
PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° MCC02/2017

# Um Framework para Proveniência de Dados

**Tassio Ferenzini Martins Sirqueira**
**Marx Lelis Viana**
**Nathalia Nascimento**
**Carlos José Pereira de Lucena**

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO**
**RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900**
**RIO DE JANEIRO - BRASIL**

Monografias em Ciência da Computação, No. MCC02/2017 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena March, 2017

# Um Framework para Proveniência de Dados

Tassio Ferenzini Martins Sirqueira[1], Marx Lelis Viana[1], Nathalia Nascimento[1]
Carlos José Pereira de Lucena[1]

[1]Pontifícia Universidade Católica do Rio de Janeiro

(tmartins, mlelis, nnascimento, lucena)@inf.puc-rio.br

**Abstract.** Data provenance refers to the historical record of the derivation of the data, allowing the reproduction of experiments, interpretation of results and identification of problems through the analysis of the processes that originated the data. Data provenance contributes to the evaluation of experiments. This paper presents a framework for data provenance using the W3C provenance data model, called PROV-DM. Such framework aims at contributing to, and facilitating, the collection, storage and retrieval of provenance data through a modeling and storage layer based on PROV-DM, yet is compatible with other representations of PROV such as PROV-O. To demonstrate the utilization of the framework, it was used in an IoT application that performs the gas classification to identify diseases.

**Keywords**: Data Provenance; PROV Framework; PROV W3C;

**Resumo**. A procedência dos dados refere-se ao registro histórico da derivação dos dados, permitindo a reprodução de experimentos, interpretação de resultados e identificação de problemas através da análise dos processos que originaram os dados. A proveniência dos dados contribui para a avaliação de experimentos. Este trabalho apresenta uma estrutura para a proveniência de dados usando o modelo de dados de procedência do W3C, chamado PROV-DM. Esse quadro visa contribuir e facilitar a coleta, armazenamento e recuperação de dados de proveniência através de uma camada de modelagem e armazenamento baseada em PROV-DM, mas é compatível com outras representações de PROV tais como PROV-O. Para demonstrar a utilização do framework, foi utilizado em um aplicativo de IoT que realiza a classificação de gases para identificação de doenças.

**Palavras-chave**: Proveniência de Dados; Framework PROV; PROV W3C;

---

**In charge of publications**

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br

## Table of Contents

# 1 Introdução

O termo "proveniência" refere-se à origem ou procedência dos dados, ou seja, é um registro do histórico de derivação de dados, que possibilita reprodutibilidade de experimentos, interpretação de resultados e diagnóstico de problemas (Lim *et al.*, 2010). A definição clássica foi descrita por Buneman *et al.* (2000), que mostra que a proveniência é a documentação complementar de um dado, contendo informações de "como", "quando", "onde" e "por que" os dados foram obtidos e "quem" obteve.

A proveniência fornece um olhar além das especificações dos domínios e sugere a adoção de modelos disciplinados, onde a informação de proveniência de dados pode ser usada para aprender ou compreender métodos e regras de design. Pode ainda auxiliar os usuários em investigações semelhantes, a fim de compreender as correlações de dados e melhorar futuras investigações (Foster, 2006). Atualmente, a proveniência dos dados é aplicada com sucesso em áreas como artes (Moreau *et al.*, 2008), bibliotecas digitais (Moreau *et al.*, 2008) e E-Ciência (Sirqueira *et al.*, 2016).

Um dos problemas de proveniência refere-se à falta de concordância quanto à abrangência dos dados a serem capturados, além da ausência de uma definição clara de como esse procedimento deve ser realizado (Marinho *et al.,* 2009). Outras questões levantadas quanto à proveniência dos dados são a confiabilidade dos dados, a integridade, a confidencialidade sobre seu uso, a disponibilidade para outras pessoas, além da eficácia em relação ao que está sendo capturado e a eficiência com que isso é feito, assegurando que todas as informações relevantes é capturado.

A proveniência dos dados deve ser tratada como algo importante e útil, registrando cada mudança nos dados. A engenharia de software experimental e outras áreas de estudos empíricos ganharão um mecanismo que os auxiliará na construção de suas bases de conhecimento para os experimentos realizados, na rastreabilidade de bugs e na reprodutibilidade dos experimentos.

O objetivo desta seção é discutir os benefícios que a proveniência de dados pode trazer, considerando os dados capturados, como eles foram modelados e armazenados, e o tipo de informação a ser obtida a partir deles. Neste contexto, apresentamos um framework (Mattsson, 2000) para a proveniência de dados (FProvW3C).

O FProvW3C visa auxiliar na coleta, armazenamento e recuperação de dados de proveniência em aplicações computacionais. Nossa abordagem é baseada no modelo de dados de proveniência do W3C, chamado PROV-DM (Missier *et al.*, 2013), e no PROV-O (Missier *et al.*, 2013), que é uma ontologia para mapear o modelo de dados. Como resultado, se um pesquisador desenvolve um aplicativo usando FProvW3C, seu aplicativo conterá uma camada Java com as classes PROV-DM e anotações. Portanto, ele pode usar nossa estrutura para lidar com as regras PROV em um prazo razoável.

Para ilustrar o uso de FProvW3C, vamos apresentar um exemplo de um sistema baseado na Internet das Coisas (IoT) (Gubbi *et al.*, 2013): um sistema que coleta dados de sensores de gás e auxilia na classificação de gases emitidos por seres humanos. Escolhemos este exemplo porque o IOT é uma abordagem empolgante e emergente que ganhou atenção tanto acadêmica como industrial.

O restante deste trabalho está organizado da seguinte forma. A Seção 2 apresenta o modelo W3C PROV (Missier *et al.*, 2013). A Seção 3 discute o trabalho relacionado. A Seção 4 detalha o FProvW3C e a Seção 5 apresenta a instância do framework. Finalmente, a Seção 6 apresenta nossas conclusões e trabalhos futuros.

## 2 PROV W3C

De acordo com Davidson *et al.* (2008), a proveniência dos dados pode ser dividida em três tipos: (i) Prospectiva: é a sequência de processos utilizados na geração de dados; (ii) Retrospectiva: são as informações obtidas durante a execução de processos de geração de dados e ambiente, e (iii) Dados de usuário: qualquer informação que o usuário considere necessária para análise futura. Além disso, a proveniência pode ser obtida de duas maneiras, de acordo com Tan *et al.* (2004), que são: (i) Preguiçoso: a proveniência é obtida a partir do momento em que sua captura é solicitada, e (ii) Ansiosa: a proveniência é obtida em todos os momentos e está prontamente disponível. A melhor maneira de coletá-lo dependerá da aplicação a ser aplicada.

Atualmente, existem dois padrões principais para a captura de dados de proveniência: (i) o modelo OPM (Moreau *et al.*, 2008b), com três vértices, cinco relações causais, e (ii) o modelo PROV (Missier *et al.*, 2013), com três vértices principais e sete relações básicas, mais complementares. Neste trabalho, o modelo de procedência utilizado foi o PROV (Missier *et al.*, 2013) pela sua amplitude e maior número de relações causais para a representação do conhecimento.

O modelo PROV (Missier *et al.*, 2013) consiste em 12 documentos que definem sua especificação, de acordo com a Fig. 1. Entre os principais documentos estão o PROV-DM, que especifica o modelo de captura de dados; O PROV-CONSTRAINTS, que é o conjunto de restrições aplicáveis ao modelo de dados (PROV-DM) e o PROV-O, uma ontologia para o mapeamento do modelo de dados.

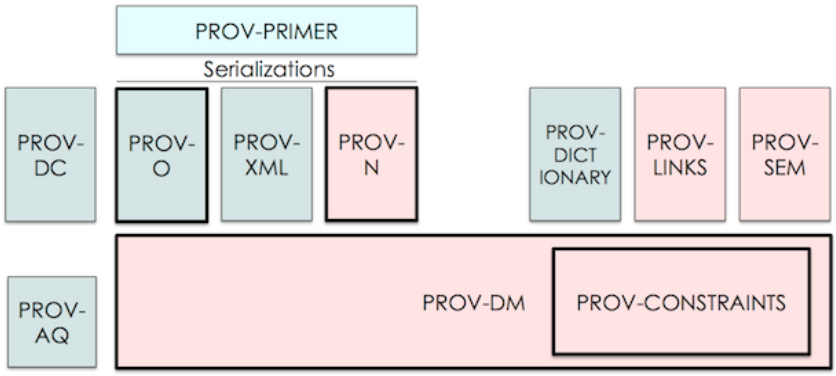



**Figura 1. Organização do modelo PROV[1].**

O PROV-DM cria uma separação de tipos e relações causais, sendo os tipos:

- Entidade: é um tipo físico, digital ou conceitual, ou algo com aspectos fixos, em que as entidades podem ser reais ou imaginárias.
- Atividade: é algo que ocorre ao longo de um período de tempo e atua em entidades.
- Agente: é algo que tem algum tipo de responsabilidade pela atividade e pela existência de uma entidade, ou pela atividade de outro agente. Agente, no modelo PROV, pode ser classificado como uma organização, uma pessoa ou um agente de software.

Em consideração às relações causais, elas são divididas em dois subconjuntos: relações primárias e secundárias (opcionais). A Figura 2 mostra as relações primárias em negrito e a Figura 3 apresenta as relações secundárias (opcional).

[1] PROV-Overview: http://www.w3.org/TR/prov-overview/

| | | Object | | |
|---|---|---|---|---|
| | | Entity | Activity | Agent |
| Subject | Entity | **WasDerivedFrom**<br>Revision<br>Quotation<br>PrimarySource<br>AlternateOf<br>SpecializationOf<br>HadMember | **WasGeneratedBy**<br>WasInvalidatedBy<br>*R T L* | **WasAttributedTo** |
| | Activity | **Used**<br>WasStartedBy<br>WasEndedBy<br>*R T L* | **WasInformedBy** | **WasAssociatedWith** *R* |
| | Agent | — | — | **ActedOnBehalfOf** |

**Figura 2. Relações Primarias do PROV[2].**

| | | Secondary Object | | |
|---|---|---|---|---|
| | | Entity | Activity | Agent |
| Subject | Entity | — | WasDerivedFrom (activity) | — |
| | Activity | WasAssociatedWith (plan) | WasStartedBy (starter)<br>WasEndedBy (ender) | — |
| | Agent | — | ActedOnBehalfOf (activity) | — |

**Figura 3. Relações secundárias do PROV[3].**

Uma das vantagens de usar o PROV e a ontologia PRV-O e que o PROV-DM pode ser representado usando OWL2 (Web Ontology Language) (Lebo *et al.*, 2013). Eles também podem ser usados para representar e trocar informações de proveniência geradas em diferentes sistemas e em contextos diferentes, como apresentado na Figura 4.

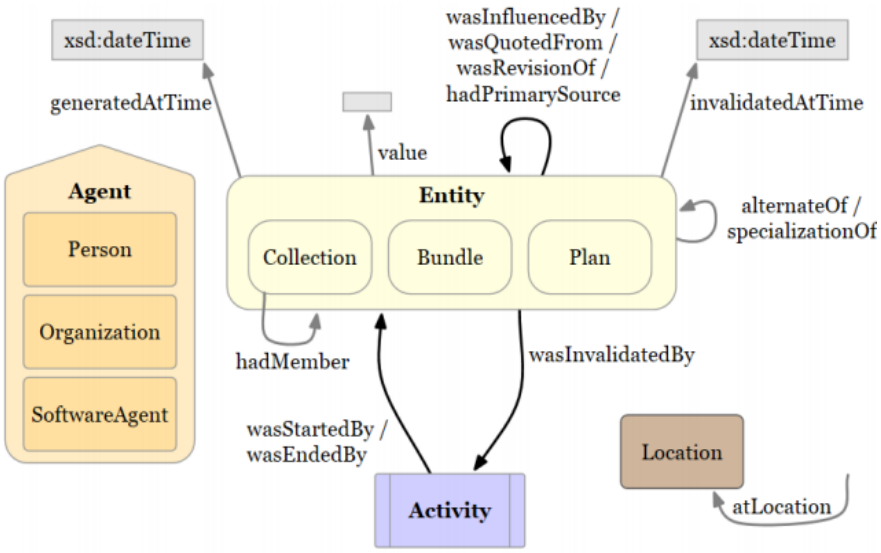



**Figura 4. PROV-O sobre o modelo PROV-DM[4].**

[2, 3] PROV-DM: http://www.w3.org/TR/prov-dm/

[4] PROV-O: http://www.w3.org/TR/prov-o/

Além disso, outra vantagem de usar o modelo PROV é com relação ao modelo de armazenamento, onde diferentes fontes de informação são convertidas em um modelo padronizado pelo W3C. Isto, por sua vez, facilita a compreensão, a rastreabilidade e a reprodutibilidade de um dado, devido ao processo que o originou. Além disso, permite anotação semântica usando o PROV-O.

A proveniência pode ser uma métrica de qualidade importante em experimentos, uma vez que o processo de derivação de dados tem implicações tanto na qualidade dos dados quanto nos erros introduzidos por dados defeituosos à medida que se propagam em outras derivações (Veregin *et al.*, 1995). Proveniência no processo de experimentação pode ajudar a aumentar a validade dos experimentos, uma vez que a confiabilidade dos dados serão monitorados.

O objetivo principal do framework FProvW3C é simplificar a captura de dados de proveniência e facilitar o uso do modelo PROV, permitindo experimentos mais confiáveis.

## 3 Trabalhos Relacionados

A proveniência pode ser usada como uma métrica de qualidade para a avaliação de dados, pois tem implicações significativas na qualidade dos dados e erros introduzidos por dados defeituosos, que aumentam à medida que as derivações são propagadas (Veregin *et al.*, 1995).

Para apoiar a pesquisa desenvolvida neste artigo, foi realizada uma busca estruturada para explorar as formas em que a proveniência dos dados tem sido aplicada na computação. Entre os resultados obtidos, foram selecionados estudos que são aplicados explicitamente à proveniência de dados com base no modelo PROV do W3C.

Começando com ProvToolbox[5] é uma biblioteca Java para criar representações PROV-DM e convertê-los entre RDF, PROV-XML, PROV-N e PROV-JSON. Apesar de trabalhar com a representação de PROV-DM, não é voltado para capturar e armazenar dados em um SGBD (Sistema Gerenciador de Banco de Dados), além disso, é um conjunto de ferramentas independentes para cada forma de representação dos dados.

O prov-api[6] é uma API Java para criar e manipular gráfos de proveniência. Atualmente, a API só implementa os termos essenciais do PROV. Ele tem duas implementações: uma usando Jena e outra baseada no SPARQL v1.1, ambas armazenando os dados em ontologias. O foco do prov-api é a inferência e consulta na ontologia PROV-O e não no armazenamento de dados usando o PROV-DM, bem como seu uso na captura de dados de proveniência.

A biblioteca PROV Python[7] é uma biblioteca que fornece uma implementação do PROV-DM em Python. Ele contém um aplicativo Django para armazenar e carregar instâncias usando o Django ORM. Apesar das preocupações com o armazenamento de dados, a biblioteca é específica para aplicações desenvolvidas em Python, diferente do que é proposto neste trabalho.

O E-SECO ProVersion apresentado por Sirqueira *et al.* (2016), é uma plataforma de gestão de *workflows* científicos, que utiliza proveniência de dados para controlar a manutenção e evolução dos *workflows* e dos dados de experimentos. Embora o aplicativo

[5] http://lucmoreau.github.io/ProvToolbox/

[6] https://github.com/dcorsar/prov-api/

[7] http://pypi.python.org/pypi/prov

trabalhe com dados de proveniência no modelo PROV-DM, ele usa uma abordagem preguiçosa para a captura de dados, fazendo uso de um serviço da web. Embora este aplicativo armazena os dados em um SGBD e mais tarde converte-lo em uma ontologia, é uma abordagem específica para *e-science*.

Já Dalpra *et al.* (2015), apresenta o que mais tarde chamou Prov-Process, uma plataforma para coleta, armazenamento e análise de dados de proveniência. No entanto, um modelo padrão deve ser usado para entrada de dados no formato ".csv" e não permite a integração com outros sistemas.

O ProvManager, apresentado por Marinho *et al.* (2009), é uma ferramenta de armazenamento e análise de dados, utiliza prolog para consultas e, como o E-SECO, não possui mecanismos de integração com outros sistemas, dependendo de uma forma particular de entrada de dados, além de sua aplicação também ser voltada para *e-science*.

Embora existam vários aplicativos voltados para a proveniência de dados, eles não possuem recursos que auxiliam os usuários na captura e armazenamento de dados. Como tal, FProvW3C é um framework desejável para experiências em ESE e outras áreas que fazem uso da análise de dados empíricos para os resultados. Na próxima seção serão apresentados os detalhes do framework.

## 4 FProvW3C - Um Framework para Proveniência de Dados

Construir uma base de conhecimento sobre um assunto não é trivial, especialmente considerando dados de experimentos empíricos. Conforme abordado por Sirqueira *et al.* (2016), a proveniência de dados é algo constante, e deve seguir todas as etapas realizadas para obter resultados concisos. Uma maneira de observar isso é considerar o ciclo de vida de um dado, onde não só os dados são importantes, mas também o processo que o originou.

O framework FProvW3C trabalha com uma abordagem ansiosa (Tan *et al.*, 2004), de modo que os dados são coletados em todos os momentos e podem ser consultados em seguida. Atualmente, em sua arquitetura, o framework FProvW3C apresenta todas as especificações do PROV-DM com as anotações em Java Persistence API (JPA), de acordo com a Figura 5.

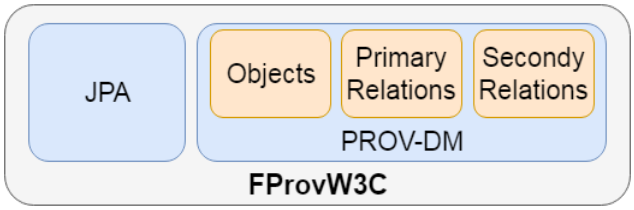



**Figura 5. Arquitetura do Framework.**

Além da modelagem, a estrutura FProvW3C traz todas as anotações de persistência. Isso reduz o trabalho dos usuários e evita erros de mapeamento por aqueles que não têm conhecimento extensivo do PROV. Ser um framework permite integrá-lo em diferentes sistemas. A estrutura de frozen-spots são os vértices principais e os hot-spots são as relações causais, ambas definidas pelo modelo PROV, e se adaptam a diferentes contextos de aplicação. O framework foi desenvolvido em Java e as anotações são baseadas em Java Persistence API (JPA), o que torna o framework independente do SGBD que será aplicado para armazenar os dados coletados, deixando essa escolha para o usuário.

O FProvW3C é um framework de mapeamento de objetos relacional encarregado de criar o banco de dados, as tabelas e seus respectivos atributos no SGBD. O modelo PROV determina as classes e como elas se relacionam, além dos atributos básicos. Desta forma,

o framework fornece as classes com os atributos básicos (frozen-spots) e, além disso, permite a criação de hot-spots, estendendo os atributos de cada classe e os relacionamentos. Estes atributos pretendem representar as características do sistema no qual FProvW3C será aplicado, de acordo com a Figura 6.

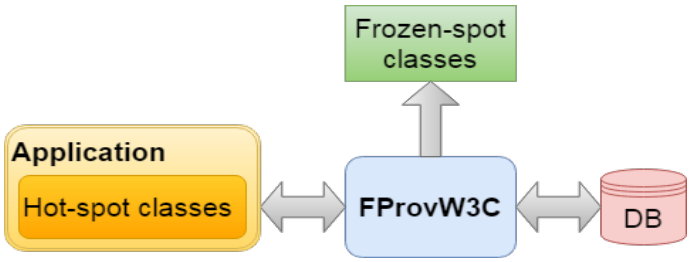



**Figura 6. Utilização e extensão do Framework**.

Como a estrutura executa o mapeamento de dados de acordo com o PROV, seu uso é facilitado para o usuário, onde com base nas relações básicas do modelo (ver Figura 7) os usuários são capazes de aplicar a proveniência de dados em suas aplicações.

Conforme ressaltado por Wohlin *et al.* (2012), a forma de coleta de dados em experimentos empíricos influencia na sua validação. A captura manual da proveniência dos dados pode distorcer os resultados levando as experiências à falha. Por outro lado, usando um modelo de dados de proveniência e fazendo com que a tarefa de captura e armazenamento seja uma responsabilidade do computador, reduzimos a possibilidade de distorção dos resultados e, além disso, criamos uma base sólida na experiência. Como resultado, facilitamos a identificação de erros, a rastreabilidade dos passos e a replicação da experimentos, uma vez que todos os dados com as respectivas etapas são registrados.

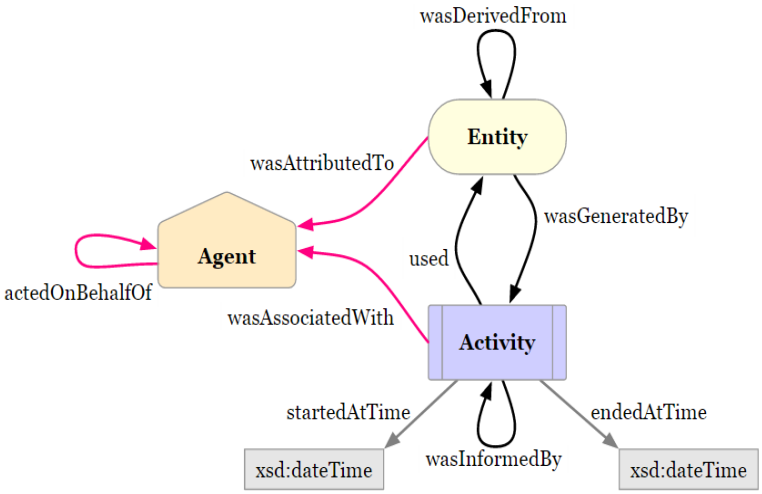



**Figura 7. Relações do PROV-DM[8].**

## 5 Cenário de Uso: Análise de Gás

A necessidade de construir plataformas para auxiliar especialistas em análises de cuidados de saúde é atualmente um problema crítico, especialmente para prevenir doenças que requerem diagnósticos invasivos, como Síndrome do Intestino Irritável (IBS) e Gas-

[8] PROV-O: http://www.w3.org/TR/prov-o/

trite. Como descrito por *(*Nascimento *et al., 2016)*, uma pessoa emite vários gases de diferentes partes do corpo (por exemplo, flatulência, eructação, exalação) e estes gases poderiam ser úteis para o diagnóstico de um conjunto de doenças intestinais e estomacais.

No entanto, existem muito poucas abordagens tecnológicas para facilitar a análise de flatulência e outros gases diariamente emitidos pelos seres humanos. Além disso, há uma necessidade de alta confiabilidade de dados nestas abordagens, uma vez que ele usa a identificação de doenças baseadas em gás, isto é, todas as informações devem ser confiáveis e os agentes não podem deixar de capturar ou classificar dados.

O framework FProvW3C foi utilizado na aplicação "Gases Device" (Nascimento *et al.*, 2016). Este aplicativo é baseado na Internet das Coisas (IoT), que usa sensores para medir gases no ambiente e usa agentes de software para fornecer a classificação de dados. A figura 9 mostra a arquitetura de aplicação Gases Device estendendo classes FProvW3C.

O sistema de monitorização dos gases (Nascimento *et al.*, 2016) é composto por um Arduino (Doukas, 2012) e pelos seguintes sensores de gás: (i) MQ-135 (sensor de gás CO2); (Ii) MQ-4 (sensor de gás metano), e (iii) MQ-8 (sensor de gás hidrogénio).

A Arduino utiliza estes sensores para medir a quantidade de gases no ambiente. Em seguida, o Arduino envia esses dados para o aplicativo Web chamado Smell App. O usuário pode acessar o aplicativo da Web para verificar a variação da quantidade de gases no ambiente após a expiração do ar. Por exemplo, o objetivo do Dispositivo de Gases é permitir que um usuário mede a quantidade de metano, hidrogênio e CO2 em sua exalação.

## 5.1 Visão geral

O FProvW3C cria uma camada intermediária entre o aplicativo e o banco de dados. Esta camada torna possível tratar e mapear os dados a serem armazenados ligando a fonte de informação a eles. Além disso, o FProvW3C cria uma base de conhecimento baseada no uso do aplicativo. Portanto, todos os dados inseridos pelos usuários ou sensores no aplicativo, bem como os dados manipulados pelo aplicativo são registrados nesta base.

A figura 8 mostra que cada vez que um sensor de gás capta a variação de um gás, a persistência dos dados na base de dados é invocada através da estrutura FProvW3C, registando a proveniência dos dados. Desta forma, toda a captura e manipulação de dados são gravados, como uma filmagem, formando a base histórica da aplicação.

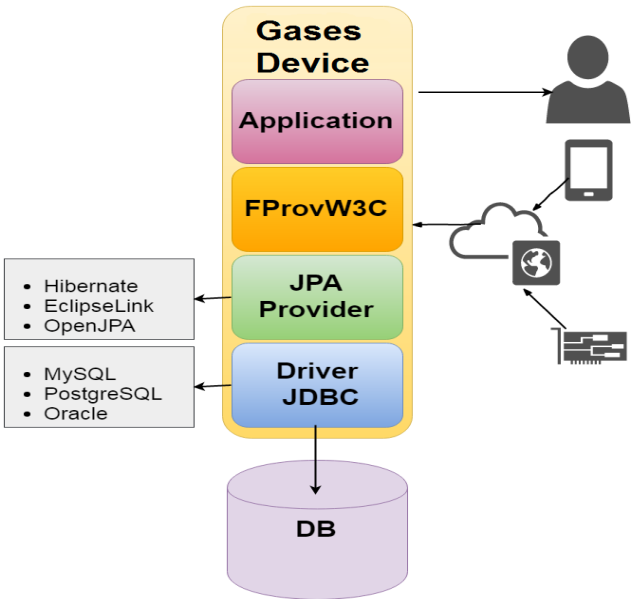



**Figura 8. Arquitetura do Gases Device com o FProvW3C.**

## 5.2 Resultados e Discussão

Os dados gravados deste aplicativo inclui informações de usuários do sistema, sensores, gases monitorados e agentes de software (Nascimento *et al.*, 2016). Os dados gravados foram vinculados às ações de usuários e agentes de software, auxiliando na rastreabilidade de cada ação executada. Por exemplo, a Figura 9 ilustra o momento de registo de um utilizador com obesidade no site Smell App.

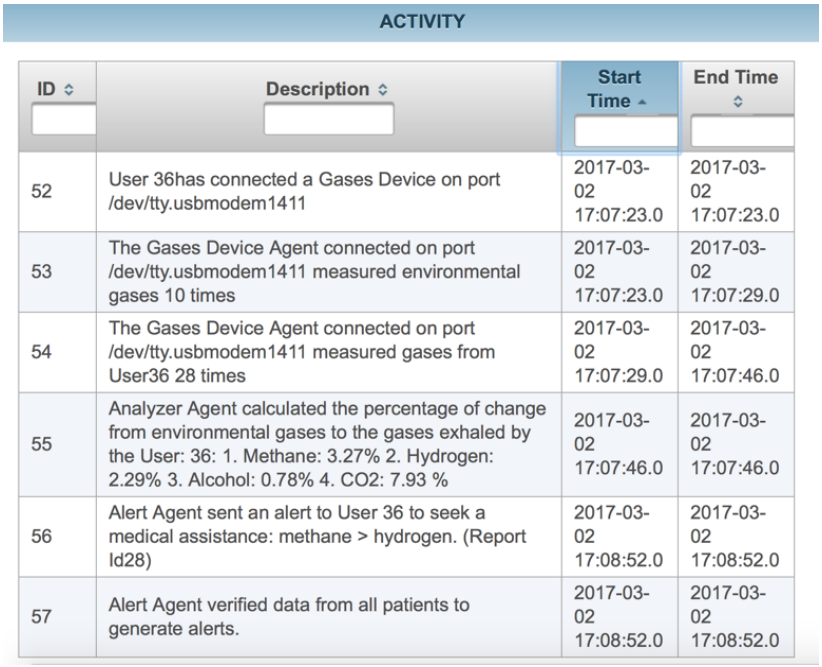
Para elaborar um banco de dados inicial para melhorar a classificação do sistema, os primeiros usuários foram solicitados informações de saúde pessoal antes de usar o Dispositivo de Gases. Conforme ilustrado na Figura 9, o sistema atribuiu o ID 36 a este utilizador registado. Em seguida, usando a estrutura do FProvW3C, foi possível rastrear todos os dados gerados durante esta operação de registro. Podemos verificar na Figura 10 que o agente de pessoa com ID de Usuário 36 foi criado com êxito, e a entidade "Obesidade" foi atribuída a este agente.


**Figura 9. Tela do Smell App.**


**Figura 10. Proveniência de dados no Smell App.**

Depois que o usuário se registrou, ele conectou um Dispositivo de Gases ao sistema e iniciou um relatório de expiração. Esta ação iniciou três agentes de software: (i) Gases Device Agent, que coletou dados do Arduino; (ii) Agente Analisador, que pré-processou e salvou dados no banco de dados, e (iii) Agente Alerta, que avaliou todos os relatórios de expiração com base nas doenças descritas em (Nascimento *et al.*, 2016) e alertas gerados.

Para avaliar a operação de um sistema multiagente não é uma tarefa trivial (Coelho *et al.*, 2007). Como os dados vêm de várias fontes, é difícil identificar a origem de um problema. No entanto, como mostrado na Figura 11, a proveniência facilitou a avaliação deste sistema multiagente, permitindo-nos rastrear todas as atividades que foram realizadas durante sua execução.



ACTIVITY

| ID | Description | Start Time | End Time |
|---|---|---|---|
| 52 | User 36has connected a Gases Device on port /dev/tty.usbmodem1411 | 2017-03-02 17:07:23.0 | 2017-03-02 17:07:23.0 |
| 53 | The Gases Device Agent connected on port /dev/tty.usbmodem1411 measured environmental gases 10 times | 2017-03-02 17:07:23.0 | 2017-03-02 17:07:29.0 |
| 54 | The Gases Device Agent connected on port /dev/tty.usbmodem1411 measured gases from User36 28 times | 2017-03-02 17:07:29.0 | 2017-03-02 17:07:46.0 |
| 55 | Analyzer Agent calculated the percentage of change from environmental gases to the gases exhaled by the User: 36: 1. Methane: 3.27% 2. Hydrogen: 2.29% 3. Alcohol: 0.78% 4. CO2: 7.93 % | 2017-03-02 17:07:46.0 | 2017-03-02 17:07:46.0 |
| 56 | Alert Agent sent an alert to User 36 to seek a medical assistance: methane > hydrogen. (Report Id28) | 2017-03-02 17:08:52.0 | 2017-03-02 17:08:52.0 |
| 57 | Alert Agent verified data from all patients to generate alerts. | 2017-03-02 17:08:52.0 | 2017-03-02 17:08:52.0 |

**Figura 11. Proveniência de dados no Smell App: Atividades dos Agentes de Software e dos Usuários.**

Esta seção mostra que a proveniência dos dados pode ser usada para uma ampla gama de propósitos em aplicações computacionais. Além disso, quando rastreamos a execução do sistema para entender sua operação, também usamos a proveniência de dados para verificar erros em dados classificados por agentes e em dados coletados de sensores.

## 6 Conclusão e Trabalhos Futuros

Registrando a proveniência de dados é uma necessidade em vários cenários, especialmente aqueles que têm execução complexa. É necessário ter um histórico de cada um dos passos. O modelo PROV tem por objetivo armazenar dados de procedência de forma detalhada, focalizando as responsabilidades dos agentes em cada item de proveniência.

Este trabalho propõe a estrutura FProvW3C, projetada para capturar e armazenar dados proveniência usando o modelo PROV do W3C. A proveniência dos dados ajuda a rastrear a origem dos dados e os processos de derivação que ocorreram entre a origem dos dados e o estado em que os dados são encontrados atualmente. Considerando que o modelo de proveniência contribui para avaliar a qualidade dos dados e consequentemente o processo que o gerou, isso ajuda a aumentar a validade das experimentos desde que os dados sejam monitorados.

Entre os trabalhos relacionados (ver Seção 3), podemos ver a preocupação com a representação dos dados, mas a forma de captura e armazenamento é omitido. Tanto a captura como o armazenamento são atividades críticas que permitem que os dados sejam reutilizados em novas pesquisas ou como uma ferramenta de validação.

A procedência dos dados pode ser utilizada na construção de bases de conhecimento que ajudem na: (i) rastreabilidade de ações; (ii) identificação de erros; (iii) acompanhamento das etapas de um estudo, e (iv) viabilidade de replicação para verificar os resultados.

A proveniência dos dados ainda é pouco explorada pela engenharia de software experimental e deve ser melhor analisada, pois pode fazer a diferença na comprovação dos resultados obtidos. Conforme apresentado por (Travassos *et al.*, 2002), fazer ciência sem validação empírica adequada pode criar estudos falhos e construir bases de conhecimento sem credibilidade.

Para futuros trabalhos, esperamos expandir o framework FProvW3C para que ele possa converter os dados capturados para uma ontologia seguindo o modelo PROV-O. Por exemplo, é possível estender FProvW3C para suportar: (i) semântica e sintaxe de dados, e (ii) a ontologia de PROV. Além disso, procuramos explorar a proveniência dos dados em sistemas multiagentes, registrando cada informação sobre o agente, suas relações com outros agentes e com o ambiente externo. Como tal, poderíamos capturar informações do processo de tomada de decisão do agente, expandindo o trabalho de Van *et al*. (1999) e poderíamos usar as informações obtidas para ajudar a rastrear erros e responder a perguntas sobre o comportamento do agente, uma vez que é uma entidade autônoma.

## Referências


BUNEMAN, Peter; KHANNA, Sanjeev; TAN, Wang-Chiew. **Data provenance: Some basic issues**. In: International Conference on Foundations of Software Technology and Theoretical Computer Science. Springer Berlin Heidelberg, 2000. p. 87-93.

COELHO, Roberta et al. **Jat: A test automation framework for multi-agent systems**. In: Software Maintenance, 2007. ICSM 2007. IEEE International Conference on. IEEE, 2007. p. 425-434.

CUNHA, Francisco et al. **JAT4BDI: An Aspect-Based Approach for Testing BDI Agents**. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on. IEEE, 2015. p. 186-189.

DALPRA, Humberto LO et al. **Using Ontology and Data Provenance to Improve Software Processes**. In: ONTOBRAS. 2015.

DAVIDSON, Susan B.; FREIRE, Juliana. **Provenance and scientific workflows: challenges and opportunities**. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008. p. 1345-1350.

DOUKAS, Charalampos. **Building Internet of Things with the ARDUINO**. CreateSpace Independent Publishing Platform, 2012.

GUBBI, Jayavardhana et al. **Internet of Things (IoT): A vision, architectural elements, and future directions**. Future generation computer systems, v. 29, n. 7, p. 1645-1660, 2013.

LEBO, Timothy et al. **Prov-O: The prov ontology**. W3C recommendation, v. 30, 2013.

LIM, Chunhyeok et al. **Prospective and retrospective provenance collection in scientific workflow environments**. In: (SCC), 2010 IEEE International Conference on Services Computing. IEEE, 2010. p. 449-456.

MARINHO, A.; WERNER, Cláudia Maria Lima; MURTA, Leonardo Gresta Paulino. **ProvManager: uma abordagem para gerenciamento de proveniência de workflows científicos**. In: Workshop de Teses e Dissertações em Engenharia de Software, XXIII SBES. 2009.

MATTSSON, Michael. **Evolution and composition of object-oriented frameworks**. 2000. Tese de Doutorado. Blekinge Institute of Technology.

MISSIER, Paolo; BELHAJJAME, Khalid; CHENEY, James. **The W3C PROV family of specifications for modelling provenance metadata**. In: Proceedings of the 16th International Conference on Extending Database Technology. ACM, 2013. p. 773-776.

MOREAU, Luc et al. **The open provenance model: An overview**. In: International Provenance and Annotation Workshop. Springer Berlin Heidelberg, 2008b. p. 323-326.

MOREAU, Luc et al. **The provenance of electronic data**. Communications of the ACM, v. 51, n. 4, p. 52-58, 2008a.

MOREAU, Luc; FOSTER, Ian. **Provenance and Annotation of Data: International Provenance and Annotation Workshop**, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers. Springer Science & Business Media, 2006.

NASCIMENTO, Nathalia Moraes; VIANA, Marx Leles; DE LUCENA, Carlos José Pereira. **An IoT-based Tool for Human Gas Monitoring**. XV Congresso Brasileiro de Informática em Saúde, 2016 p. 96-98.

SIRQUEIRA, Tássio F. M. et al. **E-SECO ProVersion: An Approach for Scientific Workflows Maintenance and Evolution**. Procedia Computer Science, v. 100, p. 547-556, 2016.

TAN, Wang Chiew. **Research problems in data provenance**. IEEE Data Eng. Bull., v. 27, n. 4, p. 45-52, 2004.

TRAVASSOS, Guilherme Horta; GUROV, Dmytro; AMARAL, E. A. G. G. **Introdução à engenharia de software experimental**. UFRJ, 2002.

VEREGIN, Howard; LANTER, David P. **Data-quality enhancement techniques in layer-based geographic information systems**. Computers, Environment and Urban Systems, v. 19, n. 1, p. 23-36, 1995.

VEREGIN, Howard; LANTER, David P. **Data-quality enhancement techniques in layer-based geographic information systems**. Computers, Environment and Urban Systems, v. 19, n. 1, p. 23-36, 1995.

WOHLIN, Claes et al. **Experimentation in software engineering**. Springer Science & Business Media, 2012.