CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2008; **20**:409–418
Published online 2 November 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1233

Special Issue: The First Provenance Challenge

SUMMARY

The first Provenance Challenge was set up in order to provide a forum for the community to understand the capabilities of different provenance systems and the expressiveness of their provenance representations. To this end, a functional magnetic resonance imaging workflow was defined, which participants had to either simulate or run in order to produce some provenance representation, from which a set of identified queries had to be implemented and executed. Sixteen teams responded to the challenge, and submitted their inputs. In this paper, we present the challenge workflow and queries, and summarize the participants' contributions. Copyright © 2007 John Wiley & Sons, Ltd.

1. INTRODUCTION

The term *provenance* is commonly used in the context of art to denote the documented history or the chain of ownership of an art object. Provenance helps to determine the authenticity and therefore the value of art objects. If the provenance of data produced by computer systems could be determined as it can be for some works of art, then users, in their daily applications, would be able to interpret and judge the quality of data better [1]. In particular, the scientific and grid communities consider that, in order to support reproducibility, workflow management systems will be required to track and integrate provenance information as an integral product of the workflow [2]. Several surveys of provenance are available [3–5].

Against this background, the *International Provenance and Annotation Workshop* (IPAW'06), held on 3–5 May 2006 in Chicago, involved some 50 participants interested in the issues of data provenance, process documentation, data derivation, and data annotation [6,7]. During a session on provenance standardization, a consensus began to emerge, whereby the provenance research community needed to understand better the capabilities of the different systems, the representations they used for provenance, their similarities, their differences, and the rationale that motivated their designs. Hence, the first Provenance Challenge was born, and from the outset, the challenge was set up to be *informative* rather than *competitive*. In this editorial, we describe the challenge and provide a view on the contributions by the participating teams.

2. THE PROVENANCE CHALLENGE

The challenge was defined by Simon Miles and Luc Moreau (University of Southampton), and Mike Wilde and Ian Foster (University of Chicago/Argonne Nat. Lab.) in May 2006; it was then reviewed by a larger group, including Juliana Freire (University of Utah) and Jim Myers (NCSA), before a public review period by the IPAW'06 participants. It was published on 19th June and concluded with a two-day workshop, at GGF 18, in Washington, DC, on 13–14 September 2006.

2.1. Instructions to participants

The aim of the Provenance Challenge is to establish an understanding of the capabilities of available provenance-related systems and, in particular, examine the following:

- the representations that systems use to document details of processes that have occurred;
- the ability of each system to answer provenance-related queries;
- what each system considers to be within scope of the topic of provenance (regardless of whether the system can yet achieve all problems in that scope).

To help achieve the aim, a simple example workflow was defined to form the basis of the challenge. This workflow is inspired from a real experiment [8] in the area of functional magnetic resonance imaging (fMRI). Here, the term *workflow* [9] is used to denote a series of procedures performed in a system, each taking some data as input and producing other data as output. The procedures and workflows in the challenge problem are not defined in terms of any particular technology (e.g. EXE files, Web Services, in the case of procedures; BPEL, compiled executable, batch file, in the case of workflows). Instead, participants could adopt their technology of choice.

Our focus in this challenge was on provenance and not on running the experiment. Hence, to facilitate take-up, we allowed challenge participants to implement procedures such as 'dummies', i.e. as fake procedures that make use of the input, output, and intermediate data we provided, take the right input and produce the right output, but do not execute the real code. Alternatively, participants could execute the real workflow after installing the necessary libraries.

Different systems use different representations for provenance information. In order to explore the capabilities of these different representations, we also defined a set of core queries, and asked participants to show how they addressed these queries.

Challenge participants were invited to upload the following information to the Provenance Challenge TWiki [10], to then allow comparison:

- Representation of the workflow in their system.
- Representation of provenance for the example workflow.
- Representation of the result of the core queries.
- Contributions to a matrix of queries vs systems, indicating for each whether: (1) the query can be answered by the system; (2) the system cannot answer the query currently but considers it relevant; (3) the query is not considered relevant to the project.

Each participant was also invited to optionally contribute the following:

- additional queries (beyond the core queries) that illustrate the scope of their system;

- extensions to the example workflow that the participant feels illustrate the unique aspects of their system;
- any categorization of queries that the project considers to have practical value.

2.2. The Provenance Challenge fMRI workflow

The purpose of the challenge workflow is to create population-based 'brain atlases' from the fMRI Data Center's archive of high-resolution anatomical data. Specifically, for an input fMRI data set, it produces average images along the axis X, Y, and Z, after aligning each input sample with a reference image. The workflow comprises procedures and data items flowing between them, respectively, shown as ovals and rectangles in Figure 1. The workflow can be seen as having five stages, where each stage is depicted as a horizontal row of the same procedure in the figure. Note that the term 'stage' is introduced only to describe the workflow; we do not specify how 'stages' should be realized in a concrete implementation.

Individual procedures employ the automated image registration (AIR) suite (`bishopw.loni.ucla.edu/AIR5/index.html`) to create an averaged brain from a collection of high-resolution anatomical data, and the FSL suite (`www.fmrib.ox.ac.uk/fsl`) to create 2D images across each sliced dimension of the brain. In addition to the data items shown in the figure, there are other inputs to procedures (constant string options), details of which can be found on the Provenance Challenge TWiki [10].

The inputs to a workflow are a set [11] of new brain images (Anatomy Image 1–4) and a single reference brain image (Reference Image). All input images are 3D scans of a brain of varying resolutions, so that different features are evident. For each image, there is the actual image and the metadata information for that image (Anatomy Header 1–4).

The stages of the workflow are as follows:

1. For each new brain image, `align_warp` compares the reference image for determining how the new image should be warped, i.e. the position and shape of the image adjusted, to match the reference brain. The output of each procedure in the stage is a *warp parameter set* defining the spatial transformation to be performed (Warp Params 1–4).
2. For each warp parameter set, the actual transformation of the image is done by `reslice`, which creates a new version of the original new brain image with the configuration defined in the warp parameter set. The output is a `resliced` image.
3. All the `resliced` images are averaged into one single image using `softmean`.
4. For each dimension (X, Y, and Z), the averaged image is sliced, with the utility `slicer`, to give an atlas data set, i.e. a 2D atlas along a plane in that dimension, taken through the centre of the 3D image.
5. Each atlas data set is converted into a graphical atlas image using (the ImageMagick utility) `convert`.

2.3. Core provenance queries

In addition to the workflow, the challenge specified an initial set of provenance-related queries. These queries, based on the authors' experience [4], identify typical patterns of querying found in provenance systems.

DOI: 10.1002/cpe

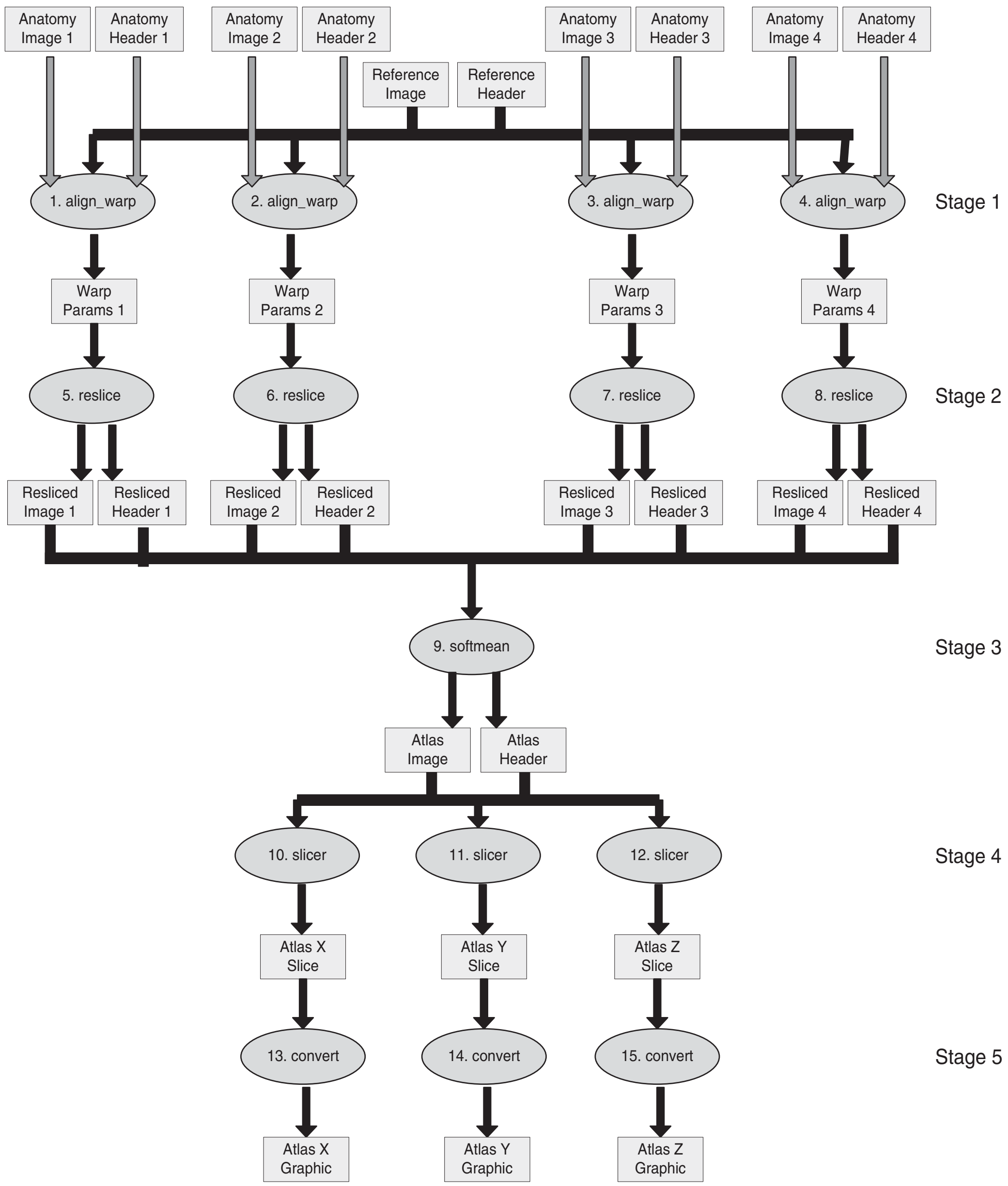

Figure 1. The Provenance Challenge workflow.

Q1. Find the process that led to Atlas X Graphic/everything that caused Atlas X Graphic to be as it is. This should tell us the new brain images from which the averaged atlas was generated, the warping performed, etc.

Q2. Find the process that led to Atlas X Graphic, excluding everything prior to the averaging of images with softmean.

Q3. Find the Stage 3–5 details of the process that led to Atlas X Graphic.

Q4. Find all invocations of procedure `align_warp` using a 12th order nonlinear 1365 parameter model (see model menu describing possible values of parameter '-m 12' of `align_warp`) that ran on a Monday.

Q5. Find all atlas graphic images outputted from workflows where at least one of the input Anatomy Headers had an entry global maximum = 4095. The contents of a header file can be extracted as text using the scanheader AIR utility.

Q6. Find all output averaged images of softmean (average) procedures, where the warped images taken as input were `align_warp`ed using a 12th order nonlinear 1365 parameter model, i.e. 'where softmean was preceded in the workflow, directly or indirectly, by an `align_warp` procedure with argument -m 12.'

Q7. A user has run the workflow twice, in the second instance replacing each procedure (convert) in the final stage with two procedures: `pgmtoppm`, then `pnmtojpeg`. Find the differences between the two workflow runs. The exact level of detail in the difference that is detected by a system is up to each participant.

Q8. A user has annotated some anatomy images with a key-value pair centre = UChicago. Find the outputs of `align_warp` where the inputs are annotated with centre = UChicago.

Q9. A user has annotated some atlas graphic images with a key-value pair where the key is studyModality. Find all the graphical atlas sets that have metadata annotation studyModality with values speech, visual or audio, and return all other annotations to these files.

3. AN ANALYSIS OF CONTRIBUTIONS TO THE PROVENANCE CHALLENGE

Following its publication, 16 teams responded to the challenge and submitted an entry to the Provenance Challenge TWiki [10]. This special issue contains each participating team's contribution to the challenge. Teams are referred to by the name of their system: Redux [12], Mindswap [13], Karma [14], JP [15], myGrid [16], VisTrails [17], ES3 [18], ZOOM [19], RWS [20], COMAD [21], PASS [22], SDG [23], NCSD2K and NCSCI [24], VDL [25], OPA [26], Wings/Pegasus [27].

In this section, we introduce a classification of the different approaches to help the reader gain a better understanding of provenance systems and their differences. To contrast the different approaches, we have identified a set of criteria, which have been grouped according to two categorizations.

Categorization 1 is concerned with the broad characteristics of provenance systems, such as the environment in which they are embedded and the technologies they use. Such systems are usually developed in the context of research projects that have specific foci: understanding research motivations is also useful for appreciating some design decisions. Given that the purpose of provenance systems is to build a computer-based representation of provenance that can be queried and reasoned over, Categorization 2 groups criteria pertaining to such representations; these criteria allow the reader to extract some of the fundamental concepts underpinning representations, and therefore, capabilities of systems. All findings are summarized in Figure 2.

DOI: 10.1002/cpe

	Redux	Mindswap	Karma	JP	myGrid	VisTrails	ES3	ZOOM	RWS	COMAD	PASS	SDG	NCSD2K	NCSCI	VDL	OPA	Wings/Pegasus
1. Characteristics of Provenance Systems																	
1.1 Execution Environmer	Workflow system	Web	Workflow system	Workflow syst and job submission	Workflow system	Workflow system	Operating system	Workflow system	Workflow system	Workflow system	Operating system	Workflow system	Visual Program. Env	Workflow system	Workflow system	Technolog independe	Workflow system
(actual system)	WWF		XBaya and BPEL	EGEE glite (inc Condor Dagman and JDL)	Taverna and Xscufl	VisTrails system	IDL and Bash	Technology independent	Kepler, Ptolemy	Kepler, Ptolemy	Linux	Kepler, Ptolemy	D2K	Cyber Integrator	VDS		Wings and Pegasus
1.2 Execution Environment (for the challenge)				EGEE VOCE VO		VisTrails system and shell		SQL scripts			Shell script				Grid	Java	Grid
1.3 Provenance Representation	RDBMS	OWL	XML View and RDBMS	key-value pairs in RDBMS	RDF	XML View and RDBMS	XML view	RDBMS	Internal	Internal and XML view	Internal	RDF (external) and SAM/Web (internal)	RDF	RDF	RDBMS	Internal and XML view	OWL and RDBMS
1.4 Query Language	SQL	SPARQL	Karma query API + SQL	JPIS + JPPS queries + Perl	ProQA + RDQL +TriQL	vtPQL and SQL	ES3 Queries	SQL + transitive closure	Internal Graph QL	Internal Graph QL	Custom	Semantic Extended DASL (SEDASL)	iTQL	iTQL	SQL +XML Querying + text querying	XQuery, PQuery + Java	SPARQL + SQL
1.5 Research Emphasis	E/R/S/Q	R/S/Q	E/R	R/S/Q	E/R/Q	E/R/S/Q	R/S/Q	S/Q	E/R/Q	E/R/Q	E/R/S	R/S/Q	Q	Q	E/R/S	R/Q/S	E/R/S/Q
1.6 Challenge Implementation	Run	Run	Run	Run	Simulated	Run	Run	Simulated	Partial	Partial	Run	Partial	Run	Run	Run	Partial	Run
2. Properties of Provenance Representation																	
2.1 Includes workflow representation	yes	no	no	yes	yes	yes	no	yes	yes	yes	no	yes	no	no	yes	no	yes
2.2 Data Derivation vs Causal Flow of Events	E	D E	D E	E	D	D E	D E	D E	D E	D E	D E	D	D	D	D	D E	D
2.3 Arbitrary annotations in scope/implemented	(+AS/+AI)	(+AS/+AI)	(-AS/+AI)	(+AS/+AI)	(+AS/+AI)	(+AS/+AI)	(+AS/-AI)	(-AS/+AI)	(+AS/-AI)	(+AS/-AI)	(+AS/+AI)	(+AS/+AI)	(+AS/+AI)	(+AS/+AI)	(+AS/+AI)	(-AS/+AI)	(+AS/+AI)
2.4 Time supported/required	(+TS/+TR)	(+TS/+TR)	(+TS/+TR)	(+TS/-TR)	(+TS/+TR)	(+TS/+TR)	(+TS/+TR)	(+TS/+TR)	(+TS/-TR)	(-TS/-TR)	(+TS/-TR)	(+TS/-TR)	(+TS/-TR)	(+TS/-TR)	(+TS/+TR)	(+TS/-TR)	(+TS/+TR)
2.5 Naming required (if yes, then what)	keys for ports and data	no	GUIDs for data, services, workflows	no	lsids for all data	no	UUID assigned to every provenance relevant object	Unique ids assigned to every provenance relevant object	yes for tokens	yes for collections and their members	yes, unique ids for files per volume	unique ids (urls or lsid) for data to be tracked	URIs	URIs	logical file names	no	logical file names and file domain metadata attributes
2.6 Tracked data, and granularity	port level (I/O) but not their contents	all data	Any GUID assignable data	arbitrary	I/O of any type, but not their contents	changes to workflow specification and execution log	file or process	I/O objects of any type (data items and collections)	uniform streams of tokens	collections hierachica trees of het. data	file or process	I/O of any type, but not their contents	any relationshi (content not stored)	any relationshi (content not stored)	file	anything	files and nested file collections
2.7 Abstraction mechanims	layered provenanc model	grouping on files, processes, and workflows	grouping on data products, processes, and nested workflows		aspects	layered provenance model	script or job	user view of composite steps			N/A				data sets and nested procedure	tracers	reusable templates - workflow instances - execution details

Figure 2. Summary of contributions.

3.1. Categorization 1: Characteristics of provenance systems

C1.1 Execution environment: In many cases (but not all), provenance systems are embedded in a specific execution environment. The most common environments are workflow systems and operating systems. When embedded in a single execution environment, provenance representation may become (though does not have to be) dependent on the execution technology. On the one hand, such approaches may offer opportunities for optimization, which indeed were exploited by some teams. On the other hand, it makes representations technology specific, and brings difficulties if applications are composed of several execution environments.

C1.2 Execution environment (*for the challenge*): When systems allow for multiple execution environments, we indicate which one was actually used for the challenge.

C1.3 Representation technology: Provenance is represented and stored using a range of technologies, including relational databases (RDBMS), semantic web technologies (RDF, OWL), and internal private formats. Several systems also expose provenance according to an XML view.

C1.4 Query language: Systems offer query interfaces that operate over the stored representation of provenance. In some systems, the supported language is standard, whereas for others, it is purpose-built.

C1.5 Research emphasis: Teams have different research objectives when investigating provenance concepts. Their research may focus variously on techniques for *executing* (E) workflows such as the one defined in the challenge; *recording* (R) a description of a process being executed; *storing* (S) descriptions of process in persistent storage; and/or *querying* (Q) stored descriptions, in a way that captures the user's interest.

C1.6 Challenge implementation: Some teams executed the challenge workflow (run); others executed the challenge with fake image processing components (partial), making use of the data and intermediary results published with the challenge definition; finally, others fully simulated its execution (simulated).

3.2. Categorization 2: Properties of provenance representation

At some level of abstraction, provenance captures a notion of a causal graph, explaining how a data product or event came to be produced in an execution. However, there are variations on this theme, as indicated by the following criteria:

C2.1 Includes workflow representation: Some systems assume that an explicit representation of a workflow is part of the provenance representation, whereas others do not have such an assumption, and hence rely on other means for describing executions.

C2.2 Data derivation vs causal flow of events: Some systems describe derivation of data (e.g. conversion was applied to '.pgm' input to produce '.gif' output), whereas others document causal flow of events (e.g. writing of a file is followed by its opening for reading). Some are capable of characterizing both data- and event-oriented views.

C2.3 Annotations: Annotations entered by users may provide valuable information pertaining to data products or executions. While most systems were able to support the challenge queries related to annotations, not all systems considered annotations to be in the scope of provenance. In the matrix `+AS` (resp. `-AS`) denotes that annotations are in scope of provenance (resp. not in scope),

DOI: 10.1002/cpe

whereas +AI (resp. -AI) indicates annotations were implemented (resp. not implemented) for the challenge queries.

C2.4 Time: A representation of provenance does not have to include time, but it is perceived that it is practical for users to be able to refer to time. Therefore, most systems support a notion of time, so that users can refer to executions or data products according to the time they took place or were produced. However, this requirement brings the challenge of identifying which clock to use, given that distributed clocks may return different times. In the matrix, +TS (resp. -TS) indicates that time is supported for challenge queries (resp. not supported), whereas +TR (resp. -TR) time denotes that time is required (resp. not required) for capturing a correct representation of provenance.

C2.5 Naming: In order to be able to identify data products, some systems require each product to be identified by a unique name, typically created during workflow execution; such a name can then be used to query about the provenance of data products. Other systems do not require names to be assigned, but see the identification of data items as a query in itself.

C2.6 Tracked data, *granularity*: Systems are capable of tracking the provenance of different kinds of data; some introduce restrictions on the granularity of data they can track the provenance of. For instance, systems may or may not deal with collections, files, bytes, or bits.

C2.7 Abstraction mechanisms: When processes or data products are complex, it is useful to describe them with different levels of abstractions, sometimes hiding details of execution or representation, and at other times providing them. Some provenance systems provide support for this, by introducing new concepts in their provenance representation.

4. CONCLUSIONS

The rest of this special issue consists of papers describing the different systems summarized in Figure 2. We judge that the Provenance Challenge was highly successful, as measured by the number of participating teams, the quality of their submissions, and the discussions that resulted during the workshop. A number of lessons were learned from the challenge.

- At times, provenance queries were considered ambiguous. In future, it would be interesting to specify them better, more precisely and unambiguously, and to characterize the performance implications of the queries.
- While most participating teams could tackle all queries, it is unclear yet whether they all obtained the same or equivalent answers.
- The community lacks consistent and coherent terminology for provenance-related concepts [28]. A consistent terminology would help outsiders to easily grasp issues and compare systems.

Following the discussions at the two-day workshop, the provenance research community has decided to organize a second Provenance Challenge to address some of these issues in a systematic manner.

REFERENCES

1. Moreau L, Groth P, Miles S, Vazquez J, Ibbotson J, Jiang S, Munroe S, Rana O, Schreiber A, Tan V, Varga L. The provenance of electronic data. *Communications of the ACM* 2007.

2. Deelman E, Gil Y (eds.). Workshop on the challenges of scientific workflows. *Technical Report*, Information, Sciences Institute, University of Southern California, May 2006.
3. Bose R, Frew J. Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys* 2005; **37**(1):1–28.
4. Miles S, Groth P, Branco M, Moreau L. The requirements of recording and using provenance in e-science experiments. *Journal of Grid Computing* 2007; **5**(1):1–25. http://eprints.ecs.soton.ac.uk/10269/.
5. Simmhan Y, Plale B, Gannon D. A survey of data provenance in e-science. *SIGMOD Record* 2005; **34**(3):31–36.
6. Bose R, Foster I, Moreau L. Report on the International Provenance and Annotation Workshop (IPAW'06). *SIGMOD Records* 2006; **35**(3):51–53. http://www.sigmod.org/sigmod/record/issues/0609/sigmod-record.september2006.pdf [September 2006].
7. Moreau L, Foster I (eds.). *Provenance and Annotation of Data—International Provenance and Annotation Workshop, IPAW 2006* (*Lecture Notes in Computer Science*, vol. 4145). Springer: Berlin, 2006.
8. Zhao Y, Dobson J, Foster I, Moreau L, Wilde M. A notation and system for expressing and executing cleanly typed workflows on messy scientific data. *SIGMOD Record* 2005; **34**(3):37–43.
9. Fox GC, Gannon D. Special issue: Workflow in grid systems. *Concurrency and Computation*: *Practice & Experience* 2006; **18**(10):1009–1019. http://dx.doi.org/10.1002/cpe.1019.
10. http://twiki.ipaw.info/bin/view/Challenge/FirstProvenanceChallenge [June 2006].
11. Head D, Snyder AZ, Girton LE, Morris JC, Buckner RL. Frontal-hippocampal double dissociation between normal aging and Alzheimer's disease. *Cerebral Cortex*, September 2004. fMRI Data Center Accession Number: 2-2004-1168X. DOI: 10.1093/cercor/bhh174.
12. Barga RS, Digiampietri LA. Automatic capture and efficient storage of escience experiment provenance. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1235.
13. Golbeck J, Hendler J. A semantic web approach to tracking provenance in scientific workflows. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1238.
14. Simmhan YL, Plale B, Gannon D. Querying capabilities of the karma provenance framework. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1229.
15. Krenek A, Sitera J, Matyska L, Dvorak F, Mulac M, Ruda M, Salvet Z. Glite job provenance—A job-centric view. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1252.
16. Zhao J, Goble C, Stevens R, Turi D. Mining Taverna's semantic web of provenance. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1231.
17. Scheidegger C, Koop D, Santos E, Vo H, Callahan S, Freire J, Silva C. Tackling the provenance challenge one layer at a time. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1237.
18. Frew J, Metzger D, Slaughter P. Automatic capture and reconstruction of computational provenance. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1247.
19. Cohen-Boulakia S, Biton O, Cohen S, Davidson S. Addressing the provenance challenge using zoom. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1232.
20. Ludäscher B, Podhorszki N, Altintas I, Bowers S, McPhillips TM. Models of computation and provenance, and the RWS approach. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1234.
21. Bowers S, McPhillips TM, Ludäscher B. Provenance in collection-oriented scientific workflows. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1226.
22. Seltzer M, Holland DA, Braun U, Muniswamy-Reddy K-K. Passing the provenance challenge. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1227.
23. Schuchardt K, Gibson T, Stephan E, Chin G Jr. Applying content management to automated provenance capture. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1230.
24. Futrelle J, Myers J. Tracking provenance semantics in heterogeneous execution systems. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1253.
25. Clifford B, Foster I, Hategan M, Stef-Praun T, Wilde M, Zhao Y. Tracking provenance in a virtual data grid. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1256.
26. Miles S, Groth P, Munroe S, Jiang S, Assandri T, Moreau L. Extracting causal graphs from an open provenance data model. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1236.
27. Kim J, Deelman E, Gil Y, Mehta G, Ratnakar V. Provenance trails in the wings/pegasus system. *Concurrency and Computation*: *Practice and Experience* 2007; DOI: 10.1002/cpe.1228.
28. Moreau L. Usage of 'provenance': A tower of babel. Towards a concept map—Position paper for the Microsoft life cycle seminar, Mountain View, 10 July 2006. *Technical Report*, University of Southampton, June 2006.

LUC MOREAU, BERTRAM LUDÄSCHER,
ILKAY ALTINTAS, ROGER S. BARGA, SHAWN BOWERS, STEVEN CALLAHAN,
GEORGE CHIN JR., BEN CLIFFORD, SHIRLEY COHEN, SARAH COHEN-BOULAKIA,

Susan Davidson, Ewa Deelman, Luciano Digiampietri, Ian Foster, Juliana Freire,
James Frew, Joe Futrelle, Tara Gibson, Yolanda Gil, Carole Goble,
Jennifer Golbeck, Paul Groth, David A. Holland, Sheng Jiang, Jihie Kim,
David Koop, Ales Krenek, Timothy McPhillips, Gaurang Mehta, Simon Miles,
Dominic Metzger, Steve Munroe, Jim Myers, Beth Plale, Norbert Podhorszki,
Varun Ratnakar, Emanuele Santos, Carlos Scheidegger, Karen Schuchardt,
Margo Seltzer, Yogesh L. Simmhan, Claudio Silva, Peter Slaughter,
Eric Stephan, Robert Stevens, Daniele Turi, Huy Vo, Mike Wilde,
Jun Zhao, Yong Zhao