Corroboration via Provenance Patterns

Lina Barakat
King's College London, UK
lina.barakat@kcl.ac.uk

Phillip Taylor Nathan Griffiths
University of Warwick, UK
{Phillip.Taylor, Nathan.Griffiths}@warwick.ac.uk

Simon Miles
King's College London, UK
simon.miles@kcl.ac.uk

Abstract

In today's distributed and heterogeneous systems, provenance data is becoming increasingly important for understanding process flow, tracing how outputs came about, and enabling users to make more informed decisions based on such outputs. However, within such systems, the sources (computational or human) that generate provenance may belong to different stakeholders operating under different policies. Thus, being autonomous and self-interested, these stakeholders may claim untrue data to protect their interests (e.g. to justify bad performance). In response, this paper proposes a corroboration methodology for verifying a claim made by a source, via confirming it against the claims of other sources. In particular, given a claim in PROV, this claim is generalised to varying levels of abstraction, deriving two types of provenance templates, namely *confirmation patterns*, capturing the information to be confirmed, and *witness patterns*, capturing the relevant witnesses. These patterns are utilised to find relevant evidence, among the reports of others, that supports the claim, and to respectively estimate the reliability degree of the claim. The proposed corroboration methodology is illustrated via a case study in the service provision domain.

Keywords Corroboration, Provenance Abstraction, Confirmation Pattern, Witness Pattern, Provenance Template

1. Introduction

The advances in network and communication technologies have enabled the emergence of complex, distributed computing systems, where the interacting parties are independent, heterogenous, and reside at different sites. Such systems bring many advantages to various parties including organisations, businesses, societies and individuals. For example, individual users connected to open networks have access today to a vast amount of information, goods and services. Likewise, enterprises can utilise such distributed communication capabilities not only to advertise and sell their products and services to end-users in an efficient and cost-effective manner, but also to automate their interactions with their trading partners (e.g. suppliers).

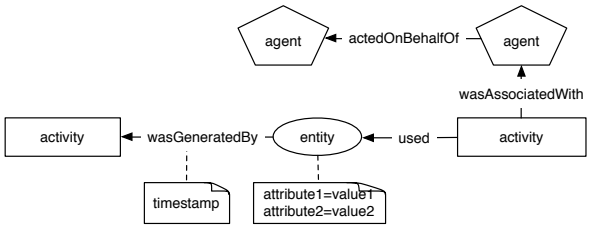

Figure 1. PROV graph illustrating the key elements

Within such systems, provenance data is important for understanding the processes under which interactions took place, tracing the context of achieved result data, and providing individuals with useful information to support their decision making in selecting future interaction partners. The PROV model (Moreau and Groth 2013) provides a suitable solution for generating (and interpreting) provenance information by system members. A PROV document describes in a queryable form the causes and effects within a particular past process of a system as a directed graph with annotations. A visualisation of such a graph, showing PROV's key elements, is shown in Figure 1. In summary, an *activity* is something that has taken place, making *use* of or *generating entities*, which could be data, physical or other things. *Agents* are parties that were responsible for (*associated with*) activities taking place, and one agent may have been *acting on behalf of* another in this responsibility. Activities, entities and agents (graph nodes) may be annotated with key-value *attributes* describing features that the elements had. *Timestamps* can also be added to show when entities were used or generated by activities.

However, system members (which we henceforth refer to as provenance *sources*) are potentially autonomous and self-

TaPP 2017, June 22-23, 2017, Seattle, Washington.

interested entities, acting to maximise their own utilities. Hence, when asked to supply provenance data, they may claim untrue events in order to protect their interests and increase their own profit. For example, in a service-oriented marketplace, a service provider may try to justify a poor performance by falsely claiming to be affected by some freak circumstances that are out of its control, in order to avoid reputation loss.

This paper contributes towards solving this problem by introducing a *corroboration methodology* that seeks evidence in the provenance data of other sources to assess the truthfulness of a provenance claim made by a source. The methodology involves deriving corroboration patterns at multiple abstraction levels to direct the search for evidence, estimating the degree of uncertainty underlying the evidence found, and incorporating the evidence found and its associated uncertainty into an overall reliability score for the claim. This *reliability score* reflects the degree of confidence that the claim is true. The rest of the paper is organised as follows. The definition of a claim in presented in Section 2. Sections 3 and 4 introduce the proposed corroboration methodology, and illustrate this methodology via a service provision case study, respectively. A discussion with related work is provided in Section 5, and finally Section 6 concludes the paper.

2. Claim Definition

A provenance report supplied by a source can be viewed as a collection of *claims*, each corresponds to a particular event conducted/experienced by a source. The *reliability* of a claim from a source can be assessed via comparing the claim against the reports of other sources. Events conducted/experienced by a source may vary in their degrees of observability by others. Some events might be local, i.e. private to the claiming source or observed by a small community around it (e.g. the event of sub-contracting a task might only be observed by the delegator and the delegatee). Other events might be more global, i.e. observed by a larger community of sources (e.g. a storm affecting shipment of goods should potentially be observed by a large community). Therefore, when assessing the truthfulness of a claim, it is important to identify which sources qualify as *relevant witnesses* for the claim, and to judge the reliability of the claim accordingly. For example, a claim confirmed by 3 out of 4 relevant independent witnesses should be regarded as more reliable than that supported by 3 out of 15 relevant independent witnesses.

For this purpose, two parts are distinguished in a claim, a *main* part and a supplementary *context* part. The main part captures the core idea expressed by the claim. The context part gives extra context information related to the claim, indicating the witnesses that are potentially relevant for assessing the claim. For example, a service provider X may claim that its service execution Y, which occurred around time T at location L, was influenced by event Z. In this case, the occurrence of the event is the core idea (main part), while the time and location details are additional context information (context part). This context part indicates that the providers relevant for judging this claim's truthfulness are those that operated (provided services) around time T and around location L. Generally, what constitutes the main part and the supportive context part of a claim is application dependent, and can be pre-defined via templates at design time. We refer to the main part and the context part of a claim as M and W, respectively. That is, $claim = (M, W)$.

Each part of a claim, main or context, is made up of one (or more) clauses. Claims within a provenance report may overlap in their clauses, e.g. the same clause may be shared among several claims. Each clause is assumed to be of the form $rel(c_1, c_2)$, where rel is a relationship connecting between concepts c_1 and c_2, where c_1 is the subject and c_2 is the object (same order as in PROV-N for properties). Each concept c is associated with a type (semantic class), $type(c)$, and possibly an individual, $referent(c)$, that is an instance of this class. That is, each concept c is of the form $type(c) : referent(c)$. If concept c is not associated with a specific individual, $referent(c) = *$ indicating a generic individual.

If we were using PROV to document provenance, a clause is mapped to PROV data model, as follows. Each concept c is either a PROV node (entity, activity, or agent) or an attribute value. Each relationship rel is either a PROV property connecting between PROV nodes (e.g. used, wasGeneratedBy, etc.) or an attribute key. We do not restrict the representation to basic PROV nodes and properties, but assume these are potentially extended to model provenance in different application domains. In particular, $type(c) \in T$, where $T = T_e \cup T_a \cup T_g \cup T_p$ is the set of concept types in the application domain. These concept types are structured in a lattice according to the is-a (subclass) relation, with all being the universal type, and $null$ being the absurd (minimal type). Sub-lattices T_e, T_a, and T_g contain the concept types that are subclasses of *prov:Entity*, *prov:Activity*, and *prov:Agent*, respectively, while T_p is the set of all primitive types for attribute values. Similarly, $rel \in R$, where R is the set of relationships in the application domain extending PROV basic properties and attributes (these relationships may or may not be structured in a lattice).

3. Corroboration Methodology

As indicated earlier, in our approach, the reliability of a claim from a source is assessed via comparing the claim against the reports of other sources. In particular, the reliability score of a claim is the ratio of the number of relevant witnesses confirming the claim to the total number of relevant witnesses, among existing sources. Considering that the set of available sources is S, the set of relevant witnesses for the claim is $S_W \subseteq S$, and the set of confirming relevant

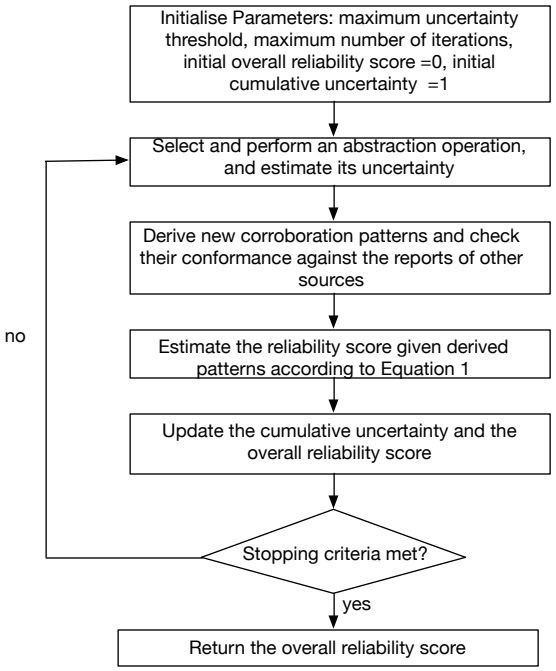

Figure 2. Flowchart of the Corroboration Methodology

witnesses is $S_{M+W} \subseteq S_W$, the reliability score rel, of the claim $claim$, is given by,

$$rel(claim) = \frac{|S_{M+W}|}{|S_W|} \in [0, 1] \quad (1)$$

Given knowledge of the claim and of set S, our aim is thus to derive sets S_W and S_{M+W}. We denote our estimations of these sets as $\overline{S}_W$ and $\overline{S}_{M+W}$, respectively. Ideally, the estimation process should ensure both a high *recall* and a high *precision* with respect to each estimated set. High recall indicates that the estimation is comprehensive and does not miss out important evidence. To increase recall, we should aim to increase the ratios $\frac{|\overline{S}_W \cap S_W|}{|S_W|}$ (witness recall) and $\frac{|\overline{S}_{M+W} \cap S_{M+W}|}{|S_{M+W}|}$ (confirmation recall). High precision, on the other hand, indicates that the estimation does not produce irrelevant evidence. To increase precision, we should aim to increase the ratios $\frac{|\overline{S}_W \cap S_W|}{|\overline{S}_W|}$ (witness precision) and $\frac{|\overline{S}_{M+W} \cap S_{M+W}|}{|\overline{S}_{M+W}|}$ (confirmation precision). Generally, there is an inverse relationship between recall and precision. Improving recall can be achieved by loosening the search criteria, which would increase the probability of incorporating irrelevant evidence into the estimated sets, thus negatively affecting precision, and vice versa.

The estimation of sets S_W and S_{M+W} is conducted by identifying appropriate search criteria over the provenance reports from other sources. We refer to such search criteria as *corroboration patterns*. Initially, these corroboration patterns can be derived by applying *abstraction* operation on the claim. This abstraction corresponds to generalising references to the identity of the claim's source so that these can be mapped to the identities of other sources. It transforms the claim into a more suitable query form which can be evaluated to true in the reports of others. Assume that $claim_1 = (M_1, W_1)$ is the result of applying such an abstraction operator op_1 on the original $claim = (M, W)$, denoted $claim \xrightarrow{op_1} claim_1$.

To improve recall, the claim can undergo a sequence of *other* abstraction operations. That is,

$$claim \xrightarrow{op_1} claim_1 \xrightarrow{op_2} claim_2 \quad ... \quad \xrightarrow{op_k} claim_k$$

All abstraction operations, $claim_{i-1} \xrightarrow{op_i} claim_i$, ensure that $claim_{i-1} \Rightarrow claim_i$ (with $claim_0 = claim$), i.e. that the satisfaction of $claim_{i-1}$ implies the satisfaction of $claim_i$. However, these operations may incur information loss, and thus do not necessarily ensure that $claim_i \Rightarrow claim_{i-1}$. That is, a report satisfying a more generic claim form $claim_i$ does not necessarily satisfy a more specific version $claim_{i-1}$. Hence, corroboration patterns derived from abstracted forms of the claim could retrieve irrelevant evidence, resulting in decreased precision. We capture the information loss (and respective precision loss) incurred by abstraction operation $claim_{i-1} \xrightarrow{op_i} claim_i$, via modelling the *uncertainty* underlying implication $claim_i \Rightarrow claim_{i-1}$. We denote this uncertainty as

$$\mu(claim_i \Rightarrow claim_{i-1}), \quad (2)$$

which can be seen as a measure of the information that need to be added to $claim_i$ in order to achieve satisfaction of $claim_{i-1}$. This uncertainty is accounted for when estimating the overall reliability score of the original claim.

The proposed corroboration process is summarised in Figure 2. In each iteration of the process, an abstraction operator is selected and applied to the current version of the claim, estimating its associated uncertainty. This is followed by deriving new corroboration patterns from the resulting abstracted form of the claim, checking their conformance against the reports of other sources, and calculating the reliability score from the respectively estimated sets $\overline{S}_W$ and $\overline{S}_{M+W}$. The overall reliability score of the original claim is then updated with respect to the iteration outputs. The process terminates either when a maximum number of iterations is reached, or when the current cumulative uncertainty exceeds a predefined maximum threshold. These steps are further detailed in the following sections.

3.1 Abstraction Operators

We consider three types of abstraction operators, namely individual generalisation, concept type generalisation, and clause detachment. These operators are detailed next.

The *individual generalisation* operator, $ig(claim, c)$, is applied to a concept c of claim $claim$. This operator generalises concept c by replacing $referent(c)$ with a more

generic individual. For example, concept <*Location: Area X of City Y*> can be generalised to concept <*Location: City Y*>. The most generic individual is individual $*$ that matches any value. Abstracting source identity is achieved by this operator.

The *concept type generalisation* operator, $tg(claim, c)$, is applied to a concept c of claim $claim$. This operator generalises concept c by replacing $type(c)$ with one of its superclasses in the concept type lattice T. For example, concept <*StormFreakEvent:* $*$> can be generalised to concept <*WeatherFreakEvent:* $*$>.

Finally, the *clause detachment* operator, $cd(claim, cls)$, generalises claim $claim$ by eliminating clause cls from this claim.

The uncertainty μ (of Equation 2) underlying an abstraction operator can be provided by the user in the form of uncertainty policies. For example, the user may annotate each *is-a* relation in the domain lattice with a penalty indicating the loss in information incurred when moving up the lattice according to this relationship. Similarly, the user may annotate each clause in the claim with a penalty that is correlated with the importance of the clause for the user. In the absence of such domain-dependent user-defined policies, automated means can be utilised instead to estimate uncertainties. For example, given that a class t has three sub-classes t_1, t_2, t_3 in the domain lattice, then generalising t_1 to t can be associated with an uncertainty of $\frac{1}{3}$. Similarly, eliminating a clause from a claim of k clauses can be associated with an uncertainty of $\frac{k-1}{k}$, which assigns equal importance to all clauses.

Selecting which abstraction operation to apply at each iteration of the process can be again either guided by the user, or reliant on some uncertainty minimisation algorithm (a simple form of which is to select the abstraction operation with the minimum underlying uncertainty at each iteration).

3.2 Corroboration Patterns and Conformance Check

As indicated earlier, the estimation of sets S_W and S_{M+W} is conducted by identifying appropriate search criteria over the provenance reports from other sources, which are referred to as corroboration patterns. In particular, we distinguish between two types of corroboration patterns, *witness pattern* and *confirmation pattern*. A *witness pattern*, $ptrn_W$, is an abstracted provenance graph characterising the witnesses relevant for assessing the truthfulness of a claim made by a source. A *confirmation pattern*, $ptrn_{M+W}$, is an abstracted provenance graph capturing the information to be confirmed by others in order to assess the truthfulness of a claim made by a source. Given the current abstracted claim form, $claim_i = (M_i, W_i)$, following the abstraction operator op_i selected at the current iteration, the witness and confirmation patterns for the current iteration are thus W_i and $M_i + W_i$ (entire $claim_i$), respectively.

Based on this, sets S_W and S_{M+W} for the current iteration can be estimated as follows.

$$\overline{S}_W = \{s \in S \mid rprt(s) \Rightarrow ptrn_W\} \quad (3)$$

$$\overline{S}_{M+W} = \{s \in S \mid rprt(s) \Rightarrow ptrn_{M+W}\} \quad (4)$$

where $rprt(s)$ is the report supplied by source s, and $rprt(s) \Rightarrow ptrn$ indicates that report $rprt(s)$ implies (satisfies) pattern $ptrn$. The definition of this implication is provided below.

Definition. A provenance graph $rprt$, satisfies (implies) a pattern graph $ptrn$, denoted $rprt \Rightarrow ptrn$, if there exists a projection (mapping) π from $ptrn$ to $rprt$, denoted $\pi(ptrn)$, such that $\pi(ptrn)$ is a sub-graph of $rprt$ satisfying all the following:

1. for each graph node n in $ptrn$, $\pi(n)$ is a graph node in $rprt$ with the same or a more restricted class, and the same or a more restricted individual (note that the generic individual $*$ is considered similar to any individual);
2. for each connecting property pr in $ptrn$, $\pi(pr)$ is the same property in $rprt$; and
3. if nodes n_1 and n_2 in $ptrn$ are connected via property pr, then $\pi(n_1)$ and $\pi(n_2)$ in $rprt$ are connected via property $\pi(pr)$.

3.3 Overall Reliability Score

The *cumulative uncertainty* $\tilde{\mu}$, after applying a sequence of abstraction operators $op_1 op_2 ... op_k$ on the original claim $claim$, is estimated as

$$\tilde{\mu}(claim_k \Rightarrow claim) = \prod_{i=1}^{k} \mu(claim_i \Rightarrow claim_{i-1}) \quad (5)$$

where $claim_0 = claim$.

Based on this, the *overall reliability score* of the original claim $claim$, accounting for all the performed abstraction operations $op_1 op_2 ... op_k$, is estimated as

$$overallRel(claim) = \sum_{i=1}^{k} wt(claim_i) \times rel(claim|claim_i) \quad (6)$$

where: $rel(claim|claim_i)$ is the reliability score of $claim$ (as indicated by Equation 1) given that the corroboration patterns for estimating sets S_W and S_{M+W} are derived from abstract form $claim_i$; and $wt(claim_i)$ corresponds to the relative weight (importance) of $claim_i$ for estimating the overall reliability of the original claim, and is given as

$$wt(claim_i) = \frac{\tilde{\mu}(claim_i \Rightarrow claim)}{\sum_{j=1}^{k} \tilde{\mu}(claim_j \Rightarrow claim)} \quad (7)$$

That is, evidence derived from abstractions that incurred more information loss would have a lower impact on the overall score that that derived from abstractions with less information loss.

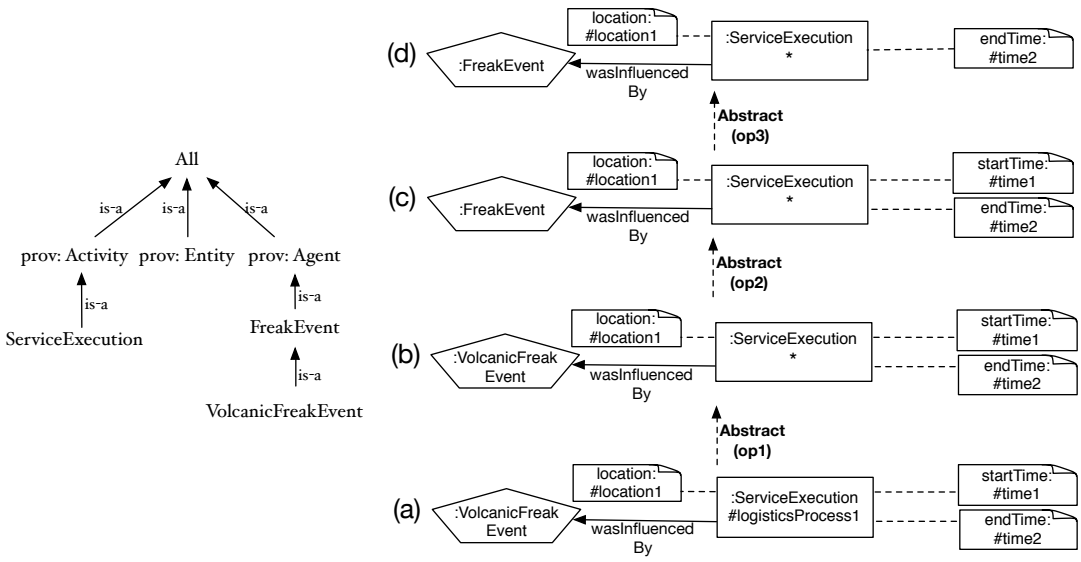

Figure 3. Multi-level Abstraction of a Claim

4. Case Study

In this section, we illustrate our corroboration model via an example from the *service provision* domain. In a service-oriented system, individuals rely on external providers to execute services for them. Knowledge of the *circumstances* under which past service provisions took place gives individuals useful information to support their decision making in selecting a future provider. Examples of circumstances that may affect the provision of a service by a provider include occurrence of freak events and sub delegation (Miles and Griffiths 2015). The PROV standard provides a suitable solution for recording and interpreting information on the circumstances of various types underlying a service provision. However, service providers may claim circumstances, which did not occur in reality, in order to justify their occasional poor performance. A potential solution to this problem is to compare a provider's claim against the claims made by other providers in order to assess its truthfulness.

An example *freak event claim* in PROV, supplied by a provider, is depicted in Figure 3(a). It reports a volcanic freak event experienced by logisticsProcess1, within time period $[time_1, time_2]$, at location $location_1$. Note that class *ServiceExecution* is-a *prov:Activity*, and *VolcanicFreakEvent* is-a *prov:Agent*. The clauses of the *main* and *context* parts of this claim are given in Table 1. The context part indicates that the providers relevant for judging this claim's truthfulness are those operated (provided services) around time frame $[time_1, time_2]$, at locations close to $location_1$.

Initial corroboration patterns for this claim, i.e. the witness pattern and confirmation pattern, are derived from the claim after applying an *individual generalisation* operator to abstract specific provider information (see Figure 3(b)). Further abstractions could also be applied to improve recall. For example, we can increase the range of search for evidence by applying *concept type generalisation* operator to generalise *VolcanicFreakEvent* to *FreakEvent* (see Figure 3(c)). The result can be further abstracted by applying *clause detachment* operator to eliminate the starting time restriction (see Figure 3(d)).

Once the corroboration patterns at each abstraction level are derived, sources whose reports imply these patterns are retrieved. A possible way to implement this implication relationship is via translating the corroboration patterns into a SPARQL query over the provenance reports of other sources.

5. Discussion and Related Work

In the proposed corroboration methodology, we have presented three example abstraction operations to a claim. However, the proposed methodology is not limited to these operations and can be extended to incorporate other abstraction operators. In fact, a number of approaches in the literature are concerned with abstracting away specific details from a provenance graph, via generating views (Danger et al. 2015), summaries (Moreau 2015), or abstractions (Missier et al. 2015) over the provenance graph. Some of these techniques can be incorporated into our methodology as additional abstraction operators. For example, we can introduce another abstraction operator, *node grouping*, which replaces a set of nodes in the graph with a new abstract node, as proposed by Missier et al. (2015). This would also require adjusting our definition of the implication relationship to account for sub-graph substitutions (Buneman et al. 2016). That is, it would require allowing a node in a pattern graph to be projected into a sub-graph in a specific report graph.

Table 1. Clauses of the Example Claim

Part	Clauses
Main	wasInfluencedBy(ServiceExecution: logisticsProcess1, VolcanicFreakEvent)
Context	startTime(ServiceExecution: logisticsProcess1, xsd:dateTime: time1) endTime(ServiceExecution: logisticsProcess1, xsd:dateTime: time2) location(ServiceExecution: logisticsProcess1, xsd:string: location1)

The claim template at each abstraction level may also have some similarity with the template language of PROV-TEMPLATE (Moreau et al. 2017). Specifically, the highest level of abstraction for a concept in our approach is $\langle$*Entity:* $*\rangle$, $\langle$*Activity:* $*\rangle$, or $\langle$*Agent:* $*\rangle$, potentially obtained via an *individual generalisation* and a sequence of *concept type generalisation* operators. Provenance nodes at such abstraction level are placeholders for any value, and in that sense are similar to the notion of *variables* in PROV-TEMPLATE (Moreau et al. 2017).

Our corroboration checking can also be considered related to the area of checking compliance to policies/rules using provenance. In particular, a provenance-based Compliance Framework is proposed by Aldeco-Pérez and Moreau (2010), in which past information processing is compared against defined policies to which the processing should comply. The algorithms proposed in this framework are specific to information processing requirements, and do not handle pattern generalisation nor reliability score estimation based on multiple witnesses required in the context of our problem. Corroboration is a different problem to compliance as, in the former, you do not have a canonical source of what should be true.

When estimating the reliability score of a claim (Equation 1), we have assumed *independence* among witnesses. However, members comprising a complex distributed system may in fact experience different types of dependencies among each other. For example, in a service-oriented marketplace, two different providers might outsource their subtasks to the same sub-providers at similar times, making their experiences similar and thus their testimonies redundant. We can account for such dependencies (redundancies) among witnesses in a similar manner to the channel weighting for a multi-version fault tolerant system (Townend et al. 2005). In particular, the contribution of each witness can be *weighted* based on the degree of its independence from other witnesses. Equation 1 can be rewritten to account for such weighting as,

$$rel(claim) = \frac{\sum_{s \in S_{M+W}} weight(s)}{\sum_{s \in S_W} weight(s)} \quad (8)$$

Finally, the concepts of abstraction and confirmation check presented in this paper share similarity with the concepts of query expansion (Carpineto and Romano 2012) and relevance models (Crestani and Lalmas 2001) from Information Retrieval (IR). Query expansion techniques reprocess a user's original query in order to improve search effectiveness, while the goal of a relevance model is to find the set of relevant documents that satisfy a query (information need) expressed by a user. In particular, a document is regarded as relevant to a query if the query can be inferred by the document. For example, some semantic IR models (Kheirbek and Chiaramella 1995) represent queries and documents as conceptual graphs, with relevance being assessed via the conceptual graph's projection operation (Mugnier 1995; Sowa 2013).

6. Conclusion

In this paper, we have presented a corroboration methodology capable of assessing the reliability degree of a provenance claim. The methodology derives suitable corroboration patterns for confirming the claim against the reports of relevant witnesses. To improve witness recall, the search space for witnesses is iteratively increased via applying abstraction operations on the search patterns. The respective uncertainties associated with such abstractions are accounted for and incorporated into the overall reliability score of the claim.

Acknowledgments

This work was part funded by the UK Engineering and Physical Sciences Research Council as part of the Justified Assessments of Service Provider Reputation project, ref. EP/M012654/1 and EP/M012662/1.

References

R. Aldeco-Pérez and L. Moreau. A provenance-based compliance framework. In *Third Future Internet Symposium (Future Internet - FIS 2010)*, pages 128–137, 2010.

P. Buneman, A. Gascón, and D. Murray-Rust. Composition and substitution in provenance and workflows. In *Proceedings of the 8th USENIX Conference on Theory and Practice of Provenance*, TaPP'16, pages 40–43, 2016.

C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1–50, 2012.

F. Crestani and M. Lalmas. Logic and uncertainty in information retrieval. In *Lectures on Information Retrieval*, pages 179–206. 2001.

R. Danger, V. Curcin, P. Missier, and J. Bryans. Access control and view generation for provenance graphs. *Future Generation Computer Systems*, 49(C):8–27, 2015.

A. Kheirbek and Y. Chiaramella. Integrating hypermedia and information retrieval with conceptual graphs formalism. In *Hypertext - Information Retrieval - Multimedia (HIM)*, pages 47–60, 1995.

S. Miles and N. Griffiths. Incorporating mitigating circumstances into reputation assessment. In *Proceedings of the Second International Workshop on Multiagent Foundations of Social Computing (MFSC 2015)*, pages 77–93, 2015.

P. Missier, J. Bryans, C. Gamble, V. Curcin, and R. Danger. *ProvAbs: Model, Policy, and Tooling for Abstracting PROV Graphs*, pages 3–15. 2015.

L. Moreau. Aggregation by provenance types: A technique for summarising provenance graphs. In *Proceedings Graphs as Models (ETAPS 2015 workshop)*, pages 129–144, 2015.

L. Moreau and P. T. Groth. *Provenance: An Introduction to PROV*. Synthesis Lectures on the Semantic Web: Theory and Technology, 2013.

L. Moreau, B. V. Batlajery, D. Huynh, D. Michaelides, and H. Packer. A templating system to generate provenance. *IEEE Transactions on Software Engineering*, 2017.

M. L. Mugnier. On generalization/specialization for conceptual graphs. *Journal of Experimental & Theoretical Artificial Intelligence*, 7(3):325–344, 1995.

J. F. Sowa. From existential graphs to conceptual graphs. *International Journal of Conceptual Structures and Smart Applications*, 1(1):39–72, 2013.

P. Townend, P. T. Groth, and J. Xu. A provenance-aware weighted fault tolerance scheme for service-based applications. In *Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, pages 258–266, 2005.