SciProvMiner: Captura de Proveniência Utilizando Recursos Web Semânticos para Ampliação do Conhecimento Gerado e Otimização do Processo de Coleta

Tatiane O. M. Alves[1], Regina Braga[1], Fernanda Campos[1], José Maria N. David[1]

[1] Mestrado em Ciência da Computação, Departamento de Ciência da Computação, Universidade Federal de Juiz de Fora – Juiz de Fora, MG – Brasil
{tatiane.ornelas, regina.braga, fernanda.campos, jose.david}@ufjf.edu.br

Resumo. Prover informação histórica de experimentos científicos com o objetivo de tratar o problema de perda de conhecimento do cientista sobre o experimento tem sido o foco de diversas pesquisas. No entanto, o apoio computacional ao experimento científico em larga escala encontra-se ainda incipiente. Este trabalho tem o intuito de colaborar para as pesquisas nessa área, apresentando a arquitetura SciProvMiner, cujo principal objetivo é coletar proveniência prospectiva e retrospectiva de experimentos científicos fazendo uso de recursos Web semânticos para otimizar o processo de captura das informações de proveniência e aumentar o conhecimento do cientista sobre o experimento realizado. Para isso, SciprovMiner utiliza ontologias em conjunto com arquiteturas orientadas a serviços para ampliar o conhecimento do cientista sobre os dados de proveniência.

Palavras-chave: Web Semântica Serviços Web, Ontologia, Proveniência, e-Science, OPM.

1 Introdução

Computação em larga escala tem sido amplamente utilizada como metodologia para a realização de pesquisa científica. Existem vários casos de sucesso em muitos domínios, incluindo física, bioinformática, engenharia e ciências [Wong et al. 2005]. Neste contexto, apesar de o conhecimento científico continuar sendo gerado de forma tradicional, *in-vivo* e *in-vitro*, nas últimas décadas experimentos científicos passaram a utilizar procedimentos computacionais para simular seus próprios ambientes de execução, dando origem à modalidade de experimentos científicos *in virtuo* [Marinho 2011]. Além disso, até mesmo os objetos e os participantes de um experimento passa-

ram a ser simulados, surgindo a categoria de experimentos *in silico*. Essas computações em larga escala, que sustentam um processo científico, são geralmente referidas como e-Science.

Mattoso et al. (2008) identificaram diversos desafios para prover apoio computacional ao desenvolvimento de ciência em larga escala, embasados no segundo dos grandes desafios da SBC (Sociedade Brasileira de Computação). Dentre os desafios de e- Science apresentados neste trabalho encontra-se o de prover informação histórica dos experimentos científicos, em vistas a tratar o problema de perda de conhecimento do cientista sobre o experimento.

O trabalho apresentado neste artigo, SciProvMiner, tem como objetivo principal especificar uma arquitetura para apoiar a coleta e gerência da proveniência de dados e processos no contexto de experimentos científicos processados através de simulações computacionais, respaldado na hipótese de que a captura e a gerência de dados de proveniência irá fornecer ao cientista informações importantes a respeito do experimento realizado, com o potencial de auxiliá-lo a formar uma visão da qualidade, da validade e da atualidade acerca da informação produzida. A arquitetura proposta provê uma camada de interoperabilidade capaz de interagir com os SGWfC (Sistemas de Gerenciamento de Workflows Científicos) tendo como finalidade capturar as informações de proveniência prospectiva e retrospectiva geradas a partir de workflows científicos. Para isso, utiliza um coletor de proveniência baseado em tecnologia de serviços Web, o que torna o mecanismo de coleta independentemente do SGWfC no qual o workflow foi desenvolvido. A arquitetura também provê uma camada de consulta aos dados coletados utilizando recursos tais como ontologia e máquina de inferência para ampliar o conhecimento do cientista a respeito do experimento realizado, disponibilizando para ele informações além daquelas explicitamente informadas na captura. Além disso, a utilização desses recursos Web Semânticos possibilita que o procedimento de captura dos dados de proveniência seja otimizado, diminuindo o processamento necessário para que a tarefa seja realizada bem como o trabalho do cientista na instrumentalização do workflow que terá a proveniência capturada.

Alguns requisitos importantes do SciProvMiner são: i) Independência em relação aos modelos de dados e processos adotados pelos SGWfC existentes; ii) Aplicabilidade direcionada a experimentos científicos; iii) Emprego de tecnologias da Web Semântica para representação e consulta aos dados de proveniência; iv) Possibilidade de ajuste no mecanismo de captura para permitir o controle sobre o impacto dos processos de coleta e persistência dos dados de proveniência no desempenho da execução dos experimentos científicos; v) Captura da proveniência prospectiva e retrospectiva utilizando o modelo OPM; vi) Uso de recursos Web semânticos para inferência de informações significativas e implícitas nos dados de proveniência coletados.

Este artigo está estruturado da seguinte maneira: Na seção 2 é apresentada a fundamentação teórica. Na seção três são descritos os trabalhos relacionados. Na sessão 4, o SciProvMiner é detalhado. A seção 5 apresenta uma prova de conceito e a seção 6 conclui o trabalho.

2 Fundamentação Teórica

Proveniência de dados pode ser definida como informação que auxilia a determinar a derivação histórica do produto de dado, a partir de suas fontes de origem, sendo considerado um componente essencial para permitir reprodutibilidade do resultado, compartilhamento e reuso de conhecimento pela comunidade científica [Freire et al. 2008]. Com o objetivo de facilitar a interoperabilidade de proveniência entre diversos SGWfCs heterogêneos, foi desenvolvido o modelo de proveniência denominado *Open Provenance Model*(OPM), concebido como resultado do primeiro e segundo episódio da série *Provenance Challenge* realizados em 2006 e 2007 [Moreau et al. 2011]. Mais recentemente foi desenvolvido outro modelo de proveniência denominado PROV pelo grupo incubador de proveniência da W3C com o mesmo objetivo do OPM (http://www.w3.org/TR/prov-overview/). Ambos os modelos têm o foco apenas na proveniência retrospectiva, visando representar os acontecimentos passados e não os acontecimentos futuros. Porém, de acordo com Lim et al. (2011), ambos os tipos de proveniência, prospectiva, que captura a especificação abstrata do workflow como uma receita para derivação de dados futuros, e retrospectiva, que captura a execução do workflow e as informações de derivação dos dados, fornecem informações importantes para a análise de resultados científicos. Como resultado, muitas consultas relacionadas à especificação do workflow não podem ser respondidas baseadas somente no modelo OPM e nem somente no modelo PROV.

Como o presente trabalho objetiva capturar ambos os tipos de proveniência, foi realizada uma extensão do *Open Provenance Model* (OPM) para englobar também a proveniência prospectiva. O modelo OPM foi escolhido para ser utilizado no SciProvMiner, por ser, até o presente momento, o modelo de proveniência mais utilizado, e por possuir em sua documentação regras de completude e inferências bem definidas que são implementadas na ontologia que o SciProvMiner utiliza em sua camada de consulta Web Semântica aos dados de proveniência. No entanto, com algumas mudanças específicas, o SciProvMiner pode ser adaptado para o modelo PROV.

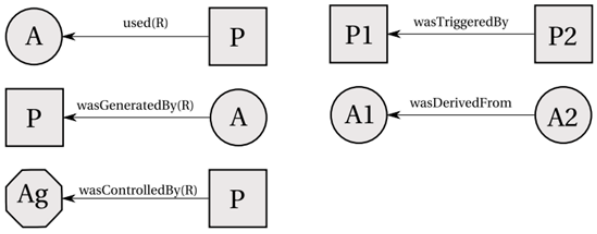

Fig. 1. Arestas no OPM: origens são efeitos e destinos são causas [adaptado de Moreau et al, 2011]

No modelo OPM, grafos de proveniência são composto por três tipos de nós: i) Artefatos: representam um dado de estado imutável, que pode ter um corpo físico, ou uma representação digital em um sistema de computador. ii) Processos: representam ações realizadas ou causadas por artefatos, e resultam em novos artefatos. iii) Agen-

tes: representam entidades contextuais agindo como um catalisador de um processo, permitindo, facilitando, controlando, ou afetando sua execução.

A Figura 1, adaptada de [Moreau et al. 2011], ilustra essas três entidades e suas possíveis formas de relacionamento, chamadas também de dependências. À esquerda da Figura 1, as duas primeiras arestas expressam que um processo usou um artefato e que um artefato foi gerado por um processo. Essas duas dependências representam relacionamentos de derivação de dados. A terceira aresta indica que um processo foi controlado por um agente. Diferente das outras duas primeiras arestas mencionadas, esta representa um relacionamento de controle. A quarta aresta é usada em situações nas quais não se sabe exatamente quais artefatos foram utilizados por um processo, porém se sabe que este processo utilizou algum artefato gerado por outro processo. Com isso, pode-se dizer que o processo foi inicializado por outro processo. De maneira análoga, a quinta aresta é utilizada em situações que não se sabe qual processo gerou um determinado artefato, porém se sabe que esse artefato foi derivado de outro artefato. Esses dois últimos relacionamentos são recursivos e podem ser implementados a partir de regras de inferência. Portanto, a partir deles, é possível determinar a sequência de execução dos processos ou o histórico de derivação que originou um dado.

Alguns modelos de proveniência usam a tecnologia da Web Semântica tanto para representar quanto para consultar informações de proveniência. Linguagens da Web Semântica, tais como RDF e OWL fornecem uma maneira natural de modelar grafos de proveniência e habilidade de representar o conhecimento complexo, tais como anotações e metadados. O benefício da abordagem através da Web Semântica consiste na possibilidade de integração de quaisquer fontes de dados na base de conhecimento [Golbeck e Hendler 2008].

Como este trabalho tem por característica utilizar recursos da Web Semântica para aumentar o conhecimento do cientista a respeito do experimento realizado, e o modelo OPM disponibiliza a *Open Provenance Model Ontology* (OPMO) que captura os conceitos do modelo OPM, foi realizada uma extensão desta ontologia, denominada OPMO-e, de forma a modelar o conhecimento acerca da proveniência prospectiva além da retrospectiva, já contemplada na OPMO. Além disso, foram implementadas regras ontológicas baseadas no conceito de cadeia de propriedades (*property chain*) disponível na OWL2 (http://www.w3.org/TR/owl2-overview/), com os seguintes objetivos: i) Viabilizar a implementação das regras de completude definidas no modelo OPM, que não eram passíveis de serem capturadas utilizando apenas conceitos de proveniência retrospectiva. Foram construídas propriedades a partir da combinação de propriedades relativas à captura da proveniência prospectiva em conjunto com propriedades relacionadas à captura da proveniência retrospectiva; ii) Implementar as inferências em múltiplos passos definidas na documentação do modelo OPM [Moreau et al. 2011]; iii) Implementar regras de otimização que dispensam a instrumentalização de certas dependências causais, por torná-las passíveis de serem inferidas; iv) Implementar inferências a respeito da proveniência prospectiva.

A criação dessas regras tem o objetivo de aumentar o conhecimento do cientista sobre o experimento realizado, inferindo informações que não foram explicitamente fornecidas pelo usuário e tornando possível a otimização do processo de captura de

proveniência e a consequente diminuição do trabalho do cientista para instrumentalizar o workflow.

3 Trabalhos Relacionados

Poucos trabalhos focam na captura de proveniência utilizando um modelo padrão e no uso de funcionalidades da Web Semântica. Em Marinho [2011], propõe-se uma arquitetura para gerência de proveniência de dados denominada ProvManager. Tanto a arquitetura do SciProvMiner quanto a do ProvManager têm como foco a captura da proveniência em workflows com recursos heterogêneos e distribuídos baseados na invocação de serviços Web. O SciProvMiner provê uma infraestrutura baseada na Web Semântica para representação e consulta aos metadados de proveniência, o que lhe confere um maior poder de expressividade para inferência de conhecimento novo. Outra característica do SciProvMiner não encontrada no ProvManager é a capacidade deste em explorar as regras de completude e inferências válidas no modelo OPM para fornecer ao cientista conhecimento implícito.

Em Lim et al. (2010) é proposta uma extensão do modelo OPM para modelar proveniência prospectiva, além da retrospectiva já suportada no OPM nativo. O SciProvMiner também utiliza uma extensão do OPM para suportar a modelagem de proveniência prospectiva. Porém, a arquitetura proposta em Lim et al. (2010) não provê infraestrutura baseada na Web Semântica. Em [Lim et al. 2010] também é proposta a implementação das regras de completude e inferência definidas no modelo OPM através de *Views* do banco de dados, no entanto apenas uma das regras de completude definidas no modelo OPM é implementada enquanto no SciProvMiner as três regras definidas no modelo são implementadas, fornecendo ao usuário maiores possibilidades de explorar o conhecimento acerca do experimento realizado. Em [Lim et al. 2011] é proposto o OPMProv, para integração de proveniência de workflows desenvolvidos por diferentes SGWfCs e consulta a esses dados. No entanto, a integração é viável para SGWfCs que possuem mecanismos próprios para capturar a proveniência de forma compatível com o modelo OPM. O SciProvMiner também armazena a proveniência capturada a partir de diversos SGWfCs, no entanto, o ScieProvMiner realiza a captura da proveniência, enquanto o OPMProv não captura a proveniência, apenas traduz a proveniência capturada pelo SGWfC no banco de dados relacional do OPMProv. Outra diferença do SciProvMiner para o OPMProv é que o OPMProv não captura a proveniência prospectiva.

Cuevas-Vicenttin et al. (2012) propõem um modelo que estende o OPM, denominado D-OPM. Em Cuevas-Vicenttin et al. (2012) os workflows devem ser especificados em uma linguagem proprietária criada por eles, denominada KPN Language, e a interoperabilidade com SGWfCs é alcançada através de wrappers que estão sendo desenvolvidos para SGWfCs tais como Kepler, Taverna e Vistrails, enquanto no SciProvMiner os workflows são desenvolvidos nos próprios SGWfCs e a captura da proveniência é realizada através de instanciações de serviços Web do SciProvMiner dentro do SGWfC no qual o workflow foi modelado. Bowers et al. (2012) apresentam uma abordagem que usa regras explícitas definidas pelo usuário para

inferir dependência de dados dos rastros da execução do workflow, gerando as informações de proveniência. No entanto, a proposta não usa um modelo de proveniência padrão como OPM ou PROV. Em [Chebotko et al. 2010], é discutida uma abordagem para o gerenciamento de proveniência que integra tecnologias Web Semânticas com SGBD. Essa abordagem usa tecnologias da Web Semântica. O SciProvMiner também adota tecnologias da Web Semântica junto com armazenamento em SGBD, além de tratar com problemas relacionados à interoperabilidade fornecendo uma ferramenta de instrumentalização que captura os dados de proveniência de maneira independente do formato de proveniência de qualquer SGWfC. Por fim, em [Amann et al. 2013] é apresentada uma abordagem que gera dados de proveniência usando documentos XML. Esta abordagem é similar ao SciProvMiner por considerar tecnologias da Web Semântica para inferir conhecimento sobre proveniência de dados no contexto de workflows científicos, mas ela não trabalha provendo uma forma de captura de proveniência a partir de workflows modelados em SGWfCs com é feito no SciProvMiner, e sim inferindo informações de proveniência a partir de documentos XML gerados pela plataforma WebLab. O modelo de proveniência padrão utilizado em [Amann et al. 2013] é o PROV. No entanto, o SciProvMiner terá brevemente suporte ao PROV.

4 SciProvMiner- Arquitetura para Coleta, Armazenamento e Consulta de Proveniência de Dados

Conforme dito anteriormente, a arquitetura do SciProvMiner se baseia no uso de recursos da Web Semântica para fornecer maior conhecimento ao pesquisador acerca da proveniência capturada e para prover funcionalidades que aumentem o poder de análise do cientista sobre o experimento realizado, diminuindo ainda o esforço de instrumentalização necessário para que a proveniência do workflow seja capturada como um todo, além de oferecer ao pesquisador um ambiente fundamentado na interoperabilidade para a gerência e a consulta à proveniência de dados heterogêneos e distribuídos.

O SciProvMiner está inserido na linha de pesquisa em e-Science do Núcleo de Engenharia do Conhecimento (NEnC) da Universidade Federal de Juiz de Fora (UFJF). Nesse contexto, os estudos desenvolvidos procuram avançar em temas abordados anteriormente pelo NEnC, incluindo-se o arcabouço CelOWS [Matos et al. 2009] e as arquiteturas SASAgent [Felipe et al. 2011] e ComposerScience [Silva et al. 2011].

Uma representação da arquitetura do SciProvMiner considerando-se um cenário típico de aplicação é apresentada na Figura 2. Em um ambiente colaborativo interconectado através de uma grade computacional, os experimentos podem ser conduzidos de forma conjunta por grupos de cientistas que se encontram em centros de pesquisa localizados remotamente. Desta forma, os pesquisadores podem modelar workflows científicos a partir de SGWfCs distintos (como Kepler, Taverna e Vistrails) e cuja execução requer o acesso a repositórios heterogêneos (dados relacionais, semiestruturados, etc.) e distribuídos em uma grade computacional.

Neste cenário, o SciProvMiner especifica um mecanismo de instrumentalização baseada na tecnologia de serviços Web, que torna o mecanismo de coleta independentemente do SGWfC no qual o workflow foi desenvolvido, para coleta da proveniência prospectiva e retrospectiva dos diversos componentes dos workflows científicos que irão conduzir o experimento colaborativo. Este mecanismo de instrumentalização tem como objetivo coletar informações geradas durante o processo de execução do workflow e enviar esses metadados para um repositório de proveniência gerenciado pelo SciProvMiner. As informações de proveniência prospectiva e retrospectiva capturadas são persistidas em um banco de dados relacional e ficam disponíveis para serem consultadas pelo usuário através da interface construída no SciProvMiner.

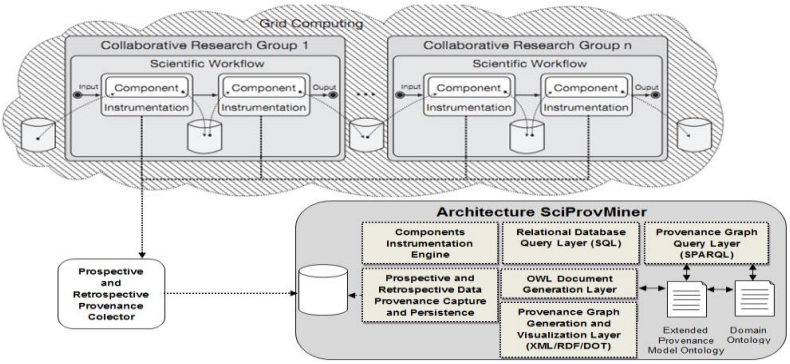

Fig. 2. Arquitetura do SciProvMiner

Considerando ainda a Figura 2, as camadas "Provenance Graph Generation and Visualization Layer", "OWL Document Generation Layer" e "Provenance Graph Query Layer" englobam as maiores contribuições do trabalho desenvolvido. A "camada de geração e visualização do grafo de proveniência" (Provenance Graph Generation and Visualization Layer na Figura 2) possui diversas atribuições interrelacionadas. A partir da base de dados relacional, as informações persistidas segundo o modelo OPM são processadas com o objetivo de se obter uma representação em memória do grafo de proveniência correspondente a cada execução de um workflow científico e dos modelos de execução deste workflow. Esta camada permite ainda a construção de uma representação visual do grafo de proveniência retrospectiva gerado. Este processo inicia-se a partir da serialização do grafo de proveniência na sintaxe de uma linguagem de marcação. Este arquivo constitui a base para a conversão do grafo para uma linguagem específica de descrição de gráficos, a partir da qual é possível produzir saídas em um formato padrão de visualização.

A "camada de geração de documento OWL" (*OWL Document Generation Layer*) da arquitetura do SciProvMiner tem o objetivo de, a partir dos dados de proveniência prospectiva e retrospectiva contidos na base de dados relacional, construir o documento OWL baseado na ontologia OPMO-e, que represente as informações de proveniência prospectiva e retrospectiva do workflow relacionado, de forma que na "camada de consulta ao grafo de proveniência" (*Provenance Query Graph Layer* na Figura 2), as consultas à ontologia OPMO-e, formuladas a partir da linguagem de consulta

SPARQL, possam ser realizadas e máquinas de inferência possam ser aplicadas sobre esse documento a fim de efetuar deduções sobre os metadados de proveniência.

A captura da proveniência prospectiva permite ao SciProvMiner a possibilidade de modelar na ontologia OPMO-e as regras de completude definidas no modelo OPM [Moreau et al. 2011], regras estas que se propõem a encontrar componentes do modelo que não foram informados de maneira explícita pelo usuário (artefato ou processo, dependendo da regra) e que, se descobertos, podem aumentar o conhecimento do cientista acerca do experimento realizado. Assim, foram estendidas as classes e propriedades da ontologia OPMO para comportarem a representação do modelo OPM estendido, que captura informações de proveniência prospectiva e retrospectiva, com o objetivo de aumentar o poder de processamento semântico pelas máquinas de inferência. Além disso, esta funcionalidade de captura da proveniência prospectiva confere ao SciProvMiner a possibilidade de otimização da instrumentalização do workflow pelo cientista. Como exemplo, pode ser citada a possibilidade de sublimação da instrumentalização da dependência causal *wasGeneratedBy*, uma vez que esta pode ser inferida pela ontologia OPMO-e, com base em regras definidas por cadeia de propriedades que levam em consideração informações de proveniência prospectiva em conjunto com informações de proveniência retrospectiva. Por questões de espaço, o detalhamento da captura de proveniência perspectiva e retrospectiva do SciProvMiner não poderá ser apresentado neste artigo.

4.1 Ontologia OPMO-e

Conforme já ressaltado, uma característica importante da arquitetura do SciProvMiner é a possibilidade de utilização das regras de completude e inferências válidas no modelo OPM sobre os dados de proveniência coletados, com o objetivo de enriquecer o conhecimento do cientista acerca da proveniência capturada, provendo informações que não foram explicitamente informadas pelo usuário. Para incorporar tais regras ao SciProvMiner, foi realizada uma extensão do modelo OPM com suporte a captura da proveniência prospectiva, e foram expandidas as classes e propriedades da ontologia OPMO para comportarem a representação do modelo OPM estendido. As novas regras implementadas na ontologia OPMO, aumentam a capacidade das máquinas de inferência em processar conhecimentos, além de permitirem a diminuição do esforço do cientista no momento da instrumentalização do workflow.

Neste cenário, a ontologia OPMO foi estendida e, com alguns ajustes, todas as inferências em múltiplos passos puderam ser capturadas pela ontologia, bem como a regra de completude que diz que uma aresta *wasTriggeredBy* pode ser obtida a partir da existência das arestas *used* e *wasGeneratedBy*. Por exemplo, a inferência em múltiplos passos *wasTriggeredBy**, p1 $\rightarrow$* p2, afirmando que o processo p1 foi disparado pelo processo p2 (*wasTriggeredBy*), se p1 usou um artefato que foi gerado (*wasGeneratedBy*) por p2 (possivelmente usando múltiplos passos), ou p1 foi derivado de um artefato (possivelmente utilizando múltiplos passos) que foi gerado por p2, não estava definida na ontologia OPMO original. Para defini-la, foi adicionada à ontologia OPMO-e uma propriedade denominada *wasTriggeredBy** utilizando o conceito de cadeias de propriedade da OWL2 (*property chains*), que define uma propriedade em termos de outras. Esta propriedade foi então definida em termos das seguintes cadeias

de propriedades: "*used⋆ ∘ wasGeneratedBy subproperty of wasTriggeredBy⋆″* e "*used⋆ ∘ wasGeneratedBy subproperty of wasTriggeredBy⋆″* **.**

No entanto, a regra de completude que esconde a introdução de processo e a regra de completude que esconde a introdução de artefato não são passíveis de serem capturadas utilizando apenas dados de proveniência retrospectiva, visto que apenas com estas informações não é possível saber qual seria o processo ou artefato a ser incluído. Para a cobertura dessas regras foram criadas na ontologia propriedades a partir dos recursos de *property chain*s, utilizando informações de proveniência prospectiva e retrospectiva.

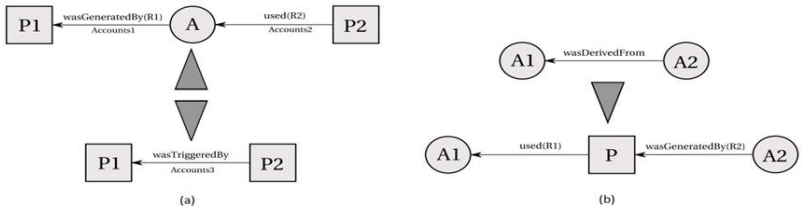

Fig. 3. Completude: (a)Eliminação e Introdução de Artefato; (b) Introdução de Processo [Moreau et al. 2011]

Por exemplo, a regra de completude Introdução de Artefato, ilustrada na Figura 3(a), afirma que a introdução de um artefato permite estabelecer que uma aresta *wasTriggeredBy* está escondendo a existência de algum artefato A usado por P2 (A *used* P2) e gerada por P1 (A *wasGeneratedBy* P1). Duas propriedades foram criadas para esta regra de completude, uma para definir a dependência causal *used* entre o processo P2 e o artefato A (P2 *used* A), denominada *usedOneStepArtifactIntroduction*, subpropriedade de *used*, e outra para definir a dependência causal *wasGeneratedBy* entre o artefato A e o processo P1 (A *wasGeneratedBy* P1), denominada *wasGeneratedByOneStepArtifactIntroduction*, subpropriedade de *wasGeneratedBy*. Ambas foram criadas em termos de cadeia de propriedades, que relacionam propriedades da proveniência prospectiva com propriedades da proveniência retrospectiva. Desta forma, somente foi possível a implementação dessas regras por conta do SciProvMiner capturar tanto a proveniência prospectiva e retrospectiva baseada no modelo OPM, o que não é possível com os outros sistemas disponíveis na literatura.

A regra de completude Eliminação de artefato (Figura 3(a)) também foi definida na ontologia OPMO-e através da propriedade *wasTriggeredByOneStep* construída a partir da cadeia de propriedades "*used⋆ ∘ wasGeneratedBy subproperty of wasTriggeredByOneStep".* A regra de completude Introdução de Processo, definida na documentação do modelo OPM, ilustrada na Figura 3(b), do mesmo modo que a Introdução de Artefato, ilustrada na Figura 3(a), também foi definida considerando propriedades da proveniência prospectiva relacionadas às propriedades da proveniência retrospectiva, tendo sido definida um propriedade chamada *usedOneStepProcessIntroduction*, subpropriedade used e uma propriedade denominada *wasGeneratedByOneStepProcessIntroduction*, subpropriedade de *wasGeneratedBy* para representar a regra. Apesar de em [Moreau et al. 2011] não estar definida explicitamente a regra de completude de Eliminação de Processo, em notas de rodapé da documentação do modelo OPM é sugerido que, se existir uma anotação indicando que todas as

saídas de um processo do workflow são dependentes de todas as suas entradas, então a inferência inversa, isto é, a eliminação de processo, pode ser afirmada. O SciProvMiner fornece ao usuário a possibilidade de, no momento da instrumentalização do workflow, definir se para aquele workflow essa afirmação é verdade e, se for, é associada uma versão da ontologia que implementa essa regra. Essa regra é implementada através da propriedade *wasDerivedFromOneStepProcessElimination*, sob a cadeia de propriedades "wasGeneratedBy ○ used".

Desta forma o SciProvMiner passou a ab*ranger* todas as regras de completude e inferência válidas no modelo OPM, fornecendo ao usuário ferramentas que aumentam o seu conhecimento acerca do experimento realizado.

Além disso, com o objetivo de diminuir o trabalho do cientista no momento da instrumentalização do workflow, foi implementada uma propriedade denominada *wasGeneratedbyRPP,* construída a partir do conceito de cadeia de propriedades, relacionando propriedades da proveniência prospectiva com retrospectiva em sua formação, que infere a dependência causal *wasGeneratedBy* sem que esta tenha sido explicitamente instrumentalizada pelo usuário. A idéia por trás desta regra é que se um artefato A foi disponibilizado para ser utilizado por outros processos através de uma porta de saída P de uma tarefa T, e sendo o processo P a representação da execução da tarefa T, então pode ser inferido que o artefato A foi gerado pelo processo P.

Além das regras acima especificadas, foram adicionadas à ontologia OPMO-e propriedades baseadas em *property chains* que relacionam apenas propriedades de proveniência prospectiva para enriquecimento da ontologia, como, por exemplo, as propriedades *likeSourceComponent** *likeDestinationComponent** que inferem respectivamente a cadeia de componente antecessores e a cadeia de componentes sucessores de um determinado componente.

Como pode ser visto, o SciProvMiner explora tecnologias de serviços Web para o desenvolvimento da ferramenta de captura de proveniência prospectiva e retrospectiva de dados, de forma a garantir a independência da ferramenta de coleta do SciProvMiner em relação ao SGWfC utilizado pelo cientista e estende a ontologia OPMO de forma a enriquecer semanticamente o conhecimento do cientista a respeito do experimento realizado, trazendo informações que não seriam possíveis de serem obtidas caso as regras de completude não tivessem sido especificadas.

5 Prova de Conceito

Com o objetivo de mostrar a aplicabilidade da abordagem aqui proposta foi utilizado o workflow Load-PC3, desenvolvido no SGWfC Kepler no Third Provenance Challenge (PC3) [SIMMHAN et al. 2011] para responder ao desafio proposto neste evento. O Load-PC3 é baseado no workflow científico de carga do Pan-STARRS, que armazena arquivos de dados dentro de um banco de dados relacional para o projeto Pan-STARRS. Este workflow é um workflow real e é adequado para flexionar as diferentes características do modelo OPM.

Para a instrumentalização completa do workflow foram adicionadas 41 instâncias de serviço Web de instrumentalização, sendo 35 instâncias com o método *used*, duas com o método *wasGeneratedBy*, uma com o método *initialConfiguration*, uma com o método *wasTriggeredBy* e uma com o método *wasDerivedFrom*. Com essa instru-

mentalização é possível exemplificar todas as regras de completude e inferência do modelo OPM sendo inferidas. Se não fosse a capacidade do SciProvMiner em inferir relações que não foram explicitamente informadas pelo usuário seriam necessárias pelo menos mais 21 instâncias de *wasGeneratedBy*, 36 instâncias de *wasDerivedFrom* e outras 21 instâncias de *wasTriggeredBy*, para ser possível obter a mesma gama de informações. Desta forma fica comprovada a capacidade do SciProvMiner em otimizar o processo de instrumentalização do workflow sem perda de conhecimento fornecido ao usuário a respeito da proveniência capturada.

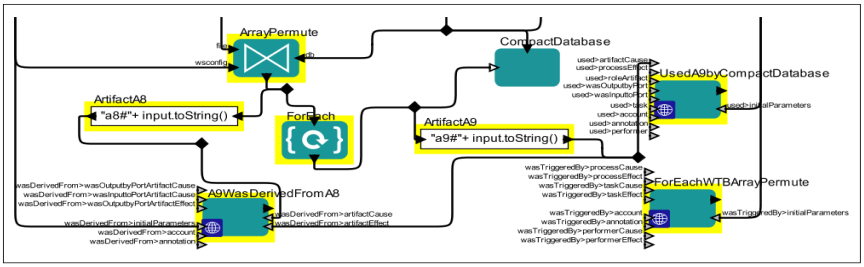

Fig. 4. Workflow Load-PC3 instrumentalizado (visão parcial)

Por questões de espaço mostramos na Figura 4 apenas um corte da instrumentalização do workflow Load-PC3 no SGWfC Kepler. Na Figura 4 estão ilustradas todas as instâncias de instrumentalização que foram definidas no workflow para os artefatos e processos relacionados às tarefas *ArrayPermute* e *ForEach*. A instância de instrumentalização *A8WasDerivedFromA8*, indicando que A9 foi derivado do artefato A8, captura além da dependência causal definida no modelo OPM entre esses artefatos, as informações de proveniência prospectiva relacionadas a por qual porta o artefato A8 saiu da tarefa *ArrayPermute* e por qual porta da tarefa *ForEach* este artefato entrou, bem como por qual porta de *ForEach* o artefato A9 se torna disponível para ser usado por outras tarefas, além das informações das tarefas nas quais os processos estão relacionados. A Instância de instrumentalização *ForEachWTBArrayPermute* captura a dependência causal *wasTriggeredBy* entre os processos *ForEach* e *ArrayPermute*, e as tarefas relacionadas a esses processos. Já a instância de instrumentalização *UsedA9ByCompactDatabase* captura a dependência causal *used* entre o artefato A9 e o processo relacionado à tarefa *CompactDatabase*, bem como as informações de proveniência prospectiva relacionadas à tarefa *compactDataBase* e as portas de entrada e saída por onde o artefato A9 é consumido pela tarefa *CompactDataBase* e gerado pela tarefa *ForEach*.

A Figura 5 exemplifica duas inferências obtidas a partir das informações de proveniência capturadas pelo SciProvMiner. Através da Figura 5(a) pode ser visto que apesar de não ter sido instrumentalizado nenhuma dependência causal *wasGeneratedBy* entre o processo *ForEach* e o artefato A9 (Figura 4), ela foi inferida. Da mesma forma, na Figura 5(b) pode ser notada a dependência causal *used* entre o processo *ForEach* e o artefato A8, apesar desta relação não ter sido explicitamente informada pelo usuário na instrumentalização, conforme mostrado na Figura 4. Essas inferências foram obtidas a partir da implementação da regra de completude Introdução de Pro-

cesso definida na documentação do modelo OPM que foi implementada na ontologia OPMO-e desenvolvida neste trabalho. Como pode ser visto, a partir do conhecimento informado pelo usuário de que o artefato A9 foi derivado do artefato A8, o *reasoner* inferiu que o artefato A9 foi gerado pelo processo *ForEach* e que o Processo *ForEach* usou o artefato A8, informações que não foram explicitamente informadas na instrumentalização do workflow. Da mesma forma, pela regra de completude Introdução de Artefato, na Figura 6 o *reasoner* infere a dependência causal *ForEach used* A8 (Figura 6(a)) e A8 *wasGeneratedBy ArrayPermute* (Figura 6(b)). Outra inferência que pode ser constatada na Figura 6(b) é que o artefato A8 foi classificado como *NoUsersInput*, ou seja, artefato que foi gerado por alguma tarefa do workflow. Pela Figura 7 pode-se observar a inferência da dependência causal *CompactDatabase wasTriggeredBy ForEach*, que também não havia sido explicitamente informada pelo usuário.

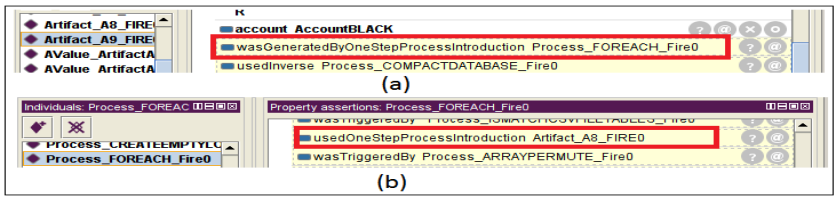

Fig. 5. Inferência da Regra de Completude Introdução de Processo

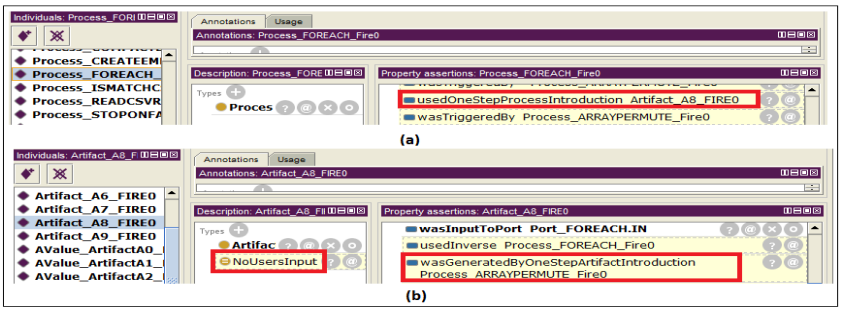

Fig. 6. Inferência da Regra de Completude Introdução de Artefato

Os exemplos acima relatados são apenas de algumas das diversas inferências possíveis de serem obtidas através da utilização do SciProvMiner para captura e consulta aos dados de proveniência coletados de um workflow. Deve ser destacado que todas as dependências causais *wasGeneratedBy* podem ser sublimadas da instrumentalização do workflow, por serem inferidas pela ontologia OPMO-e. Além disso, o SciProvMiner implementou todas as regras de inferências em múltiplos passos definidas no modelo OPM além de outras regras relacionadas a proveniência prospectiva, dando subsídio para serem respondidas questões relacionadas à especificação do workflow, como por exemplo qual é o componente sucessor imediato de um componente, qual é o antecessor imediato de um componente, quais componentes estão conectados, quais tarefas foram executadas pelo workflow, visto que se ocorrer algum erro durante a execução do workflow, alguma tarefa pode deixar de ser realizada, quais

tarefas deixaram de ser executadas pelo workflow em caso de algum erro na execução, entre outras. Essas duas últimas perguntas fazem parte das questões do *Third Provenance Challenge*, que não puderam ser respondidas por nenhuma equipe baseadas apenas no modelo OPM [LIM et al. 2011].

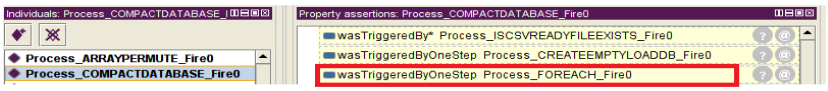

Fig. 7. Inferência de wasTriggeredBy

6 Conclusão

Conforme ressaltado anteriormente, um dos desafios considerados estratégicos na área de e-Science é o de prover informação histórica dos experimentos científicos, em vistas a tratar o problema de perda de conhecimento do cientista sobre o experimento.

Neste contexto, o SciProvMiner, tem como objetivo principal especificar uma arquitetura para apoiar a coleta e gerência da proveniência de dados e processos, respaldado na hipótese de que a captura e a gerência de dados de proveniência irá fornecer ao cientista informações importantes a respeito do experimento realizado. O SciProvMiner, por capturar a proveniência prospectiva trouxe de ganho a possibilidade de responder a consultas relacionadas à especificação do workflow. Além disso, a proveniência prospectiva foi utilizada como base para a formação de regras na ontologia OPMO em conjunto com a proveniência retrospectiva, tais como *wasGeneratedByRPP*, que possibilitou a inferência da dependência causal *wasGeneratedBy* entre um artefato e um processo, mesmo sem ter sido explicitamente informada pelo usuário. Esta interação de informações de proveniência prospectiva e retrospectiva também tornou possível que fosse inferido o "artefato escondido" na regra de completude Introdução de Artefato e também o "processo escondido" na regra de completude Introdução de Processo, ambas definidas na documentação do modelo OPM [MOREAU et al. 2011], sendo que apenas com informação de proveniência retrospectiva não seria possível a descoberta dessas informações implícitas.

7 Referências Bibliográficas

1. AMANN, B.; CONSTANTIN, C.; CARON, C.; GIROUX, P. WebLab PROV: computing fine-grained provenance links for XML artifacts. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops, p. 298-306 (2013)
2. BOWERS, S.; MCPHILLIPS, T.; LUDASCHER, B. Declarative rules for inferring fine-grained data provenance from scientific workflow execution traces. In: Provenance and Annotation of Data and Processes, p. 82-96 (2012)
3. CHEBOTKO, A.; LU, S.; FEI, X.; FOTOUHI, F. RdfProv: A relational rdf store for querying and managing scientific workflow provenance. Data & Knowledge Engineering, Elsevier, v. 69, n. 8, p. 836-865 (2010)
4. CUEVAS-VICENTTIN, V.; DEY, S.; WANG, M. L. Y.; SONG, T.; LUDASCHER, B. Modeling and querying scientific workflow provenance in the d-opm. In: Proceedings of

the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, p. 119-128 (2012)
5. FELIPE M. L., SILVA, L, MATOS, E., BRAGA, R., CAMPOS, F.: SASAgent: An agent based architecture for search, retrieval and composition of scientific models. Computers in Biology and Medicine , vol. 1, pp. 1-14 (2011)
6. FREIRE, J.; KOOP, D.; SANTOS, E.; SILVA, C. T. Provenance for Computational Tasks: A Survey, Computing in Science & Enginnering, v. 10, n. 3, p. 11-21 (2008).
7. LIM, C., LU, S., CHEBOTKOT, A., FOTOUHI, F.: Prospective and retrospective provenance collection in scientific workflow environments. Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010, art. n. 5557202, pp. 449-456 (2010)
8. LIM, C.; LU, S.; CHEBOTKO, A.; FOTOUHI, F. Storing, reasoning, and querying opm-compliant scientific workflow provenance using relational databases. Future Gener. Comput. Syst., Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 27, n. 6, p. 781-789 (2011).
9. MARINHO, A. S.: ProvManager: Uma Abordagem para Gerenciamento de Proveniência de Experimentos Científicos. Dissertação de mestrado, Engenharia de Sistemas e Computação, UFRJ/COPPE, Rio de Janeiro, RJ, Brasil (2011)
10. MATOS, E. E., CAMPOS, F., BRAGA, R., PALAZZI, D.: CelOWS: an ontology based framework for the provision of semantic Web services related to biological models. Journal of Biomedical Informatics , vol. 43, pp. 125-136 (2009)
11. MATTOSO, M., WERNER, C., TRAVASSOS, G., BRAGANHOLO, V., MURTA, L.: Gerenciando experimentos científicos em larga escala. Seminário Integrado de Software e Hardware, pp. 121–135, Sociedade Brasileira de Computação: Porto Alegre, RS, Brazil (2008)
12. MOREAU, L., CLIFFORD, B., FREIRE, J., FUTRELLE, J., GIL, Y., GROTH, P., KWASNIKOWSKA, N., MILES, S., MISSIER, P., MYERS, J., PLALE, B., SIMMHAN, Y., STEPHAN, E., DEN BUSSCHE, J. V.: The Open Provenance Model core specification (version 1.1), Future Generation Computer Systems, vol. 27 Issue 6, pp.743 –756 (2011)
13. SILVA, L. M., BRAGA, R., CAMPOS, F.: Composer-Science: A semantic service based framework for workflow composition in e-Science projects. Information Sciences, pp. 186-208 (2011)
14. SIMMHAN, Y.; GROTH, P.; MOREAU, L. Special section: The third provenance challenge on using the *Open Provenance Model* for interoperability. Future Generation Computer Systems, Elsevier, v. 27, n. 6, p. 737-742 (2011)
15. WONG, S. C., MILES, S., FANG, W., GROTH, P. and MOREAU, L.: Provenance-based Validation of E-Science Experiments. In: 4th International Semantic Web Conference (ISWC), 6-10 November 2005, Galway, Ireland, pp. 801-815 (2005)