ISBN 978-89-6211-492-8 98560

전자 데이터에 대한 이력관리 기술 조사

남덕윤

KISTI 슈퍼컴퓨팅본부 차세대연구환경개발실

목차

1. 서론

디지털 자료가 보편화된 요즘 전자 데이터의 Provenance에 대해 생각해 보고자 본 문서를 작성한다. 계산과학의 시뮬레이션 수행과 관련된 데이터의 관점에서 살펴보고자 하며, 특히 계산 시뮬레이션 결과 데이터에 대한 provenance를 고려하려 한다.

Provenance는 예술품의 문서화된 역사 (documented history) 에 대한 미술품 연구에서 잘 이해되어 있다. 문서화된 역사에서 객체(object)는 학자들이 그것의 중요성과 다른 작업과 관계된 내용을 이해하고 정당하게 평가할 수 있게 하는 권위를 갖게 된다. 예술품에 있어서 증명된 역사가 부족한 경우들은 연구하는 사람들로 부터 의심을 받는다. 컴퓨터 시스템에 의해 생산된 데이터의 provenance를 알 수 있다면, 사용자들은 문서들이 어떻게 모여지고, 시뮬레이션 결과들이 어떻게 결정되고, 제정적인 분석이 어떻게 수행되었는지 이해할 수 있다 [1]. 이를 위해 컴퓨터 응용 프로그램들은 provenance-aware가 되어야 하며, 데이터의 provenance는 추출되고, 분석되고, 추론될 수 있다.

2. Provenance의 의미

Oxford english dictionary에서 Provenance는 다음의 내용으로 설명하고 있다.

- The fact of coming from some particular source or quarter
- The history or pedigree of a work of art, manuscript, rare book, etc.; concretely, a record of the ultimate derivation and passage of an item trough its various owners

여기서 우리는 특정 출처로부터 item의 특정 상태로 가는 유래 (derivation)를 provenance로 간주할 수 있다. 이러한 유래에 대한 서술들은 사용자의 개인적인 관심에 따라, 다른 형태를 취하거나 다른 속성들이 강조될 수 있다. 예를 들어, 예술작업에 있어서 provenance는 일반적으로 소유권의 연속 (chain of ownership)을 식별하거나, painting의 실제 상태가 허용되는 다양한 복원 방법의 연구를 통해 보다 잘 이해 될

수도 있다.

Provenance에 대한 computer 기반 표현 (computer-based representation)은 electronic data를 믿어야 할지 말아야 할지 결정하길 원하는 사람들에게는 매우 중요하다. 이상적으로는 사용자들이 이전 계산의 재수행을 통해 그들의 결과를 재생산하고, 왜 같은 입력값과 함께 외형상 동일한 수행들이 다른 결과를 생산하는지 이해하고, 어떤 데이터 셋, 알고리즘, 또는 서비스들이 그들의 유도과정에 포함되어 있는지 결정할 수 있어야 한다.

e-Science나 business 측면에서, 사용자, 검토자, 감사, 심지어 조정자들이 특정 규제나 방법들에 의해 생산된 결과를 도출하는 과정을 검토해야만 한다. 게다가 그들은 주어진 권한하에서 결과들이 서비스들이나 데이터베이스들로부터 독립적으로 도출되었다는 것을 증명해야 한다. 또한 정밀한 기술적 특징을 같는 실험기계들에 의해 source에서 데이터를 취득했다는 것을 확립해야 한다.

어떤 사용자들은 오늘 당장 그러한 작업을 수행해만 할 때, 그렇게 할 수 없거나, 완전하지 않은 상태에서 약간만 수행하는 수도 있다. 왜냐하면 기저에 깔려있는 원리가 완전히 조사되지 않았고 시스템도 그러한 요구사항을 지원하도록 설계되지 않았기 때문이다. 주목할 사항은 전자 데이터는 전형적으로 사용자와 검토자 또는 규제자가 필요한 증명을 하도록 도와주는 historical information을 포함하지 않는다는 점이다. 이에 추가적인 정보나 실행시에 발생하는 것들을 기록해 놓은 process documentation을 획득해야 한다.

Process documentation과 소유권의 기록은 electronic data와 예술품에 대응된다. Provenance-aware applciation은 process documentation을 만들고, provenance store에 이를 저장한다. Provenance store 는 process documentation의 long-term persistent, secure 저장소이다. 이것의 역할은 다양한 물리적 보급을 조정한다. 예를 들어 provenance store는 autonomous service 이거나 (보다 scalable 하기 위해) 분산 저장의 federation일 수 있다.

그림 1은 Provenance life cycle 예제이다. Process documentation이 기록되었을 때, 데이터 결과의 provenance 는 provenance store에 질의함으로써 추출될 수 있고,

사용자의 니즈에 맞게 분석될 수 있다. Provenance store와 그 내용은 managed, maintained, or curated 할 필요가 있을 수 있다.

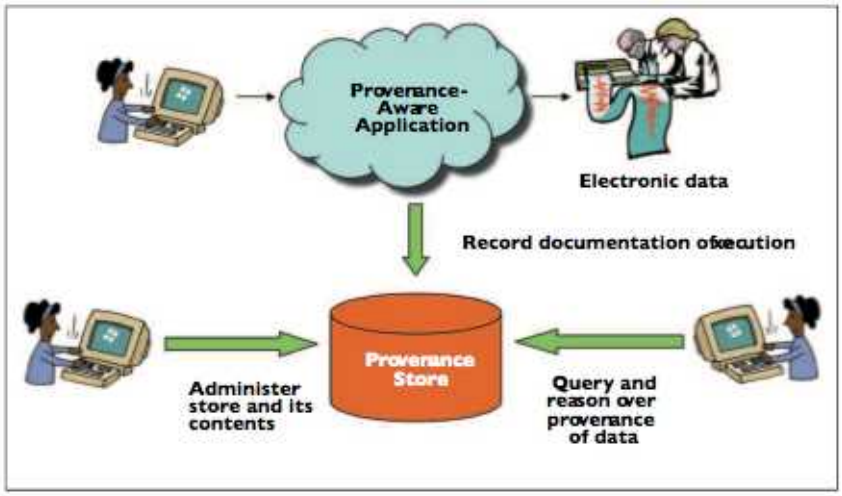

그림 1. Provenance life cycle 예제 [1]

3. 컴퓨터 공학 분야별 Provenance 연구 관점

Computer science 분야에서도 다양한 분야에서 provenance에 대한 연구를 진행해 왔다.

- Database 분야

Database 분야에서, provenance는 data annotation의 내용과 하부 DB들에서 "source" 데이터와 관련된 view에서 정보를 추적하게 도와주는 수단으로써 data warehouse에서 연구가 진행되었다 [2, 3]. DB 연구에서는 Wang-Chiew Tan의 글[4]이 전반적인 개요를 제공하고 있다.

- Scientific workflow system 분야

Scientific workflow system에서, provenance는 반복성을 보장하고, 값비싼 재계산을 피하기 위해 취급되었다 [5, 6]

- Bioinformatics 및 다른 Scientific DB 분야

Bioinformatics와 다른 scientific DB들에 있어서는, DB의 변경 이력을 기록하는 provenance 정보는 과학적인 가치를 결정하기 위해 유용한 것으로 생각 되었다 [7].

- Security 분야

Security 측면에서, provenance는 networked system에서의 data에 대한 integrity를 제공하는 문제의 도전적인 부분으로 고려되고 있다 [8].

- Semantic web system 분야

Semantic web system에서, provenance는 추론 기반 검색 결과들의 의미를 이해하는데에 도움을 주기 위해 사용자에게 제공될 필요가 있는 "proofs"나 "explanations"의 형태로써 연구되고 있다 [9].

4. Open model for process documentation

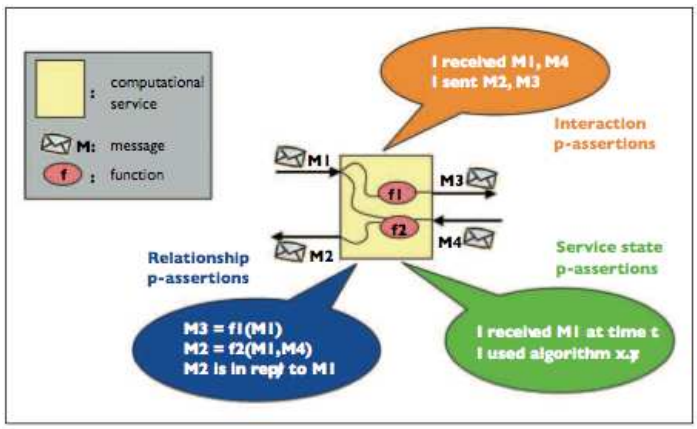

그림 2. p-assertion 예제

많은 응용프로그램을 위한 Process documentation은 하나로 표현될 수 없으며, 실행 중간에 연속적으로 표현되어야 한다. 프로세스의 문서는 프로세스에 포함된 서비스에 의해 만들어 지는 p-assertaion 집합을 포함한다. p–assertion은 process에 포함된 개별 프로그램에 의한 선언으로 3가지 타입이 있다. 서비스 사이의 데이터 플로우를 표현하는 Interaction p-assertion, 서비스 상태를 표현하는 service state p-assertion, 서비스 내의

데이터 플로우를 표현하는 relationship p-assertion이다. 이러한 flow들이 실행 시의 인과관계 및 data dependencies를 표현하며, DAG 을 구성한다. 그림 2에서 하나의 computation service에 대한 p-assertion을 볼 수 있다.

Internal service states는 (서비스의 성능이나 정확성 같은) 실행의 nonfunctional 특징과 이 서비스들이 계산한 결과들의 본질을 이해하기 위해 필요하다. 이에 service-state p-assertion은 특정 interaction의 문맥에서 내부 상태에 대해 서비스에 의해 제공되는 문서이다. Service-state p-assertion은 다양하며,계산에서 서비스에 의해 사용되는 디스크 용량과 CPU time, action이 발생할 때의 local time, 생산된 결과의 floating-point precision, 또는 application-specific state description을 포함한다.

Provenance-aware applciations가 interoperable하게 하기 위해서는, shared data model에 따라 process documentation이 구조화 되는것이 중요하다. 사우스햄턴 대학에서 제안한 documentation 모델의 개방성은 응용 기술들에 독립적으로 설계되었다. 이러한 특징들은 process documentation이 application service들에 의해 자동으로 생산되고 provenance queries가 표현될 수 있는 open format으로 표현될 수 있도록 한다. 그림 3은 Organ Transplant Management (OTM) system 예제로, 여기서 p-assertion들과 실제 프로세스를 정의한 것을 알 수 있다.

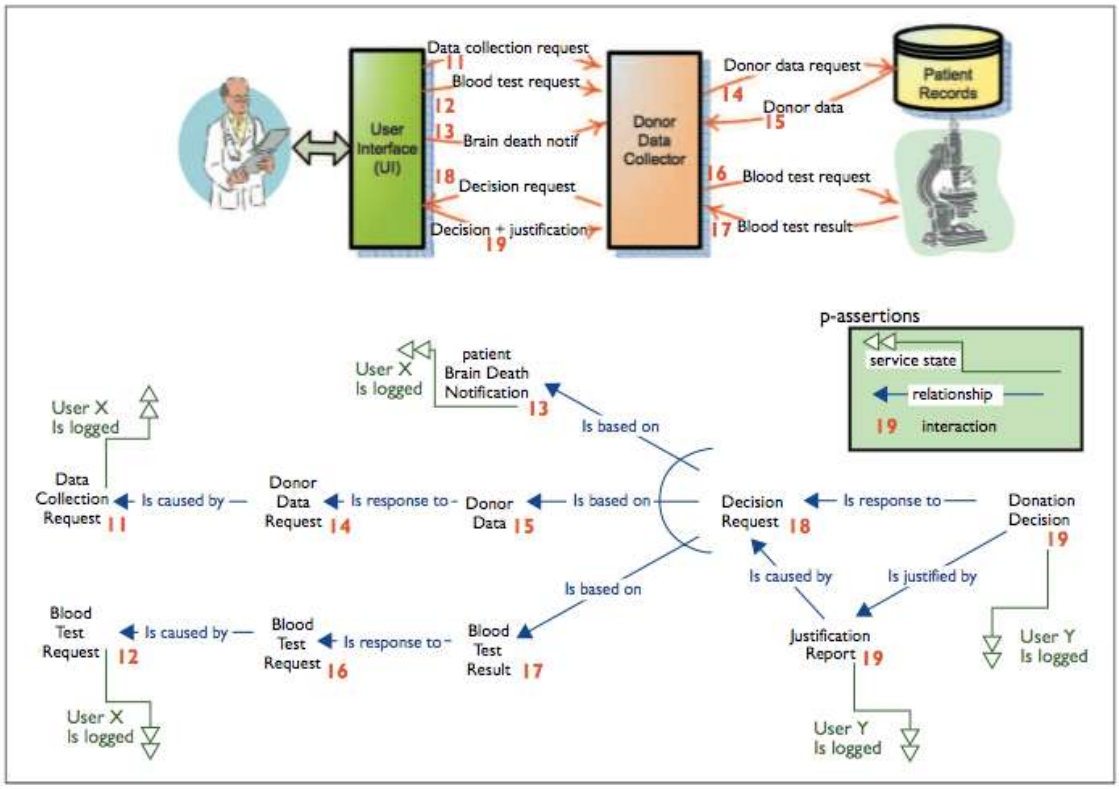

그림 3. Organ Transplant Management (OTM) 예제 [1]

5. Querying the provenance of electronic data

Provenance queries는 electronic data의 provenance를 획득하는 경우를 목표로한 process documentation에 대한 user-tailored queries이다. 이러한 문맥에서, 사용자에게 관심이 있는 data item은 특징지어져야 한다. 데이터가 변하기 쉽다면, 사용자가 찾기 원하는 실행 시점에 따라 그것의 provenance 또는 history가 다를 수 있다. Provenance query는 메시지를 보내고 받는 것과 같은 주어진 문서화된 이벤트에 따라 data item을 식별할 수 있어야만 한다.

Data item이 최종적으로 무엇이냐에 대한 모든 것에 대한 full detail은 매우 방대할 수 있다. 예를 들어 실험 결과에 대한 full provenance는 거의 항상 성분들을 생산하는 어떤 성분들에 대한 provenance와 실험에서 사용하는 장비와 소프트웨어와 함께, 실험에서 성분들을 생산하는 과정의 description을 포함한다. documentation을 활용할 수 있다면, full provenance는 최종적으로 시작 시점이나 적어도 provenance awareness의 특정 시점으로 돌아갈 술 있는 과정들의 세부 내용들을 포함한다.

사용자들은 provenance query를 통해 과정 상에서, 특히 data flow DAG에서의

가로지르는 역 그래프를 수행하고, query-specified scope에 따라 종료되는, 관심의 범위(scope of interest)를 표현할 수 있어야 한다. Query output은 DAG subset이다. Scoping은 관계, 중간 결과들, 서비스들 또는 하부 프로세스들의 유형에 기반할 수 있다.

6. 활용 예시: 시뮬레이션 모사

여기서 Provenance 기능의 일부로 시뮬레이션의 모사에 대한 내용을 제안한다. 유체역학분야 교육에 활용되는 e-AIRS 시스템은 연구용으로 쓰일 때와 다른 사용자 패턴이 존재한다. 실습에서 활용될 때, CFD 시뮬레이션 코드가 결정론적 알고리즘이고 동일한 입력값, 동일한 모델로 수행하는 경우 동일한 결과를 보일 것이다. 예를 들어, 보통 학생들은 실습시간에 조교가 지시하는 내용대로 실습을 수행함으로 동일한 시뮬레이션에서 동일한 결과를 볼 것이다. 이에 컴퓨팅 자원은 유한한 상황에서 시뮬레이션 결과를 빨리 확인하기 위해서는 기존의 결과 값을 확인하게 함으로써 시스템의 성능 및 실습의 효과를 개선할 수 있다. 이에 본 절에서는 기존의 시뮬레이션 결과를 확인해 볼 수 있는 시뮬레이션 수행 모사 기능을 제안한다.

교육에 사용될 때의 문제점은 학생들의 일부가 짧은 시간 내에 동일한 작업, 즉 동일한 알고리즘, 모델, 입력값을 활용한 작업이 컴퓨팅 자원에 반복 실행된다는 점이다. 이로 인해, 컴퓨팅 자원에는 동일한 결과를 얻기 위한 작업 수행으로 과부하가 걸리게 되고, 실습의 목적을 달성하기 위한 노력보다는 결과를 기다림으로써 생기는 낭비 시간이 늘어나게 된다. 실습의 목적을 이루기 위해, 시뮬레이션을 수행하기 위한 전처리 과정은 동일하게 수행토록 한 다음, 컴퓨팅 자원에 작업을 제출할 시, 기 수행된 시뮬레이션 결과가 있다면 이를 바로 확인하게 해 줌으로써, 시뮬레이션 종료 시까지 기다리는 시간을 줄일 수 있다.

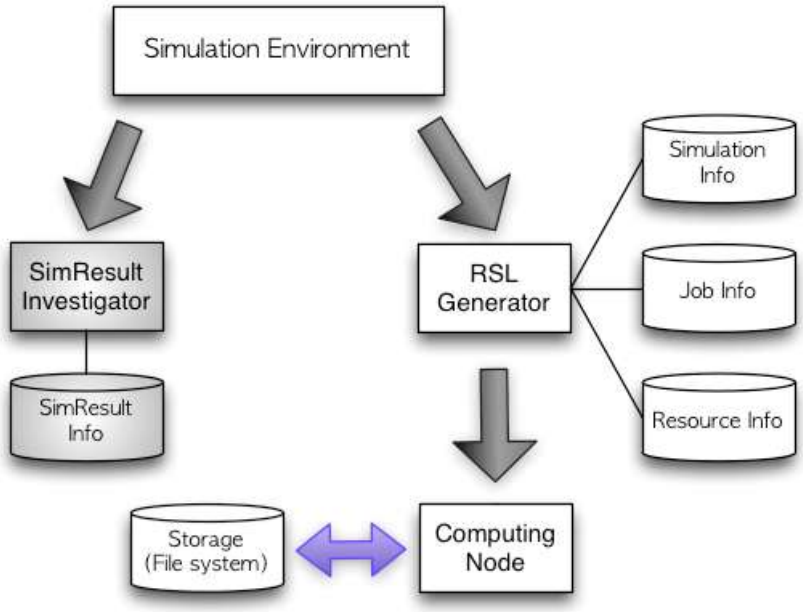

그림 4. 시뮬레이션 모사 환경

그림 4는 시뮬레이션의 모사를 위한 환경이다. 그림의 오른편은 기존의 Globus를 활용한 그리드 자원에서의 작업실행 환경이다. eAIRS에서는 Simulation, Job, Resource에 대한 정보를 가지고 있다. 사용자가 시뮬레이션 수행을 위한 작업 제출을 하는 경우, RSL Generator는 작업 제출을 위한 정보들을 수집하여, 작업제출서인 RSL을 만들어 컴퓨팅 자원에 제출한다. 그러면 컴퓨팅 자원에서는 시뮬레이션을 실행하고 그 결과를 저장장치에 저장한다.

시뮬레이션 모사를 위해 저장장치에 저장된 작업 수행을 위한 정보, 수행 결과들에 대한 시뮬레이션 결과 정보(SimResult Info)를 저장해 둔다. 그러면 사용자가 작업 제출을 할 경우, 컴퓨팅 자원에 대한 시뮬레이션 수행을 바로 지시하는 것이 아니라, 기존에 수행된 동일한 시뮬레이션 결과가 있는지 조사한다. 이를 SimResult Investigator에서 시뮬레이션 결과에 대한 SimResult 정보를 검색한다. 동일한 시뮬레이션 수행 결과가 있다면, SimResult Investigator는 Storage에서 사용자의 account 내의 local storage로 시뮬레이션 결과 파일을 복사해 둔다. 그러면 사용자는 시뮬레이션을 수행한 것과 똑같은 형태의 결과를 얻게 되며, 이러한 과정은 eAIRS 내부의 서버 단에서 수행하는 작업이다. 이 기능을 통해, 앞서 언급했듯 시스템 성능 및 실습 효과를 개선할 것으로 기대한다.

7. 관련 연구

여기에 소개되는 방법들은 데이터 모델과 인터페이스의 개방형 스펙을 작성하는데에 기본으로 사용되었던 complete architectural specification [10]을 결과로 하는 광대한 requirement analysis [11]에서 도출했다. 개방형 접근방법은 Web services, command-line executables, and monolithic executables과 같은 다수의 기술들이 포함된 복잡한 분산 응용프로그램의 문서화를 가능하게 한다. 또한 복잡한 provenance queries의 표현은 데이터와 사용된 기술들에 독립적인 프로세스들을 식별하게 한다.

Virtual Data System과 myGrid는 provenance에 대한 지원을 제공하는 scientific workflows에 대한 실행 환경들이다. 이들은 p-assertion 에 적합한 데이터 모델을 이용하는 workflow 제정자의 관점에서 문서를 생산하는데 초점을 맞추고 있다. 이들은 compact process documentation을 획득하게 하는 각자의 workflow language를 가정한다. Process documentation을 위해 개방형 데이터 모델을 채택함으로써, 이러한 시스템들은 provenance queries를 연이어서 실행하는 이종의 응용프로그램으로 통합될 수 있을 것이다.

데이터베이스 커뮤니티에서 또한 provenance에 대해 연구가 진행되고 있으나, 다른 가정을 하고 있다. 예를 들어 결과의 원천을 식별하기 위해 queries를 역추(reversed)하는 것이 가능하다고 가정한다. Provenance queries의 특정한 인스턴스로 값을 가짐으로써, 다른 종류의 provenance가 보여진다 [12].

Harvard Univ.에서 개발한 Provenance Aware Storage System은 OS에서 파일 시스템의 이벤트를 획득함으로써 실행에 대한 문서화를 자동으로 생산하도록 설계되었다. 모든 다른 접근방법들처럼, small-grain documentation을 capture하는 것은 scalability and performance challenge를 포함한다. 이에 사용자에 대해 적당한 추상화 수준에서 정보를 도출하는 것은 어렵다.

8. 결론

오늘날의 IT 관점에는 실행 중에 결과와 서비스들을 발견하면서, 개방되어 있으며 동적으로 구성되는 응용 프로그램들이 포함된다. 사용자들은 그들은 응용프로그램의 electric data에 대해 신뢰할지 안할지를 알아야 한다. 그런 까닭에 이는 생산으로 이끈 포로세스가 기술된 provenance에 의해 충족되어져야 한다.

이러한 비전을 이루기 위해, 기술과 관련 없이, 응용프로그램들이 사용자의 니즈에 맞게 제작된 provenance queries를 수행하기 위해 사용되어 질 수 있는 개발형 데이터 모델에서의 실행을 문서화하는 개방형 접근 방식이 필요하다. 학자들이 그들의 문서화된 역사를 연구함으로써 예술 작품들에 감사할 수 있는 것과 같이, 사용자들은 provenance queries 덕분에 electronic data에 대한 신뢰를 획득할 수 있을 것으로 기대한다.

[참고자료]

[1] Luc Moreau , Paul Groth , Simon Miles , Javier Vazquez-Salceda , John Ibbotson , Sheng Jiang , Steve Munroe , Omer Rana , Andreas Schreiber , Victor Tan , and Laszlo Varga, "The provenance of electronic data," Communications of the ACM, v.51 n.4, p.52-58, April 2008.

[2] Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. VLDB Journal, 14(4):373–396, 2005.

[3] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Syst., 25(2):179–227, 2000.

[4] Wang-Chiew Tan. Provenance in databases: Past, current, and future. IEEE Data Eng. Bull., 30(4):3–12, 2007..

[5] Rajendra Bose and James Frew. Lineage retrieval for scientific data processing: a survey. ACM Comput. Surv., 37(1):1–28, 2005

[6] Yogesh Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. SIGMOD Record, 34(3):31–36, 2005.

[7] Peter Buneman, Adriane Chapman, and James Cheney. Provenance management in curated databases. In SIGMOD 2006, pages 539–550, 2006.

[8] INFOSEC hard problem list. Technical report, INFOSEC Research Council, 2005. http://www.infosec-research.org/-docs_public/20051130-IRC-HPL-FINAL.pdf.

[9] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Rob McCool. Knowledge provenance infrastructure. IEEE Data Eng. Bull., 26(4):26–32, 2003.

[10] Groth, P., Jiang, S., Miles, S., Munroe, S., Tan, V., Tsasakou, S., and Moreau, L. D3.1.1: An Architecture for Provenance Systems. Technical Report. University of Southampton, Southampton, U.K., Feb. 2006; eprints.ecs.soton.ac.uk/12023/

[11] Miles, S., Groth, P., Branco, M., and Moreau, L. The requirements of recording and using provenance in e-science experiments. Journal of Grid Computing 5, 1 (Mar. 2007), 1–25

[12] Buneman, P., Khanna, S., and Tan, W.-C. Why and where: A charac- terization of data provenance. In Proceedings of Eighth International Conference on Database Theory Vol. 1973 of Lecture Notes in Computer Science (London, Jan. 4–6). Springer, Heidelberg, 2001, 316–330