Modelling Provenance Collection Points and their Impact on Provenance Graphs

David Gammack, Steve Scott, Adriane P. Chapman

Marymount University, Arlington, VA USA dgammack@marymount.edu
The MITRE Corporation, McLean, VA USA {slscott, achapman}@mitre.org

Abstract: As many domains employ ever more complex systems-of-systems, capturing provenance among component systems is increasingly important. Applications such as intrusion detection, load balancing, traffic routing, and insider threat detection all involve monitoring and analyzing the data provenance. Implicit in these applications is the assumption that "good" provenance is captured (e.g. complete provenance graphs, or one full path). When attempting to provide "good" provenance for a complex system of systems, it is necessary to know "how hard" the provenance-enabling will be and the likely quality of the provenance to be produced. In this work, we provide analytical results and simulation tools to assist in the scoping of the provenance enabling process. We provide use cases of complex systems-of-systems within which users wish to capture provenance. We describe the parameters that must be taken into account when undertaking the provenance-enabling of a system of systems. We provide a tool that models the interactions and types of capture agents involved in a complex systems-of-systems, including the set of known and unknown systems in the environment. The tool provides an estimation of quantity and type of capture agents that will need to be deployed for provenance-enablement in a complex system that is not completely known.

Keywords: provenance, lineage, Agent Based Modelling, modelling and simulation, complex systems

1 Introduction

Provenance, the record of creation, update and activities that influence a piece of data, is used to: understand if data was produced correctly (according to published methodology, or according to policy); detect suspicious behavior within complex systems; and, enable trust during cross-organizational collaboration [3]. The utility of the provenance stream for these purposes is tied to what information is actually collected, and how far through the system the provenance can "see".

In our experience, when approached by government organizations seeking to become provenance aware, the first question becomes: How much of the system must

be provenance aware in order to utilize the provenance data stream in the desired manner? One of the first considerations is how many capture agents are needed to have good coverage of the system of systems. The next question is, which system(s), if provenance-capture enabled, will give the most "bang for the buck"?

In other words, an analysis of the system of systems with the utility of the final provenance data stream is required to put capture points at appropriate places. Unfortunately, in many cases, the "full workflow" isn't known because of *system complexity* and *human cognitive load*. In the case of *system complexity*, large numbers of systems exist to form a complex system of systems and no one person knows all systems and their interactions. The people required to administer a system know about their system, but not who is using it or why. The running joke in one IT center: Q: "How do you know who is using your system?" A: "Turn it off and listen in the hallway to who starts screaming." Meanwhile, the users of the system, see it merely as a tool and have no knowledge where the underlying data resides and what other systems may be called by the backend. Additionally, *human cognitive load* also plays a part in creating an inaccurate picture of the system-of-systems. When asked, users can faithfully recite their top-used tools, but become increasingly vague and forgetful of lesser used systems.

In this work, we extend a modelling tool designed to simulate the provenance data stream within a complex system of systems, with varying types and distributions of capture agents. The original work [12] was limited in that it allowed a user to specify the *number* of systems in a system-of-systems, but merely arranged that number upon a grid and randomly assigned how provenance was captured at each point. This work is a marked divergence from [12]; it is designed to take in the major systems used in a workflow that can be described with fidelity by users. It also models the "grey areas", those parts of the system that are not well-remembered or documented. The modelling tool allows a user to specify the known systems, and how they are expected to connect. Using this as a base, the tool will run simulations over that base with augmentations of "grey systems". The contributions of this paper are as follows:

1. A simulation tool that analyzes how much provenance is captured in a system of systems, including expansion of those systems to unknown and unspecified "grey systems".
2. A real-world use case from the US healthcare system that motivates the need for anticipating "grey areas" in provenance capture.
3. An application of the tool that shows how the system performs over likely scenarios.

Section 2 discusses related work, in particular capturing provenance, and Agent Based Modelling. In Section 3, we present foundations that describe the complex system of systems and motivate this work. Section 4 contains a real world US healthcare use case. The architecture of our system and details on its implementation are in Section 5, while in Section 6, the model is executed over various sample systems. Finally, in Section 7 we conclude and highlight future work.

2 Related Work

Provenance has been touted as a tool to assist with scientific collaboration [19, 20, 25]. Unlike the system-of-systems we describe above, most of these systems [4, 20, 25, 29] constrain the user to a single management system. Many "execution platforms" can be used, but with a central management system and user-defined system-of-systems, provenance tracking can occur with a single capture point within these workflow managers. [3, 9] describe provenance-based techniques for assessing data trustworthiness. However, in order to use the provenance, as described above, it must be captured.

2.1 Provenance Capture

As described in [10], we note that there are classes of provenance capture agents.

Coordination-points: In some systems-of-systems, there are "coordination points". Coordination points are systems or software that provide natural bottlenecks. Typically, these are systems that help order, transmit and manage data and jobs. Examples of current coordinate point capture points include MapReduce [24], UNIX kernel [22], GIT [23], and Enterprise Service Bus (ESB) [2]. Workflow (and yes/no workflow) management systems such as [4, 20, 25, 29] are also good coordination-points. Dynamic instrumentation has the same effect as a coordination point on a given system [26].

Application-based: In some cases, an application is so heavily used that it is beneficial to expend the resources to capture provenance information from just that application. Examples include [5, 11, 21].

Manual: Many standards, such as [1], include provenance as components of the required metadata; in many instances, much of that information is populated by hand by a data curator. Unless the user is particularly motivated to capture provenance, manual capture points have very low capture rates. Of particular interest are hybrid approaches in which the application itself is somewhat provenance-enabled, but the user makes the final decision as to what is important and needs to be stored [19].

Provenance reconstruction work, such as [16] is not considered in this work since the accuracy of this provenance is not always equal to that obtained via making applications and systems provenance aware.

2.2 Provenance Simulation

There have been several efforts to simulate provenance information. Typically, these revolve around creating samples of provenance artifacts. [8] creates sample provenance graphs based on workflows and user rules. The PLUS system [3] has DAGAholic, a tool that creates provenance graphs of specified size, connectivity, sub-graph patterns (tree, star, etc.). Sample provenance graphs such as [7, 10] are available on GitHub ProvBench. To our knowledge, there is no simulator that simulates capture points for a system of systems.

2.3 ABMs

Agent-Based Models (ABMs) are a type of distributed computational simulation in which a set of autonomous, goal-seeking, perceptive agents interact with each other and with their environment in order to achieve some outcome. ABMs originated in the field of computer science, in particular with multi-agent system (MAS) and distributed artificial intelligence (DAI). In DAI applications, a problem is defined such that it can be addressed in parallel by the efforts of multiple independent agents. In MAS applications, a number of agents address a problem in parallel by passing messages among one another in a shared environment. ABMs combine the parallel and distributed inter-agent communications of MAS applications with two-dimensional Cellular Automata (CA) models that are used to form an artificial landscape with which the agents may interact as well [13].

ABMs have been used in a variety of domains, including computational economics [27], auction markets [14], social network analysis [17, 18], and public policy modeling [6], to name a few.

In this study, an ABM is used as a computational platform to study the capture of provenance information in a complex system of systems. The model consists of an abstract landscape containing a network of arbitrarily connected agents, where the agents are used to represent systems in an interconnected set of systems that share data. The network also includes provenance monitoring agents as well, which represent systems that are capable of detecting and logging provenance relevant transactions. As time progresses in the simulation, information artifacts flow through the network, and are subject to simulated update events which may or may not be observed by a provenance monitoring agent, depending on the configuration of the original network and the placement of provenance monitoring capabilities in the network. The ABM is used as a modeling platform to study the interactions among systems in order to provide quantitative metrics on the level of provenance obtained with a particular distribution of provenance capture agents. Based on analysis of the simulated information flow, a predefined agent connection topology, and a particular placement of provenance monitoring systems in the simulation, various provenance management topologies can be quantitatively compared and evaluated.

3 Foundations

To reiterate the problem, when looking to understand a system of systems, interviews and code reviews are performed to understand systems touched and their interactions. The techniques uncover a set of found, known systems, N_F, and their interactions, E_F. We assume that the actual graph of interconnected systems, S, is:

$$S \supseteq (N_F, E_F)$$

S may also contain grey systems (i.e., those that are not well-remembered or documented), N_G, and as yet unfound connections E_G. Thus, the number of systems in the system of systems, S, is bounded by:

$$|N_F| \leq |N| \leq |N_F| + |N_G|$$

Meanwhile, the number of interactions, I, possible is bounded by:

$$|E_F| \leq |I| \leq (|N_F| + |N_G|)^2$$

Each I is an interaction between systems over which data may flow during the execution of a user's tasks; each I represents a possible edge in a provenance graph. Even with low numbers of grey systems, there can be large numbers of interactions that are missing from the provenance record. [10] contains provenance datasets that highlight how a poor choice in capture agents create holes in the provenance record and re-enforce the grey system's absence from the record. Unfortunately, as we discuss in subsequent sections, the systems-of-systems we are concerned about have a large number of N_F and a possibly large number of N_G. In addition, each system, either known or grey, has the potential to be provenance-enabled through a capture agent. We utilize the general categories described in Section 2: coordination point, application, and manual.

The goal is to determine where the best systems are to place capture agents, and type of capture agent. We assume that capture agents are expensive to build, deploy and maintain, and thus wish to minimize the number of capture agents while still capturing "good" provenance. There are several choices for "good provenance", depending on the desired usage of the provenance. An example set of "good" evaluations could be:

- 100% of all provenance is captured
- 80% of all possible provenance is captured
- At least 1 complete path between a source and sink exists

In the degenerate case, if we assume that every system in S can only support an application-based capture point, then it is easy to determine the minimum number of capture agents by inspection. If "good" is complete provenance, then every system must be provenance enabled; if "good" is 1 path from source to sink, then the shortest path can be provenance-enabled, etc. However, the problem quickly becomes trickier. Assuming that any system can be made a coordination point, that is, it can record the provenance for itself and all systems connected to it, then asking if there is a way to capture complete provenance in fewer than k capture points becomes the NP-complete Vertex Cover problem. For this reason, we utilize a modelling and simulation approach.

4 Use Case: Provenance in the US Healthcare System of Systems

In the US Healthcare system, key stakeholders include Patients, Providers, Payers, and Public Health. Each stakeholder has different incentives, and each maintains information about medical events, but at different levels of detail and for various reasons. The medical records and associated healthcare information illustrate the problems found in data provenance in a complex system of systems.

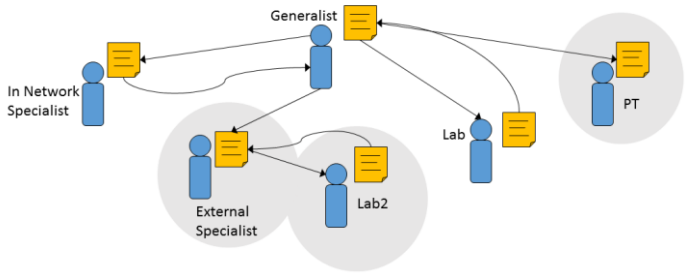

Fig. 1. Depiction of the medical record as it moves through providers. Some providers modify the record and send it back to the generalist's system. Others become an off-shoot. Systems in grey are part of the complex system of systems, but are not easily identified.

Consider the situation in which a patient is seeking medical care for a non-acute condition, for which an episode of care typically spans 6-8 months and involves coordination with generalists and specialists. Assume that the patient has medical insurance coverage under a private plan or a state sponsored health care exchange managed under the Affordable Care Act (ACA). Assume too, that the patient's main provider is part of a hospital-based physician's group. The patient first seeks care from his primary care physician. An initial set of medical records is generated based on this encounter. Based on a preliminary diagnosis, the patient is referred to two specialists and to a lab. The lab and one of the specialists is also a part of the hospital-based group. Because the traffic between these entities is high, the IT and physicians understand how the patient's record gets passed between them. If provenance capture was desired, it would be trivial to analyze these interactions and find the appropriate provenance capture points. Unfortunately, the second specialist and the special lab she sent our patient to are not in the hospital's network; it is unlikely that either of these systems would be remembered for inclusion in a large-reaching provenance system. Assume the patient is advised to seek physical therapy, and choses to do so close to work, not at the hospital. Again, the patient's medical record is passed to the physical therapist, but the therapist in effect runs a grey system – one essential to tracking the movement of the patient's record, but outside the well-worn and understood tracks of the hospital's in-network system. Fig. 1 shows the movement of the patient's health record, and both the known and grey systems in the complex system of systems.

5 The Model and Tool

The provenance capture simulator is built using NetLogo [28]. It has two operating modes: default, as described in [12] and user-assisted as we describe here. The model incorporates:

- Types of capture agents:
  - Coordination points capture provenance for themselves and any systems that interact with them.

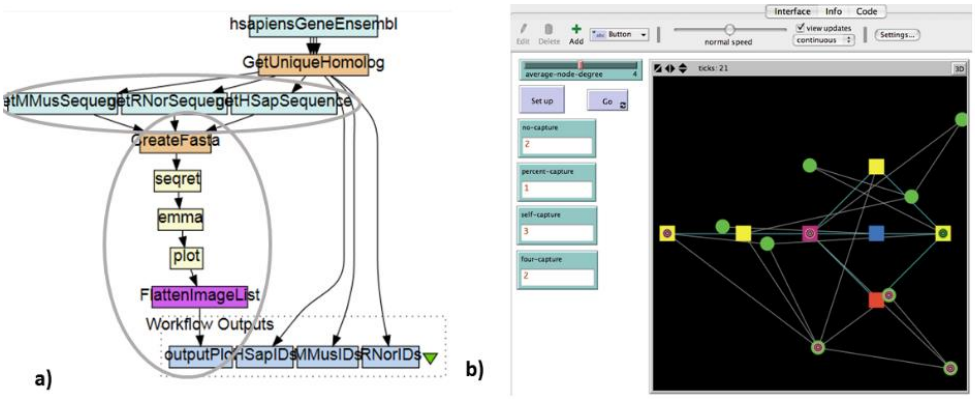

Fig. 2. a) Sample of a myExperiment workflow, BiomartAndEMBOSSAnalysis. The grey systems in this case are demarked by grey circles; b) A sample execution of the tool using the systems in 2a. Colored boxes and circles are known systems and grey systems respectively. Along the left, the user can specify the number of coordination-points, application-based and manual capture points to be used amongst the grey systems during the permutations, or the system can run through permutations randomly choosing capture types.

  - Application-based capture provenance for themselves only
  - Manual provenance capture based on a human recording the provenance. We assume that humans are lazy provenance agents, and only capture 10% of the time
  - None: no capture agent exists in this system
- Known systems, N_F, as specified by the user
- The interactions between the known system, E_F
- The capture agent type for each known system, chosen from: coordination point, application, manual, or none.
- A set number of unknown, grey, systems, $|N_G|$
- A maximum number of possible connections between grey systems
- The average number of connections a grey system may have, as specified by the user
- A user defined number of coordination-point application-based, manual and do-nothing capture agents to be distributed among the grey systems.
- The probability a piece of data will be directed along a new path. Data will always go through the set paths, and probability of P that it will also move down a second or third path at any node that has a degree > 1.

Our goal is to produce provenance streams that are useful. Obviously, usefulness is defined by the ultimate usage of the provenance data stream. For instance, for intrusion detection, seeing as close to 100% of the provenance as possible would detect the greatest number of possible anomalies. Determining trustworthiness based on data path similarity as in [9] we are looking for a complete path from source to sink for every data item.

The user of the simulation system specifies the known systems, and the types of capture agents they can support. Remember, only certain types of systems can be a coordination point (e.g. service bus, web proxy, message router, etc.), and otherwise

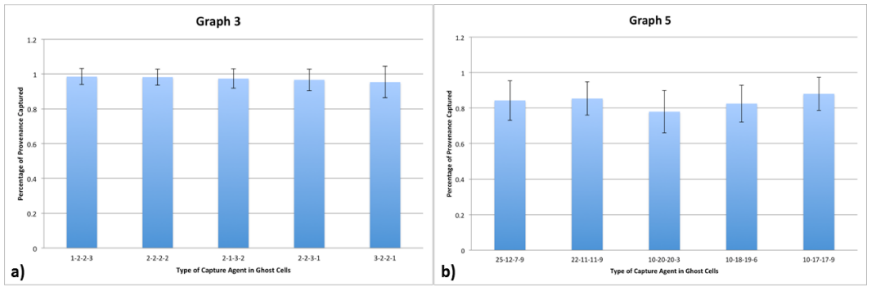

Fig. 3. Workflow 3 from **Table 1** run with varying ratios of capture types; b) Workflow 5 from **Table 1** run with varying ratios of capture types. The x-axis indicates the quantity of each type of capture agent: No Capture – Manual Capture – Application-based – Coordination Point.

can only be an application-based capture agent. Additionally, the user can specify how the systems are actually connected. Using this as a backbone, the system then randomly generates grey systems and their connections across that backbone. **Fig. 2** shows a sample execution.

Using this execution, and defining the allowed number and type of application agents, the output provenance can be analyzed to see if it meets the user's specification of "good" provenance. The simulation itself runs through permutations of grey system configurations, allowing the best, worst and average amount of provenance a given system of systems can capture to be highlighted.

6 Results and Discussion

The point of this section is to show that the tool can be run over any user-specified system-of-systems with different configurations, connections, grey-systems and capture agents. To this end, the tool is run on a standard laptop, and realistic system configurations were used, although chosen with little attention to the systems themselves. The system was run on a Mac with OSX 10.9.5 4. The system has a core processor speed of 2.7GB with 16GB RAM. Despite being run on a modest laptop, the simulations took on average less than a minute to run.

In an attempt to find descriptions of real, complex systems-of-systems, we turn to pre-defined workflows in myExperiment [15]. While all of the workflow technologies have strong provenance capture as discussed in Section 2, and have no need to analyze what systems involved may need capture agents, the workflows themselves provide a nice, bounded set of realistic system configurations. To showcase this tool, we chose first 5 workflows from myExperiment that satisfied the "runnable" facet, and chose 1 path from source to sink to represent known systems. All others are considered grey systems. Within the known systems, any system with > 3 edges in the workflow, is considered a coordination-point. For all other known systems, we rotate through application-based, manual and none, assigning them at random to the known systems. **Fig. 2** shows an example workflow from myExperiment that was translated into known

and unknown systems. We chose a path from source to sink as the known systems. The remaining systems we circled in grey. When actually executing the tool over this description, the edges between grey systems will be lost and substituted in randomly, since by definition we do not know any of the grey systems or their interactions. **Table 1** describes the workflows chosen for the tool showcase.

Table 1. The MyExperiments wokflows used to create reasonable system of system connections for showcasing the tool

	Name	Total Systems	$\|N_F\|$	Known Coord / App / Manual / None	$\|N_G\|$	Grey Coord-point
1	Trivial US Healthcare Example	6	3	1 / 1 / 1 / 1	3	0
2	Unnested_qtl_pathway_3 by Antoon Goderis	12	4	1 / 1 / 1 / 1	8	1
3	BiomartAndEMBOSSAnalysis by Alan Williams	13	5	1 / 2 / 1 / 1	8	1
4	EBI_ClustalW2 by Hamish McWilliam	21	4	2 / 1 /1 / 1	17	1
5	PathwaysandGeneannotationsforQTLregion by Paul Fisher	61	8	2 / 2 / 2 / 2	53	8

Using this sampling of complex system of systems, we can explore the functionality of the tool. First we take #3 and #5 from Table 1 and show how user knowledge and graph connectivity can impact how easy or hard it is to get "good" provenance.

Fig. 3 shows snapshots of runs for #3 and #5 in which we vary the ratios of capture agent types. We do not show all combinations of coordination-point, application-based, manual and no capture, just a small selection. For graph #3, almost half of the systems are known, and those known systems form a direct path from source to sink (see Figure 2). Given the assignment of coordination points, the set of known provenance systems is guaranteed to meet the criteria of two of our "good" provenance metrics (at least 1 path through the graph, and 80% provenance captured). Figure 3a shows this in detail. No matter the provenance capture points deployed in the unknown systems, "good" provenance capture given what is known about the system is very likely. In other words, the tool provides a checkpoint for whether the user must invest more time and effort hunting down and provenance enabling the last few grey systems that may exist. In this case, the answer is a resounding "no" and the user may feel confident that the provenance from her system of systems is "good". On the other hand, the execution over graph #5 tells a different story. In the case of #5, only a very small subset of systems is known (e.g. because a user has just started exploring what to provenance-enable). **Fig. 3**b shows a spread of possible distributions of coordination-points, application-based, manual and no capture through the grey systems. While the ratios chosen represent the same spread as in **Fig. 3**a, the resulting provenance is not

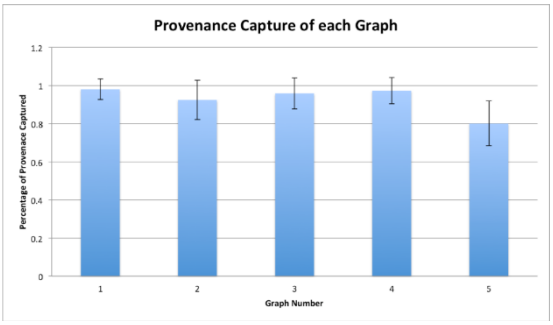

Fig. 4. Each graph from **Table 1** has been run with grey system capture agents in the following ratio: 10% coordination point, 30% application-based, 30% manual, 30% no capture.

necessarily as good. The tool helps the user recognize that better consideration of grey systems is in order.

Although it is expected that this tool will be used by an individual trying to provenance enable a very specific complex system, and hence will be used as we described above, varying numbers and types of capture points, we wish to highlight that the tool can function over any setup of size and number of provenance graphs. **Fig. 4** shows the amount of provenance captured when running each complex system from **Table 1**, using the following arrangement. The known systems and their capture types is fixed. Of the grey systems, 10% are coordination points, 30% are application-based, 30% are manual and 30% are no capture agent defined. As is expected, our very simple graph #1 that has only a few grey systems does very well. It is supported by the larger number of systems that are known and provenance enabled. At the other end of the spectrum is graph #5 that has a greater set of grey systems than known systems. In other words, the tool can help a user estimate how much provenance will be captured in a complex system, and allow the user to determine if that is good enough.

7 Future Work and Conclusion

Until now, we have made the assumption that all provenance capture agents cost the same amount to implement and maintain, and we are merely attempting to maximize the provenance captured while reducing the overall number of required provenance capture agents. Unfortunately, this assumption is incorrect. In our experience, creating a provenance capture agent for a commercial off the shelf (COT) tool, such as Palantir[1] or SpectorSoft [10], is harder and more costly than open source (OS), such as MuleSoft ESB [2] because the code is proprietary and not always easily accessible. Worse, there is no community of developers willing and able to provide insight into how the code is organized. Given that the cost of capture agents can vary widely, a natural extension to this model would focus on minimizing the cost of capture agent creation instead of

[1] www.palantir.com

minimizing the number of capture agents. In other words, a utility function needs to be created that minimizes cost and maximizes "good" provenance capture.

In this work, we describe complex systems-of-systems that users wish to capture provenance within. Because understanding these systems is difficult, and only a subset of systems is typically identified by users, we introduce the notion of a grey system. We provide a tool that models the set of known and grey systems, altering the interactions among all component systems and types of capture agents involved. Using this tool, an estimation of quantity and type of capture agents that will need to be deployed can be found.

8 References

[1] "North American Profile of ISO19115:2003 - Geographic Information - Metadata." NAP Metadata Working Group 2005.

[2] M. D. Allen, A. Chapman, B. Blaustein, and L. Seligman, "Provenance Capture in the Wild," in *IPAW*, 2010.

[3] M. D. Allen, A. Chapman, L. Seligman, and B. Blaustein, "Provenance for Collaboration: Detecting Suspicious Behaviors and Assessing Trust in Information," *CollabCom*, 2011.

[4] I. Altintas, O. Barney, and E. Jaeger-Frank, "Provenance Collection Support in the Kepler Scientific Workflow System," *IPAW*, pp. 118-132, 2006.

[5] H. U. Asuncion, "Automated data provenance capture in spreadsheets, with case studies," *Future Generation Computer Systems*, vol. 29, pp. 2169-2181, 2013.

[6] S. C. Bankes, "Tools and techniques for developing policies for complex and uncertain systems," *Proceedings of the National Academy of Sciences*, vol. 99, pp. 7263–7266, 2002.

[7] K. Belhajjame, J. Zhao, D. Garijo, A. Garrido, S. Soiland-Reyes, P. Alper, and O. Corcho, "A Workflow PROV-Corpus based on Taverna and Wings," in *ProvBench*, J. M. G.-P. Khalid Belhajjame, Satya Sahoo, Ed., 2013.

[8] C. Caron, B. Amann, C. Constantin, and P. Giroux, "WePIGE: The WebLab Provenance Information Generator and Explorer," *EDBT*, 2014.

[9] D. L. Chenyun Dai, Murat Kantarcioglu, Elisa Bertino, Ebru Celikel, Bhavani Thuraisingham, "Query Processing Techniques for Compliance with Data Confidence Policies," in *SDM*, 2009, pp. 49-67.

[10] G. B. Coe, R. C. Doty, M. D. Allen, and A. Chapman, "Provenance Capture Disparities Highlighted through Datasets," *Theory and Practice of Provenance*, 2014.

[11] H. Conover, R. Ramachandran, B. Beaumont, A. Kulkarni, M. McEniry, K. Regner, and S. Graves, "Introducing Provenance Capture into a Legacy Data System," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, 2013.

[12] D. Gammack and A. Chapman, "Provenance Tipping Point," *Theory and Practice of Provenance*, 2015.

[13] N. Gilbert and P. Terna, "How to build and use agent-based models in social science," *Mind & Society*, vol. 1, pp. 57-72, 2000.

[14] D. Gode and S. Sunder, "Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality," *Journal of Political Economy*, pp. 119-137, 1993.

[15] A. Goderis, D. De Roure, C. Goble, J. Bhagat, D. Cruickshank, P. Fisher, D. Michaelides, and F. Tanoh, "Discovering Scientific Workflows: The myExperiment Benchmarks," *IEEE Transactions on automation science and engineering*, 2008.

[16] P. Groth, Y. Gil, and S. Magliacane, "Automatic Metadata Annotation through Reconstructing Provenance," *Third International Workshop on the role of Semantic Web in Provenance Management*, 2012.

[17] M. Jackson, "The stability and efficiency of economic and social networks," *Advances in Economic Design*, pp. 319-361, 2003.

[18] M. Jackson and A. Watts, "The evolution of social and economic networks," *Journal of Economic Theory*, vol. 106, pp. 265-295, 2002.

[19] B. Lerner and E. Boose, "RDataTracker: Collecting Provenance in an Interactive Scripting Environment," *Theory and Practice of Provenance* 2014.

[20] T. McPhillips, T. Song, T. Kolisnik, S. Aulenbach, K. Belhajjame, K. Bocinsky, Y. Cao, F. Chirigati, S. Dey, J. Freire, D. Huntzinger, C. Jones, D. Koop, P. Missier, M. Schildhauer, C. Schwalm, Y. Wei, J. Cheney, M. Bieda, and B. Ludaescher, "YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts," *International Journal of Digital Curation*, 2015.

[21] P. Missier and Z. Chen, "Extracting PROV provenance traces from Wikipedia history pages," *EDBT*, 2013.

[22] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. I. Seltzer, "Provenance-Aware Storage Systems," *USENIX*, pp. 43-56, 2006.

[23] T. D. Nies, S. Magliacane, R. Verborgh, S. Coppens, P. Groth, E. Mannens, and R. V. d. Walle, "Git2PROV: Exposing Version Control System Content as W3C PROV," *Proceedings of the 12th International Semantic Web Conference*, 2013.

[24] H. Park, R. Ikeda, and J. Widom, "RAMP: A System for Capturing and Tracing Provenance in
MapReduce Workflows," *VLDB*, vol. 4, 2011.

[25] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. Silva, "Querying and Re-Using Workflows with VisTrails," *SIGMOD*, 2008.

[26] M. Stamatogiannakis, P. Groth, and H. Bos, "Looking Inside the Black-Box: Capturing Data Provenance Using Dynamic Instrumentation," in *Provenance and Annotation of Data and Processes*, vol. 8628, 2015, pp. 155-167.

[27] L. Tesfatsion, "Agent-based computational economics: modeling economies as complex adaptive systems," *Information Sciences*, vol. 149, pp. 262-268, 2003.

[28] U. Wilensky, "Netlogo," *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.*, vol. http://ccl.northwestern.edu/netlogo, 1999.

[29] K. Wolstencroft, R. Haines, and e. al, "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud," *Nucleic Acids Research*, vol. 41, pp. w557-w561, 2013.