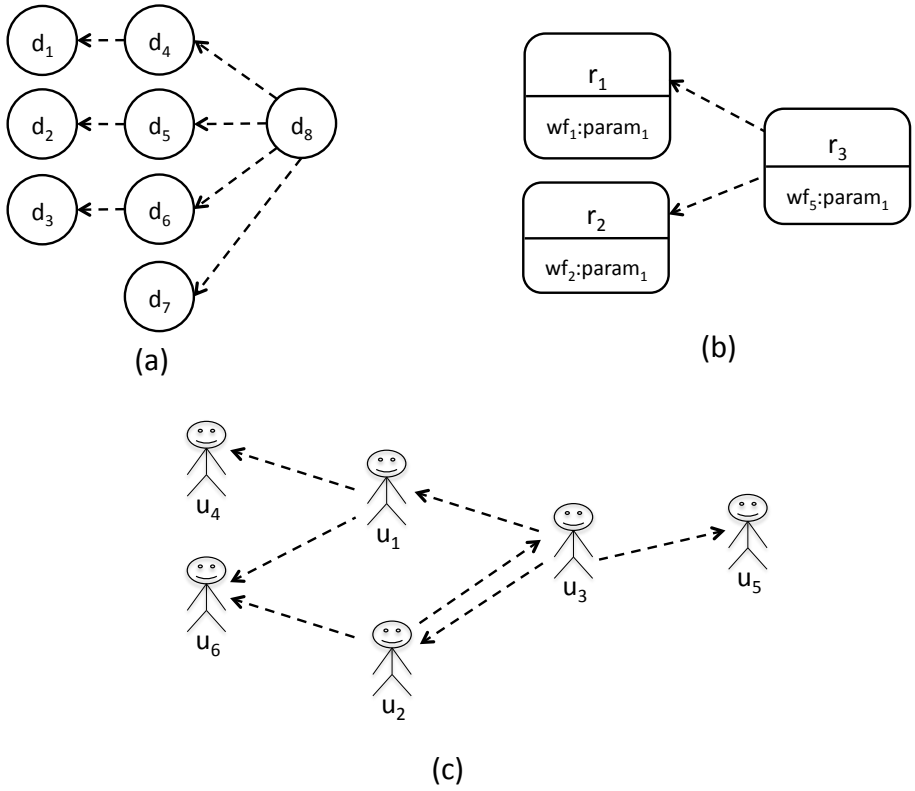


Figure 5.2: Various collaborative provenance views: (a) data dependency view; (b) run dependency view; and (c) user collaboration view (based on Figure 5.1).

In Figure 5.1(c), a run node identifies the provenance of a previous workflow run and the data dependencies between inputs and outputs of workflow execution are shown by dashed links between data nodes. One can identify the flow of workflow executions leading to a data artifact that is published as a "scientific discovery" by chaining together the interrelated runs (where outputs of runs can be used as inputs to other runs). The provenance information related to all these activities is captured in the common provenance store (see Figure 1.1).

Users who performed the workflow runs or used published data start a *collaboration* with those users who published these entities. This collaboration could be "implicit", i.e., the user who published the workflows might not know that his public workflow or data is being used by other users, or "explicit", i.e., the user who published the data or workflows is informed of their usage by other users in advance. One can identify the exact flow of workflow executions leading to a data artifact that is published as a *scientific discovery* by chaining together the interrelated runs performed by users. Using the runs as a connector in a three- dimensional space of users, workflows and data, we can generate collaborative provenance views based on the provenance of workflow runs.

5.3 Building Collaborative Provenance Views

The history of workflow runs in different user spaces $\{u_1, u_2, u_3\}$ is depicted in Figure 5.1(c) which shows the usage of published data in Figure 5.1(a) and published workflows in Figure 5.1(b). Using this extended information, we can generate views of data dependency, run (workflow execution) dependency and user collaboration, as seen in Figure 5.2(a), Figure 5.2(b), and Figure 5.2(c) respectively. Also, note that while *data dependency view* and *run dependency view* are directed and transitive, *user dependency view* is directed and non-transitive.

A *data dependency view* shows dependencies between outputs and inputs of workflow runs, which may span across multiple related workflow runs. Figure 5.2(a), shows the data dependency view for workflow execution scenario as shown in Figure 5.1(c). Here, d_8 which is the result of execution of run r_3 depends on its input dataset d_4, d_5, d_6 and d_7. d_4 is the result of execution of r_1 which depends on input data d_1. d_5 and d_6 are the result of execution of run r_2 which depends on input dataset d_2 and d_3 respectively. Combining together the entire chain of dependency gives us the complete data dependency view, spanning across data of multiple related workflow runs.

A *run dependency view* shows dependencies between runs depending on whether the runs used the output of previous runs as their inputs. Figure 5.2(b), shows the run dependency view for workflow execution scenario as shown in Figure 5.1(c). Here, run r_3 used the output data of runs r_1 (d_4) and r_2 (d_5 and d_6) as part of its input dataset, resulting in a run dependency view where r_3 depends on r_1 and r_2.

A *user collaboration view* shows whether users used entities (data, and workflows) published by other users during workflow executions. Figure 5.2(c), shows the user collaboration view for published data, published workflow and workflow execution based on the scenario as shown in Figure 5.1. User u_3 performs run r_3 where he executes workflow wf_5 with input dataset d_4, d_5, d_6, and d_7. Figure 5.2(c), shows that u_3 has three collaborative dependencies with users u_1, u_2, and u_5. He depends on: (i) u_1 as he used d_4 that was generated by r_1, which was executed by u_1; (ii) u_2 as he used d_5 and d_6 that were generated by r_2, which was executed by u_2; and (iii) u_5 since he used workflow wf_5 which was published by u_5, also he used d_7 as input which was published by u_5. User u_1 performs run r_1 where he executes workflow wf_1 with input data d_1. User u_1 has two collaborative dependencies with users u_4, and u_6. He depends on:(i) u_4 as he used workflow wf_1 which was published by u_4; and (ii) u_6 as he used d_1 as input during his workflow run r_1. User u_2 performs run r_2 where he executes workflow wf_2 with input data d_2, and d_3. User u_2 has two collaborative dependencies with users u_3, and u_6. He depends on:(i) u_3 as he used d_2, which was published by u_3; and (ii) u_6 as he used d_3, which was published by u_6. Note that there could be cycles, i.e., mutual dependency relationships, in a user collaboration view, but not all of these are true cycles. However, in the context of a collaboration that leads to a data artifact, user dependencies could really cycle. In the scenario shown in in Figure 5.1, to generate d_8, u_3 depends on a run by u_2 that depended on a data published by u_3. This creates a cyclic dependency between

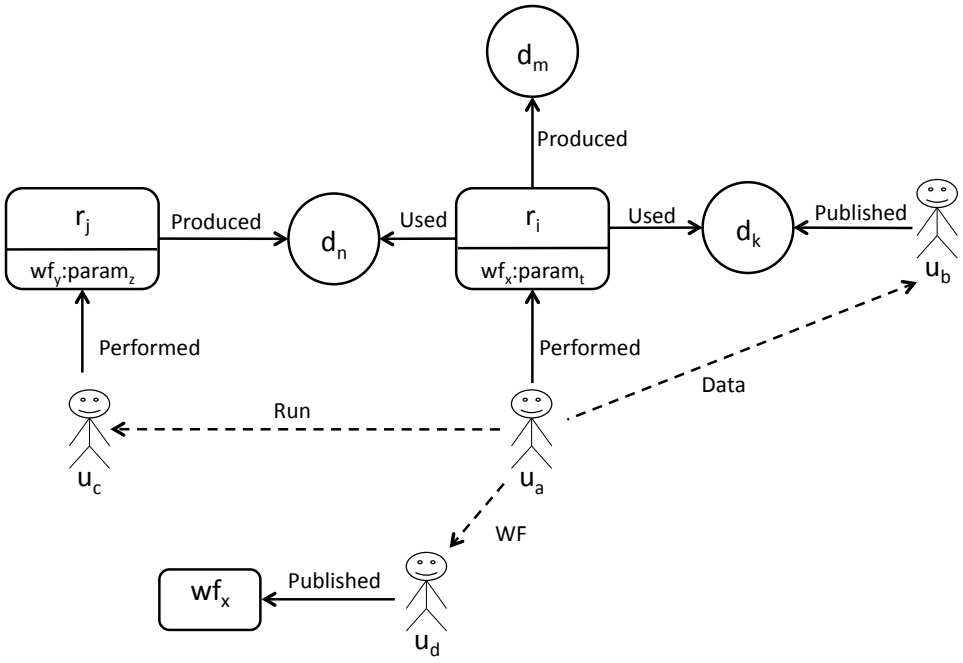


Figure 5.3: A collaborative provenance model where a user can share collaborations with other users when he performs a workflow execution and uses data and workflow published by other users (collaborations are shown with directed dashed lines).

two users.

5.4 Analyzing User Collaborations

Until now, we concentrated on a simpler form of collaborative provenance user views as a non-transitive directed graph with edges that shows the direction of collaborations between users (Altintas *et al.* 2010*d*). Below, we explain collaborative provenance attributes that can be used to provide more explicit information about these collaborations. Before we describe these specific attributes, in Figure 5.3, we present a model that shows all the collaborative relationships that a user can share with other users based on a workflow run, where he uses the data and workflow published by other users. Although, it is possible to identify other relationships in a collaborative environment, these relationships capture workflow publishing, data publishing and workflow execution related actions, and will serve as the basis of our model and examples throughout this thesis. A type of relationship that relates to workflow design is workflow attributions in the case that users extend and publish a dependent of a workflow or use sub-workflows from other users. Since we limit our provenance information only to workflow runs and don't have provenance information related to workflow design,

we left this relationship out of the model. We also assume that a workflow or data artifact is designed by a single user. The model can be extended to add more users and co-working relationships.

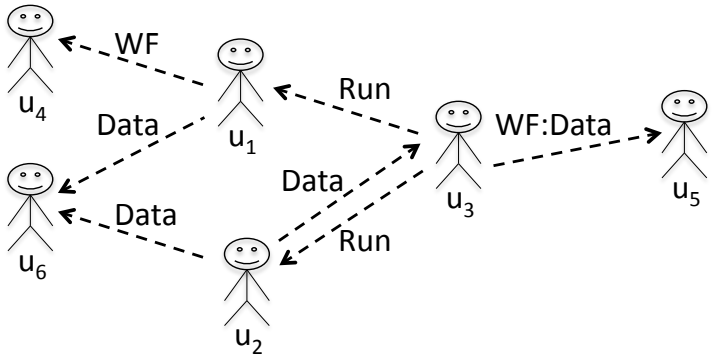
In Figure 5.3, a *workflow collaboration* (WF) is established between two users whenever the first user (u_a) executes a workflow (wf_x) that is published by the second user (u_d). A *data collaboration* (Data) is established between two users whenever the first user (u_a) performs a run (r_i) in which the input of the run (d_k) that is published by the second user (u_b). The third rule states that a *run collaboration* (Run) is established between two users whenever the first user (u_a) performs a run (r_i) in which the input of the run was generated by a run (r_j) performed by the second user (u_c). Please note that users collaborate with each other only when they execute a workflow with datasets, where either the workflow or datasets (or both) have been published by other users. In addition, there exists a special case of self-collaboration when a user uses his/her previous datasets or workflows. Based on this collaborative provenance model, we introduce three attributes, *nature*, *weight* and *self*, on top of the simple user collaboration view depicted in Figure 5.2(c). These attributes enable us to explore and analyze the user collaborations more deeply.

5.4.1 Nature of Collaboration

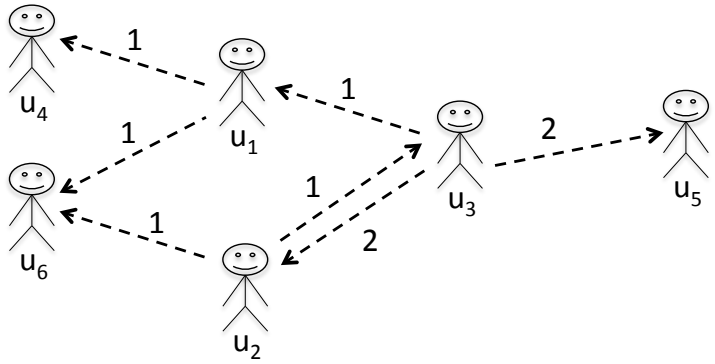
As shown in Figure 5.2(c), all the collaborations from a user to another user are reduced to a single directed edge, represented by a dashed arrow. This edge may represent multiple collaborations between the same set of users which might be based on WF, Run, and Data collaborations. We label the collaboration edges to explicitly denote the nature of collaboration between user nodes. Figure 5.4(a) shows such a user collaboration graph where the edges are labeled to denote the nature of collaboration, WF, Run and Data, e.g., collaboration edge between u_2 and u_3 is labelled as Run. Also, collaboration edge between u_2 and u_6 is labelled as Run. When there are multiple collaborative activities from one user to the other, the labels are separated with a colon ":" to explicitly denote all the collaborations between them, e.g., collaboration edge between u_3 and u_5 is labelled as WF:Data to denote that u_3 shares both WF and Data collaborations with u_5. We denote such a user collaboration graph where the edges are explicitly labelled with the nature of collaboration as C_N.

5.4.2 Weight of Collaboration

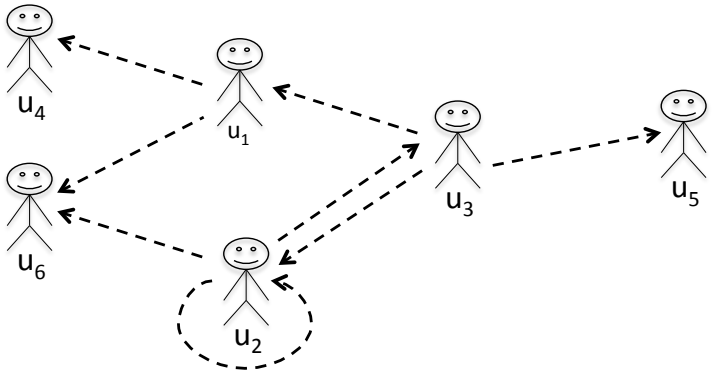
Each user collaboration edge can be assigned a weight which shows the strength of collaborations between two users based on the number of collaborative dependencies from one user to another. Each collaboration is assigned a value "1" irrespective of the nature of collaboration. Thus, the weight of the dependency between u_x to u_y is proportional to the number of collaborations between them. Figure 5.4(b) shows such a user collaboration graph where the edges are labeled to denote the weight of collaboration, e.g., edge from u_3 to u_2 has weight "2" based on the Data collaborations caused by dependency of d_8 to d_5 and d_6, where both d_5 and d_6 are published by u_2 (see Figure 5.2). Similarly, the weight of the edge from u_3



(a)



(b)



(c)

Figure 5.4: Different user collaboration graph where attributes and edges show the direction and (a) nature (C_N); (b) weight (C_W); and (c) self-collaboration (C_S) aspect, of user collaborations based on the observables in Figure 5.1.

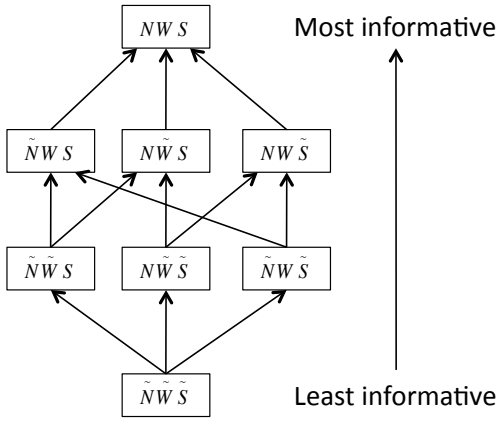
to u_5 is "2" based on the Data collaboration caused by dependency of d_8 to d_7, where d_7 was published by u_5, and WF collaboration, as u_3 ran workflow wf_5 published by u_5 (see Figure 5.2). We denote such a user collaboration graph where the edges are explicitly labelled with the weight of collaboration as C_W.

5.4.3 Self Collaboration

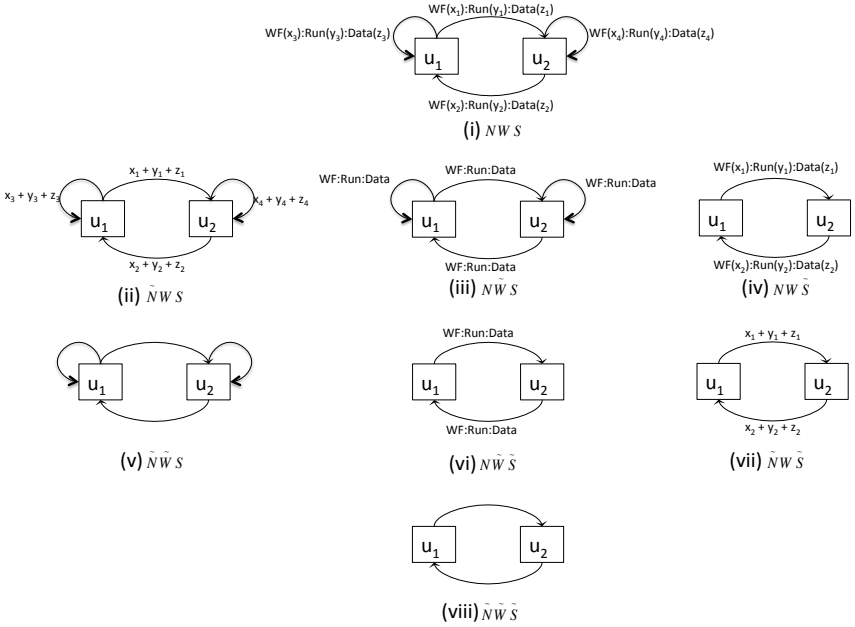
A self collaboration is a special case of collaboration where a user uses a self-published dataset, a self-published workflow or a data item that was generated by one of his previous runs, in other words, when the user establishes a collaboration relationship with himself, as he is using his own published data and workflows. Figure 5.4(c) shows such a user collaboration graph where there is "loop" on the user to denote self collaboration, e.g., u_2 has a "loop" on himself, as he uses his own published workflow wf_2 during execution of r_2. Self collaboration is by default disabled in a collaborative graph and can be activated to keep track of a user's independent research or to show how much a user made use of the workflows and data published by himself. We denote such a user collaboration graph where self collaboration is explicit as C_S.

5.5 Combining User Collaborations Attributes

The section above describes various user collaboration graphs with explicit attributes: nature (C_N), weight (C_W), and self collaboration (C_S) (see Figure 5.4). These distinct user collaboration graphs are more informative than that was shown in Figure 5.2(c), where the attributes of collaborations between users are not considered. However, the user collaboration graph can be even more informative when these attributes are combined in the same user collaboration graph. Figure 5.5 shows a bottom up view from the least informed (Altintas *et al.* 2010*d*) to most informed user dependency views. In Figure 5.5(a) N stands for the existence of the nature of collaboration attribute. Similarly, W represents the strength (weight) of collaboration attribute and S represents the self collaboration attribute. Here weight stands for the number of specific collaborations between two users per nature type when combined with the nature attribute and the sum of counts of all collaboration types when the nature attribute is disabled. $\tilde{N}$, $\tilde{W}$ and $\tilde{S}$ are used to denote the lack of these attributes respectively. NWS denotes a user collaboration graph which has all the three attributes (nature, weight, self) in the same graph, which is the most informative. Similarly, $\tilde{N}\tilde{W}\tilde{S}$ denotes a user collaboration graph which has neither of these three attributes (nature, weight, self) in the same graph. Such a graph looks like as shown in Figure 5.2(c), which is the least informative. The arrows in this figure show collaboration view transitions to add more information by enabling one more attribute. Transitions that require more than one attribute change, e.g., $\tilde{N}\tilde{W}S$ and $NW\tilde{S}$, are not shown for simplicity, but such collaborative provenance queries are possible as shown in Figure 5.5(b). Likewise, transitions between the nodes at the same level of the graph, e.g., $\tilde{N}W\tilde{S}$ and $\tilde{N}\tilde{W}S$, are not shown with arrows since these nodes are considered



(a) Combined user collaboration graphs: from C_{NWS} (least infomative) to $C_{\tilde{N}\tilde{W}\tilde{S}}$ (most informative)



(b) Combined user collaboration graphs for two users u_1 and u_2 (C_{NWS} to $C_{\tilde{N}\tilde{W}\tilde{S}}$)

Figure 5.5: (a) A bottom up view over collaborative provenance attribute combinations from least informative to most informative. (b) Detailed views of possible collaborations between two users.

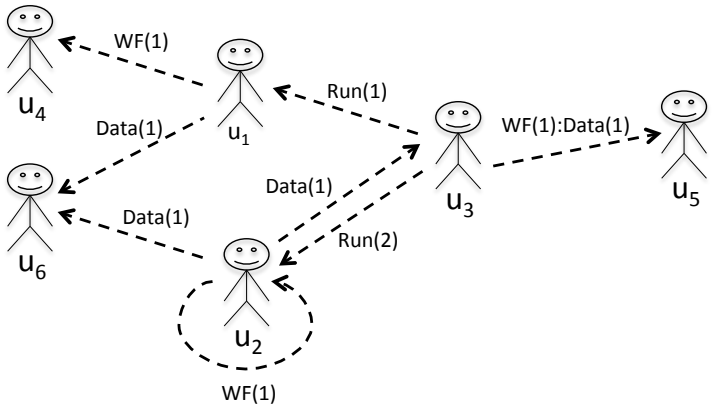


Figure 5.6: A collaboration graph where all the attributes of user collaborations are shown based on the observables for the scenario in Figure 5.1.

as equally informative containing information on an equal number of attributes.

We use Figure 5.5(b) to illustrate the feature of combining attributes of user collaborations further. Different singleton figures in Figure 5.5(b) map to corresponding positions to that of Figure 5.5(a). $\tilde{N}\tilde{W}\tilde{S}$ corresponds to (viii) where u_1 collaborates with u_2 and vice-versa. $\tilde{N}\tilde{W}S$ corresponds to (v) where it is as informative as (viii), in addition it shows self collaboration, i.e., both u_1 and u_2 collaborates with self. $N\tilde{W}\tilde{S}$ corresponds to (vi) which is as informative as (viii), in addition it shows the nature of collaboration, i.e. both u_1 and u_2 collaborate with each other as WF:Run:Data. $\tilde{N}W\tilde{S}$ corresponds to (vii) which is as informative as (viii), in addition it shows the weight of collaboration, i.e., $x_1 + y_1 + z_1$ as weight from u_1 to u_2, and $x_2 + y_2 + z_2$ as weight from u_2 to u_1, where x_i is the number of WF collaborations, y_i is number of Run collaborations, and z_i is the number of Data collaborations. $\tilde{N}WS$ corresponds to (ii) which is as informative as (v), in addition it shows the weight for each collaboration as $x_i + y_i + z_i$. $N\tilde{W}S$ corresponds to (iii) which is as informative as (vi), in addition it shows the self collaboration. $NW\tilde{S}$ corresponds to (iv) which is as informative as (vii), in addition it shows the nature for each collaboration along with weight as WF (x_i):Run (y_i):Data (z_i). Finally, NWS corresponds to (i), which is as informative as any of any of (ii), (iii), and (iv), and is basically union of (ii), (iii), and (iv). Note that Figure 5.5(b) shows all the attributes that will have an important role in the application of the proposed model, where each sub-model covers a specific area of the collaborative provenance query space, and thus making it much easier to answer such queries.

Figure 5.6 shows a C_{NWS} user collaboration graph, where all the attributes of user collaborations are shown. For instance, edge between u_3 and u_5 is labeled as WF(1):Data(1) to denote that u_3 shares one workflow and one data collaboration with u_5. Similarly, u_2 shares a self collaboration of type workflow with weight one. Figure 5.6 also shows a Run collaboration from u_3 to u_2 with weight 2 to indicate that two of the inputs to workflow runs of u_3 was generated by workflow runs by u_2.

Table 5.1: Example queries across workflow executions and collaborations

Q1	Which data artifacts were used explicilty or implicitly to generate data artifact d?
Q2	What is the data dependency graph that led to data artifact d?
Q3	Which runs were used in the generation of a data artifact d?
Q4	What is the run dependency graph that led to data artifact d?
Q5	If data artifact d is detected to be faulty, which runs were affected by d?
Q6	Which users depended on data artifact d?
Q7	Which user collaborations were involved in the derivation of artifact d_2 from artifact d_1?
Q8	Who are the potential acknowledgements for a publication of a data artifact d?

5.6 Example Collaborative Query Usecases

There are multiple scenarios and related provenance queries that collaborative provenance model makes possible to answer. These include standard provenance queries as well as queries that involve user collaborations. A number of scenarios shown in Table 5.1 were explained in detail in (Altintas *et al.* 2010*d*). In this section, we discuss the following two example scenarios in detail.

5.6.1 Acknowledgement List for Collaborators

The first example usecase is the acknowledgement of a scientific result, i..e., *data artifact*, coming out of a workflow run before it's publication in a scientific paper. Figure 5.7 shows a scenario for how an acknowledgement map for a specific data product can be generated based on WF, Data and Run collaborations. In this example, we assume that u_2 is interested in publishing the results of his run, r_2, and he is querying for the users he collaborated with in order to potentially acknowledge them in his paper. When he performs this query, e.g. through a collaborative provenance browser in the platform he's on, he gets back the user view on top that shows all the users that $\{u_2\}$ depended on to generate $\{d_5\}$ (See Figure 5.7(a)). In this case, we assume that C_{NWS} is enabled. Based on this user view, u_2 can already have a good idea about how much and in what context he worked with u_3 and u_6, and that he used a workflow of his own. If u_2 needs more information about each of these dependencies, he can explore the provenance views further. For example, the data view in Figure 5.7(c) shows that the Data(1) edge from u_2 to u_6 is inferred based on the data dependency from d_5 to d_3. Note that, u_2's r_2 didn't depend on the results of any other runs in this simple example, so in the run view (Figure 5.7(b)) no additional information other than the run, i.e., r_2, that generated d_5 is provided. Once u_2 analyzes and understands the context of these collaborations, their nature and strength, he may or may not choose to acknowledge u_3 and u_6 in his article.

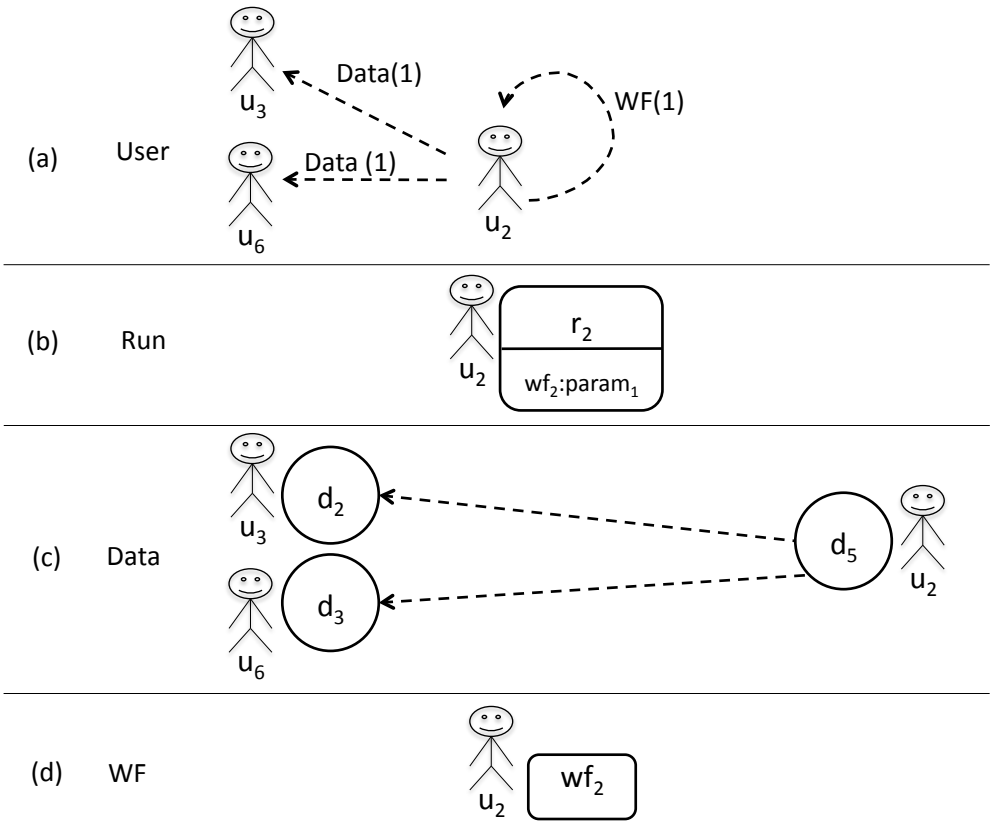


Figure 5.7: Acknowledgement view for data artifact d5 based on the observables for the scenario in Figure 5.1.

5.6.2 Usage Trail of a Data Artifact

The second example usecase is the generation of a usage trail view for a data item. There are multiple scenarios for using such a view. An obvious one is users querying for the impact of the data they published. Let's assume that a user, u_d published a dataset, d_u. Some time later, he is trying to find out how d_u was used and if it impacted any scientific results. Using the collaborative provenance views, he can generate a full trace of how d_u was used. An even more important scenario occurs once a data artifact is found to be faulty and tagged through the data stores. Let's now assume that u_d found out that d_u has a problem and tagged it as *faulty*. Using collaborative provenance queries, the system can now identify all the workflow executions and data that depended on d_u using the usage trail view, tag all the data that dependent on this data as potentially faulty with attached explanation similar to the acknowledgement veiw, and notify the users, e.g., via email, that were effected by d_u for correction.

In addition to this conceptual queries, in Chapter 7, we will provide example scientific usecases based on example studies in the CAMERA (Sun *et al.* 2010) and the Virolab (Sloot *et al.* 2009) projects, and show how collaborative views can help such studies.

5.7 Advantages of the Collaborative Provenance Approach

Advantages of a collaborative provenance approach as described in this chapter are many since it is all about recording the e-Science activities, specifically, capturing the relationships between human users, workflow executions and data. Firstly, it builds upon existing knowledge and extends it without a re-architecturing of the components in a collaborative e-Science platform. Secondly, it allows for extensions to collaborating entities, e.g., instruments and other system modules, as long as the provenance is kept as system-wide assertions that adhere to a global data model. Most importantly, while minimizing the interrupts to scientists and the way they do their work, it impacts the effectivity of the collaborative research by adding value by assisting scientific work. This value comes from being able to analyze collaborations and track the footprint of data. In sum, this approach brings together consumers and producers of data and other e-Science objects together allowing for proper tracking and acknowledgement mechanisms.

Summary

In this chapter, we introduced a new collaborative provenance model which addresses the need for inferring dependencies across multiple workflow runs and understanding user collaborations. We explain how our proposed collaborative provenance model allows to establish the attributes for the nature of user collaborations, the strength of collaborations and self collaborations. We also provide some example collaborative provenance scenarios and collaborative provenance queries. Next, we describe a data model to capture and query collaborative provenance.