Collaborative provenance for workflow-driven science and engineering

Altıntaş, İ.

Publication date
2011

7

Collaborative Provenance Usecases

"That by listening to some music, by reading some books, by looking at paintings, and most important by hanging out with one another - by collaborating with one another and creating your own network - you can achieve something that is much better than what is out there."

– David Amram

This chapter provides example applications for collaborative provenance for drug ranking drawn from ViroLab[1] and CAMERA[2] projects respectively. For Virolab, we discuss a scenario and provide how different collaborative provenance views can be used to improve the infrastructure. For CAMERA, we provide a simplified usecase from the actual infrastructure and answer some of the queries that collaborative provenance model can help with.

7.1 Virolab Virtual Patient Experiment Scenario

7.1.1 Components of the Virtual Patient Experiment

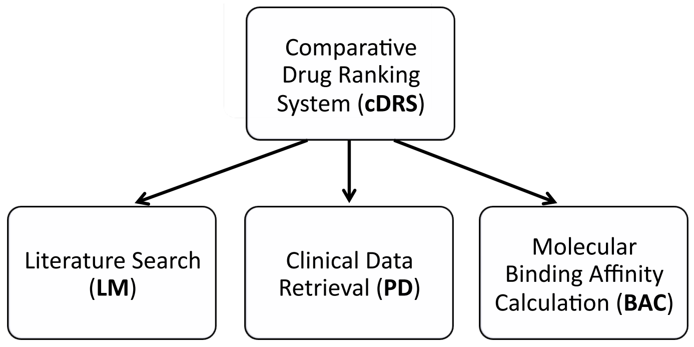
HIV is one of the most destructive pandemics in recorded history. It has been proven that use of a suitable combination of drugs tailored for the individual antiretroviral therapy leads to greater clinical success. The fundamental issue is to prescribe the most suitable combination of drugs fitting the patients viral genotype. For this purpose various Decision Support Systems (DSS) have been designed.

ViroLab (Sloot *et al.* 2009) developed a virtual laboratory (Bubak *et al.* 2007) for decision support in viral diseases treatment. Figure 7.1 demonstrates the conceptual process (See Figure 7.1 (a)) and data flowing between processes (See Figure 7.1 (b)) for a patient scenario

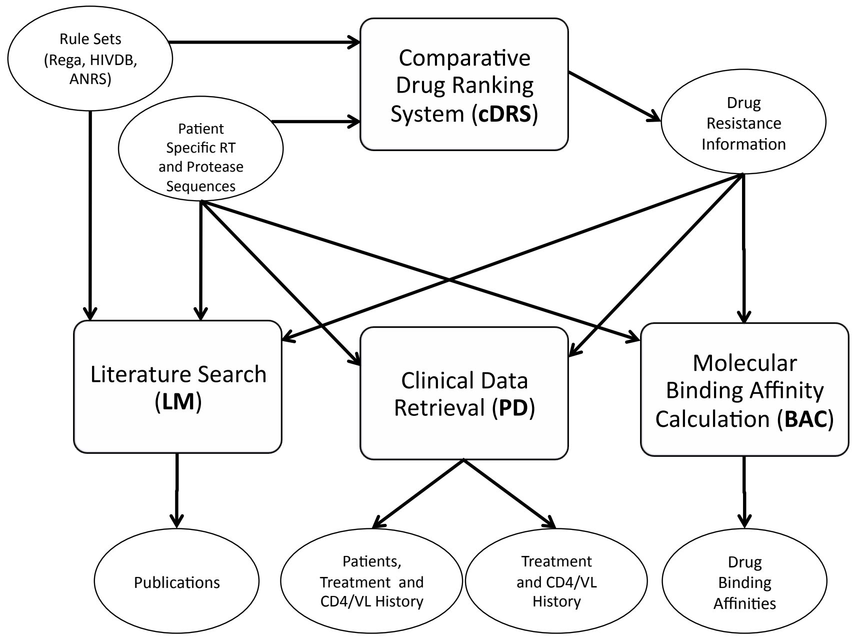
[0]This chapter is based on (Altintas *et al.* 2010*e*), (Sloot *et al.* November 2006), (Altintas *et al.* 2010*a*), (Altintas *et al.* 2010*c*), (Sun *et al.* 2010) and (Altintas *et al.* 2010*f*) co-authored by Altintas.

[1]Virolab website: `http://www.virolab.org`

[2]CAMERA website: `http://camera.calit2.net`


Comparative Drug Ranking System (**cDRS**)
Literature Search (**LM**)
Clinical Data Retrieval (**PD**)
Molecular Binding Affinity Calculation (**BAC**)

(a)


Rule Sets (Rega, HIVDB, ANRS)
Patient Specific RT and Protease Sequences
Comparative Drug Ranking System (**cDRS**)
Drug Resistance Information
Literature Search (**LM**)
Clinical Data Retrieval (**PD**)
Molecular Binding Affinity Calculation (**BAC**)
Publications
Patients, Treatment and CD4/VL History
Treatment and CD4/VL History
Drug Binding Affinities

(b)

Figure 7.1: Virtual Patient Experiment (VPE): (a) Combined Workflow and (b) Data Flow

using ViroLab. ViroLab consists of complementary, multi-level computational tools among which there are:

- *Comparative Drug Ranking System (cDRS)* is a first order logic, rule based classification system. Its rules are derived from expertise and literature. It takes patient's viral sequence (or mutations), aligns it with the default viral sequence revealing the set of mutations in the patient's viral sequence. Then, it provides the suggestions from three widely accepted DRSs (HIVdb, ANRS and REGA) for this mutation set.

- *Molecular Dynamics Simulations (BAC)* takes as an input the patient's viral mutations it calculates the binding affinity of the drug-protein complexes, from which one can derive the theoretical fitness of the virus for the suggested drug.

- *Patient Data Retrieval (PD)* is a secure patient data retrieval system. It allows the clinician to run combined-queries for patients that have the same set (or subset) of mutations to view similar patient treatment histories.

- *Literature Mining (LM)* searches all the abstracts in PubMed for the pairs of mutations and drug(s) under consideration.

The Virtual Patient Experiment (VPE) has been implemented as a Ruby script (Malawski *et al.* 2010). The provenance of all these steps are captured by the system via assertions into the virtual laboratory's provenance data store (Balis *et al.* 2009). However, these components are run as multiple set of steps for added decision support often repeating some of the steps multiple times. We can infer the overall *"combined workflow"* generated by such dynamic courses of actions using the data and run dependency views in our collaborative provenance model. In addition, in a scenario where different steps are run by different users, we can create a view over user collaborations of implicit and explicit nature.

7.1.2 Collaborative Provenance for VPE

Once the clinical researcher works with the Virolab portal, she starts with executing cDRS and, based on the results of cDRS, she may or may not choose to execute the rest of the steps in the VPE. Similarly, she may execute some of the steps multiple times, experimenting with different test results. Additionally, a couple of clinical researchers could be collaborating on the drug ranking of a patient, executing different parts of the VPE at different times. The provenance of all of these activities are captured. However, as mentioned before, the steps are currently implemented as scripts, not as workflows. For the rest of this subsection, to motivate our collborative provenance for combined workflow approach, we assume a conceptual implementation of the VPE in different scientific workflow environments, namely, Kepler (Ludäscher *et al.* 2006), Taverna (Oinn *et al.* 2006) and WSVLAM (Korkhov *et al.* 2007).

Based on the focus of each workflow system and scientific workflows implemented previously, we assume that PS and cDRS are Kepler workflows, BAC is a WSVLAM workflow,

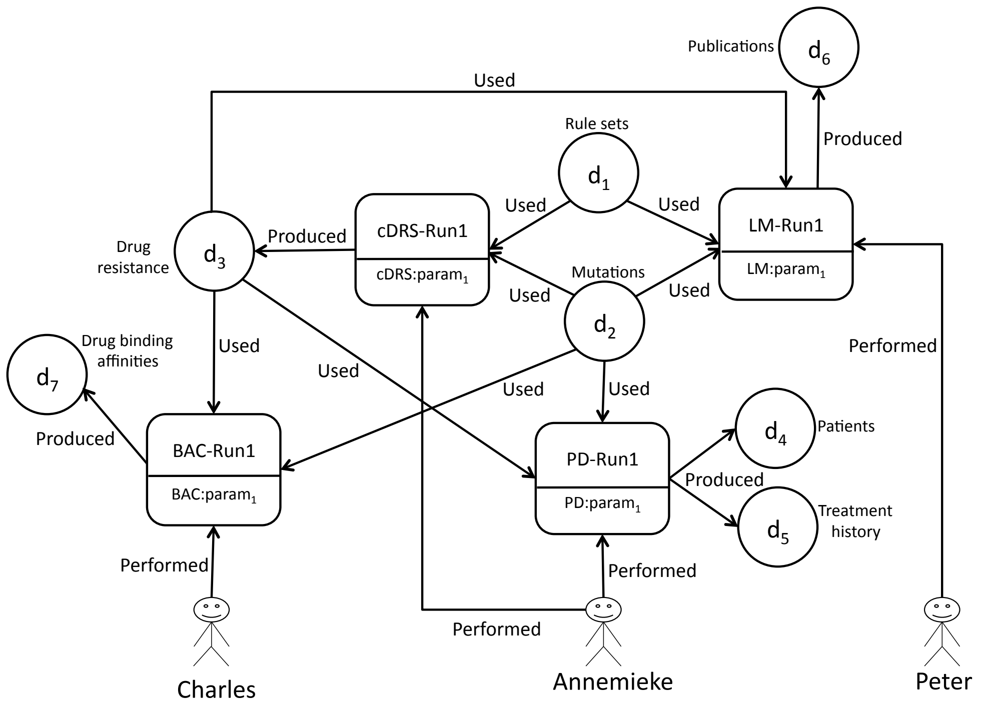


Figure 7.2: The detailed view for a VPE collaboration scenario where users Annemieke, Charles and Peter work together on different parts of the same combined workflow and depend on each others workflow executions. The figure also shows user actions along with data used and produced by these workflows.

and LM is a Taverna workflow. Also, for simplicity, we assume that all the provenance information captured by these systems conforms to a model of provenance (Ludäscher *et al.* 2008), e.g., OPM (Moreau *et al.* 2010), allowing for their interoperability. This is not an unlikely case as Kepler, WSVLAM and Taverna teams all participated in the Third Provenance Challenge[3] and was able to demonstrate their capability to collect (or export to) provenance data that conforms to OPM.

Figure 7.2 illustrates a collaborative project between three clinical researchers working in the Virolab Virtual Laboratory. Note that, in this scenario, the collaboration is of *explicit* nature, i.e., Annemieke, Charles and Peter know that they are working on different parts of the same problem and depend on workflow executions of each other. This might not be the case for other collaborations where users might not even be aware of each other, but use each others publicly published data and workflows, forming an *implicit* collaboration. Also note that, for simplicity, we assume that the workflows and initial data items, d_1 and d_2, were made available by the system as public without any associations to any users. In Figure 7.3(a),

[3]Third Provenance Challenge Wiki: `http://twiki.ipaw.info/bin/view/Challenge/ThirdProvenanceChallenge`

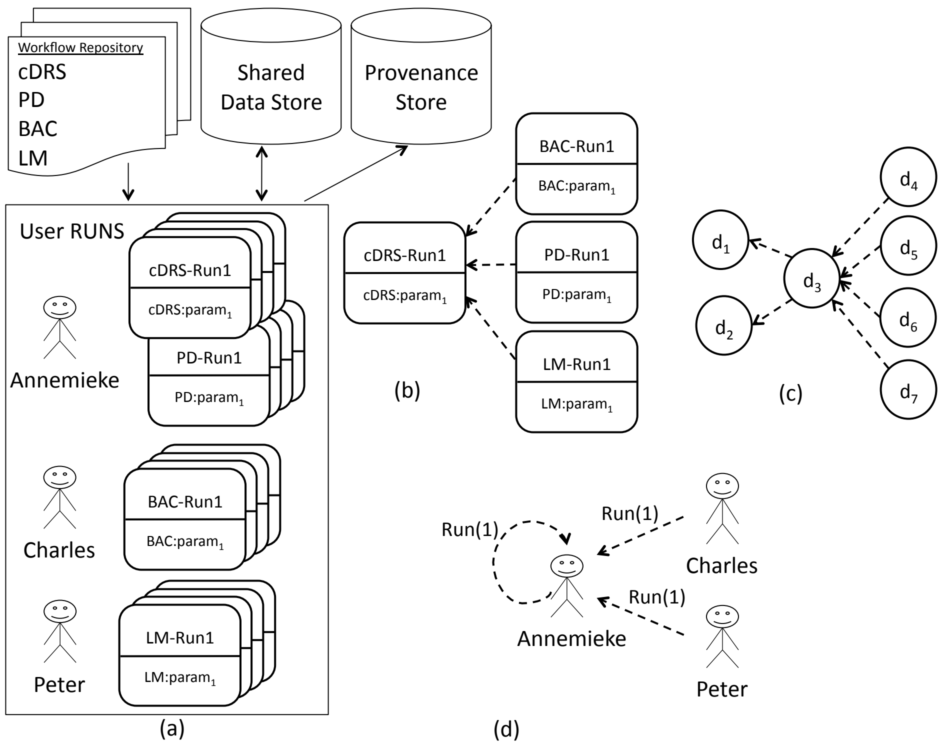


Figure 7.3: Simpler views over the VPE collaboration scenario showing: (a) Users Annemieke, Charles and Peter working on their parts of the combined workflow by running workflows and using data from the system and saving the provenance of their workflow runs.; (b) Run dependency view for Figure 7.2.; (c) Data dependency view for Figure 7.2.; (d) User collaboration view for Figure 7.2.

Annemieke executes cDRS and PD in Kepler multiple times with various inputs, Charles executes BAC workflow in WSVLAM multiple times using the data from Annemieke's cDRS runs and Peter runs LM in Taverna also using the data from Annemieke's cDRS runs, all through the virtual laboratory. (A detailed view showing data and relationships between these workflow runs is shown in Figure 7.2.) At the end of their analysis work, they get together and go through their combined results, potentially coming up with a decision or new input data to run these steps again. This process continues until a decision is made. In addition, the generated run dependency view can be saved as workflows in the system. So if Annemieke, Charles and Peter wanted to run their collaborative combined workflow again with more data, they can just enter the new data and run through the combined version of all the workflows they have done before.

A partial run dependency view for all the runs by Annemieke, Charles and Peter is illustrated in Figure 7.3(b). Based on the scenario in Figure 7.2, Figure 7.3(c) shows the data dependency view. The simple user collaboration view in Figure 7.3(d) shows the C_{NWS}

graph for this partial run dependency view. The collaborative provenance views in Figure 7.3 (b), (c) and (d) convert the otherwise complicated collaborative work graph in Figure 7.2 to three different focused views, thus, providing a simple and clear picture of the overall collaborative effort.

A very important contribution illustrated in this scenario is that such a combined workflow puts users actions and working methods as the basis for the research instead of requiring users to use built in workflows to benefit from the advantages of large e-Science infrastructures. The contribution of this thesis ties together the user actions with data and different computational entities, i.e., workflows, in an e-Science project as long as different pieces conform to a common collaborative provenance model allowing for interoperability. This approach puts the user and collaborations in the center of the conducted research and makes the users freely choose between different technologies used to solve parts of the problem at hand allowing workflows and scripts to be executed within the same infrastructure.

7.2 Collaborative Metagenomics in CAMERA

CAMERA (Community Cyberinfrastructure for Advanced Marine Microbial Ecology Research and Analysis) is a collaborative e-Science platform, wherein scientific workflows (Altintas *et al.* 2010*a*) enable the use of various community tools that are shared by a metagenomics (Committee on Metagenomics: Challenges and Functional Applications 2007) research community. CAMERA enables the microbial ecology community to manage the challenges of metagenomics analysis, and has more than 3000 users in over 75 countries worldwide.

7.2.1 Scientific Workflow-Driven Science in CAMERA

CAMERA provides a component-based infrastructure that includes the Kepler scientific workflow system (Ludäscher *et al.* 2006). The Kepler scientific workflows used within CAMERA support the interaction of automated computational tools and human inspection and interaction. Kepler is also used to record the provenance of data products stored within the central CAMERA data repository that were produced through CAMERA workflows.

CAMERA enables users to create, share, and execute workflows specific to their own experiments. Currently, the core CAMERA workflows make the following metagenomic analyses available to researchers: data quality control (specifically, QC Filter and 454 Duplicate Clustering), read assembly (454 Read Assembly), functional annotation and clustering (Metagenomic Data Annotation and Clustering), taxonomy binning (Taxonomy Binning), BLAST, and additional downstream analysis methods. The scientific goals and technical details for these workflows are explained in (Altintas *et al.* 2010*a*), (Altintas *et al.* 2010*c*) and (Sun *et al.* 2010).

Figure 7.4 shows an example scenario with different observables of shared data, workflows, and workflow runs in CAMERA, where all or part of the output of workflow runs can

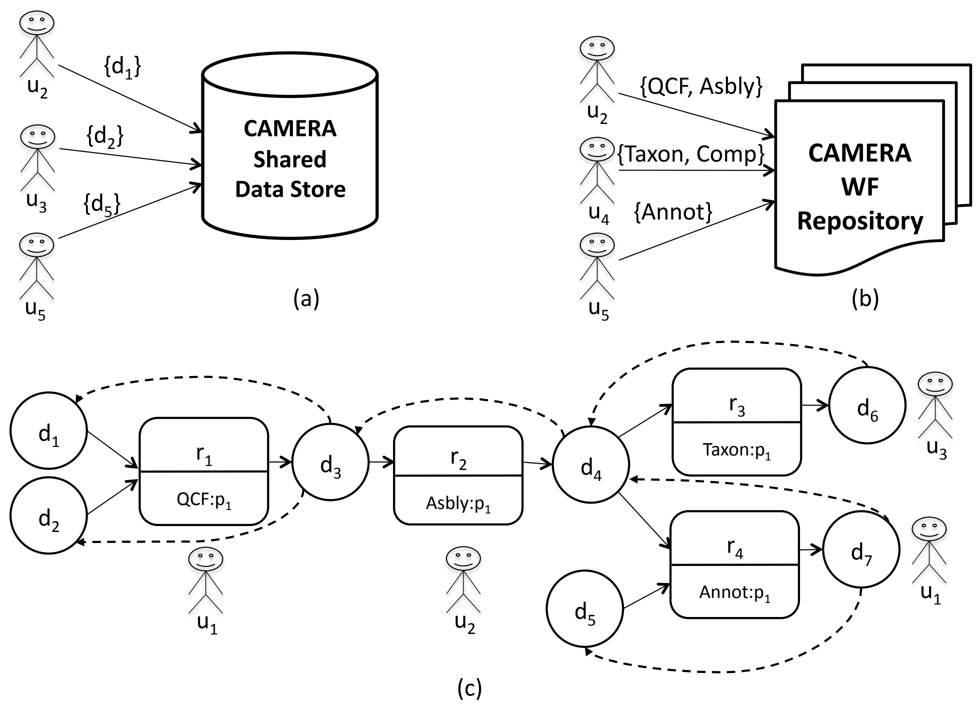


Figure 7.4: A typical scenario for different observables of shared data, workflows and workflow executions (runs) in CAMERA: (a) data $\{d_1, d_2, d_5\}$ published by users $\{u_2, u_3, u_5\}$; (b) workflows $\{QCF, Asbly, Taxon, Annot, Comp\}$ published by users $\{u_2, u_4, u_5\}$; and (c) flow graph for workflow runs (customized through their parameters, p_1) and related provenance data $\{d_1 \ldots d_7\}$ in user space $\{u_1, u_2, u_3\}$.

be used as input to subsequent runs. Figure 7.4(a) shows that datasets d_1, d_2 and d_5 are published by users u_2, u_3 and u_5, respectively, in the shared data store. Similarly, Figure 7.4(b) shows workflows $\{$*QCF*,*Asbly*$\}$, $\{$*Taxon*,*Annot*$\}$, and *Comp* being published by users u_2, u_4, and u_5, respectively, in the workflow repository. A critical aspect of the CAMERA workflow environment is that these workflows can be organized into a systematic network (a.k.a. *combined workflow*), in which outputs of one workflow execution can be used as inputs for subsequent workflows, as shown in Figure 7.5. This allows researchers to build a complete end-to-end analysis stream by choosing to use different combinations of workflows based on their specific data and analysis needs. For instance, one possible end-to-end analysis stream (see Figure 7.5) for researchers with raw sequencing data may entail: (1) use of the QC filter for data quality control (*QCF*); (2) assembly of the resultant reads to longer contigs (*Asbly*); (3) assignment of taxonomic information to each contig (*Taxon*); (4) annotation of genes against COG, Pfam, TIGRFAM, and other reference databases and clustering of genes to the

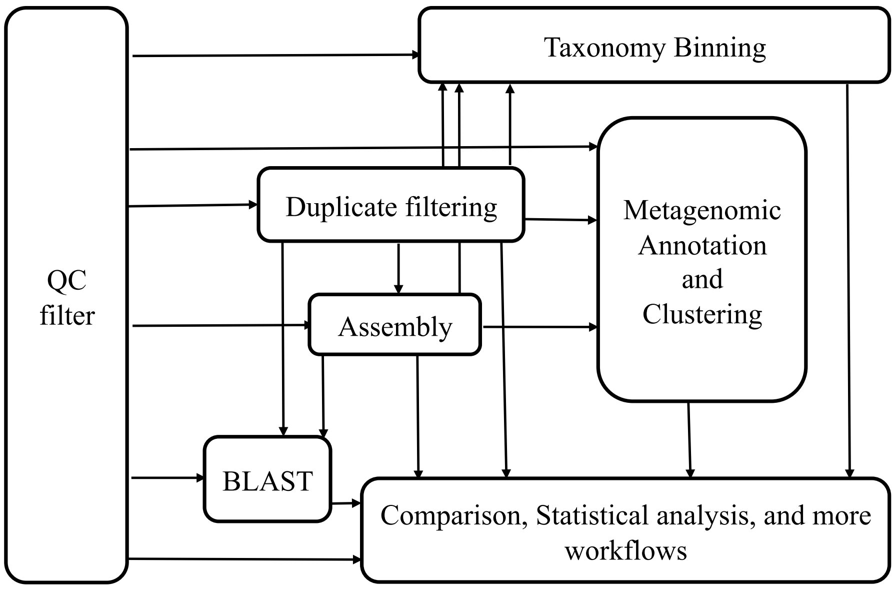

QC filter
Taxonomy Binning
Duplicate filtering
Metagenomic Annotation and Clustering
Assembly
BLAST
Comparison, Statistical analysis, and more workflows

Figure 7.5: A possible end-to-end analysis stream in CAMERA with raw sequencing data.

desired level (*Annot*); (5) execution of a statistical comparison, obtaining a comparison graph (*Comp*); and so on. The first four steps of this workflow execution scenario are illustrated in Figure 7.4(c), where 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 arcs 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).

In Figure 7.4(c), user u_1 performs a run r_1 of workflow *QCF* with parameter settings p_1 and input datasets d_1 and d_2. Run r_1 produces data d_3 as its output, and dependencies between the output and inputs of run r_1 are shown using a dashed arc. Similarly, user u_2 performs a run r_2 of workflow *Absly* with parameter settings p_1 and input dataset d_3. Run r_2 produces data d_4 as its output, and dependencies between the output and input of run r_2 are shown with a dashed arc. User u_3 performs a run r_3 of workflow $Taxon$ with parameter settings p_1, using the output data d_4 from run r_2. User u_1 also performs run r_4, using data from a previous run, i.e., d_4 from r_2, along with other published data d_5, and produces d_7 as its output. In run r_4, d_7 depends on d_4 and d_5. Although not shown in this figure, in general, the output of a run may depend on some but not necessarily all inputs of a run. This explicit dependency information is provided to our provenance schema by the underlying provenance engine and might not always be captured. For example, the Kepler Provenance Recorder uses

$$
\begin{aligned}
\texttt{publishes} &:= \{(\mathrm{d}_1,\mathrm{u}_2),(\mathrm{d}_2,\mathrm{u}_3),(\mathrm{d}_5,\mathrm{u}_5)\}\\
\texttt{workflow} &:= \{(\texttt{QCF},\mathrm{u}_2),(\texttt{Asbly},\mathrm{u}_2),(\texttt{Taxon},\mathrm{u}_4),(\texttt{Comp},\mathrm{u}_4),(\texttt{Annot},\mathrm{u}_5)\}\\
\texttt{run} &:= \{(\mathrm{r}_1,\texttt{QCF},\mathrm{u}_1),(\mathrm{r}_2,\texttt{Asbly},\mathrm{u}_2),(\mathrm{r}_3,\texttt{Taxon},\mathrm{u}_3),(\mathrm{r}_4,\texttt{Annot},\mathrm{u}_1)\}\\
\texttt{uses} &:= \{(\mathrm{r}_1,\mathrm{d}_1),(\mathrm{r}_1,\mathrm{d}_2),(\mathrm{r}_2,\mathrm{d}_3),(\mathrm{r}_3,\mathrm{d}_4),(\mathrm{r}_4,\mathrm{d}_4),(\mathrm{r}_4,\mathrm{d}_5)\}\\
\texttt{produces} &:= \{(\mathrm{r}_1,\mathrm{d}_3),(\mathrm{r}_2,\mathrm{d}_4),(\mathrm{r}_3,\mathrm{d}_6),(\mathrm{r}_4,\mathrm{d}_7)\}\\
\texttt{ddep} &:= \{(\mathrm{d}_3,\mathrm{d}_1),(\mathrm{d}_3,\mathrm{d}_2),(\mathrm{d}_4,\mathrm{d}_3),(\mathrm{d}_6,\mathrm{d}_4),(\mathrm{d}_7,\mathrm{d}_4),(\mathrm{d}_7,\mathrm{d}_5)\}\\
\texttt{ddep}^* &:= \texttt{ddep} \cup \{(\mathrm{d}_6,\mathrm{d}_3),(\mathrm{d}_6,\mathrm{d}_2),(\mathrm{d}_6,\mathrm{d}_1),(\mathrm{d}_4,\mathrm{d}_2),(\mathrm{d}_4,\mathrm{d}_1),(\mathrm{d}_7,\mathrm{d}_3),(\mathrm{d}_7,\mathrm{d}_2),(\mathrm{d}_7,\mathrm{d}_1)\}
\end{aligned}
$$

Table 7.1: Relation instances of the provenance schema corresponding to the example in Figure 7.4.

an implicit dependency mechanism that assumes that each output depends on all of the inputs where as the COMAD model provides explicit data dependencies between input and outputs. The details of COMAD's implicit data dependency mechanism is discussed in (Anand *et al.* 2009*b*). The statistical comparison step (i.e., the *comp* workflow contributed by user u_5 in Figure 7.4(b)) merges and compares the outputs of the *Taxon* and *Annot* workflow runs has been left out of the current scenario for simplicity.

An instance of the collaborative schema is shown in Table 7.1, which corresponds to the example CAMERA scenario of Figure 7.4. This instance can be used to generate the data dependency (see Figure 7.6(a)), run dependency (see Figure 7.6(b)) and user collaboration (see Figure 7.6(c)) views using `DATA-DEP`, `RUN-DEP` and C_{NWS} queries from Section 6.3, respectively.

7.2.2 Answering Example Queries

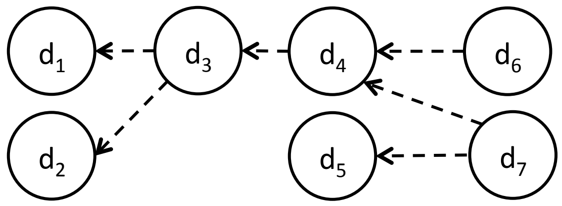
Using the collaborative provenance instance in Table 7.1 to capture the basic observables in Figure 7.4 for CAMERA, below, we give the Datalog queries for answering example collaborative provenance questions. We denote query results below via the `ans` relation.

1. Which data artifacts were used directly or indirectly to generate d_7? (*Answer:* $\{d_1, d_2, d_3, d_4, d_5\}$)

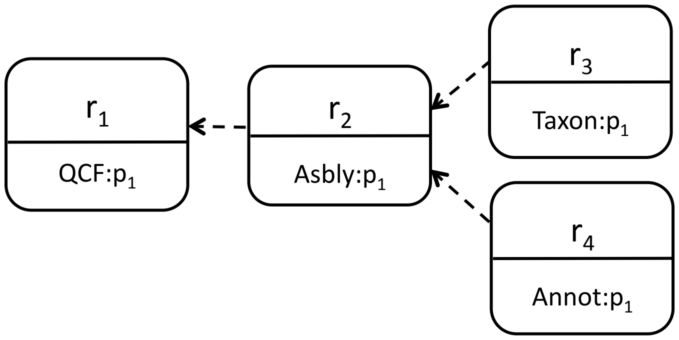
$$\texttt{ans}_1(d_{to}) \texttt{ :- } \texttt{ddep}^*(d_{to}, \mathrm{d}_7).$$

2. Which runs were used in the generation of d_6? (*Answer:* $\{r_1, r_2, r_3\}$)

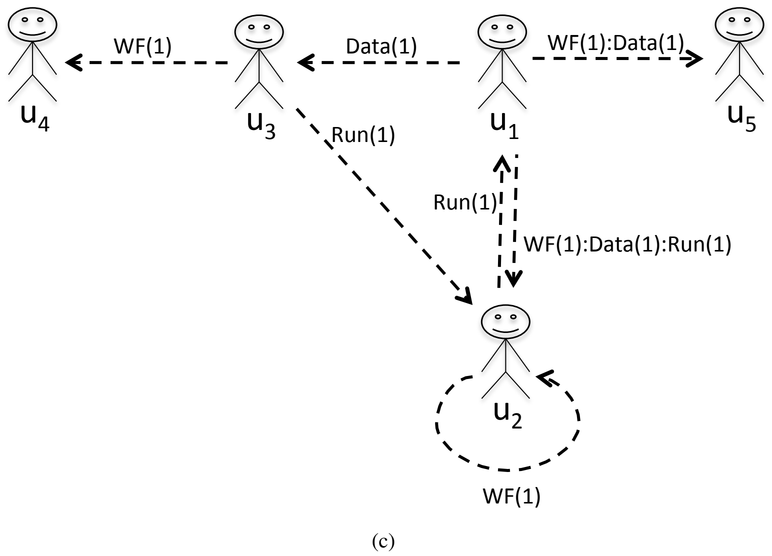
$$\texttt{ans}_2(r) \texttt{ :- } \texttt{ddep}^*(d_{to}, \mathrm{d}_6), \texttt{produces}(r, d_{to}).$$



(a)



(b)



(c)

Figure 7.6: Collaborative provenance views in CAMERA: (a) data dependency; (b) run dependency; and (c) user collaboration (based on Figure 7.4).

3. If data artifact d_2 is detected to be faulty, which users should be notified of the error? (*Answer:* $\{u_1, u_2, u_3\}$)

$$\begin{aligned}\texttt{ans}_3(u)\ \texttt{:-}\ & \texttt{ddep}^*(\texttt{d}_2, d_{from}), \texttt{produces}(r, d_{from}),\\ & \texttt{run}(r, w, u).\end{aligned}$$

4. What are all the datasets that depended on d_2, i.e. the *"impact"* of d_2? (*Answer:* $\{d_3, d_4, d_6, d_7\}$)

$$\texttt{ans}_4(d_{from})\ \texttt{:-}\ \texttt{ddep}^*(\texttt{d}_2, d_{from}).$$

5. Which users depended on data artifact d_1, directly or indirectly? (*Answer:* $\{u_1, u_2, u_3\}$)

$$\begin{aligned}\texttt{ans}_5(u)\ \texttt{:-}\ & \texttt{ddep}^*(\texttt{d}_1, d_{from}), \texttt{produces}(r, d_{from}),\\ & \texttt{run}(r, w, u).\end{aligned}$$

6. Which users did u_1 depend on, i.e., *"collaborate with"*, directly? What is the nature and strength of each collaboration? (*Answer:* u_2, u_3, and u_5 via `WF`(1):`Data`(1):`Run`(1), `Data`(1), and `WF`(1):`Data`(1), respectively)

$$\texttt{ans}_6^u(u_{to})\ \texttt{:-}\ \texttt{C}(u_{to}, e, \texttt{u}_1).$$

$$\texttt{ans}_6^n(e)\ \texttt{:-}\ \texttt{C}(u_{to}, e, \texttt{u}_1).$$

$$\begin{aligned}\texttt{ans}_6^w(e, n) := \ & \texttt{SELECT}\ e, \texttt{COUNT}\ (e)\ \texttt{AS}\ n\\ & \texttt{FROM C}\\ & \texttt{WHERE}\ u_{from} = \texttt{u}_1\\ & \texttt{GROUP BY}\ e\end{aligned}$$

Note that we use SQL above to perform the necessary grouping and aggregation for $\texttt{ans}_6^w$, which gives the number of different kinds of collaborations $e(n)$ between users u_1 and u_{to}.

7. Who are the potential acknowledgements for a publication involving d_7, i.e., which user collaborations were involved in the derivation of d_7? (*Answer:* $\{u_1, u_2, u_3, u_5\}$)

$$\begin{aligned} \texttt{ans}_7(u) \;\texttt{:-}\; & \texttt{produces}(r, \texttt{d}_7), \\ & \texttt{run}(r, w, u). \end{aligned}$$

$$\begin{aligned} \texttt{ans}_7(u_2) \;\texttt{:-}\; & \texttt{produces}(r, \texttt{d}_7), \\ & \texttt{run}(r, w, u_1), \\ & \texttt{workflow}(w, u_2). \end{aligned}$$

$$\begin{aligned} \texttt{ans}_7(u) \;\texttt{:-}\; & \texttt{ddep}^*(d_{to}, \texttt{d}_7), \\ & \texttt{produces}(r, d_{to}), \\ & \texttt{run}(r, w, u). \end{aligned}$$

$$\begin{aligned} \texttt{ans}_7(u) \;\texttt{:-}\; & \texttt{ddep}^*(d_{to}, \texttt{d}_7), \\ & \texttt{publishes}(d_{to}, u). \end{aligned}$$

$$\begin{aligned} \texttt{ans}_7(u_2) \;\texttt{:-}\; & \texttt{ddep}^*(d_{to}, \texttt{d}_7), \\ & \texttt{produces}(r, d_{to}), \\ & \texttt{run}(r, w, u_1), \\ & \texttt{workflow}(w, u_2). \end{aligned}$$

Note that a data can either be published by a user or be the result of a workflow execution. So, we formulate queries which union these conditions. The above query is explained as: (1) returns the user that created the data d_7 through execution of a workflow; (2) returns the user who published the workflow whose execution created data d_7; (3) returns users who ran workflows, such that the workflow results shared a transitive dependency ($\texttt{ddep}^*$) relation with data d_7; (4) returns users who published data such that those data shared a transitive dependency relation with data d_7; and (5) returns users who published the workflows such that their execution by other (or same) users resulted in data that shared a transitive dependency relation with data d_7.

As can be seen from the datalog queries as described above, expressing collaborative provenance queries can be very complex in any standard query language. So, we have adopted and extended QLP to enable users to easily and concisely express collaborative queries. Table 7.2 and Table 7.3 show the QLP expressions associated to these queries using the collaborative provenance extensions.

Table 7.2: Example CAMERA queries 1 through 6 expressed in Datalog and QLP extensions.

1	Which data artifacts were used directly or indirectly to generate d_7? (*Answer:* $\{d_1, d_2, d_3, d_4, d_5\}$)	$\mathtt{ans}_1(d_{to})$:- $\mathtt{ddep}^*(d_{to}, \mathrm{d}_7)$.	artifacts(* .. d_7)
2	Which runs were used in the generation of d_6? (*Answer:* $\{r_1, r_2, r_3\}$)	$\mathtt{ans}_2(r)$:- $\mathtt{ddep}^*(d_{to}, \mathrm{d}_6)$, $\mathtt{produces}(r, d_{to})$.	runs(* .. d_6)
3	If data artifact d_2 is detected to be faulty, which users should be notified of the error? (*Answer:* $\{u_1, u_2, u_3\}$)	$\mathtt{ans}_3(u)$:- $\mathtt{ddep}^*(\mathrm{d}_2, d_{from})$, $\mathtt{produces}(r, d_{from})$, $\mathtt{run}(r, w, u)$.	users(runs(d_2 .. *))
4	Which datasets depended on d_2, i.e. the *"impact"* of d_2? (*Answer:* $\{d_3, d_4, d_6, d_7\}$)	$\mathtt{ans}_4(d_{from})$:- $\mathtt{ddep}^*(\mathrm{d}_2, d_{from})$.	artifacts(d_2 .. *)
5	Which users depended on data artifact d_1, directly or indirectly? (*Answer:* $\{u_1, u_2, u_3\}$)	$\mathtt{ans}_5(u)$:- $\mathtt{ddep}^*(\mathrm{d}_1, d_{from})$, $\mathtt{produces}(r, d_{from})$, $\mathtt{run}(r, w, u)$.	users(runs(d_1 .. *))
6	Which users did u_1 depend on, i.e., *"collaborate with"*, directly? What is the nature and strength of each collaboration? (*Answer:* u_2, u_3, and u_5 via WF(1):Data(1):Run(1), Data(1), and WF(1):Data(1), respectively)	$\mathtt{ans}_6^u(u_{to})$:- $\mathtt{C}(u_{to}, e, \mathrm{u}_1)$. $\mathtt{ans}_6^n(e)$:- $\mathtt{C}(u_{to}, e, \mathrm{u}_1)$. $\mathtt{ans}_6^w(e, n) :=$ SELECT e, COUNT (e) AS n FROM C WHERE $u_{from} = \mathrm{u}_1$ GROUP BY e.	COLLAB-DEP(* ..*) From above query result of form $\langle u_{from}, e, u_{to}\rangle$, select all those u_{to} where u_{from}=u_1.

Table 7.3: Example CAMERA query 7 expressed in Datalog and QLP extensions.

7	Who are the potential acknowledgements for a publication involving d_7, i.e., which user collaborations were involved in the derivation of d_7?
	(*Answer:* $\{u_1, u_2, u_3, u_5\}$)
		`ans`$_7(u)$ `:-`
		`produces`(r, d_7),
		`run`(r, w, u).
		`ans`$_7(u_2)$ `:-`
		`produces`(r, d_7),
		`run`(r, w, u_1),
		`workflow`(w, u_2).
		`ans`$_7(u)$ `:-`
		`ddep`$^*(d_{to}, \mathrm{d}_7)$,
		`produces`(r, d_{to}),
		`run`(r, w, u).
		`ans`$_7(u)$ `:-`
		`ddep`$^*(d_{to}, \mathrm{d}_7)$,
		`publishes`(d_{to}, u).
		`ans`$_7(u_2)$ `:-`
		`ddep`$^*(d_{to}, \mathrm{d}_7)$,
		`produces`(r, d_{to}),
		`run`(r, w, u_1),
		`workflow`(w, u_2).
			users(artifacts(* .. d_7))
			users(runs(* .. d_7))
			users(workflows(* .. d_7))

Summary

Through usecase scenarios for drug ranking and bioinformatics, we demonstrated how the collaborative provenance model and queries can be applied to effectively understand and analyze users collaborations in scientific discoveries driven by scientific workflows. Next, we present an implementation of the collaborative provenance database in PostgreSQL and evaluation of the execution of collaborative views and queries using the CAMERA usecase as an application.