Universidade Federal
do Rio de Janeiro
Escola Politécnica

PUBLICAÇÃO DE PROVENIÊNCIA DE WORKFLOWS NA WEB SEMÂNTICA

Rachel Gonçalves de Castro

Projeto de Graduação apresentado ao Curso de Engenharia de Computação e Informação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheira.

Orientador: Marta Lima de Queirós Mattoso
Renan Francisco Santos Souza

Rio de Janeiro
Março de 2015

PUBLICAÇÃO DE PROVENIÊNCIA DE WORKFLOWS NA WEB SEMÂNTICA

Rachel Gonçalves de Castro

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRA DE COMPUTAÇÃO E INFORMAÇÃO.

Examinada por:

_____

Profa. Marta Lima de Queirós Mattoso, D.Sc.

_____

Renan Francisco Santos Souza.

_____

Profa. Kary Ann del Carmen Soriano Ocaña, D.Sc.

_____

Prof. Alexandre de Assis Bento Lima, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
MARÇO de 2015

Castro, Rachel Gonçalves

Publicação de Proveniência de Workflows na Web Semântica / Rachel Gonçalves de Castro. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2015.

X, 66 p.: il.; 29,7 cm.

Orientadores: Marta Lima de Queirós Mattoso e Renan Francisco Santos Souza

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia de Computação e Informação, 2015.

Referências Bibliográficas: p. 55-57.

1. Web Semântica 2. Workflows Científicos 3. Proveniência de Dados I. Mattoso, Marta Lima de Queirós. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Computação e Informação. III. Título.

À minha família.

AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais, Mônica e Maurício, as pessoas mais importantes da minha vida, por serem a minha base e minha inspiração, sempre me proporcionando o melhor, desde a educação até o carinho e conforto em momentos difíceis. À minha irmã Rebeca, pelo amor e compreensão e por tornar os meus dias mais alegres e leves. À minha avó Dina, meu exemplo de vida e superação, por sempre me apoiar e incentivar. Aos meus familiares: minhas tias, meus tios e minha prima Gabriella, por sempre acreditarem em mim.

Ao meu namorado Bernardo, pela paciência, dedicação e carinho, e por acreditar em mim e me fazer uma pessoa melhor.

Aos meus amigos, por não me esquecerem, mesmo em períodos de ausência. Em especial, ao Renan Hozumi, pelas conversas nos almoços e por sempre me incentivar. Aos amigos de ECI, por tornarem esses cinco anos mais felizes, enfrentando as dificuldades das matérias com união e parceria. Terei um carinho eterno por vocês.

À professora Marta Mattoso, por aceitar ser minha orientadora e pelas aulas que me levaram a escolher o tema deste projeto. Ao professor Alexandre Assis e à professora Kary Ocaña, por participarem da apresentação do projeto. Ao Renan Souza, pelas reuniões que viabilizaram este projeto, pelo empenho em sempre responder meus email e revisar diversas vezes o texto, procurando sempre sanar as minhas dúvidas.

Aos professores em geral de ECI, por transmitirem seus conhecimentos, contribuindo para a minha formação. Em especial, gostaria de agradecer ao professor Henrique Cukierman, com que fiz iniciação científica por um ano e meio, por expandir a minha visão sobre o curso e proporcionar as minhas primeiras experiências acadêmicas extracurriculares.

Por fim, gostaria de agradecer a todos que participaram e contribuíram, mesmo que indiretamente, para a minha formação.

Resumo do Projeto de Graduação apresentação à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheira de Computação e Informação.

Publicação de Proveniência de Workflows na Web Semântica

Rachel Gonçalves de Castro

Março/2015

Orientadores: Marta Lima de Queirós Mattoso
Renan Francisco Santos Souza

Curso: Engenharia de Computação e Informação

Devido ao crescente uso de *workflows* científicos para a execução de simulações que exigem um alto desempenho computacional, o armazenamento da proveniência é essencial para garantir maior confiabilidade e reprodutibilidade do experimento executado através do *workflow*. Apesar dos dados de proveniência existirem pelo mundo, eles geralmente não são disponibilizados publicamente de forma estruturada ou padronizada, dificultando a interoperabilidade de dados científicos de diversas áreas do conhecimento. A publicação da proveniência na Web Semântica, uma tecnologia ainda relativamente pouco difundida, mas com um reconhecido e notório potencial, é uma alternativa que permite a publicação dos dados de forma estruturada, padronizada e aberta. Este projeto propõe uma ontologia para um Sistema de Gerência de *Workflows* Científicos (SGWfC) e um sistema que permite seleção e publicação dos dados de proveniência gerados pelo SGWfC na Web Semântica.

Palavras-chave: Web Semântica, *Workflows* Científicos, Proveniência de Dados.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Computer and Information Engineer.

Publication of Workflows Provenance in the Semantic Web

Rachel Gonçalves de Castro

March/2015

Advisors: Marta Lima de Queirós Mattoso
Renan Francisco Santos Souza

Major: Computer and Information Engineering

Due to the increasing use of scientific workflows to perform simulations that require high computational performance, storing provenance is essential to ensure greater reliability and reproducibility of the experiment executed using a workflow. Although provenance data exist around the world, they are usually not published in a structured or standardized form, which makes scientific data interoperability among different knowledge fields more difficult. Despite being relatively not so popular, the Semantic Web has a notorious potential and is an alternative to enable data publication on the web in a structured, standardized and open way. This project proposes an ontology for a Scientific Workflow Management System (SWfMS) and a system that enables the provenance data generated by the SWfMS to be selected and published on the Semantic Web.

Keywords: Semantic Web, Scientific Workflows, Provenance Data.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

AMS – Alinhamento Múltiplo de Sequências

CRUD – *Create, Read, Update and Delete*

ETC – Extração, Transformação e Carga

LOD – *Linked Open Data*

MER – *Molecular Evolution Reconstruction*

OPM – *Open Provenance Model*

OPMW – *Open Provenance Model for Workflows*

OWL – *Web Ontology Language*

RDF – *Resource Description Framework*

SGBD – Sistema de Gerenciamento de Banco de Dados

SGWfC – Sistema de Gerência de *Workflows* Científicos

SQL – *Structured Query Language*

URI – *Uniform Resource Identifier*

W3C – *World Wide Web Consortium*

XML – *Extensible Markup Language*

1. Introdução

A validação de experimentos científicos pode ser uma tarefa complexa para o cientista, necessitando da manipulação e análise de um grande volume de dados. O uso de simulações é um mecanismo bem utilizado nessa fase do experimento, porém pode ter uma complexidade computacional alta. Para facilitar o desenvolvimento da experiência nessa etapa, muitas vezes são usados *workflows* científicos, que são uma abstração para modelar um conjunto de tarefas executadas em uma ordem pré-determinada.

Para garantir a confiabilidade na execução e reprodutibilidade do experimento usando o *workflow*, normalmente são coletados dados de proveniência, permitindo que o cientista saiba todo o caminho percorrido até gerar o resultado final. Com essas informações, o cientista é capaz de descrever todo o fluxo de dados, facilitando a identificação e correção de erros na execução, além de uma análise mais confiável do resultado gerado.

Essa proveniência, porém, costuma ser armazenada em bancos de dados privados, onde apenas um conjunto restrito de pessoas tem acesso à informação. A troca de informações entre cientistas de áreas semelhantes ou que queiram compartilhar as informações de seus trabalhos é dificultada por essa estrutura fechada, inibindo uma análise mais integrada dos dados. A Web Semântica, então, pode ser uma alternativa para a publicação dos dados de proveniência de um *workflow*.

A Web Semântica surgiu a partir de um projeto do W3C (*Word Wide Web Consortium*), uma organização para a padronização do desenvolvimento da *Web*, cujo diretor é Tim Berners-Lee, muito conhecido como um dos percussores da *Word Wide Web* (Berners-Lee e Cailliau, 1990). O objetivo da Web Semântica é tornar os dados

disponíveis, não apenas para a comunicação entre humanos, mas também para a compreensão de máquinas, permitindo que computadores possam, além de consumir, gerar informações (Berners-Lee, 1998).

Os dados são interligados por relacionamentos, o que permite a atribuição de mais significado. Além disso, todos eles têm um formato em comum, facilitando a interoperabilidade e possibilitando a combinação e integração de dados. Idealmente, na Web Semântica as pessoas poderiam criar bancos de dados e vocabulários que permitiriam a construção de regras para analisar melhor os dados (World Wide Web Consortium, 2005).

Assim, a publicação de dados de proveniência de *workflow* na Web Semântica permite que os dados fiquem armazenados de forma estruturada e aberta. Dados estruturados são organizados em classes que compartilham as mesmas características, possibilitando a sua maior descrição e detalhamento. A abertura dos dados facilita a interação entre cientistas e possibilita uma troca de informações, como já foi falado anteriormente.

Já o acesso aos dados publicados na Web Semântica pode ser feito através de uma linguagem parecida com SQL (*Structured Query Language*), o SPARQL. A realização de consultas SPARQL possibilita o relacionamento de informações disponíveis dentro do mesmo escopo de maneira fácil. Assim, uma análise mais enriquecida pode ser realizada, sem a necessidade de despender um esforço grande para o aprendizado de uma nova linguagem.

O objetivo deste projeto é criar um mecanismo que permita a publicação de dados de proveniência de *workflows* executados no SciCumulus (Oliveira *et al*., 2010), um sistema de gerência de *workflows* científicos, na Web Semântica. Para isso, são necessários dois importantes passos: a construção da ontologia a ser utilizada no SciCumulus, a qual nomeamos de PROV-O-Wf, e a publicação dos dados de proveniência propriamente ditos.

Para a construção da ontologia, a ideia é desenvolver uma modelagem genérica, permitindo que qualquer *workflow*, independente da área do conhecimento, consiga ter seus dados de proveniência publicados. Para a publicação propriamente dita, é necessário realizar um processo de Extração, Transformação e Carga (ETC). Isto é, todas as informações contidas no banco de dados de proveniência original, inicialmente privado, devem ser extraídas. Em seguida, os dados precisam ser transformados no formato adequado para a Web Semântica e, por fim, devem ser inseridos no banco de dados semântico, onde deverão ser disponibilizados publicamente na Web Semântica.

Esta monografia está dividida em cinco capítulos, além dessa introdução. O Capítulo 2 tem o objetivo de apresentar os conceitos fundamentais para o total entendimento dos temas abordados. O Capítulo 3 apresenta a ontologia criada para o SciCumulus, bem como conceitos e padrões usados. O processo de extração das informações do banco de dados relacionais e inserção no banco de dados semântico é mostrado no Capítulo 4. O Capítulo 5 apresenta um estudo de caso, com uma avaliação experimental e, por último, o Capítulo 6 conclui o trabalho.

2. Fundamentação Teórica

Este capítulo tem o objetivo de apresentar alguns conceitos utilizados para a realização do projeto. A seção 2.1 apresenta o *workflow* científico e os seus principais componentes. Já a seção 2.2 mostra uma breve definição de Sistemas de Gerência de *Workflows* Científicos. Em seguida, proveniência de dados e suas principais características são apresentadas, além da importância do uso de um padrão para a modelagem de informações. Por último, é introduzida a Web Semântica e alguns conceitos necessários para o seu entendimento na seção 2.4.

2.1. *Workflow* Científico

Experimentos científicos consistem na observação de um fenômeno, através da análise de dados, e o uso dos resultados obtidos para provar ou refutar uma hipótese. Devido à necessidade de organizar, processar, controlar e analisar o experimento, sua representação é feita através de um ciclo cujos passos são composição, execução e análise. Um *workflow* científico é a abstração desse processo, que permite a especificação formal dos passos a serem executados em um experimento científico (Deelman *et al*., 2009).

Constituem um *workflow* científico dois componentes principais: atividades e relacionamentos. A atividade é a execução de um programa que consome e produz dados. O relacionamento é responsável pela definição do fluxo de dados. Através da especificação de um relacionamento, as dependências entre as atividades são definidas, garantindo que a saída de uma atividade seja a entrada de uma atividade dependente.

Uma atividade também pode ser um *workflow* completo. Este arranjo é conhecido como *sub-workflow* (Sousa, 2011).

A Figura 1 é a representação de um esquema de *workflow* (Sousa, 2011), que apresenta uma atividade A e um *sub-workflow* composto de três atividades, B_1, B_2 e B_3. A partir da imagem é possível observar a dependência entre a saída da atividade A e a entrada do *sub-workfow*, representada através do relacionamento entre eles.

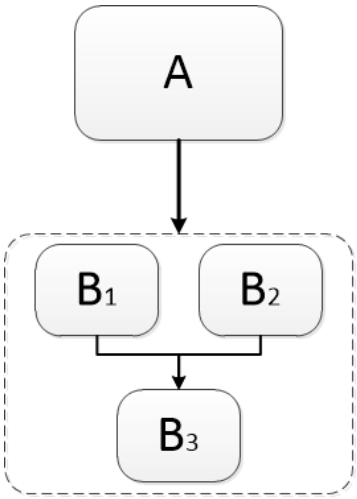

Figura 1. Esquema de um workflow científico. Adaptado de (Sousa, 2011).

2.2. Sistema de Gerência de *Workflows* Científico

Os *workflows* científicos são estruturados, executados e têm seus dados analisados através do suporte de um Sistema de Gerência de *Workflows* Científicos. Os SGWfC têm o objetivo de facilitar o trabalho do cientista, permitindo que foquem apenas na pesquisa e não nas questões computacionais (Deelman e Chervenak, 2008). .

Assim, os parâmetros que serão utilizados por um *workflow* específico, bem como a manipulação de dados, podem ser configurados a partir do SGWfC. Além disso, são capazes de coletar a proveniência dos dados e armazená-las para, posteriormente, serem

consultadas (Cruz *et al*, 2009). Alguns dos SGWfC mais conhecidos são: Taverna (Hull *et al*., 2006), Pegasus (Deelman *et al*., 2007) e VisTrails (Callahan *et al*., 2006).

2.3. Proveniência de Dados

Segundo Moreau e Groth (2013), proveniência é um registro que descreve pessoas, instituições, entidades e atividades que produzem, influenciam ou entregam um dado ou coisa. Pode ser considerado também um metadado que descreve a origem e todo caminho realizado para atingir os resultados de um experimento.

Por exemplo, ao comprar uma obra de arte, é importante saber a sua procedência, desde a sua criação, incluindo todos os antigos donos. Essa informação será essencial para estabelecer o valor da obra de arte. O mesmo acontece com dados. A proveniência é um mecanismo para garantir a qualidade e a veracidade dos dados (Simmhan *et al*., 2005). Além disso, a proveniência de dados é importante para: garantir que o experimento possa ser repetido, catalogar os resultados, evitar esforço duplicado e recuperar os dados de entrada a partir dos dados de saída (Buneman e Tan, 2007). Essa última vantagem pode ser realizada a partir de uma técnica conhecida como inversão e é útil quando alguns dados não estão disponíveis ou quando é muito custoso acessá-los.

De maneira geral, existem duas formas de proveniência: prospectiva e retrospectiva. No contexto de *workflow*, a proveniência prospectiva (Davidson e Freire, 2008) é a especificação da estrutura do *workflow*, ou seja, a instrução que deve ser seguida para atingir um resultado final. Essas informações podem ser as dependências entre os dados, os parâmetros de entrada, além das configurações do ambiente onde o experimento está sendo executado.

A proveniência retrospectiva está mais relacionada à execução, capturando informações dos passos que foram executados. Segundo Cruz *et al*. (2009), a proveniência retrospectiva pode ser considerada um *log* mais detalhado da execução de uma tarefa computacional. Exemplos de dados armazenados nesse caso são: tempo inicial e final de execução, possíveis erros que podem ter ocorrido e arquivos produzidos.

A utilidade da proveniência está diretamente relacionada à granularidade dos dados coletados (Simmhan *et al*., 2005). A granularidade é o nível de detalhamento dos dados capturados e pode ser classificada em dois tipos: grão fino e grão grosso. A proveniência de grão fino é a derivação de uma parte do resultado final (Buneman e Tan, 2007). Este tipo de informação é importante porque muitas vezes o resultado completo não está disponível e apenas uma parte do resultado pode ser de suma importância, além de ser também muito mais simples.

Já a proveniência de grão grosso, também conhecida como proveniência de *workflow*, é um histórico completo da derivação do resultado final. Esse tipo de proveniência pode incluir a interação humana e o uso de sensores e câmeras durante a execução do *workflow*.

Para garantir a interoperabilidade entre diferentes proveniências, é recomendado o uso de um modelo para representar as informações de proveniência. O uso de um padrão também permite que a proveniência seja independente da tecnologia usada pelos desenvolvedores (Moreau e Groth, 2013). Os dois principais modelos são o OPM[1] (*Open Provenance Model*) e o PROV[2].

[1] http://openprovenance.org/
[2] http://www.w3.org/TR/prov-overview/

No OPM, a proveniência é representada através de um grafo direcionado, possibilitando a identificação das dependências (Moreau *et al*., 2010). Os nós do grafo podem ser divididos em três categorias: artefato, processo e agente. O artefato é um "pedaço de estado imutável" (Moreau *et al*., 2010), o processo é uma ação que resulta ou é resultante de um artefato e agente é uma entidade capaz de gerar um processo. Já as arestas do grafo podem ser de cinco tipos: *used* e *wasGeneratedBy* relacionam um processo e um artefato, *wasControlledBy* liga um processo a um agente, *wasTriggeredBy* relaciona dois processos e *wasDerivedFrom* representa a dependência entre dois artefatos.

O PROV é um padrão muito similar ao OPM, criado pelo W3C. A diferença entre os dois é que o PROV não é estruturado na forma de um grafo, mas sim representado através de um modelo ER (Santos e Assis, 2013). É subdividido em doze documentos e os mais importantes são o PROV-DM (World Wide Web Consortium, 2013a), um padrão para modelo de dados para proveniência, e o PROV-O (World Wide Web Consortium, 2013b), usado para representação de ontologia.

O PROV-DM é o padrão definido para a representação de modelos de dados de proveniência. A sua estrutura principal é dividida em três partes: entidade, atividade e agente. A entidade é a "coisa" cuja proveniência se quer representar, podendo ser algo físico, digital ou conceitual (World Wide Web Consortium, 2013a). Atividade é um evento que ocorre durante um período de tempo, uma ação que pode gerar ou usar uma entidade. Por último, agente pode ser considerado o responsável pela existência de uma entidade, pela realização de uma atividade ou pela ação de outro agente.

As relações entre entidade, atividade e agente também são definidas pelo PROV. As entidades podem ser geradas por atividades (*WasGeneratedBy*), atribuídas a agentes (*WasAttributedTo*) ou derivar de outras entidades (*WasDerivedFrom*). Já as atividades

podem usar entidades (*Used*), estar associada a agentes (*WasAssociatedWith*) ou ser dependente de outras atividades (*WasInformedBy*). Os agentes, além dos relacionamentos já citados, podem ter responsabilidade sobre outros agentes (*ActedOnBehalfOf*). Essas são as relações mais básicas existentes entre os três elementos constituintes do PROV-DM.

O PROV-O é considerado uma especialização do PROV-DM para representação de ontologias. Possui as mesmas classes que o padrão para modelos de dados, assim como os relacionamentos citados acima. No PROV-O, as classes são representadas por retângulos, as entidades por elipses e os agentes por pentágonos, como é possível ver na Figura 2. Esta figura também está representando o tempo inicial e o final de uma atividade, duas propriedades pertencentes a essa classe.

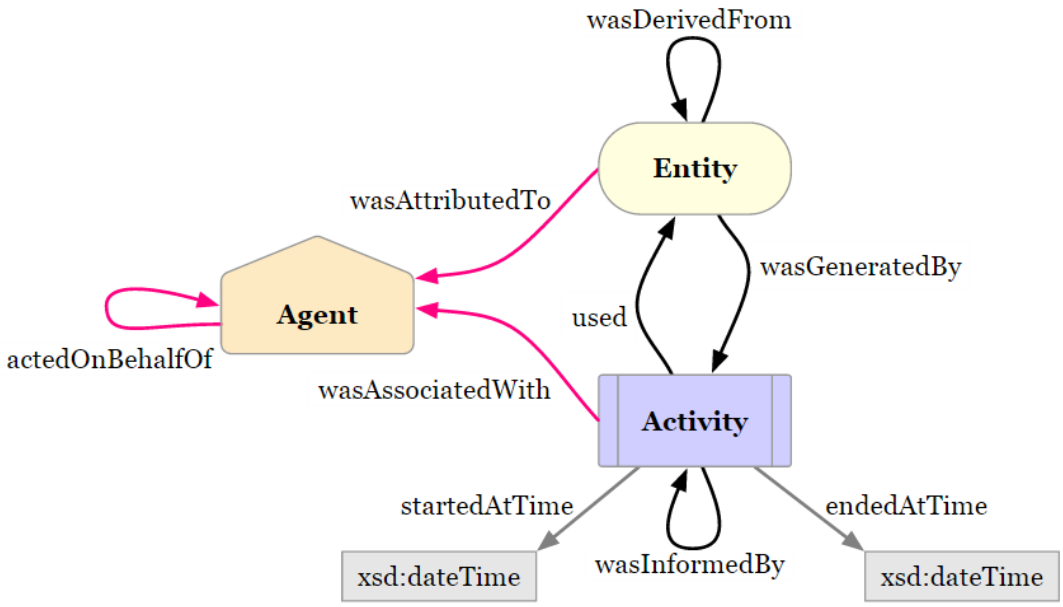

Figura 2. Diagrama com a principal estrutura do PROV-O. Retirado de (World Wide Web Consortium, 2013b).

2.4. Web Semântica

Em 1989, foi criado o *Word Wide Web*, mais comumente conhecido como *WWW* ou *Web*, um sistema acessado via Internet, cujo objetivo é o compartilhamento de informações. O principal nome dessa criação é Tim Berners-Lee, considerado o criador da Web devido a um conjunto de trabalhos que publicou. Posteriormente, como diretor do W3C foi responsável pelo projeto que fez surgir a Web Semântica, uma tecnologia que proporcionaria a facilidade de busca e correlação de informações (Berners-Lee, 1998).

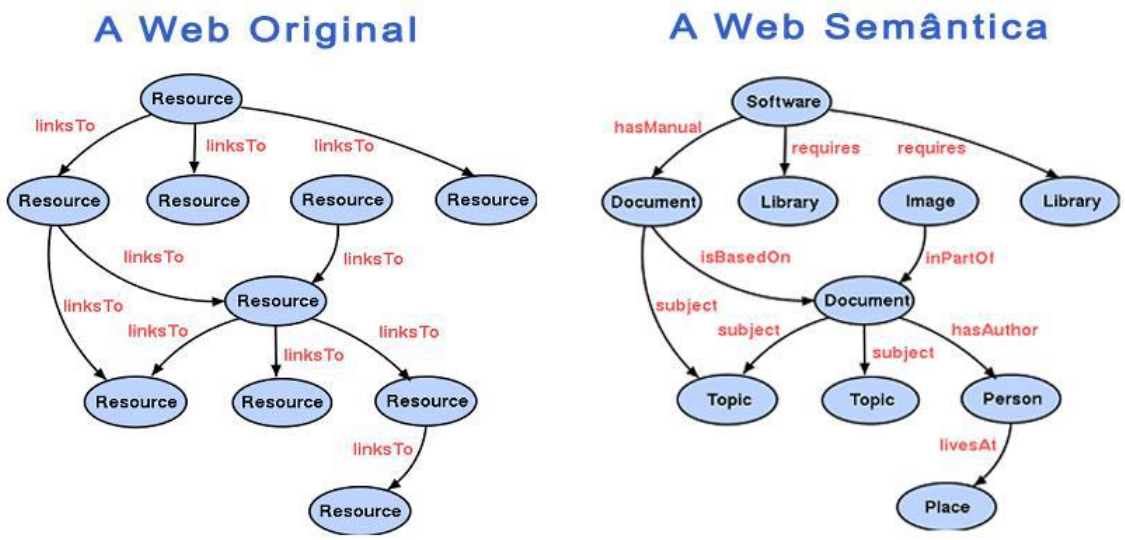

Figura 3. Comparação entre a Web Original e a Web de Dados[3].

A Web semântica é uma evolução da Web, que possibilita a comunicação não apenas entre humanos, mas também entre máquinas, permitindo que estas interpretem e gerem informações automaticamente. É também conhecida como "Web dos dados", termo utilizado pelo W3C, pois permite a criação de banco de dados com padrões e de

[3] http://ciencialaultima.blogspot.com.br/2012/12/web-30.html

forma estruturada, que facilitam a ligação dos dados. É possível observar na Figura 3 que a Web Semântica adiciona contexto às informações, associando metadados e relacionamentos.

A seguir serão apresentados alguns conceitos relacionados à Web Semântica que são importantes para o desenvolvimento do projeto.

2.4.1. Ontologia

Gruber (1993) define ontologia como "uma especificação explícita de uma conceituação". De forma mais clara, a ontologia é a forma utilizada para caracterizar o domínio que está sendo estudado através de um vocabulário controlado que reúne conceitos e relacionamentos. O conjunto de vocabulários pode ser organizado de forma hierárquica conhecida como taxonomia.

Já para a Web Semântica, a ontologia é usada para acrescentar mais significado e inteligência nas relações entre os recursos presentes na Web. Esses recursos são identificados por uma URI (*Uniform Resource Identifier*), isto é, um conjunto de caracteres que deve ser único em toda a Web Semântica. O uso de prefixos (ou *Namespaces*) também é muito comum. Eles são como um nome simplificado e menor para as URIs, que podem ser grandes e difíceis de representar.

É possível fazer uma analogia com a área de banco de dados. Da mesma forma que se cria um modelo de dados para um banco de dados relacional, é necessária a modelagem da ontologia para a criação do banco de dados semântico. Existe, porém, uma diferença entre uma ontologia e um modelo Entidade-Relacionamento: a flexibilidade (Souza, 2013). É geralmente mais simples e menos custoso modificar a estrutura de um modelo de ontologia, através da adição ou remoção de entidades na Web Semântica.

Os elementos básicos de uma ontologia são as classes e as propriedades.

- Classe: é o elemento mais básico de uma ontologia, usada para agrupar recursos com características similares. A instância de uma classe é chamada de indivíduo. Além disso, uma classe pode herdar de uma superclasse, sendo chamada então de subclasse. Ou seja, uma classe mais genérica (classe "mãe") pode ter subclasses (classes "filhas") mais específicas, que herdam as propriedades da classe "mãe".
- Propriedade: é a característica de uma classe. Assim como as classes, também é possível observar hierarquia entre as propriedades e as subpropriedades. Existem três tipos de propriedade, a de objetos, a de dados e as anotações. A propriedade de objetos é o relacionamento entre duas classes. Já a propriedade de dados relaciona uma classe a um valor. Esse valor também é conhecido como literal e pode ser uma *string*, um inteiro, entre outros. Por último, as anotações são menos importantes. São utilizadas apenas para fazer comentários, permitindo que a ontologia fique mais fácil de ser entendida.

Para descrever os elementos da ontologia no *Word Wide Web*, o W3C recomenda o uso da linguagem OWL (*Web Ontology Language*). Nesta linguagem, todos os indivíduos de uma ontologia são considerados subclasses da classe `owl:Thing` (World Wide Web Consortium, 2004). Já as classes são definidas como `owl:Class` e as propriedades de dados e de objetos como `owl:DatatypeProperty` e `owl:ObjectProperty` respectivamente, onde owl: é um prefixo para `<http://www.w3.org/2002/07/owl#>`.

2.4.2. RDF

Assim como OWL, o RDF (*Resource Description Framework*) também é uma recomendação do W3C. Neste caso, porém, é considerado um padrão para descrição e troca de instâncias e relacionamento dos recursos na web semântica (Souza, 2013).

Dois conceitos importantes para o RDF são recurso e literal. Recurso é definido como uma entidade na web de dados que pode ser identificada, nomeada ou endereçada (Souza, 2013). Já literal é o valor que uma propriedade pode assumir, podendo ser de diversos tipos. O mais comum é o uso dos tipos primitivos de dados. No exemplo que será apresentado mais abaixo, *#spiderman* é um recurso que possui o nome "Spiderman", um literal do tipo *string*.

Um grafo em RDF é definido como um conjunto de triplas. A tripla nada mais é que uma estrutura com sujeito, predicado e objeto, como é possível observar na Figura 4. Na Web Semântica, o sujeito é a representação de um recurso, o predicado é a propriedade que relaciona o sujeito ao objeto e o objeto pode ser um recurso, se for uma propriedade de objeto, ou um literal, caso a propriedade seja de dados.

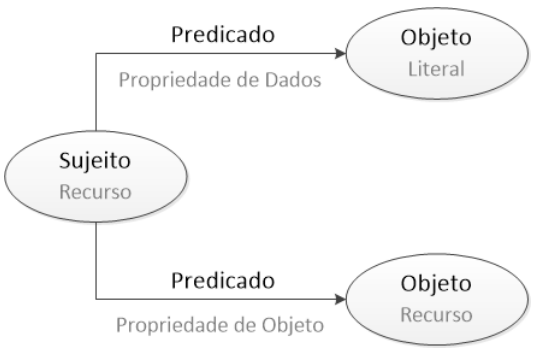

Figura 4. Tripla RDF

Para que um conjunto de triplas seja entendido por um computador, é necessário escrevê-lo em algum formato. Os mais usados são: RDF/XML, Turtle e N-Triples. O

primeiro é o formato mais conhecido e segue as mesmas regras de um XML (*Extensible Markup Language*), linguagem já bem difundida.

O Turtle (*Terse RDF Triple Language*) é um formato de fácil leitura para os humanos (Roberts, 2012), além de ocupar menos espaço de armazenamento em disco em relação aos outros citados, se for armazenado como texto plano (Souza, 2013). Como é possível ver no exemplo (World Wide Web Consortium , 2014) abaixo, toda tripla deve terminar com '.' e a separação entre sujeito, predicado e objeto é feita através de um espaço em branco. O ponto e vírgula representa que o sujeito é o mesmo, porém o restante da tripla varia. Por último, a vírgula separa dois objetos que possuem o mesmo sujeito e o mesmo predicado.

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#spiderman>
    rel:enemyOf <#green-goblin> ;
    a foaf:Person ;
    foaf:name "Spiderman", "Человек-паук"@ru .
```

Por último, o N-Triples é um formato mais simplificado do Turtle e cada tripla é representada em uma linha. Assim como o Turtle, as triplas também devem terminar com um ponto e a separação entre os elementos é feito através de um espaço em branco. No entanto, não é possível usar prefixos e sujeito e predicado devem ser repetidos mesmo que permaneçam os mesmos. É por isso que o Turtle é considerado um formato compacto, ao contrário do N-Triples.

```
    <http://example.org/#spiderman>
<http://www.perceive.net/schemas/relationship/enemyOf>
<http://example.org/#green-goblin> .
    <http://example.org/#spiderman>
<http://xmlns.com/foaf/0.1/name> "Spiderman".
```

2.4.3. SPARQL

É uma linguagem de consultas para RDF, recomendada pelo W3C desde 2008. Como um banco de dados semântico armazena triplas, o SPARQL é usado para realizar consultas nestes bancos de dados, da mesma forma que a linguagem SQL é usada com banco de dados relacionais.

A semelhança com SQL é evidente. As palavras chaves, bem como a estrutura e as funções de uma consulta em SPARQL são bem parecidas (Souza, 2013), facilitando o uso por pessoas que já dominam o uso do SQL em banco de dados relacionais.

Assim como as triplas, uma consulta SPARQL possui sujeito, predicado e obejeto, porém cada um dos três elementos pode ser uma variável. Ao substituir qualquer das variáveis por um valor, a consulta retornará as triplas que possuem o valor determinado. Caso nenhum valor seja especificado, todas as triplas presentes no banco de dados serão retornadas, como aconteceria no exemplo abaixo, caso não fosse especificado um limite de triplas. A cláusula `LIMIT` restringe o número de soluções retornadas pela consulta para vinte.

```
SELECT ?a ?b ?c
WHERE { ?a ?b ?c .}
LIMIT 20
```

Para realizar consultas buscando por literais, basta substituir o objeto pelo valor procurado. É possível também especificar apenas parte de um literal, quando este é do tipo *string*, através da cláusula `FILTER regex`. Por exemplo, na consulta abaixo, todas as triplas que possuem a palavra "web" no objeto serão retornadas. O "i" no final indica que letras maiúsculas e minúsculas são consideradas a mesma.

```
SELECT ?a ?b ?c
WHERE { ?a ?b ?c .
        FILTER regex(?c, "web", "i")}
```

Quando literal é um valo numérico, é possível restringir os valores através de uma expressão aritmética (World Wide Web Consortium, 2008). Para isso, é necessário usar a cláusula `FILTER`, como no exemplo anterior, porém sem `regex`. Por exemplo, para buscar por literais com valores menores do que 10, bastaria substituir a restrição "web" do exemplo acima por `?c < 10`.

```
PREFIX property: <http://worldbank.270a.info/property/>
PREFIX indicator: <http://worldbank.270a.info/classification/indicator/>
PREFIX sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#>
PREFIX sdmx-measure: <http://purl.org/linked-data/sdmx/2009/measure#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?CountryName ?Time ?Value WHERE {
        ?Analysis property:indicator indicator:GB.XPD.RSDV.GD.ZS  .
        ?Analysis sdmx-dimension:refArea ?Country .
        ?Country skos:prefLabel ?CountryName .
        ?Analysis sdmx-measure:obsValue ?Value .
        ?Analysis sdmx-dimension:refPeriod ?Time .
}
ORDER BY ?CountryName ?WBTime
```

É possível também realizar consultas mais estruturadas na Web Semântica. O exemplo acima apresenta uma consulta SPARQL sobre os dados do *World Bank*[4], disponibilizados em RDF. Ela recupera a porcentagem do PIB de cada país que é investida em Ciência e Tecnologia ao longo dos anos.

Essa consulta irá retornar um conjunto de triplas que informa o nome do país, o ano e o valor do investimento. Dentro da cláusula `WHERE`, a primeira linha define o indicador que será consultado, ou seja, a porcentagem do PIB investida em Ciência e Tecnologia. Em seguida, seleciona as regiões e o nome dos países das regiões, os valores do investimento e, por último, os tempos. É possível visualizar o resultado da consulta apresentada acima. Para isso, basta entrar no *endpoint*[5] do *World Bank*. A Figura 5 apresenta apenas uma amostra das triplas retornadas pela consulta.

CountryName	Time	Value
"Albania" @en	<http://reference.data.gov.uk/id/year/2007>	"0.08735" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Albania" @en	<http://reference.data.gov.uk/id/year/2008>	"0.15308" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Algeria" @en	<http://reference.data.gov.uk/id/year/2001>	"0.22846" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Algeria" @en	<http://reference.data.gov.uk/id/year/2002>	"0.36452" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Algeria" @en	<http://reference.data.gov.uk/id/year/2003>	"0.19578" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Algeria" @en	<http://reference.data.gov.uk/id/year/2004>	"0.16417" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Algeria" @en	<http://reference.data.gov.uk/id/year/2005>	"0.0666" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/1996>	"0.41749" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/1997>	"0.41959" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/1998>	"0.41131" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/1999>	"0.45337" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2000>	"0.43884" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2001>	"0.42461" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2002>	"0.38886" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2003>	"0.41013" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2004>	"0.43756" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2005>	"0.46077" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2006>	"0.49462" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2007>	"0.50793" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2008>	"0.52381" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2009>	"0.5951" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Argentina" @en	<http://reference.data.gov.uk/id/year/2010>	"0.61745" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Armenia" @en	<http://reference.data.gov.uk/id/year/1997>	"0.18692" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Armenia" @en	<http://reference.data.gov.uk/id/year/1998>	"0.22601" ^^<http://www.w3.org/2001/XMLSchema#decimal>
"Armenia" @en	<http://reference.data.gov.uk/id/year/1999>	"0.214" ^^<http://www.w3.org/2001/XMLSchema#decimal>

Figura 5. Parte do resultado da consulta SPARQL

[4] http://www.worldbank.org/
[5] http://worldbank.270a.info/sparql

3. PROV-O-Wf

O processo de publicação de dados na Web Semântica exige a realização de uma série de etapas. Para a publicação dos dados proveniência de *workflows* deste projeto, foi usado o esquema desenvolvido por Souza (2013) para a publicação de *Linked Open Data* (LOD) ou, em português, Dados Abertos Interligados. LOD é a maneira como os dados são publicados na Web Semântica.

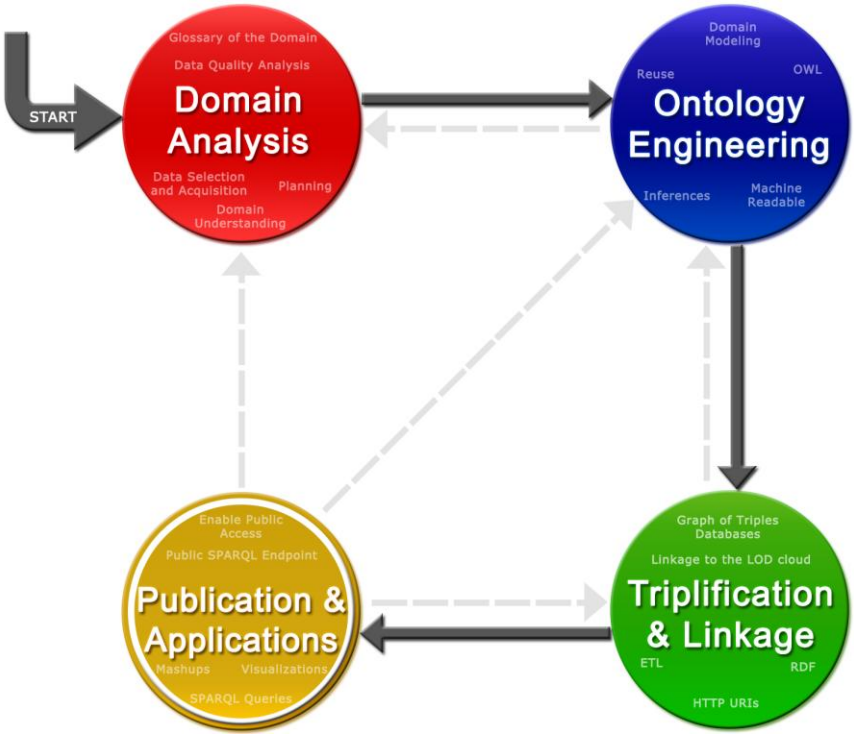

Figura 6. Processo de publicação de LOD. Retirado de (Souza *et al.*, 2014)

O processo de publicação é dividido em quatro fases, como é possível observar na Figura 6. A primeira etapa é a Análise de Domínio, onde o domínio dos dados é definido, seguida pela Engenharia da Ontologia, etapa onde é realizada a modelagem da ontologia de acordo com o domínio estabelecido. A triplificação é o processo de

instanciação dos dados utilizando a ontologia construída. Nesta etapa, triplas RDF são criadas e inseridas no banco de dados semântico. Na última etapa, Acesso Público aos Dados, já é possível realizar consultas SPARQL, pois os dados já foram publicados e estão disponíveis para acesso.

Para a primeira etapa do processo, foi realizado um estudo dos principais conceitos envolvidos no projeto: *workflow* e proveniência. Assim, a partir da análise de domínio, foi possível verificar que o domínio deste projeto são os dados de proveniência de *workflows* executados no SciCumulus. Assim, é essencial entender a estrutura do banco de dados que armazena as informações de proveniência do SciCumulus.

3.1. SciCumulus

O SciCumulus é um SGWfC responsável pela distribuição, controle e monitoração de execuções paralelas das atividades de um *workflow* científico, ou do *workflow* como um todo (Oliveira *et al*., 2010). Possui um mecanismo de coleta de dados de proveniência em tempo real, permitindo um agrupamento de execuções do *workflow* que podem ser utilizados para realizar consultas e análises.

Cada *workflow* é composto por um conjunto de atividades. As atividades possuem relações de entrada e de saída, sendo que a saída de uma pode ser a entrada de outra, constituindo a noção de dependência entre as atividades. Cada relação possui campos que recebem valores de entrada, podendo ser do tipo *float*, *string*, entre outros, ou arquivos que armazenam informações lidas por determinada atividade. Além dessa estrutura básica, o SciCumulus possui uma peculiaridade: a ideia de ativação. Cada atividade é composta por um conjunto de ativações que podem ser executadas em máquinas diferentes.

A ativação é um objeto que contem todas as informações necessárias para executar uma atividade em algum núcleo de um processador. Pode ser dividida em três etapas: instrumentação, invocação e extração (Ogasawara *et al.*, 2011). A primeira é responsável pela extração dos valores de entrada que serão usados na invocação. A segunda inicia a execução do programa e a monitora e a última extrai os valores do programa de saída e constrói o resultado que será retornada pela ativação.

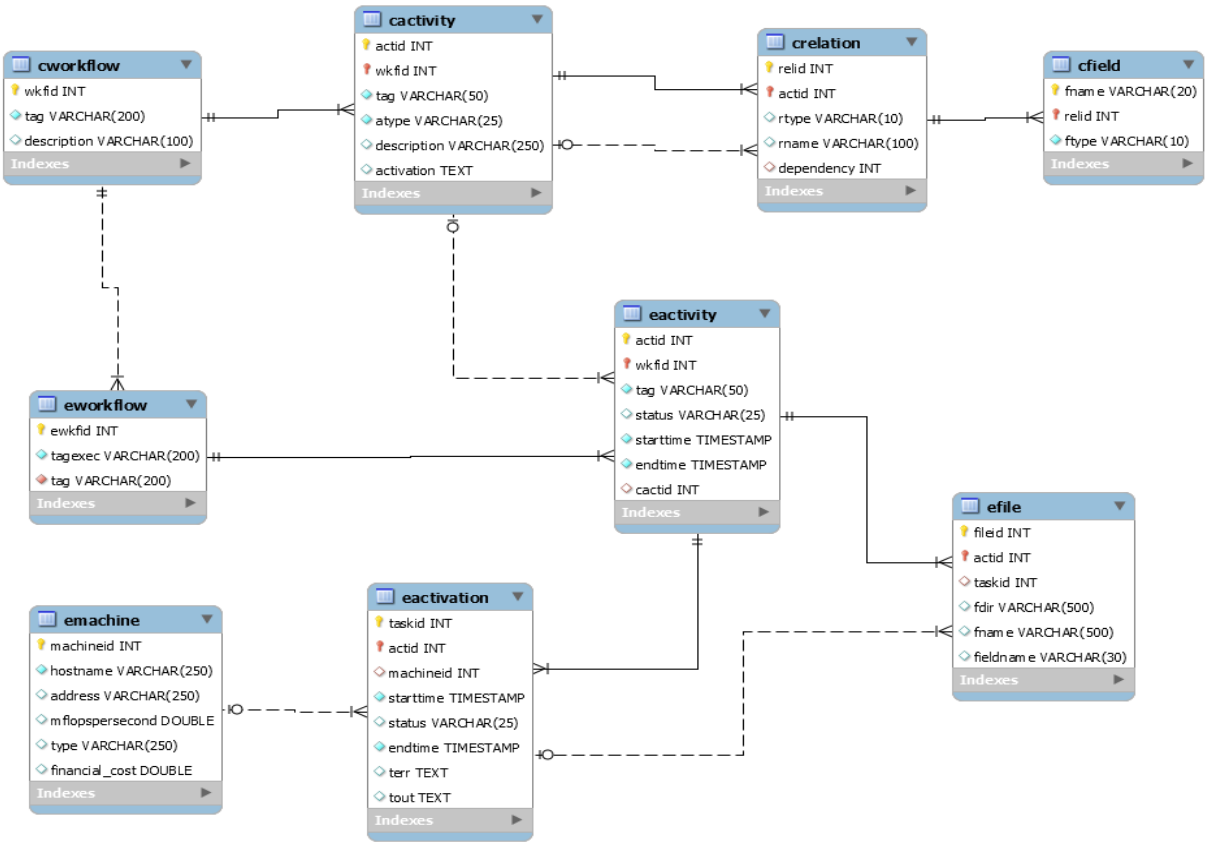

Figura 7. Modelo de dados simplificado do SciCumulus

A Figura 7 apresenta o modelo de dados simplificado da base de proveniência do SciCumulus. As tabelas presentes no modelo podem ser divididas em dois grupos: as que apresentam informações conceituais, localizadas na parte superior da figura, e as que armazenam informações de execução, localizadas na parte inferior. O primeiro

grupo é composto por *cworkflow*, *cactivity*, *crelation* e *cfield*. Já o segundo grupo possui as tabelas *eworkflow*, *eactivity*, *eactivation*, *efile* e *emachine*. A seguir será apresentada uma breve descrição do papel de cada uma destas tabelas.

- *cworkflow*: apresenta as informações mais básicas do *workflow*, como o nome e a descrição.
- *cactivity*: armazena todas as atividades que compõem cada *workflow* definido na tabela acima. Além das informações básicas como, por exemplo, o nome das atividades, esta tabela também informa o tipo de operação que cada atividade realiza, podendo ser *Map*, *SplitMap*, *Reduce*, *Filter*, *SRQuery* ou *JoinQuery* (Ogasawara *et al*., 2011).
- *crelation*: armazena as relações de entrada e de saída de cada atividade. É nesta tabela também que a dependência entre as atividades é definida.
- *cfield*: apresenta os campos de cada relação, definindo o seu nome e qual será o tipo de informação que cada um receberá.
- *eworkflow*: é a primeira tabela que irá armazenar informações sobre a execução. Todos os *workflows* que forem executados devem ser antes definidos na tabela *cworkflow*.
- *eactivity*: exibe as informações relacionadas a cada execução das atividades, que já foram definidas na tabela conceitual. Entre as informações mais relevantes está o tempo inicial e final da execução e o status, que permite saber se a atividade está executando, já finalizou ou se ocorreu algum problema.
- *eactivation*: como já foi explicado, assim como o *workflow* tem uma série de atividades, elas, por sua vez, também têm uma conjunto de ativações.

Nessa tabela estão armazenadas todas as ativações de cada atividade. É possível saber o tempo de inicio e fim, o status, além da máquina que determinada ativação foi executada e uma breve descrição do erro, caso ocorra algum problema.

- *efile*: como os campos de uma relação de entrada ou de saída podem ser um arquivo, é necessário armazenar algumas informações sobre eles. Assim, para cada ativação de uma atividade, as informações sobre o nome do arquivo, diretório e seu tamanho, por exemplo, são guardadas nessa tabela.
- *emachine*: armazena informações das máquinas do *cluster* nas quais as ativações são executadas.

Além das tabelas detalhadas acima, existem as tabelas de domínio de cada *workflow*, que estão definidas em um esquema separado, com o nome do *workflow*. Dentro desse esquema, há uma tabela para cada relação especificada em *crelation*, e suas colunas são, entre outras, os campos definidos na tabela *cfield*. Para cada ativação, os campos podem assumir valores diferentes, que são armazenados nesta tabela. Caso o campo seja do tipo *file*, há apenas o caminho do arquivo, sendo as outras informações guardadas na tabela *efile*. Esses dados de domínio são de grande importância para a análise do cientista. São neles que estão representados os resultados gerados pelo *workflow* e usados durante a execução. Logo, é onde estão contidas as informações mais valiosas e de maior interesse para a análise do cientista.

3.2. Modelagem da Ontologia

O segundo passo no processo de publicação de LOD é a engenharia da ontologia. Segundo Souza (2013), o reúso é uma característica muito importante nesta etapa. Existem conceitos que são gerais e estão presentes em diversos domínios. Logo, a chance de alguém já ter modelado o domínio em questão, ou pelo menos parte dele, é bem grande. Dessa forma, é muito importante a realização de uma pesquisa à procura de ontologias já existentes, relacionadas ao que está sendo modelado. Caso já exista alguma e seja adequada, recomenda-se utilizar o que foi encontrado, fazendo apenas modificações ou acréscimos, caso necessário.

Outro motivo para o reúso é a ideia de comunidade (Souza, 2013). Reutilizar um modelo já existente permite que um padrão seja mantido e difundido. A padronização também facilita a integração e interoperabilidade entre os dados na Web Semântica, permitindo que dados interligados sejam mais facilmente identificados e relacionados.

Assim, para a modelagem da ontologia do SciCumulus, primeiramente foram realizadas pesquisas a cerca do domínio, ou seja, sobre proveniência de *workflows*. Algumas ontologias foram encontradas ao realizar a busca e, após análise e eliminação das que modelavam *workflows* de uma área específica, foram selecionadas duas: PROV-O e OPMW (*Open Provenance Model for Workflows*).

A ontologia PROV-O, já explicada no Capítulo 2, será usada como um metamodelo[6] para a PROV-O-Wf, a ontologia que proporemos. Ademais, o PROV-O é considerado uma referência para a modelagem de proveniência e é uma recomendação do W3C. Assim, as principais classes do PROV-O, entidade, atividade e agente, serão consideradas características das classes da ontologia PROV-O-Wf.

[6] Metamodelo pode ser entendido como um modelo de dados de nível mais alto de abstração para descrever outro modelo de dados, de nível inferior.

Já o OPMW é exatamente o que este projeto visa, ou seja, uma ontologia para descrever proveniência de *workflows*, baseada no OPM. Além de ter seus fundamentos no OPM, o OPMW foi modificado, após o surgimento do PROV, para estendê-lo (Garijo *et al*., 2014a). Como é possível observar na Figura 8, também é baseado na ontologia P-Plan, uma especificação do PROV.

No PROV, *Plan* é uma entidade utilizada para representar um conjunto de ações ou passos de um ou mais agentes para atingir um objetivo (World Wide Web Consortium, 2013b). No P-Plan, ontologia criada para detalhar melhor a entidade do PROV, há três classes principais: *Plan*, *Step* e *Variable*. A primeira é uma subclasse do prov:Plan, enquanto que a segunda representa a execução das atividades planejadas (Garijo et *al*., 2014b). Por último, a classe *Variable*, identifica as variáveis de entrada e de saída das atividades do plano.

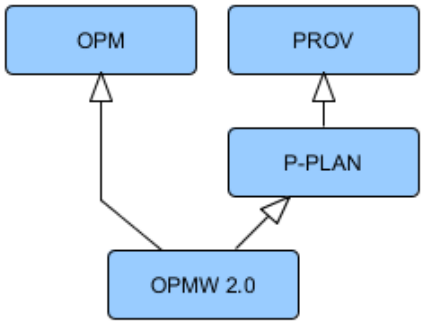

Figura 8. Extensões do OMPW. Retirado de (Garijo *et al*., 2014a).

3.2.1. OPMW

O OPMW possui seis classes, que serão descritas abaixo (Garijo *et al*., 2014a).

- *WorkflowTemplate*: é uma visão genérica do *workflow*, representando o seu projeto, que é instanciado em cada execução.

- *WorkflowTemplateProcess*: representa uma abstração das etapas de execução de um *workflow*, descrevendo os método utilizados pelo cientista. Estende *Step* de P-Plan.
- *WorkflowTemplateArtifact*: explica que tipo de artefato é usado ou gerado por *WorkflowTemplateProcess*. Estende *Variable* de P-Plan e pode ser dividido em dois tipos: *DataVariable*, que representa uma variável na especificação do *workflow*, e *ParameterVariable*, que descreve um parâmetro de entrada de uma etapa do *workflow*.
- *WorkflowExecutionAccount*: agrupa todas as declarações de execuções, representando a execução a partir da perspectiva do sistema. Estende *Account* do OPM e *Bundle*[7] do PROV.
- *WorkflowExecutionProcess*: representa a execução de uma etapa do *workflow*, definida em *WorkfowTemplateProcess*, descrevendo o método utilizado para realizar a tarefa. Estende processo do OPM e atividade do PROV.
- *WorkflowExecutionArtifact*: utilizada para representar um recurso usado ou gerado pela execução de um *workflow*. Estende artefato do OPM e entidade do PROV.

Como é possível perceber, assim como o SciCumulus, o OPMW também divide a proveniência do *workflow* em dois tipos: uma parte relacionada às informações conceituais (*template*) e uma que aborda as características da execução. Devido às diversas semelhanças, para a construção da ontologia foram utilizadas as classes

[7] *Bundle* é uma entidade utilizada para descrever um conjunto de proveniências. Logo, é a maneira utilizada para representar a proveniência da proveniência. O *Account* tem a mesma finalidade, sendo um artefato do OPM.

definidas no OPMW e algumas foram acrescentadas para englobar todos os conceitos necessários.

Tabela 1. Mapeamento do SciCumulus nas classes do OPMW

OPMW	SciCumulus
WorkflowTemplate	*cworkflow*
WorkflowTemplateProcess	*cactivity*
DataVariable	-
ParameterVariable	*cfield*
WorkflowExecutionAccount	*eworkflow*
WorkflowExecutionProcess	*eactivity*
WorkflowExecutionArtifact	*domain data*

Para construir o mapeamento mostrado na Tabela 1, observou-se as definições de cada classe do OPMW e as comparou às descrições das tabelas do SciCumulus. A primeira classe, *WorkflowTemplate*, pode ser usada como a representação de *cworkflow*, pois esta tabela representa o conceito de um *workflow* e todas as execuções devem estar associadas a este modelo.

As etapas descritas pela classe *WorkflowTemplateProcess* podem ser consideradas atividades definidas para um *workflow*, pois também representam componentes de um *workflow* que têm suas execuções atreladas à execução do *workflow*. Assim, podemos dizer que *WorkflowTemplateProcess* representa a tabela *cactivity*, pois ambas têm um objetivo conceitual. Já as relações de entrada e saída das atividades possuem uma série de campos, que também podem ser chamados de parâmetros. A classe *ParameterVariable*, um tipo de *WorkflowTemplateArtifact*, representa exatamente os campos de uma relação, ou seja, pode ser associada à tabela *cfield*. Não foi possível

encontrar, no entanto, nenhuma tabela do SciCumulus similar à classe *DataVariable*, logo ela não foi utilizada na ontologia.

Em relação à parte de execução, a tabela *eworkflow* armazena todas as execuções dos *workflows*, da mesma forma que a classe *WorkflowExecutionAccount*. Já *WorkflowExecutionProcess* também descreve as etapas de um *workflow*, exatamente como a classe *WorkflowTemplateProcess*, porém voltada para a execução. Logo, pode ser associada à tabela *eactivity*. Por último, *WorkflowExecutionArtifact* representa um recurso gerado ou consumido. No SciCumulus, os valores dos campos de entrada e saída do *workflow* são armazenadas nas tabelas de domínio. Assim, esta classe pode representar os valores dos campos armazenados nas tabelas de domínio.

3.2.2. PROV-Wf

Na seção 3.2.1 foram apresentadas as duas ontologias usadas para a construção do PROV-O-Wf. Além dessas ontologias, também foi utilizado como base para o seu desenvolvimento o PROV-Wf, um modelo de dados para proveniência retrospectiva apresentado por Costa *et al*. (2013), que utiliza o PROV-DM como padrão.

O fato do PROV-Wf utilizar o PROV-DM permitiu um mapeamento para o PROV-O de maneira rápida e simples. Apesar dessa vantagem, algumas modificações foram realizadas para possibilitar a utilização das classes do OPMW.

Como é possível observar na Figura 9, o modelo de dados está dividido em três partes: os componentes de cor branca representam a estrutura do *workflow*, os de cor cinza escuro são a execução e os de cinza claro representam a configuração do ambiente (Costa *et al*, 2013). Todos os elementos também estão identificados com os tipos do PROV, localizados na parte superior de cada classe. Além de entidades, planos,

atividades e agentes, um apresenta o tipo *SoftwareAgent*. Esse último é uma subclasse de agente que representa a execução de um *software*.

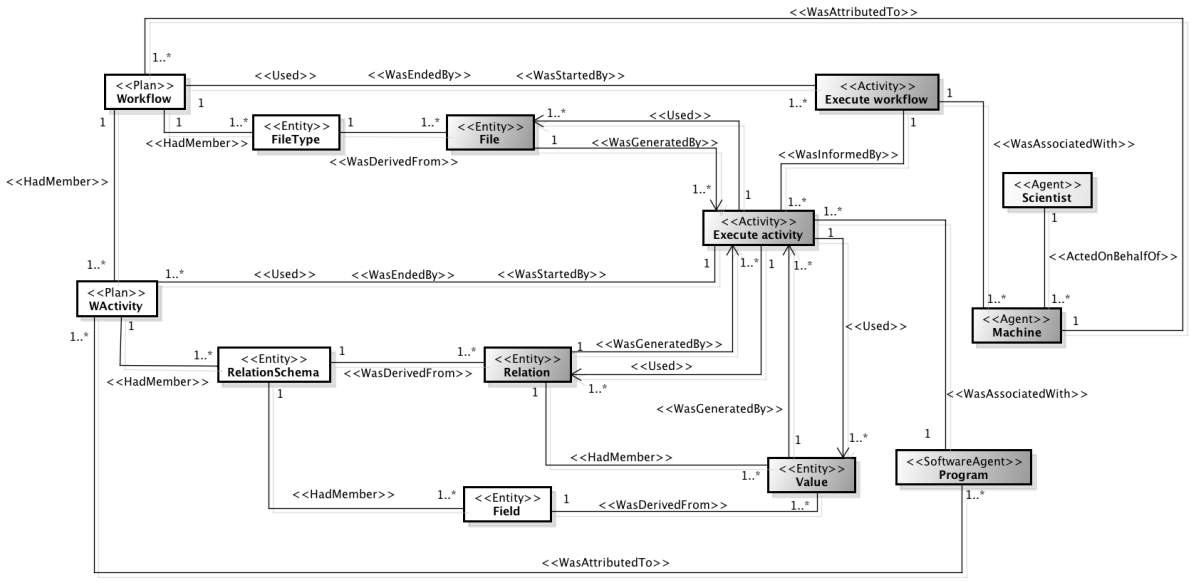

Figura 9. Modelo de dados PROV-Wf. Retirado de (Costa *et al*, 2013)

O primeiro passo para a utilização do PROV-Wf foi mapear seus componentes nas tabelas do SciCumulus. Para isso, foi realizada uma comparação entre este modelo de dados e o apresentado na Figura 7. O resultado desta análise está descrito na Tabela 2

Para este mapeamento, procurou-se seguir o esquema de cores já apresentado e as explicações proporcionadas pelos criadores do PROV-Wf. Alguns componentes são facilmente mapeáveis, como o *WActivity* e o *Workflow*. Ambos têm cor branca no modelo e são identificados como *Plan*. Por isso é possível perceber que representam a parte conceitual, listando um conjunto de características básicas de cada *workflow* e de cada atividade, respectivamente.

Tabela 2. Mapeamento do PROV-Wf nas tabelas do SciCumulus

Prov-Wf	SciCumulus
WActivity	*cactivity*
Workflow	*cworkflow*
RelationSchema	*crelation*
Relation	Tabelas do domínio
Field	*cfield*
Value	Tabelas do domínio
FileType File	*efile*
Execute workflow	*eworkflow*
Execute activity	*eactivation*
Program	*eactivity*(*)
Machine	*emachine*
Scientist	-

RelationSchema se relaciona com *WActivity* da mesma forma que acontece entre *crelation e cactivity* no modelo de dados do SciCumulus, além de estar pintado de branco. Costa *et al*. (2013) também explica que *RelationSchema* é utilizado para expressar os dados consumidos por uma execução, podendo ser definido com um conjunto de campos que são representados pelo componente Field. A partir dessa informação, é possível concluir que, além de *RelationSchema* ser similar a *crelation*, *Field* corresponde à a tabela *cfield*.

A entidade *Value* representa "o conjunto de valores de um campo" (Costa *et al*., 2013). Logo, é possível perceber que este componente não está especificado no modelo de dados do SciCumulus, mas sim no esquema de domínio do *workflow*. Representa os valores atribuídos a cada campo de cada relação. O mesmo acontece com a entidade *Relation*, representante das tabelas que recebem os nomes das relações, onde estão inseridos os valores.

Já o componente *File* também poderia ser mapeado na tabela *cfile*, pois o primeiro representa os arquivos consumidos e produzidos, além de ser pintado de cinza escuro, cor que representa a execução. Porém, relacionado ao *File*, há a entidade *FileType*, cujo objetivo é apenas informar o tipo de arquivo esperado. Não há uma tabela usada apenas para essa informação, uma vez que *efile* armazena todos os dados importantes de um arquivo. Logo, a tabela *efile* engloba as duas entidades.

Em relação às atividades representadas no PROV-Wf, pode-se perceber que *Execute workflow* representa a tabela *eworkflow*. O mesmo não acontece com *Execute activity*. A princípio, é possível deduzir que este componente deve ser mapeado na tabela *eactivity*. Porém, ao observar os seus relacionamentos, concluímos que, na verdade, representa a tabela *eactivation*. *Execute activity* está relacionada diretamente às entidades *File* e *Value*. No entanto, no SciCumulus, os arquivos e valores dos campos nas relações são gerados e usados pelas ativações. Apesar de indiretamente as atividades também usarem esses valores, acreditamos que o correto é mapear esta entidade como *eactivation*.

Por último, em relação aos agentes, associamos *Machine* à *emachine*, apesar de não ficar explícito o porquê de *Machine* estar se relacionando a *Execute workflow*, uma vez que no SciCumulus cada ativação, e não cada *workflow*, pode ter diferentes máquinas. Ademais, o componente *Program* foi, de certa forma, mapeado na *eactivirty*, pois cada execução de uma atividade chama um programa diferente. Não é possível, porém, relacionar ambos diretamente, porque *eactivity* possui um tempo inicial e um final, característica de uma atividade no PROV, não de um agente. Já *Scientist* ainda não é representado, de nenhuma forma, no modelo de dados de proveniência mais recente do SciCumulus.

A falta de representação desses últimos componentes pode ser considerada uma limitação do modelo de dados de proveniência do SciCumulus. Não é possível saber os responsáveis pelos *workflows* e suas execuções, o que pode dificultar a explicação de determinados problemas. O mesmo acontece com o programa, uma vez que o SciCumulus se limita a armazenar a chamada do programa via linha de comando na tabela *cactivity*. Nenhuma outra informação é armazenada, como, por exemplo, a versão do programa ou em que linguagem é escrito, que poderia facilitar a investigação de possíveis erros ou permitir a reconstrução da execução de forma mais clara e detalhada.

A partir desse mapeamento e do construído para o OPMW, a ontologia pode ser modelada. Todas as classes do OPMW que puderam ser associadas a alguma tabela foram utilizadas. Para as tabelas restantes, classes novas foram criadas. Além disso, procurou-se sempre seguir os padrões do PROV-O para representações. Logo, para entidades foram usadas elipses, atividades foram representadas com retângulos e para os agentes foram usados pentágonos.

As classes que foram criadas, ou seja, que não pertencem à ontologia OPMW são: *Scientist*, *Machine*, *Program*, *TemplateRelation*, *ExecutionTask* e *File*. É possível observar que todas as novas classes vêm precedidas do prefixo prov-o-wf, para mostrar que foram criadas para esta ontologia. As outras classes recebem o prefixo opmw para ficar evidente que pertencem àquela ontologia.

Em relação às propriedades de objeto, procurou-se usar as que já são definidas no OPMW. Na falta de uma relação, a segunda opção foi procurar nas propriedades definidas pelo PROV-O. Por último, caso nenhuma existente se adequasse, uma nova propriedade era criada. Foi o caso do *dependsOn*, propriedade utilizada para representar a dependência entre as atividades. A ontologia completa está representada na Figura 10.

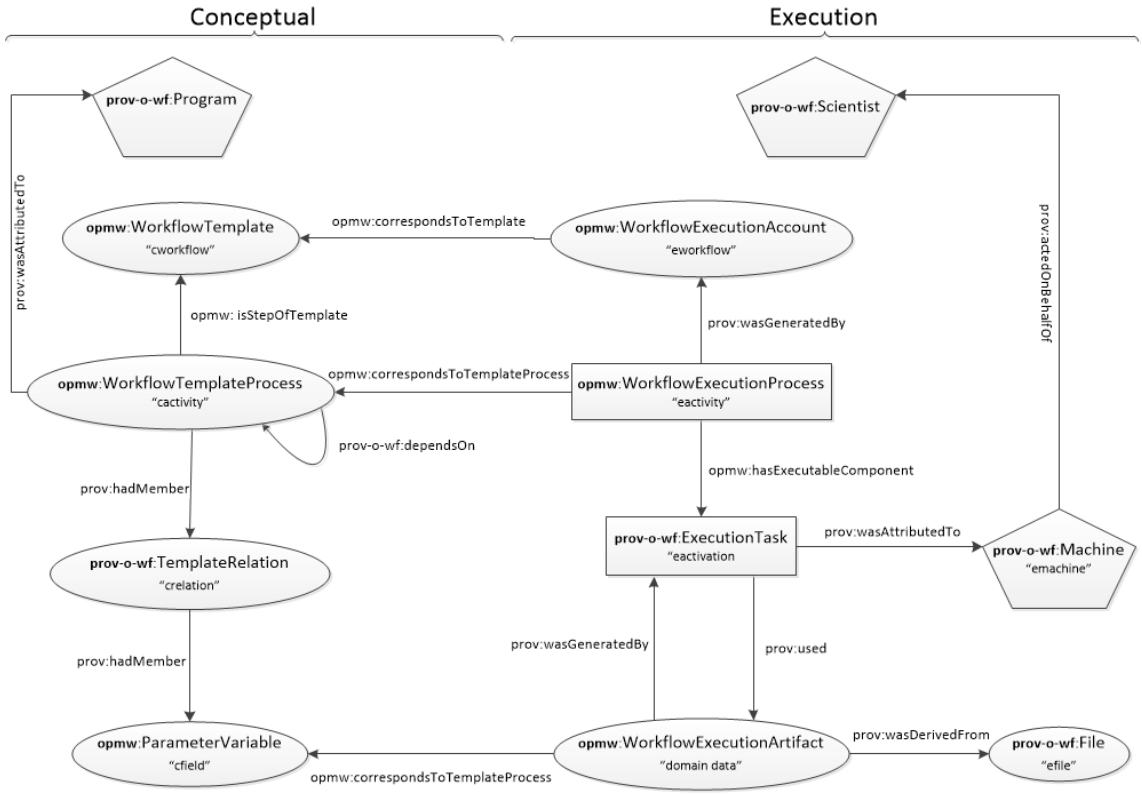

Figura 10. Ontologia PROV-O-Wf

É importante observar, também, que alguns relacionamentos existentes no PROV-Wf não foram utilizados ou foram substituídos, para uma melhor adequação das classes do OPMW e devido a ajustes realizados, baseados no mais atual modelo de dados do SciCumulus, mostrado na Figura 6. Por exemplo, na ontologia, *Machine* está relacionada à classe *ExecutionTask*, que representa a tabela *eactivation*, ao contrário do PROV-Wf, em que está relacionada à *eworkflow*. Isso foi feito porque, como já dito anteriormente, as ativações de uma atividade de um *workflow* são executadas de maneira distribuída em máquinas diferentes do *cluster*.

Ainda em comparação com o PROV-Wf, outra modificação realizada foi em relação às propriedades de objeto da classe *File*. Como as tabelas de domínio já armazenam informações sobre os arquivos consumidos e gerados pelo *workflow*

(comumente armazena-se o caminho dos arquivos consumidos ou gerados durante a execução do *workflow* nas tabelas de domínio) e os arquivos são representados pela classe *File*, podemos entender que os dados de domínio dos arquivos são derivados dos arquivos e, consequentemente, da classe *File*. Logo, essa entidade da ontologia se relaciona diretamente com a classe *WorkflowExecutionArtifact*, através do relacionamento *wasDerivedFrom,* quando os dados de entrada e saída forem do tipo *file*.

Apesar de anteciparmos e indicarmos na ontologia PROV-O-Wf, esse último relacionamento não poderá ser considerado para a etapa de ETC que será vista no próximo capítulo. Utilizando o modelo de dados atual do SciCumulus, não é possível relacionar a tabela *efile* à tabela *crelation,* mesmo que apenas de forma indireta. No SciCumulus atual, em *efile*, só há armazenado o nome do campo ao qual o arquivo pertence. Porém, a chave primária da tabela *cfield* é composta pelo nome do campo e pelo id da relação, pois o nome pode ser o mesmo para a relação de entrada ou de saída de uma atividade. Assim, como não há nenhuma informação sobre *crelation* na tabela *efile*, não é possível saber à qual relação determinado arquivo está associado. Por exemplo, determinada atividade possui uma relação de entrada com id=1 e campos de nome a e b, sendo o campo b do tipo *file*. A relação de saída, com id=2, é composta pelos campos cujos nomes são b e c, sendo o primeiro também do tipo *file*. Nas tabelas de domínio, os campos da primeira relação estarão armazenados em uma tabela e os da segunda estarão em outra. Os que são arquivos também serão representados na tabela *efile*, e possuem associados a eles apenas o nome do campo que, no caso, é b para ambos. Logo, não é possível saber qual dos dois arquivos representa o arquivo consumido por uma relação de entrada e qual foi gerado por uma relação de saída. Assim, apesar de já existir uma tabela para representação dos arquivos no modelo de

dados atual do SciCumulus, o único relacionamento da classe *File* da PROV-O-Wf não será considerado na etapa de ETC, da mesma forma que acontecerá com a classe *Scientist*.

Tabela 3. Propriedades de Dados

PROV-O-Wf	Colunas	Propriedades de Dados
opmw:WorkflowTemplate	wkfid	prov-o-wf:hasId
	tag	prov-o-wf:hasName
	description	prov-o-wf:hasDescription
opmw:WorkflowTemplateProcess	actid	prov-o-wf:hasId
	tag	prov-o-wf:hasName
	atype	prov-o-wf:hasType
prov-o-wf:TemplateRelation	relid	prov-o-wf:hasId
	rtype	prov-o-wf:hasType
	rname	prov-o-wf:hasName
opmw:ParameterVariable	fname	prov-o-wf:hasName
	ftype	prov-o-wf:hasType
opmw:WorkflowExecutionAccount	ewkfid	prov-o-wf:hasId
	tagexec	prov-o-wf:hasName
opmw:WorkflowExecutionProcess	actid	prov-o-wf:hasId
	status	opmw:hasStatus
	starttime	prov:startedAtTime
	endtime	prov:endedAtTime
prov-o-wf:ExecutionTask	taskid	prov-o-wf:hasId
	starttime	prov:startedAtTime
	endtime	prov:endedAtTime
	status	opmw:hasStatus
opmw:WorkflowExecutionArtifact	id	prov-o-wf:hasId
	value	opmw:hasValue
prov-o-wf:Machine	machineId	prov-o-wf:hasId
	hostname	prov-o-wf:hasName

Para modelar as propriedades de dados, foram selecionadas as colunas com informações mais relevantes de cada tabela. As propriedades de dados já existentes no PROV e no OPMW relacionadas às informações das colunas selecionadas foram usadas. Os dados não modelados por nenhuma das duas ontologias precisaram ter propriedades de dados criadas. Como é possível obervar na Tabela 3, existiam poucas propriedades que representassem as colunas das tabelas. Logo, praticamente todas foram criadas para completar a ontologia.

A Figura 11 apresenta a taxonomia do PROV-O-Wf, construída para permitir uma melhor visualização da estrutura hierárquica das classes.

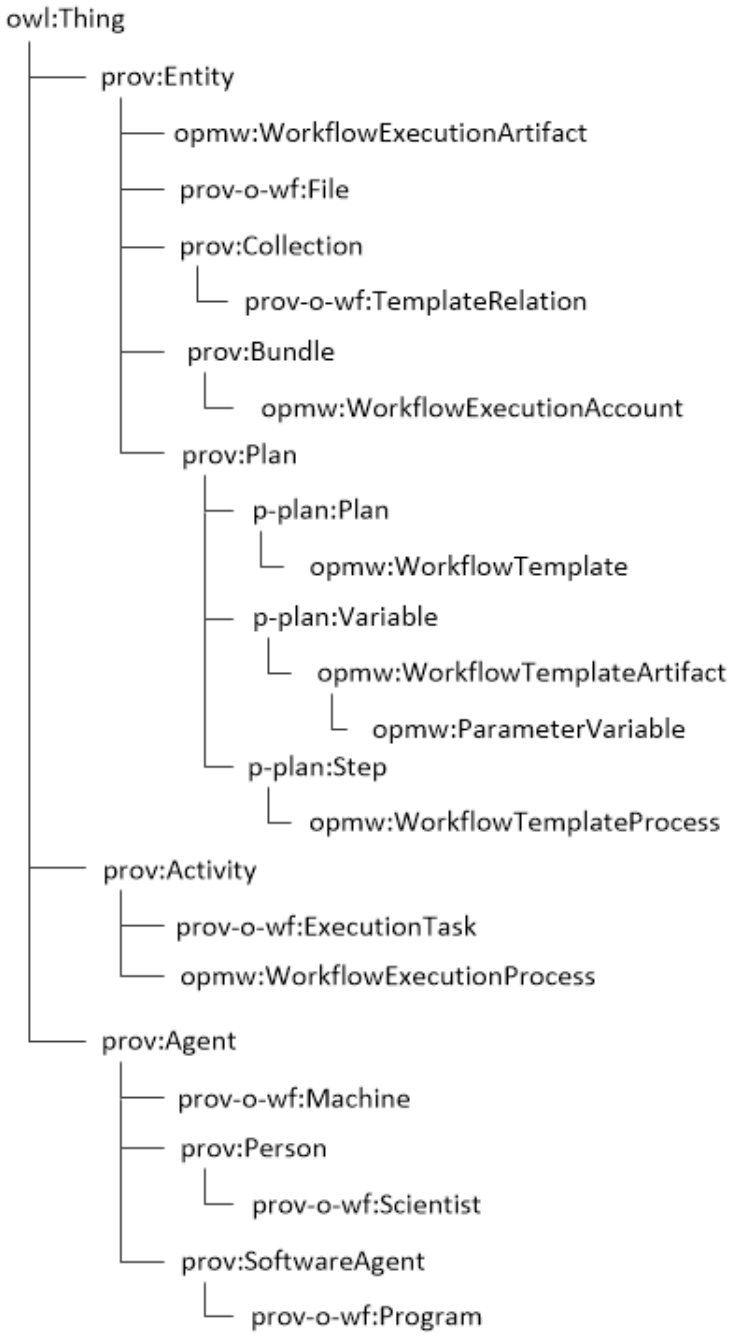

Figura 11. Taxonomia da ontologia PROV-O-Wf

4. Processo de Extração, Transformação e Carga

Seguindo o processo de publicação de dados na Web Semântica proposto por Souza (2013), o próximo passo a ser seguido é o projeto de triplificação (i.e., instanciar triplas em RDF).

A primeira etapa do projeto de triplificação é a escolha do Sistema de Gerenciamento de Banco de Dados RDF (SGBD RDF). Esses sistemas são usados, entre outras finalidades, para realizar funções CRUD (*Create, Read, Update and Delete*) nos dados RDF (Souza, 2013). Para este trabalho, foi escolhido o SGBD RDF Virtuoso[8].

Escolhido do SGBD RDF, a próxima etapa é carregar os dados no repositório. Como no nosso caso os dados estão disponibilizados em um banco de dados relacional e queremos gerar dados em RDF para instanciar dados utilizando a ontologia proposta no capítulo anterior, foi necessária a realização do processo de ETC. Nele, os dados precisaram ser extraídos da base de proveniência (relacional) do SciCumulus, triplificados usando a ontologia e carregados no repositório do Virtuoso.

Para facilitar o processo de ETC, foi construída uma interface Web, possibilitando a interação entre os dados disponíveis para publicação e o cientista que quer publicar seus resultados na Web Semântica. Esta interface permite que o usuário escolha quais dados de determinado *workflow* serão publicados. Assim, caso alguma informação armazenada seja sigilosa, por exemplo, não precisa ser necessariamente disponibilizada, trazendo uma maior maleabilidade e flexibilidade para o cientista.

Dessa forma, para a publicação, basta que o cientista realize um cadastro ou, caso já possua um, faça um *login* na área de inserção de dados. Em seguida, é necessária a

[8] http://virtuoso.openlinksw.com/

escolha da base de dados onde estão guardados os dados do *workflow* a ser publicado. Por último, à medida que as informações vão aparecendo, o cientista pode ir selecionando o que será publicado na Web Semântica.

A seguir será explicado de forma mais detalhada como foi realizada cada etapa do processo de ETC.

4.1. Extração

Para realizar a extração dos dados, foram construídas consultas SQL para retornar as informações necessárias. Inicialmente, o cientista deve escolher qual *workflow* quer publicar. Todos os nomes cadastrados na tabela *cworkflow* são listados e um deve ser escolhido. Com o id do *workflow* escolhido, é possível realizar uma consulta para retornar todas as execuções armazenadas na tabela *eworkflow* e todas as atividades pertencentes àquele *workflow* guardadas na tabela *cactivity*. A partir deste ponto já é possível selecionar as execuções do *workflow* e as atividades que serão publicadas.

Em posse dessas duas novas informações, o próximo passo é realizar consultas na tabela *eactivity*, para retornar as execuções das atividades escolhidas, relacionadas às execuções do *worklfow*, também escolhidas. Também já é possível consultar a tabela *crelation*, para obter todas as relações de entrada e saída listadas para as atividades selecionadas. Esse processo de construção de consultas é repetido, de acordo com o fluxograma abaixo.

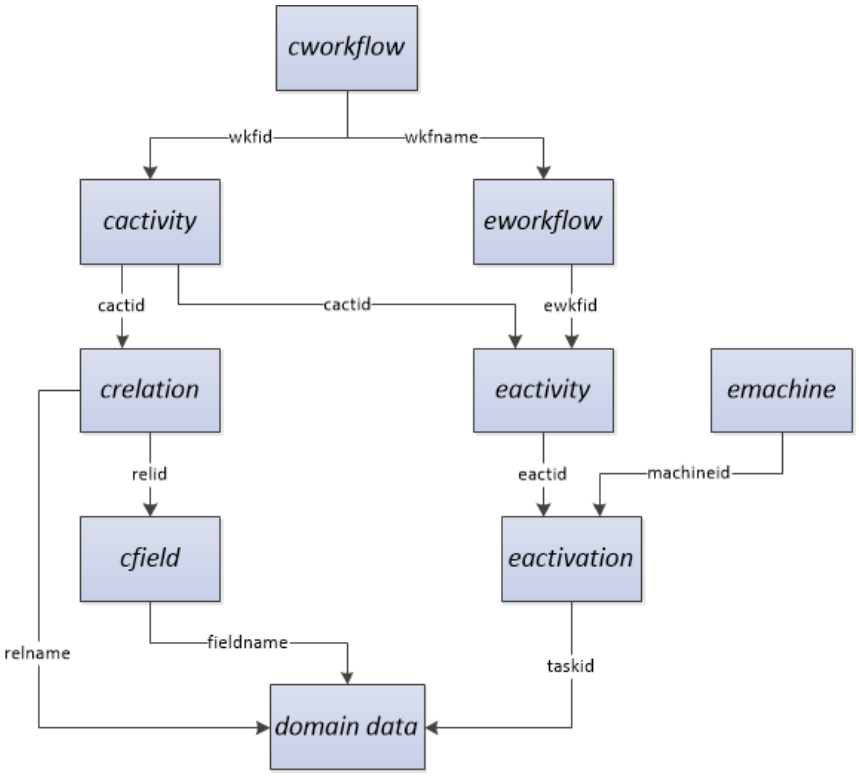

Figura 12. Fluxograma para extração de dados do SciCumulus

Em relação aos dados de domínio, a construção da consulta é um pouco mais complexa. A consulta é feita nas tabelas com os nomes das relações escolhidas e de acordo com os campos associados. As colunas que são retornadas têm o nome dos campos selecionados e os dados devem estar relacionados às ativações escolhidas pelo cientista para a publicação.

4.2. Transformação

Como já dito anteriormente, a transformação é usada para criar as triplas RDF a partir dos dados extraídos do banco de dados relacional. Nesta etapa, é essencial a criação de identificadores para compor as URIs. Neste projeto, apenas para fins de demonstração, foi estabelecido que todos os recursos iniciam suas URIs com `http://`

www.cos.ufrj.br/~scicumulus/resource#, que chamaremos de URI_BASE. Além disso, todas as informações escolhidas para publicação são triplificadas, de acordo com a classe da ontologia a qual pertence e com suas propriedades.

Primeiramente, para cada tabela selecionada na interface, é necessária uma URI com o objetivo de representar uma instância da classe da ontologia. A Tabela 4 apresenta como as URIs das instâncias são criadas. Além da construção mostrada abaixo, todas as URIs são precedidas pela URI_BASE e pela URI da classe *WorkflowTemplate*. Essas triplas devem possuir o relacionamento http://www.w3.org/1999/02/22-rdf-syntax-ns#type, cujo prefixo é rdf:type. Por fim, o objeto é a classe à qual a instância pertence.

Tabela 4. URIs para as instâncias das classes da ontologia

Classe Prov-Wf	URI
opmw:WorkflowTemplate	“workflow_” + Name + “-“ + Id
opmw:WorkflowTemplateProcess	“cactivity_” + Name + “-“ Id
prov-o-wf:TemplateRelation	“crelation_”+Name + “-“ + Id
opmw:ParameterVariable	“cfield_”+Name + “-“ relationId“
opmw:WorkflowExecutionAccount	“eworkflow_” + Name + “-“+Id
opmw:WorkflowExecutionProcess	“eactivity_” + Starttime + “-“ + Id
prov-o-wf:ExecutionTask	“eactivation_” + Starttime + “-“ + Id
opmw:WorkflowExecutionArtifact	“domain_”+fieldname+“-“+“relationId“+”-“+ Id
prov-o-wf:Program	“program_”+ Name
prov-o-wf:Machine	“emachine_” + Hostname + “-“ + Id
prov-o-wf:Scientist	-
Prov-o-wf:File	-

Após criar as instâncias de cada classe, é necessário triplificar os relacionamentos entre as tabelas e os relacionamentos com os dados existentes nas colunas de interesse. No primeiro caso, os sujeitos e os objetos serão as instâncias das classes, cujas URIs já

foram definidas acima. Os relacionamentos serão as propriedades de objetos da ontologia desenvolvida. No segundo caso, os sujeitos também serão as instâncias, enquanto que os objetos serão os valores que estavam armazenados no banco de dados. Os predicados serão as propriedades de dados.

As classes e propriedades da ontologia também possuem URIs. Os elementos que são da ontologia OPMW, possuem a URI `http://wwww.opmw.org/ontology/` e o seu prefixo é `opmw:`. Já para a ontologia PROV-O, a URI utilizada é `http://wwww.w3.org/ns/prov#` e o seu prefixo é `prov:`. A ontologia PROV-O-Wf, utiliza URI `http://wwww.scicumulus.com/ontologies/prov-o-wf#` e prefixo `prov-o-wf:`.

4.3. Carga

O último passo no processo é a carga de todas as triplas RDF criadas na etapa anterior e a inserção delas no SGBD RDF Virtuoso. Para isso, todas as triplas são salvas em um arquivo do tipo N-TRIPLES, formato já explicado no capítulo 2. Este arquivo então é carregado no repositório de triplas.

A partir deste ponto, todas as informações do *workflow* que foram escolhidas já estão na Web Semântica e acessíveis publicamente, caso o domínio do servidor que hospeda o banco de dados semântico seja público. Logo, os dados publicados poderão ser descobertos através de mecanismos de busca semântica, interligados através de técnicas de LOD e consultados através de consultas SPARQL, caso o SPARQL Endpoint[9] esteja público.

[9] http://www.w3.org/TR/rdf-sparql-protocol/

A interface construída também disponibiliza a visualização do que foi escolhido para publicação. Está aplicação de visualização foi construída utilizando JavaScript e a biblioteca gráfica D3.js[10]. Após pressionar o botão para publicar os dados, uma imagem da instância da ontologia com as classes e suas propriedades de objetos é gerada dinamicamente e exibida. Essa ferramenta possibilita que o cientista visualize a estrutura dos dados publicados, facilitando a análise e a construção de consultas SPARQLS, por exemplo.

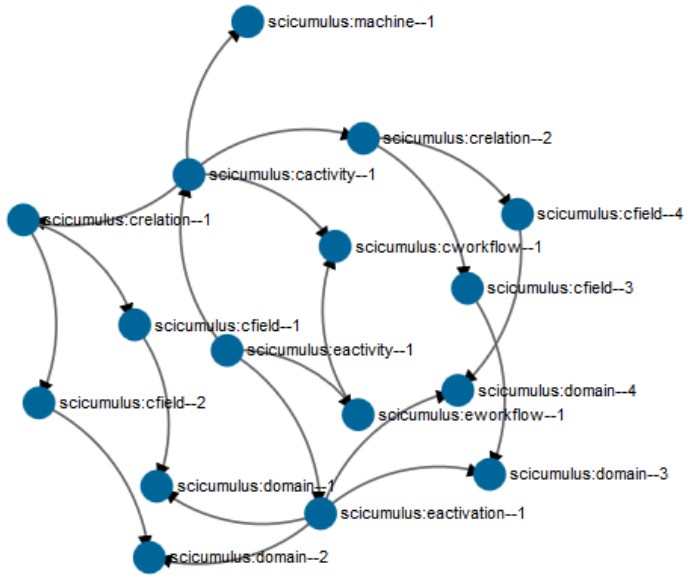

Figura 13. Exemplo de visualização gerada

A Figura 13 apresenta a visualização de um *workflow* genérico. As posições dos nós são flexíveis, porém limitadas pelo tamanho fixo das arestas, possibilitando uma rearrumação dos nós. Como é possível observar, as propriedades de objeto não estão representadas com os respectivos predicados, assim como não são exibidas as propriedades de dados. Essa escolha foi feita para permitir uma visualização da

[10] http://d3js.org/

ontologia como um todo de forma mais clara. No entanto, é possível observar essas informações faltantes ao clicar em um nó. Uma tabela então aparecerá para complementar os dados da classe selecionada. A Figura 14 apresenta um exemplo das informações exibidas quando foi selecionada a classe *scicumulus:eactivity—1*.

Class: scicumulus:eactivity--1	
Data Property	
id	1
name	eactivity
Object Property	
prov:wasGeneratedBy	range: scicumulus:eworkflow--1
opmw:correspondtoTemplateProcess	range: scicumulus:cactivity--1
opmw:hasExecutableComponent	range: scicumulus:eactivation--1

Figura 14. Informações adicionais exibidas ao selecionar uma classe

Assim, o usuário da interface pode interagir com os dados que foram publicados de uma forma visual, facilitando o entendimento da ontologia aplicada para o seu *workflow*.

5. Estudo de caso

Este capítulo tem o objetivo de apresentar um estudo de caso para demonstrar o que foi desenvolvido ao longo do projeto, ou seja, a publicação dos dados de proveniência na Web Semântica. Para isso, será utilizado o SciEvol, um *workflow* científico da área da bioinformática (Ocaña *et al*., 2012). Neste capítulo, primeiramente será apresentado o *workflow* em questão e, em seguida, será descrito todo o processo realizado para a publicação dos dados.

5.1. SciEvol

O SciEvol é um *workflow* científico usado para a análise MER (*Molecular Evolution Reconstruction*). Experimentos MER são usados para a inferência de relacionamentos evolutivos entre indivíduos, populações, espécies e entidades de taxonomia elevada, utilizando dados moleculares (Ocaña *et al*., 2012). Por ser composto por um conjunto de programas científicos, este tipo de experimento pode ser modelado como um *workflow* científico.

O uso de *workflows* é importante, pois o gerenciamento e a análise desse experimento necessitam de um alto desempenho computacional e o processamento de uma grande quantidade de dados. Por esses motivos, foi criado o SciEvol, *workflow* que utiliza os dados de entrada para estimar parâmetros e testar hipóteses. Ele é composto por um conjunto de atividades, como pode ser observado na Figura 15.

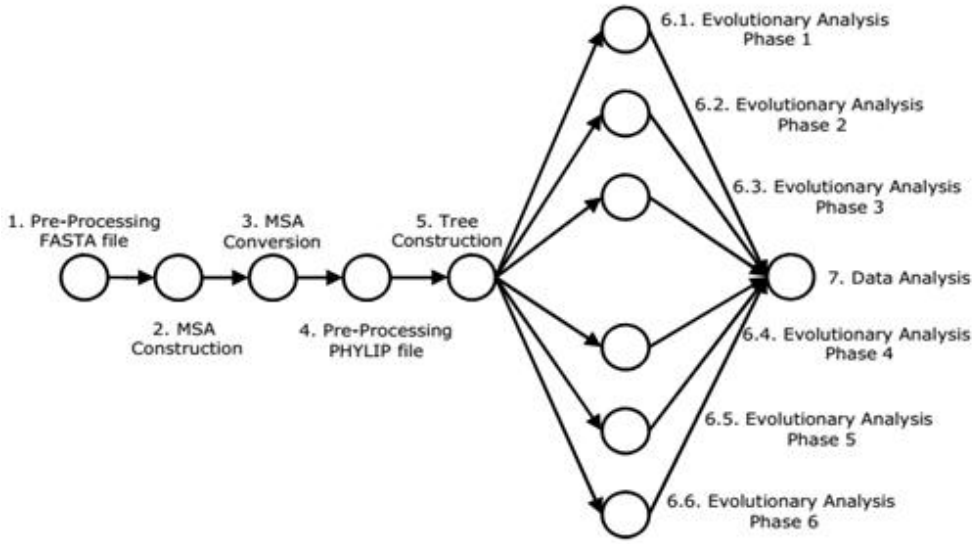

Figura 15. Atividades que compõe o SciEvol. Retirado de (Ocaña *et al*., 2012)

A seguir será apresentada uma explicação rápida das atividades do *workflow* SciEvol (Ocaña *et al*., 2012).

- Pré-processamento dos arquivos Multi-Fasta: remove os códons de parada dos dados de entrada através de um *script* Python.
- Construção do AMS: constrói o alinhamento das sequências múltiplas, utilizando o programa MAFFT. Seu dado de entrada é um arquivo multi-fasta pré-formatado e sua saída é o AMS (Alinhamento Múltiplo de Sequências) no formato FASTA.
- Conversão do AMS: transforma o AMS gerado na atividade anterior para o formato PHYLIP, utilizando um programa chamado ReadSeq.
- Pré-processamento do arquivo PHYLIP: *script* Python utilizado para formatar o arquivo PHYLIP. Uma das alterações feitas, por exemplo, é inserção de um parâmetro no final da primeira linha do arquivo.

- Construção da árvore: utilizando um programa chamado RAxML, constrói a árvore filogenética, em um formato conhecido como Newick.
- Análise evolutiva: a análise evolucionária é realizada da sexta atividade até a décima primeira, executando seis fases do MER relacionadas à substituição dos modelos de códons.
- Análise dos dados: nesta última atividade, todos os arquivos de saída das últimas seis atividades são processados, disponibilizando informação suficiente para a manipulação e análise estatística do cientista.

5.2. Publicação dos dados de proveniência do SciEvol

O primeiro passo para a publicação dos dados de proveniência do SciEvol foi a execução do *workflow* no SciCumulus. A execução foi realizada localmente assim como o armazenamento dos dados, pois o objetivo é apenas a realização de um estudo de caso para o projeto desenvolvido.

As atividades descritas na seção acima são representadas de uma maneira diferente nos dados de proveniência. A tabela abaixo apresenta um mapeamento dessas atividades.

Tabela 5. Mapeamento das atividades do SciEvol

Diagrama do SciEvol	Atividade
Pré-processamento dos arquivos Multi-Fasta Construção do AMS	mafft
Conversão do AMS	readseq
Pré-processamento do arquivo PHYLIP	formatPhylip
Construção da árvore	raxml

Análise evolutiva	codemlM0 codemlM1 codemlM2 codemlM3 codemlM7 codemlM8
Análise dos dados	merge analyses compile

Em posse dos dados de proveniência, foi necessário realizar uma análise de quais informações deveriam ser publicadas, baseando-se na ontologia proposta. É importante observar que este passo só é possível pois foi construída uma interface que permite a escolha dos dados por parte do cientista. Caso contrário, tudo o que foi definido como propriedade de dados e de objeto seria inserido na Web Semântica.

A Figura 16 apresenta a interface de uma das etapas para a publicação dos dados na Web Semântica, e será usada apenas para exemplificar o funcionamento da ferramenta desenvolvida. Como é possível observar, após escolher qual *workflow* será publicado, uma tabela exibe todos as propriedades de dados relacionadas à classe *WorkflowTemplate*. O cientista pode então decidir se quer ou não publicar os dados. Ao finalizar suas escolhas, os dados da próxima classe, no caso *WorkflowExecutionAccount*, serão exibidos. Esse processo se repete para todas as outras classes da ontologia até que todos os dados de interesse estejam selecionados.

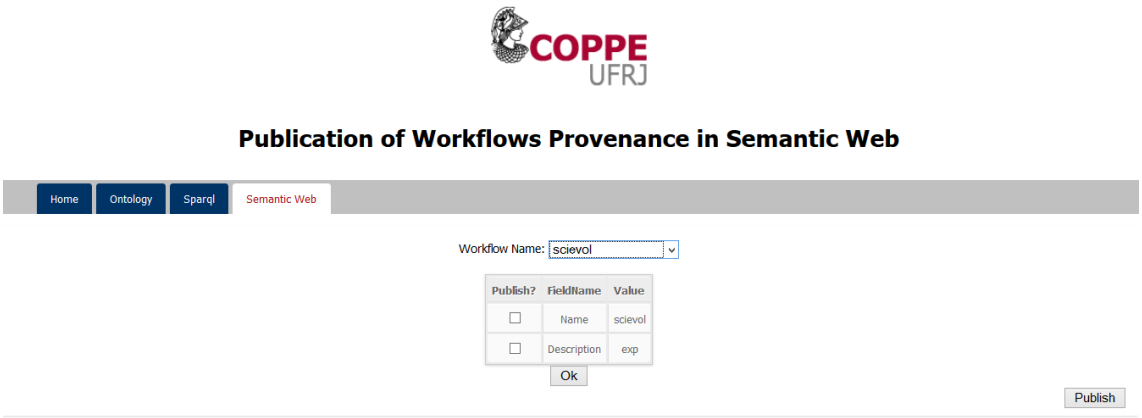

Figura 16. Interface para a publicação na Web Semântica

O SciEvol possui duas atividades, *readseq* e *formatPhylip*, que possuem o objetivo apenas de limpar os dados e colocá-los no formato correto. Ou seja, é apenas um pré-processamento dos dados. Dessa forma, as informações relacionadas a essas atividades não possuem um grande valor de análise que desperte o interesse de outros cientistas. Logo, não é necessário publicar os seus dados de proveniência.

Assim, a base de proveniência utilizada possui um *workflow* conceitual (*cworkflow*) e uma execução de *workflow* (*eworkflow*). Apenas os dados de proveniência de onze atividades serão publicados, tanto em relação à parte conceitual (*cactivity*) como de execução (*eactivity*). Há duas ativações para cada atividade (*eactivation*), todas executadas na mesma máquina (*emachine*). Todas as relações (*crelation*) e os respectivos campos (*cfield*) das atividades escolhidas serão publicados, assim como os dados de domínio. Em relação às propriedades de dados, todas elas serão inseridas, pois, para o SciEvol, não há motivo para não publicá-las.

A Figura 17 apresenta uma pequena parte dos dados do SciEvol que foram inseridos no banco de dados semântico. Esta imagem tem um objetivo didático, criada

apenas para exemplificar o uso da ontologia em um *workfow*. Não foi possível colocar todas as informações, pois são muito numerosas e impediriam boa visualização.

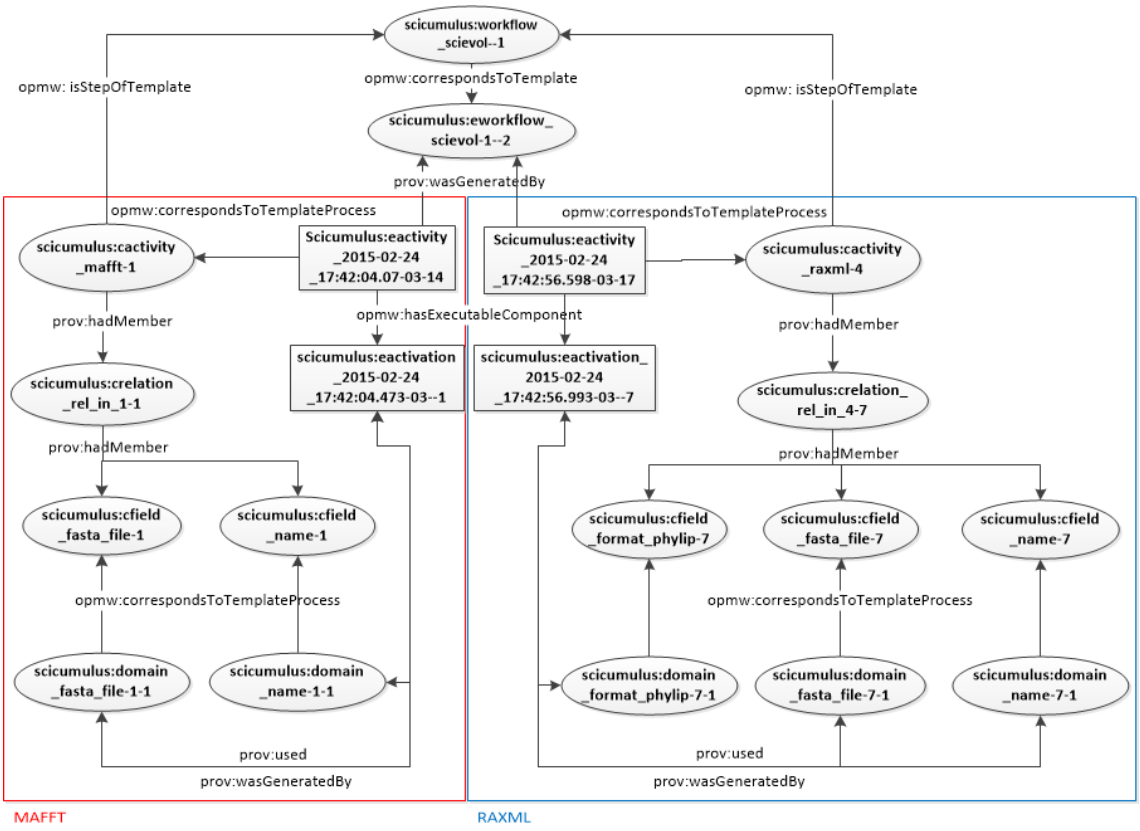

Figura 17. Parte da instância da ontologia para o SciEvol

Está figura é a representação de um pedaço da instância da ontologia para o *workflow* em questão. As duas primeiras atividades do SciEvol escolhidas para publicação estão sendo mostradas na Figura 17: *mafft* e *raxml*. Para não tornar o desenho confuso, estão sendo representadas apenas as relações de entrada, uma ativação e os dados de domínio correspondentes a essa ativação. A ontologia inicia com as classes relacionadas ao *workflow*, seguida pelas classes que definem as atividades. Cada atividade possui uma relação de entrada (e uma de saída, omitida no desenho) e sua

execução é composta por pelo menos uma ativação. Cada relação é constituída por campos, que possuem valores gerados e usados por cada ativação.

Ao observá-la, é possível notar a grande quantidade de informação armazenada, principalmente se levar em consideração a instância com todas as atividades. Logo, se tornam evidentes os benefícios de se estruturar os dados, a partir da criação de relacionamentos entre informações antes parcialmente soltas. Essas ganham um maior significado e, consequentemente, facilitam a interpretação por parte da pessoa que está analisando os dados.

A publicação na Web Semântica permite que consultas SPRQLS sejam realizadas, como já foi dito anteriormente. Em um banco de dados relacional, algumas consultas possibilitam a busca de informações básicas do *workflow*, como quais são as atividades que compõem o *workflow* e os seus respectivos tempos de execução, por exemplo. A mesma informação pode ser obtida a partir do banco de dados semântico, porém retornada na forma de triplas RDF, estruturando os dados e acrescentando mais significado através das relações. Além do fato dos dados já estarem públicos na web, o que não ocorre no SGBD relacional.

Outra facilidade proporcionada pela Web Semântica é a análise dos dados de domínio. Como no SGBD relacional essas informações ficam armazenadas em diferentes tabelas, a realização de uma análise conjunta é mais complexa e requereria diversas junções. Com consultas SPARQLs, no entanto, é possível acessar todos os dados de domínio juntos, de maneira simplificada e sem a necessidade de um grande conhecimento da linguagem. Basta usar a consulta apresentada na seção 2.4.3, especificando o valor do predicado para *prov:used*, como é possível ver abaixo. Os dados retornados são um conjunto de triplas exibindo os valores de entrada e saída e as respectivas ativações.

```
@prefix prov:  <http://wwww.w3.org/ns/prov#> .
SELECT ?a ?b ?c
WHERE { ?a prov:used ?c }
```

Além das consultas básicas usadas em um SGBD relacional, a Web Semântica permite uma análise mais integrada com dados de proveniência de outros *workflows*, como já foi dito anteriormente. Esse tipo de consulta exigiria que outros SGWfCs utilizassem uma ontologia seguindo os padrões do W3C, como a ontologia proposta neste projeto. Entretanto, essa área ainda é pouco explorada, dificultado, no momento, a elaboração de um exemplo de análise integrada dos dados de diferentes fontes. Para exemplificar, poderia ser realizada uma comparação entre os dados de domínio de diferentes *workflows*. Para isso, seria necessário apenas que ambas as ontologias possuísse a classe *WorkflowExecutionArtifact*.

A partir do que foi mostrado neste capítulo, é possível observar que a publicação dos dados de proveniência de um *workflow* na Web Semântica pode ser realizada facilmente, possibilitando uma nova forma de análise dos dados com o uso de consultas SPARQL. A visualização da ontologia proporcionada pela interface também possibilita novos modos de interação com os dados já publicados.

6. Conclusão

O uso de *workflows* científicos pode ser considerado uma alternativa para simulações computacionais complexas centradas em dados. Apesar de exigir uma série de configurações por parte de cientistas que muitas vezes não possuem grande contato com a computação, a sua usabilidade pode ser facilitada com o armazenamento de dados de proveniência.

A proveniência contem informações dos *workflows* que já foram executados, criando, desta forma, um histórico de execuções. Além de facilitar o uso futuro, a proveniência garante uma maior confiabilidade para o *workflow*, pois permite que o caminho percorrido seja refeito a partir dos dados armazenados. Assim, devido a sua grande utilidade, tornar os dados de proveniência de *workflows* científicos públicos e de uma forma estruturada, pode trazer grandes vantagens para cientistas de diferentes áreas.

Já a Web Semântica é uma tecnologia em desenvolvimento, pois ainda há um caminho a ser percorrido para atingir o que foi proposto por seu idealizador. No entanto já é possível observar um crescente esforço, principalmente por parte dos acadêmicos, para desenvolver tecnologias que contribuem com o constante crescimento da Web Semântica.

Tendo em mente a importância do acesso a dados de proveniência de *workflows* científicos e o potencial da Web Semântica, foi desenvolvida uma ontologia genérica com base no modelo de dados de proveniência do SciCumulus, importante passo para a publicação de LOD. A ontologia proposta neste projeto foi a principal contribuição para os dados desse SGWfC. Devido a essa importância, investiu-se tempo considerável pesquisando por padrões de proveniência, já apresentados no Capítulo 2 e por

ontologias já existentes. Todo o projeto procurou seguir padrões mais difundidos para facilitar a eventual integração entre os dados. Como consequência, pode-se dizer também que o modelo de dados do SciCumulus foi mapeado em duas ontologias de grande importância e de grande uso: o PROV-O e o OPMW.

A criação de uma interface para a inserção dos dados no banco de dados semântico também proporcionou maior facilidade para a execução do processo de publicação na Web Semântica. Com isso, os cientistas possuem maior incentivo para disponibilizar a proveniência dos seus *workflows*, pois a mesma permite que as informações sejam filtradas de acordo com a conveniência de cada publicador.

Por último, para a comprovação da amplitude da ontologia e da interface desenvolvidas, utilizou-se um *workflow* científico real e não um construído especificamente para este projeto.

Procurando-se por trabalhos relacionados, é possível perceber que não há muitos projetos que envolvam Web Semântica, *workflow* e proveniência. A partir de uma mapeamento realizado por Cruz *et al*. (2009), observou-se que, entre os SGWfC mais conhecidos, apenas três possuem algo relacionado à Web Semântica: Taverna, Pegasus e View. Desses, só foi possível encontrar a ontologia do Taverna, nomeada de MyGrid. Essa ontologia, no entanto, não é genérica, focando apenas em *workflows* da área da bioinformática. Além disso, a modelagem não utiliza padrões já difundidos, como o PROV.

A publicação de proveniência de *workflows* ainda é pouco explorada, dificultado a realização de uma análise mais integrada. Acreditamos, porém que esta área irá se desenvolver, devido principalmente aos potenciais benefícios proporcionados pela Web Semântica, apresentados ao longo de todo o projeto.

A partir do que foi apresentado, pode-se concluir que as principais contribuições deste projeto são de grande importância. Além de ser um trabalho pioneiro na inserção da Web Semântica no SciCumulus, aborda um tema ainda pouco explorado, unindo proveniência de dados de *workflows* científicos a Web Semântica. A criação de uma ontologia genérica, reutilizando ontologias já existentes, não limita a publicação de dados à determinada área, permitindo a expansão do uso da Web Semântica na área de SGWfC. O mesmo papel é exercido pela interface para o cientista, uma vez que facilita a publicação da proveniência na Web Semântica.

6.1. Trabalhos Futuros

Como trabalho futuro, consultas de domínio complexas poderiam ser mais exploradas, como por exemplo fazer consultas SPARQL que consultem dados de dois *workflows* de bioinformática dentro do próprio SciCumulus. Como o estudo de caso deste trabalho utilizou apenas um *workflow* científico, não foi possível realizar essa abordagem.

O outro trabalho futuro está voltado para a interface, construída com o objetivo de facilitar a publicação dos dados pelos cientistas. Como o objetivo principal deste projeto era a construção de uma ontologia para o SciCumulus e a realização do processo de ETC, não foi possível despender muito tempo na apresentação da interface. Assim, como trabalho futuro, é necessário melhorar a usabilidade da interface. Para *workflows* muito grandes, necessita-se de um tempo razoável para conseguir publicar todos os dados. Além disso, o *design* da interface também precisa ser melhorado, para proporcionar uma aplicação mais agradável aos olhos do usuário.

Referências Bibliográficas

Berners-Lee, T., (1998), " An attempt to give a high-level plan of the architecture of the Semantic WWW". Disponível em : http://www.w3.org/DesignIssues/Semantic.html

Berners-Lee, T., Cailliau, R., (1990), " WorldWideWeb: Proposal for a HyperText Project ". Disponível em : http://www.w3.org/Proposal.html

Buneman, P., Tan, W.-C., (2007), "Provenance in databases". In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, p. 1171–1173, New York, NY, USA.

Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., Vo, H. T., (2006), "VisTrails: visualization meets data management". In: SIGMOD International Conference on Management of Data, p. 745–747, Chicago, Illinois, USA.

Costa, F., Silva, V., Oliveira, D., et al., (2013), " Capturing and Querying Workflow Runtime Provenance with PROV: a Practical Approach". In: Proceeding of the Joint EDBT/ICDT 2013 Workshops, Genoma, Italy.

Cruz, S. M. S. da, Campos, M., Mattoso, M., (2009), "Towards a Taxonomy of Provenance in Scientific Workflow Management Systems". In: IEEE International Workshop on Scientific Workflows, Los Angeles, California, United States.

Davidson, S. B., Freire, J., (2008), "Provenance and scientific workflows: challenges and opportunities". In: ACM SIGMOD international conference on Management of data, p. 1345–1350, Vancouver, Canada.

Deelman, E., Chervenak, A., (2008), "Data Management Challenges of Data-Intensive Scientific Workflows". In: CCGRID ’08, p. 687–692.

Deelman, E., Gannon, D., Shields, M., Taylor, I., (2009), "Workflows and e-Science: An overview of workflow system features and capabilities", Future Generation Computer Systems, v. 25, n. 5, p. 528–540.

Deelman, E., Mehta, G., Singh, G., Su, M.-H., Vahi, K., (2007), "Pegasus: Mapping Large-Scale Workflows to Distributed Resources", Workflows for e-Science, Springer, p. 376–394.

Garijo, D., Gil, Y., (2014a), "The OPMW-PROV Ontology ". Disponível em: http://www.opmw.org/model/OPMW/.

Garijo, D., Gil, Y., (2014b), "The P-PLAN Ontology". Disponível em: http://vocab.linkeddata.es/p-plan/.

Gruber, T., (1993), " A Translation Approach to Portable Ontology Specifications", Knowledge Systems Laboratory, Standford, California, USA.

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., Oinn, T., (2006), "Taverna: a tool for building and running workflows of services", Nucleic Acids Research, v. 34, n. 2, p. 729–732.

Moreau, L., Groth, P., (2013) "Provenance: an introduction to PROV. Synthesis Lectures on the Semantic Web: Theory and Technology", Morgan & Claypool Publishers.

Moreau, L., Clifford, B., Freire, J., et al., (2010) "The Open Provenance Model Core Specification (v1.1)", Future Generation Computer Systems.

Ocaña, K., Oliveira, D., Horta, F., et al., (2012) "Exploring molecular evolution reconstruction using a parallel cloud-based scientific workflow". In: Proceedings of the 2012 Brazilian Symposium on Bioinformatics.

Ogasawara, E., Dias, J., Oliveira, D., et al., (2011), "An Algebraic Approach for Data-Centric Scientific Workflows", Proceedings of the VLDB Endowment, v. 4, n. 12, pp. 1328-1339.

Oliveira, D., Ogasawara, E., Baião, F., Mattoso, M., (2010), "SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows". In: 3rd International Conference on Cloud Computing, p. 378–385, Washington, DC, USA.

Roberts, R., (2012), "Understanding RDF serialisation formats". Disponível em: http://blog.swirrl.com/articles/rdf-serialisation-formats/.

Santos, E., Assis, V., (2013), "Avaliação de Consultas Executadas sobre Bases de Dados de Proveniência Distribuídas".

Simmhan, Y., Plale, B., Gannon, D., (2005), "A survey of data provenance in e-science".

Sousa, V., (2011), "Simiflow: Uma Arquitetura para Agrupamentos de Workflow por Similaridade".

Souza, R., (2013), "Processo de Publicação de Dados Abertos Interligados e Aplicação a Dados de Desempenho de Rede".

Souza, R., Cottrell, L., White, B., et al., (2014), "Linked Open Data Publication Strategies: Application in Networking Performance Measurement Data".

World Wide Web Consortium, (2004), "OWL Web Ontology Language". Disponível em : http://www.w3.org/TR/owl-guide/.

World Wide Web Consortium, (2013a), "PROV-DM: The PROV Data Model". Disponível em: http://www.w3.org/TR/2013/REC-prov-dm-20130430/.

World Wide Web Consortium, (2013b), "PROV-O: The PROV Ontology". Disponível em: http://www.w3.org/TR/2013/REC-prov-o-20130430/.

World Wide Web Consortium, (2014), "RDF 1.1 Turtle". Disponível em: http://www.w3.org/TR/turtle/.

World Wide Web Consortium, (2008), " SPARQL Query Language for RDF". Disponível em: http://www.w3.org/TR/rdf-sparql-query/.

World Wide Web Consortium, (2005), "Technical report development process: Maturity level for work in progress". Disponível em: http://www.w3.org/2005/10/Process-20051014/tr.html#q73.