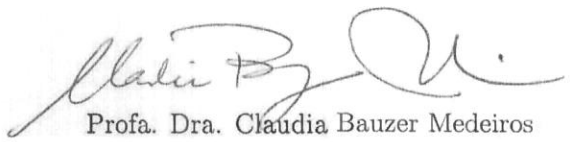
Instituto de Computação
Universidade Estadual de Campinas

Mecanismos de anotação semântica para workflows científicos

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Arnaldo Francisco Vitaliano Filho e aprovada pela Banca Examinadora.

Campinas, 03 de julho de 2009.



Profa. Dra. Claudia Bauzer Medeiros
Instituto de Computação - UNICAMP
(Orientadora)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Crisllene Queiroz Custódio – CRB8 / 7966

Vitaliano Filho, Arnaldo Francisco
V831m Mecanismos de anotação semântica para workflows científicos / Arnaldo Francisco Vitaliano Filho -- Campinas, [S.P. : s.n.], 2009.

Orientadora : Claudia Maria Bauzer Medeiros
Dissertação (Mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Fluxo de trabalho. 2. Semantica e processamento de dados. 3. Gerenciamento da informação. I. Medeiros, Claudia Maria Bauzer. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Mechanisms of semantic annotation for scientific workflows.

Palavras-chave em inglês (Keywords): 1. Workflows. 2. Semantic and data processing. 3. Information management.

Área de concentração: Banco de Dados

Titulação: Mestre em Ciência da Computação

Banca examinadora: Profa. Dra. Claudia Maria Bauzer Medeiros (IC-Unicamp)
Prof. Dr. Luciano Antonio Digiampietri (EACH-USP)
Profa. Dra. Marta L. Queirós Mattoso (UFRJ)
Profa. Dra. Maria Beatriz Felgar de Toledo (IC-Unicamp)

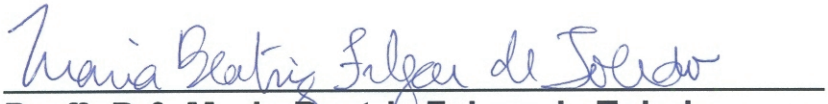
Data da defesa: 03/07/2009

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 03 de julho de 2009, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Luciano Antonio Digiampietri
EACH/USP



Profª. Drª. Maria Beatriz Felgar de Toledo
IC / UNICAMP

Profª. Drª. Claudia Maria Bauzer Medeiros
IC / UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Mecanismos de anotação semântica para workflows científicos

Arnaldo Francisco Vitaliano Filho[1]

Julho de 2009

Banca Examinadora:

- Profa. Dra. Claudia Maria Bauzer Medeiros
  Instituto de Computação - UNICAMP (Orientadora)
- Profa. Dra. Marta L. Queirós Mattoso
  COPPE - UFRJ
- Profa. Dra. Maria Beatriz Felgar de Toledo
  Instituto de Computação - UNICAMP
- Prof. Dr. Luciano Antonio Digiampietri
  EACH - USP
- Profa. Dra. Islene Calciolare Garcia (suplente)
  Instituto de Computação - UNICAMP

[1]Suporte financeiro de: Bolsa Capes – 2007, Bolsa Fapesp (Processo 07/53611-6) 2007 – 2008

Resumo

O compartilhamento de informações, processos e modelos de experimentos entre cientistas de diferentes organizações e domínios do conhecimento vem aumentando com a disponibilização dessas informações e modelos na Web. Muitos destes modelos de experimentos são descritos como workflows científicos. Entretanto, não existe uma padronização para a sua descrição, dificultando assim o reaproveitamento de workflows e seus componentes já existentes.

A dissertação contribui para a solução deste problema com os seguintes resultados: a análise dos problemas relativos ao compartilhamento e projeto cooperativo de workflows científicos na Web, análise de aspectos de semântica e metadados relacionados a estes workflows, a disponibilização de um editor Web de workflows usando padrões WFMC e, o desenvolvimento de um modelo de anotação semântica para workflows científicos. Com isto, a dissertação cria a base para permitir a descoberta, reuso e compartilhamento de workflows científicos nas Web. O editor permite que pesquisadores construam seus workflows e anotações de forma online, e permite o consequente teste, com dados externos, do sistema de anotações.

Abstract

The sharing of information, processes and models of experiments is increasing among scientists from many organizations and areas of knowledge, and thus there is a need for supply mechanisms of workflow discovery. Many of these models are described as scientific workflows. However, there is no default specification to describe them, which complicates the reuse of workflows and components that are available.

This thesis contributes to solving this problem by presenting the following results: analysis of issues related to the sharing and cooperative design of scientific workflows on the Web; analysis of semantic aspects and metadata related to workflows, the development of a Web-based workflow editor, which incorporates our semantic annotation model for scientific workflows. Given these factors, this work creates the basis to allow the discovery, reuse and sharing of scientific workflows in the Web.

Agradecimentos

Gostaria de agradecer a

Meus pais e meu irmão, por todo o apoio, incentivo e amor.

Minha orientadora, por me mostrar que conseguimos atingir nossos objetivos se o fizermos com excelência, ética e paixão.

Meus colegas do LIS e colegas de faculdade, por partilharem seus conhecimentos e amizade.

Os velhos e novos amigos, pela amizade e por fazerem valer a pena todo o esforço, estudo e trabalho.

Capes, CNPq e FAPESP (Processo 07/53611-6) por financiarem esta dissertação.

A banca, Prof. Luciano Antonio Digiampietri, Profa. Maria Beatriz Felgar de Toledo e Profa. Marta L. Queirós Mattoso, pelas sugestões para aprimoramento do texto e do trabalho.

Sumário

Lista de Figuras

Capítulo 1

Introdução

Uma das premissas do trabalho cooperativo é a possibilidade de compartilhamento e reuso de documentos, processos e modelos. Quando este trabalho ocorre na Web, a cooperação, o compartilhamento e o reuso exigem vários mecanismos sofisticados que levem em conta aspectos específicos a ambientes distribuídos. Exemplos de problemas associados são a integridade dos objetos compartilhados, seu versionamento, replicação e identificação na Web.

Uma área em que há grande demanda de soluções para estes problemas é a chamada *e-Science* [21]. O termo denota o desenvolvimento de atividades e experimentos científicos usando a Internet. Há um grande número de domínios científicos hoje associados à e-Science, principalmente aqueles ligados a simulações matemáticas (física), processos biológicos (bioinformática, neurologia) ou gerenciamento de grandes volumes de dados coletados dinamicamente por sensores sofisticados (astronomia, física de partículas). Em todos esses domínios, está crescendo a prática de experimentos científicos projetados e executados virtualmente, de forma detalhada, trazendo desafios adicionais ao cenário de compartilhamento e reuso. Estas questões vêm sendo abordadas sob diversos prismas - por exemplo, definição de padrões para intercâmbio e publicação de dados, ou uso de ontologias para facilitar cooperação entre grupos com vocabulários distintos.

No trabalho científico, no entanto, além de compartilhamento de dados e documentos, há necessidade de disponibilizar procedimentos e modelos. Em particular, estes últimos vêm sendo freqüentemente descritos por meio dos chamados *workflows científicos*. Na Web, workflows científicos são usados para especificação e execução de simulações nos chamados "grids" de computadores. O workflow contém a especificação (por vezes incompleta) de atividades a serem executadas. Cada atividade corresponde à ativação de uma ou várias ferramentas computacionais, que são executadas por nós do grid ou, freqüentemente, pela ativação de serviços Web.

O problema de compartilhamento de modelos e processos é assim transformado no

problema de disponibilização de workflows científicos. A dificuldade, neste caso, consiste em proporcionar mecanismos que facilitem tal disponibilização. Em cenários típicos de e-Science, vários grupos de cientistas, de diferentes especialidades e vocabulários, precisam interagir intercambiando seus workflows. As soluções existentes se baseiam no conhecimento prévio, por parte dos cientistas, de repositórios que contenham os workflows que são potencialmente de interesse. Além do mais, tais workflows recebem identificadores muito particulares, de conhecimento de poucos grupos de pesquisa, sendo recuperados diretamente a partir desses identificadores. Isto diminui a probabilidade do real compartilhamento e divulgação de modelos e processos, que ficam restritos a grupos que trabalham em um mesmo projeto.

Esta dissertação aborda esta questão. O objetivo principal é especificar e desenvolver um conjunto de mecanismos que permitam o gerenciamento de workflows científicos na Web, com vistas, principalmente, a facilitar buscas e descobertas de workflows científicos na Web. Este gerenciamento deve levar em consideração aspectos previamente mencionados, como a sua anotação semântica para posterior recuperação.

A pesquisa partiu da plataforma para workflows científicos denominada WOODSS, desenvolvida no Laboratório de Sistemas de Informação (LIS) do IC- UNICAMP. Este sistema permite o armazenamento desse tipo de workflows, em diferentes níveis de abstração, em um banco de dados.

As principais contribuições desta dissertação são:

- Estudo dos problemas relativos ao compartilhamento e projeto cooperativo, na Web, de modelos de experimentos científicos expressos como workflows científicos;
- Projeto de um mecanismo de identificação e recuperação de workflows na Web, incluindo aspectos de anotação semântica. Com isto, cria a base para realização de descobertas de workflows científicos na Web;
- Desenvolvimento de um protótipo para anotação de workflows. Baseado no modelo de workflows do WOODSS, o protótipo permite edição, anotação e busca de workflows na Web.

A dissertação está organizada da seguinte forma. O Capítulo 2 introduz alguns conceitos utilizados na pesquisa, além de apresentar trabalhos correlatos na área. O Capítulo 3 apresenta o modelo de anotação semântica, e como ele se relaciona com o modelo de workflows. O Capítulo 4 apresenta os aspectos de implementação do protótipo e alguns casos de uso de inserção e busca de workflows. O Capítulo 5 contém as conclusões e trabalhos futuros.

Capítulo 2

Conceitos Básicos e Trabalhos Correlatos

2.1 Introdução

Este capítulo apresenta conceitos básicos que foram utilizados neste trabalho, além de alguns trabalhos correlatos. A seção 2.2 apresenta sistemas de workflows, dando ênfase a workflows científicos, que são o modo como a modelagem de experimentos científicos é representada na nossa abordagem. A seção 2.3 apresenta o modelo de workflows científicos adotado pelo sistema WOODSS, desenvolvido no LIS - Unicamp, que é o modelo utilizado neste trabalho. A seção 2.5 apresenta conceitos de Web Semântica, mostrando como documentos genéricos disponibilizados na Web devem ser anotados de modo a facilitar o entendimento tanto de humanos quanto de agentes de software. A seção 2.4 apresenta o conceito de Ontologias, uma maneira de se expressar semântica em sistemas na Web. Por fim, a seção 2.7 apresenta alguns trabalhos relacionados a esta dissertação.

2.2 Sistemas de Workflows

É crescente o uso de sistemas baseados em workflows tanto para descrever processos corporativos quanto científicos. As vantagens mais conhecidas da aplicação dessas técnicas incluem o aumento na eficácia na utilização de recursos, a automação de tarefas e a melhora em aspectos organizacionais como a diminuição de falhas de alocação ou paralisação de tarefas pelo aumento da capacidade de gerenciamento [2, 44]. Além das vantagens relacionadas com o gerenciamento de tarefas individuais, tem-se percebido vantagens relacionadas ao gerenciamento de dados e de processos como um todo [36, 37]. Muitas vezes, workflows são usados como meios para invocar e compor serviços, desta forma exe-

cutando tarefas complexas (ver, por exemplo, a discussão em [10]). Neste trabalho, o foco de interesse é em sistemas de workflows científicos. Os conceitos relacionados a esses sistemas são melhor definidos a seguir.

Processos científicos são o foco da dissertação, ou seja, processos relacionados ao desenvolvimento de experimentos científicos. Processos podem envolver atividades em locais diferentes e demandar cooperação de vários parceiros. Isso exige aprimorar a representação para especificações de workflows, além de uma maneira mais eficiente de se anotar tais workflows, para que estes sejam melhor indexados, facilitando assim a sua disponibilização em sistemas distribuídos na Web. Dessa necessidade surgiram alguns consórcios para tratar de aspectos de padronização. Dentre eles, os de maior destaque são a Workflow Management Coalition (WfMC) [56] e a Business Process Management Initiative (BPMI) [7].

2.2.1 Conceitos Básicos

Esta seção define alguns conceitos básicos relacionados a workflows, a partir do modelo de referência da WfMC, uma organização internacional que busca promover e desenvolver o uso de workflows, propondo modelos e padrões para a construção dos mesmos. Ela existe desde 1993, e teve um papel importante na definição inicial dos conceitos básicos, de um modelo de referência que engloba representação de dados e de uma arquitetura genérica para sistema de workflows [25], para prover interoperabilidade entre diferentes sistemas de workflows. Este modelo já foi estendido de várias maneiras - e.g. para prover mais informações semânticas ou facilitar o compartilhamento de experimentos científicos. Além disso, para aumentar a interoperabilidade, várias ferramentas e sistemas invocados por workflows são agora encapsulados por Serviços Web.

Entretanto, a busca por especificações genéricas prejudicou a padronização, deixando lacunas importantes como a representação de algumas estruturas complexas em workflows. A BPMI (*Business Process Management Initiative*) é uma iniciativa mais recente que busca tratar da representação e colaboração interinstitucional por meio de processos de negócios (representados como workflows). Essa entidade tem um foco maior em aspectos computacionais e tem como objetivo dar apoio à colaboração. Conceitos básicos definidos pela WfMC e BPMI são: processo, workflow e atividade.

Processo. Um processo é um conjunto de um ou mais procedimentos interdependentes que, coletivamente, cumprem um objetivo, normalmente dentro do contexto de uma estrutura organizacional definindo papéis (funções) e relacionamentos. É usual a divisão entre processos de negócio e processos científicos, diferenciados pelo contexto em que estão inseridos, respectivamente: envolvendo trocas comerciais e/ou monetárias; e contemplando a realização de experimentação científica.

Workflow. Um workflow representa a automação de um processo, parcial ou completamente, durante a qual documentos, informações ou tarefas são passadas de um participante a outro para a realização de alguma ação de acordo com um conjunto de regras procedurais. Pode-se diferenciar dois estados de um workflow, como notado em [4]: estático, que é uma representação de um processo; e dinâmico, onde um modelo é instanciado e posto em execução sendo alimentado com dados para a obtenção de resultados concretos. Um Sistema Gerenciador de Workflows, ou SGWf, é um sistema automatizado para definir e instanciar workflows, gerenciando sua execução. Essa execução pode ocorrer em um ou mais motores de execução, que são capazes de interpretar as definições de um workflow e interagir com os participantes desse workflow (humanos ou sistemas automáticos).

Atividade. Uma atividade é uma descrição de uma parte do trabalho a ser realizado dentro de um processo. Uma atividade pode ser básica (ou atômica), representando uma ação indivisível para o SGWf, ou pode ser um sub-fluxo, composto de outras atividades. Além disso, uma atividade pode ser automatizada ou manual (dependente de intervenção humana). Atividades são os elementos básicos da construção de workflows.

2.2.2 Workflows Científicos

Um workflow científico é a especificação de um processo que descreve um experimento científico [28], e vem sendo cada vez mais utilizado como um meio de especificar e coordenar a execução de experimentos que envolvem participantes de diferentes organizações. Tais workflows permitem a representação e o apoio a tarefas complexas que utilizam fontes de dados heterogêneas e diferentes tipos de softwares, e se diferenciam de workflows corporativos em diversos aspectos.

Uma área em que há grande uso de workflows científicos é a chamada e-Science [34, 21]. Há um grande número de domínios científicos hoje associados à e-Science, como física, biologia [54], astronomia ou medicina. Em todos esses domínios, está crescendo a prática de experimentos científicos projetados e executados virtualmente.

A documentação de um experimento científico necessita de tratamento especial, pois tal tipo de experimento é caracterizado por um alto grau de flexibilidade e apresenta uma quantidade muito maior de incertezas e exceções do que processos de negócio comuns. Workflows científicos estendem as funcionalidades de workflows de negócio, cobrindo os seguintes aspectos: incompletude, reuso parcial, abandono/repetição e modificações dinâmicas, e rastreamento de processos inválidos. Além disso, em certas áreas da ciência não há consenso bem definido de como algumas tarefas devem ser executadas. Com isso, existe um grande número de possibilidades de composição de workflows para uma mesma tarefa. Alguns workflows podem não terminar corretamente, ou forne-

cer resultados errôneos ou não conclusivos. É preciso, então, que além dos registros de execuções bem sucedidas, registros de execuções defeituosas também sejam mantidos.

A motivação para o gerenciamento de workflows em aplicações científicas é auxiliar o controle de experimentos, e disponibilizar para os usuários informações sobre como experimentos são realizados [32]. Mais recentemente, tais motivos foram reforçados pelo uso de workflows na Web, como apoio ao trabalho cooperativo.

2.3 Modelagem de workflows científicos no WOODSS

O WOODSS (WOrkflOw-based spatial Decision Support System) é um sistema baseado em workflows científicos [32], cujo objetivo, inicialmente, era auxiliar a tomada de decisões em processos que envolvem informações geoespaciais. Houve uma evolução, e, atualmente, o WOODSS suporta modelos em qualquer área de conhecimento.

A dissertação utiliza o modelo definido por Pastorello [29], adotado no sistema WOODSS. Este modelo permite armazenar em um único repositório a representação de workflows e seus componentes, em diferentes níveis de abstração. A diferenciação entre os níveis de abstração e a possibilidade de se estender modelos abstratos para modelos concretos é uma característica fundamental dos mecanismos de anotação semântica desenvolvidos nesta dissertação. O modelo utiliza padrões de design [1], além de ser compatível com o modelo de referência proposto pela WfMC [25]. Sua generalidade e concepção estão discutidos no trabalho de Pastorello [29].

O modelo considera diferentes níveis de abstração de workflows. Como mostrado na Figura 2.1, retirada de [29], os níveis considerados são os seguintes: (i) componentes abstratos, (ii) workflows abstratos, e (iii) workflows concretos ou executáveis.

Este modelo permite a construção de workflows de forma incremental. O usuário deve: (i) especificar os tipos das atividades utilizadas na construção do workflow; (ii) combinar as atividades para se obter um workflow abstrato; (iii) associar as atividades a serviços web ou códigos executáveis, e a fontes de dados; (iv) executar o workflow por meio de uma máquina de execução [39].

O primeiro nível, de especificação abstrata de componentes, Figura 2.1(a), descreve componentes que podem ser utilizados em um workflow, i.e., atividades, seus dados de entrada e de saída. O segundo nível, Figura 2.1(b) refere-se à especificação de um workflow abstrato, obtido por meio da combinação de atividades, definidas no nível anterior, que são ligadas umas às outras pelos seus conectores. No terceiro nível, Figura 2.1(c), o workflow é instanciado, já possui as ligações com as fontes de dados, e estão especificados quais agentes devem ser invocados para realizar as operações das atividades. Neste nível, o workflow é dito ser *executável* ou *concreto*.

Tais especificações de workflows são armazenadas em um banco de dados relacional,

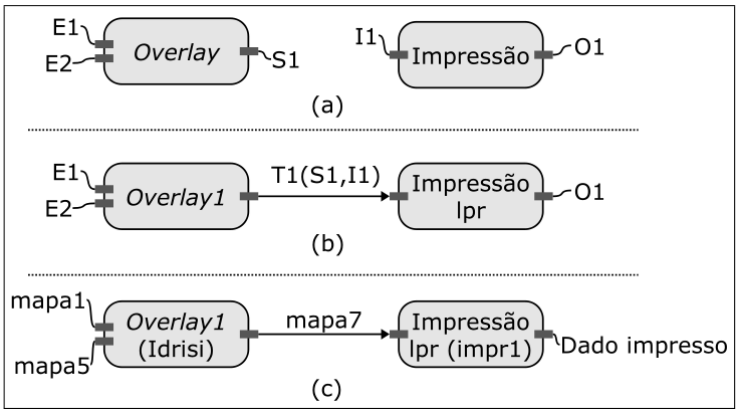


Figura 2.1: Níveis de abstração na especificação de um workflow [29]

cujo modelo entidade-relacionamento é apresentado na Figura 2.2. O modelo de dados é dividido em 5 grupos lógicos, que agrupam entidades relacionadas aos diversos níveis de abstração de workflows. Os grupos são descritos a seguir:

- **Workflow** é o cerne do modelo. Nele estão as entidades que representam o nível *workflows abstratos*. Um workflow (*Workflow*) é constituído por atividades (*Activity* - *BasicActivity*), que por sua vez possuem conectores (*ActivityConnector*). Um conector de workflow (*WorkflowConnector*) representa as portas de entrada e saída de um workflow. As atividades se interligam por meio dos conectores formando transições (*Transition*), que representam o fluxo de execução do workflow. Por fim, cada atividade é executada por algum agente de software ou humano segundo um papel (*Role*).

- **Data** encapsula os dados que são utilizados no workflow. Cada conector de atividade possui um tipo de dados (*DataType*), e a entrada/saída do dado (*Data*).

- O **Component** representa o nível de *workflows concretos*. São representados componentes de software (*Content*), serviços Web (*WebService*) ou qualquer outro componente que realiza a tarefa especificada por uma atividade. Um componente (*DC*) possui uma interface de serviços (*Interface*) que realiza uma série de operações (*Operation*). Uma operação realiza as tarefas especificadas em uma Atividade (definida no modelo abstrato). Cada conector da atividade é representado por um parâmetro (*Parameter*) da operação.

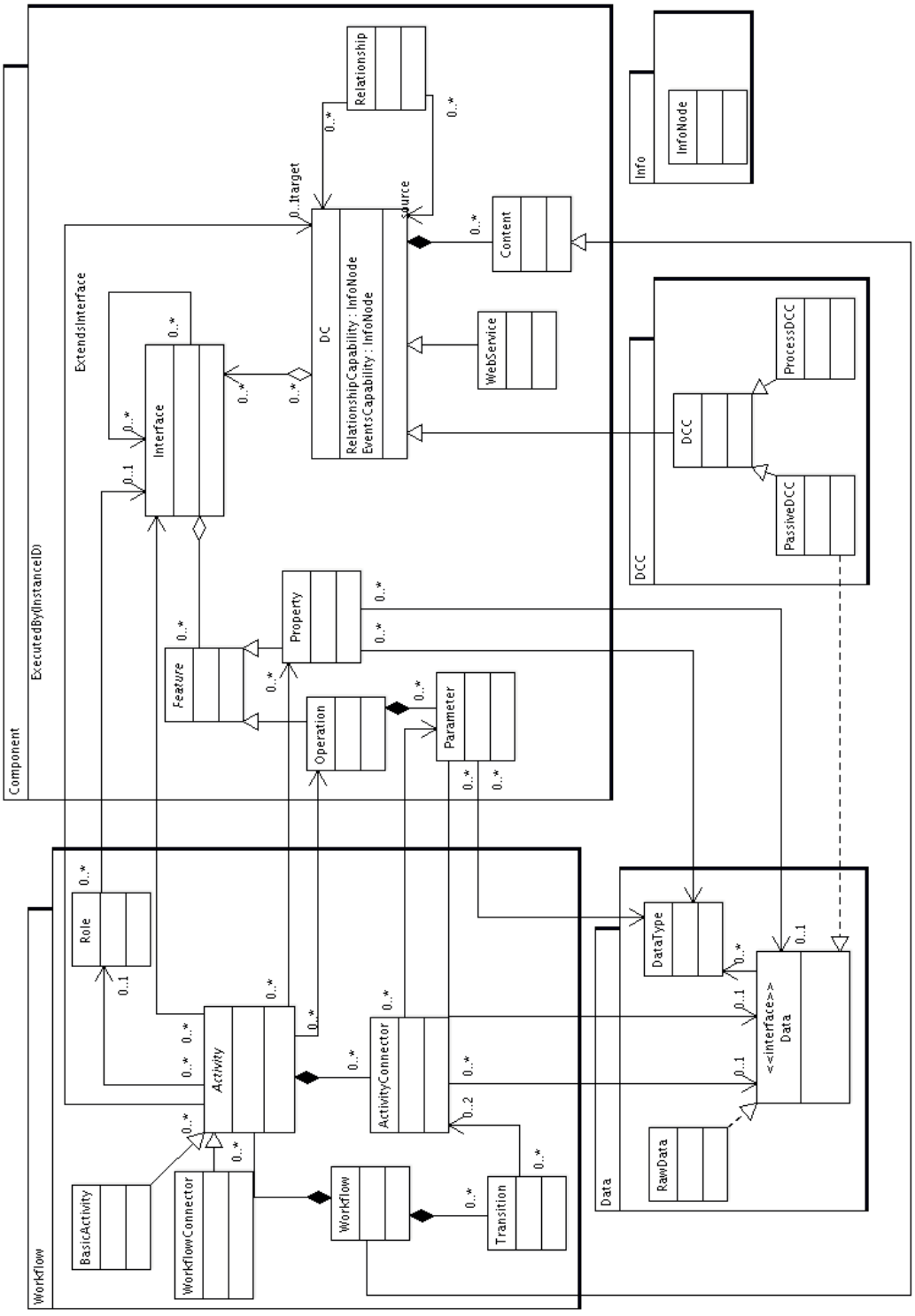


Figura 2.2: Modelo de dados - WOODSS

- **DCC** representa um modelo de encapsulamento. Dentre outros artefatos, um workflow pode ser encapsulado no chamado *Digital Content Component*, definido em [52].
- **Info** representa informações textuais (*InfoNode*) que podem ser associadas a qualquer entidade do modelo.

Para a implementação do protótipo foram consideradas todas as classes dos grupos *Workflow* e *Data*, além das classes *Parameter, Operation, Property, Feature, Interface* e *DC* do módulo *Component*. Essas são as entidades necessárias para se mapear especificações de workflows (nos 3 diferentes níveis de abstração) em um banco de dados relacional. As demais entidades fogem ao escopo deste trabalho.

2.4 Ontologias

Uma ontologia define um domínio, ou, mais formalmente, especifica uma conceitualização acerca dele [23]. Um tipo comum de ontologias são aquelas organizadas em hierarquias de conceitos (ou taxonomias). Como ontologias não refletem nenhum formalismo específico, e representam com freqüência um vocabulário comum entre usuários e sistemas, são consideradas como a materialização de um certo nível de conhecimento.

Podemos, também, definir o termo ontologia a partir dos requisitos para possibilitar sua aplicação em informática:"*Uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada.*" [48].

Ontologias estão presentes em muitos sistemas, ferramentas e produtos de manipulação de informação, sendo representadas como hierarquias de palavras-chave, conceitos, e muitas outras formas. Contudo, ontologias devem possuir um significado e abrangência muito mais profundos do que as simples hierarquias de conceitos e palavras-chave empregadas por muitas máquinas de busca. Segundo [16], há vários tipos de ontologias:

- *Ontologias de representação*: definem, de forma declarativa, as primitivas de representação como frames, axiomas, atributos e outros;
- *Ontologias gerais*: trazem definições abstratas necessárias para a compreensão de aspectos do mundo, como tempo, processos, papéis, espaço, seres, coisas, etc.
- *Ontologias centrais (core ontologies)* ou *genéricas de domínio*: definem os ramos de estudo de uma área ou conceitos mais genéricos e abstratos desta área.
- *Ontologias de domínio*: tratam de um domínio mais específico de uma área genérica de conhecimento, como direito tributário, microbiologia, etc.

- *Ontologias de aplicação*: procuram solucionar um problema específico de um domínio, como por exemplo, identificar doenças do coração, a partir de uma ontologia de domínio de cardiologia. Normalmente, referenciam termos de uma ontologia de domínio.

O modo como ontologias são definidas e elaboradas vem sendo utilizado em estudos que buscam desenvolver ferramentas de apoio à construção de ontologias e às atividades-fim que as utilizam. Estas ferramentas compõem ambientes para a manipulação de ontologias: editores, servidores, repositórios e ferramentas para consolidação e tradução de ontologias, que têm como objetivo facilitar a construção, disponibilização e compartilhamento de ontologias, que podem ser utilizadas por diferentes grupos de pesquisa de áreas afins, ainda que distantes geograficamente.

No contexto de sistemas gerenciadores de workflows, ontologias podem ser utilizadas por mecanismos de anotação. Modelos de anotação devem ter uma compreensão comum de conceitos e seus significados. Ontologias podem satisfazer este requisito, provendo uma representação de conceitualizações compartilhadas de um domínio genérico, neste contexto, representação de processos científicos, além de um vocabulário controlado e compartilhado que pode ser utilizado na comunicação entre aplicações [24].

2.5 Web Semântica

A Web Semântica é uma extensão da Web, na qual o conteúdo disponível na Web pode ser expressado em diversas formas, podendo ser lido por agentes de software, permitindo assim que informações possam ser pesquisadas, compartilhadas e integradas de forma mais fácil [55]. Esta preocupação tem propiciado pesquisas em ambientes adequados à publicação de dados, principalmente por causa da possibilidade de se acoplar semântica a esses dados por meio de descrições/metadados. Mais recentemente, ontologias estão aparecendo como uma solução complementar para permitir que grupos distintos compartilhem recursos (ver Seção 2.4).

Entretanto, o processo de construção da Web Semântica ainda está no estágio inicial; do ponto de vista de implementação, ainda é necessária intervenção humana na publicação e escolha das fontes de dados para alimentar aplicações. Ainda pela pouca maturidade da tecnologia, a maioria dos padrões ainda não estão definidos, mas alguns deles, como linguagens para ontologias, já apareceram. Segundo [47], os seguintes passos mostram, em alto nível, a direção para onde a Web Semântica está convergindo: (i) provimento de uma sintaxe comum, para o entendimento de agentes de software, (ii) estabelecimento de um vocabulário comum, (iii) acordo sobre uma linguagem lógica, (iv) utilização da linguagem para troca de informações.

A Figura 2.3, retirada de [47], mostra a estrutura da pilha de padrões para descrição de dados em camadas:

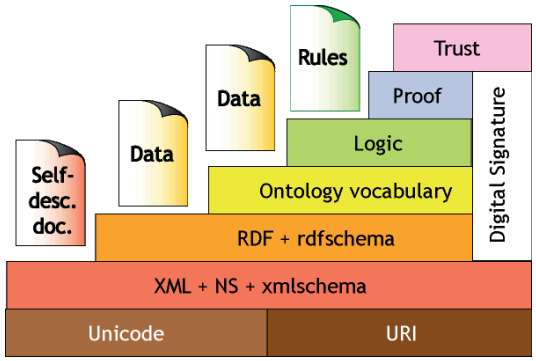


Figura 2.3: Padrões para a Web Semântica [47]

Nas duas camadas inferiores, é provida uma sintaxe comum. Na primeira camada se encontram os URI's *(Unified Resource Identifiers)*, que provêm um meio de identificação unívoca de recursos abstratos ou físicos. A codificação UNICODE é utilizada para a compatibilidade no processamento de texto. Na segunda camada encontra-se o padrão XML *(eXtensible Markup Language)* utilizado para a representação semi-estruturada dos dados, o padrão associado XMLSchema, que define uma gramática para documentos XML válidos, e os NS*(namespaces)*, utilizados para remover qualquer ambiguidade entre termos com o mesmo nome, mas de domínios diferentes.

A camada de RDF *(Resource Description Framework)* pode ser vista como a primeira camada que faz parte, de fato, da Web Semântica. RDF provê a base semântica, sendo a fundação para o processamento de metadados e o provimento da interoperabilidade de aplicações que trocam informação na Web [49]. A camada de Vocabulário de Ontologia descreve, de forma não ambígua e formal, conceitos, características, e relacionamentos entre conceitos, de forma a construir uma base de conhecimento e a terminologia utilizada para descrever documentos na Web. Para tal descrição, utiliza linguagens como a OWL *(Web Ontology Language)*, padrão proposto pela W3C. A camada de Lógica estabelece um sistema lógico, por meio do qual a camada de Prova pode realizar inferências sobre os dados representados em camadas inferiores. A camada de Assinatura Digital dá aos dados um certificado, garantindo sua origem, por exemplo. A Assinatura Digital combinada com a Prova assegura a validade da informação a ser entregue na camada de Confiança.

A dissertação utiliza desde a primeira camada até a camada de vocabulários de ontologias. Dentro dos padrões de Web Semântica, as camadas Lógica, Prova e Confiança

precisam ser melhor desenvolvidas. Tópicos de segurança e confiabilidade fogem ao escopo da dissertação.

2.6 Anotações Semânticas

De acordo com NISO [41] um metadado é uma informação estruturada que descreve, explica, localiza, ou seja, facilita a identificação, uso ou gerência de uma informação. Metadados – muitas vezes chamado de dado sobre dado ou informação sobre informação – descreve uma informação, em parte, no todo ou em uma coleção.

"Anotar" significa adicionar uma nota, comentar. Em computação, uma anotação é utilizada para descrever um recurso (geralmente, uma anotação é representada na forma textual) a o que este recurso representa, em termos de conceitos formais (e.g., utilizando entidades de uma ontologia) [33]. Uma anotação é representada como um conjunto de metadados que provêm a cada entidade anotada uma referência para uma ontologia única na Web (e.g. uma URI).

Em outras palavras, anotações identificam formalmente entidades por meio do uso de conceitos e relacionamentos entre conceitos, de uma maneira que possam ser processados por máquinas. Uma maneira de se promover interoperabilidade entre sistemas, é se utilizando entidades de uma ontologia de domínio como estes conceitos. Por exemplo, uma anotação pode relacionar a palavra laranja que ocorre em um texto com uma ontologia que identifica esta palavra como um conceito abstrato de fruta (ao contrário do conceito de cor). Isso ajuda a remover ambiguidades de significado. O aumento da qualidade de informações retornadas em uma busca e interoperabilidade são alguns dos beneficios da adoção de anotações semânticas.

Nós consideramos duas proposições para descrever as especificações de workflows: metadados e anotações. Metadados possuem ma estrutura bem definida, e anotações são notas adicionadas como comentários ou explicações, sem nenhuma estrutura definida ou limite de valores. Entretanto a grande flexibilidade de anotações não facilita processamentos automatizados, ao contrário de metadados, que mesmo sendo menos flexíveis, são utilizados em funções de indexação e busca.

Nós combinamos ambas soluções em anotações semânticas, as quais possuem estruturas de metadados e parte de seus conteúdos são definidos por referências a ontologias escolhidas arbitrariamente pelos usuários. As ontologias podem ser estendidas conforme necessidade, o que provê maior informações sobre o contexto. Automação é disponibilizada por operações relacionadas a ontologias (e.g. alinhamento de termos). Definimos anotações semânticas a seguir.

Unidades de Anotação. Uma unidade de anotação é uma tripla <o,p,v>, onde **o** representa o objeto sendo descrito, **p** representa uma propriedade que o descreve, e **v**

representa um valor atribuído à propriedade.

Anotação Semântica. Uma anotação semântica M é um conjunto de uma ou mais unidades de anotação, com no mínimo uma unidade tendo como seu objeto a entidade sendo descrita. Uma anotação semântica é materializada como um grafo RDF, que é representado como um conjunto de triplas RDF (objeto - propriedade - valor).

Nossa definição de anotação semântica entra na categoria de formal e explícita, similar à definição de [38].

2.7 Trabalhos Correlatos

Nesta seção apresentamos trabalhos relacionados a associações de semântica a workflows, e apresentamos alguns sistemas gerenciadores de workflows existentes.

DAGMan e Pegasus [17] são dois sistemas que são comumente referenciados como sistemas de workflows, e têm sido aplicados em ambientes de Grid. DAGMan provê um *engine* que gerencia *jobs* Condor, organizados como grafos direcionados acíclicos (DAGs), nos quais cada aresta corresponde a uma precedência explícita de uma tarefa. Ambos sistemas focam o agendamento e execução de tarefas que levam muito tempo para executar.

Taverna [43] é um sistema de workflow de código aberto, particularmente focado em aplicações e serviços de bioinformática, baseado na linguagem XScufl (*XML Simples Conceptual Unified Flow*). Kepler [35] é sistema de workflows científicos que contém uma ferramenta de modelagem visual de workflows escrita em Java. Triana [11] é um sistema de workflows para coordenação e execução de coleções de serviços. O sistema myExperiment [15] é um sistema de criação e compartilhamento de workflows que se baseia em tags para que usuários compartilhem e encontrem workflows.Todos estes sistemas possuem alguns tipos de interface visual que permite composição gráfica de workflows.

A evolução dos próprios workflows é vital em análises científicas. VisTrails [9, 3] possui essa noção de evolução, e implementa um mecanismo de histórico, que mantém versões de um workflow, e de seus dados gerados e fornecidos. Isso permite que cientistas retrocedam a etapas anteriores, apliquem diferentes dados, e com isso, comparem os resultados das diferentes versões.

A Microsoft Windows Workflow Foundation (WWF) [53] oferece um *framework* genérico para a criação e execução de workflows. Este *framework* permite integrar diferentes componentes dentro de uma aplicação, permitindo que um workflow seja integrado de forma nativa em uma aplicação. A ideia fundamental do WWF é o fato de que cada atividade é modelada como um programa que pode ser parado e reiniciado de forma arbitrária, e a invocação de uma atividade é organizada de maneira assíncrona. Essas características permitem que a execução de um workflow seja armazenado de maneira persistente, e reiniciada em um tempo arbitrário posterior.

Swift [57] é um sistema que combina workflows científicos com computação paralela. É uma ferramenta de programação paralela para especificação, execução e gerenciamento rápidos e confiáveis de workflows. Swift utiliza uma solução estruturada para especificação, agendamento e execução. Consiste em uma linguagem simples, chamada SwiftScript, utilizada para especificação de computações paralelas complexas, baseadas em conjuntos de dados dinâmicos, de grande escala e representados em diversos formatos.

Esta dissertação utiliza técnicas de Web semântica e ontologias para especificar o mecanismo de anotação de workflows, visando facilitar o reuso de especificações de experimentos por cientistas de diferentes domínios e instituições. As dissertações de [31, 50] defendidas no IC estão diretamente ligadas a estes itens. Seus focos são recuperação de modelos armazenados em sistemas de workflows e utilização de metadados para anotação de workflows.

Kaster [31] combina técnicas de inteligência artificial (*Case-Based Reasoning – CBR*) e sistemas espaciais de apoio à decisão (*Spatial Decision Support Sistems – SDSS*) para recuperação de modelos ambientais, armazenados como workflows. CBR é uma técnica que reutiliza casos precedentes para resolver problemas. A idéia básica consiste em considerar workflows como *"cases"* de experimentos e usar técnicas de inteligência artificial para comparar os casos. Esta solução de recuperação é mais flexível que a busca por palavras-chave. Entretanto, esta técnica é muito cara computacionalmente e limita o compartilhamento a usuários de um mesmo domínio e perfil, devido a restrições de vocabulário e metodologia de trabalho.

Rocha [50] se concentra em especificar um padrão de metadados associados a workflows científicos para a documentação de atividades de planejamento ambiental, associado ao WOODSS. Seu padrão aproveita características de alguns padrões existentes, como *Dublin Core* [27] e FGDC [14]. Em adição, o WOODSS foi estendido, de forma a acoplar os metadados aos workflows. O padrão proposto fornece uma base para a interoperabilidade, resolvendo problemas no nível sintático. Entretanto, a interoperabilidade semântica não é abordada no trabalho. Os metadados são descritivos, sem qualquer estrutura hierárquica entre conceitos, e sem vocabulários pré-definidos, como em ontologias. O tratamento semântico das informações não é realizado.

Ambos os trabalhos visam facilitar a identificação e reuso de modelos relacionados a atividades ambientais, que, apesar de serem abrangentes, se restringem a um único domínio.

Em [52] os autores propõem um modelo para a construção de artefados digitais reutilizáveis, os DCCs (*Digital Content Components*). Os componentes podem encapsular tanto descrições de processos executáveis em um computador (e.g. sequência de instruções ou planos) como conteúdos centrados em dados (e.g. arquivos texto ou vídeos). O modelo utiliza ontologias descritivas e taxonômicas para especificar os artefatos. Esta

descrição permite que sejam adicionados metadados que agregam semântica aos componentes. Além disso, outro foco do trabalho é a descoberta de DCCs para reuso. Em uma busca por componentes, o modelo utiliza técnicas de *Ranking* para comparar a consulta com metadados dos componentes procurando por similaridades.

Nosso trabalho propõe a especificação de um mecanismo de anotação mais genérico do que em [50] e [8], não se restringindo ao domínio ambiental. Como em [52], adotamos ontologias para especificar metadados utilizados na anotação dos workflows.

2.8 Conclusões

Este capítulo apresentou os principais conceitos que serão utilizados no decorrer da dissertação. Foram apresentados workflows científicos, que são utilizados como meio de representação de experimentos científicos de forma virtual. Além disso, foi apresentado o conceito de Web Semântica, em especial sua necessidade de estratégias de anotação, utilizadas para agregar informações relevantes, de forma a facilitar a recuperação e busca de workflows. Foi apresentado o conceito de ontologias, que podem ser usadas na implementação de anotações semânticas em sistemas web. E por último, o capítulo apresentou alguns trabalhos relacionados a sistemas de workflows, identificação, anotação e recuperação de workflows científicos.

Capítulo 3

Anotação Semântica em Workflows Científicos

Este capítulo apresenta o modelo proposto de anotação semântica em workflows científicos, que visa facilitar sua identificação e recuperação. Este modelo foi concebido para ser utilizado em conjunto com a modelagem de workflows em vários níveis de abstração, descrita na seção 2.2. A seção 3.1 dá uma visão geral do modelo. A seção 3.3 apresenta o modelo de anotação semântica referente ao nível de componentes de um workflow. A seção 3.4 apresenta o modelo de anotação referente aos níveis de workflows abstratos. A seção 3.5 discute perspectivas de busca, dada a presença das anotações. Por fim, as conclusões são apresentadas na seção 3.7.

3.1 Visão Geral do Modelo

A seção 2.2 mostra que workflows científicos podem ser descritos em diferentes níveis de abstração. A diferenciação entre os níveis de abstração e a possibilidade de se estender modelos abstratos para modelos concretos são características fundamentais dos mecanismos de anotação semântica desenvolvidos neste trabalho, permitindo a usuários anotar e buscar desde componentes de workflows até workflows executáveis. A seção 2.3 apresentou o modelo de workflows utilizado no WOODSS com 3 níveis principais de abstração de workflows: (i) componentes abstratos, (ii) workflows abstratos, e (iii) workflows concretos.

O modelo de anotação proposto endereça os 2 primeiros níveis de abstração, de forma a garantir que as características específicas de cada nível sejam cobertas. No nível de componentes, as anotações devem focar as atividades, seus conectores e os tipos de dados associados aos conectores. No nível de workflows abstratos, são descritos os metadados relativos ao workflow (e.g., domínio de aplicação e informações sobre autores e organizações).

3.2 Unidade de Anotação

Como visto na Seção 2.6, uma anotação semântica é um conjunto de unidades de anotação, representadas por triplas RDF. Uma unidade de anotação é definida por uma tupla do tipo <o,p,v>(*objeto, propriedade, valor*) [30]. Nosso modelo estende esta definição. Consideramos uma unidade de anotação, uma tupla do tipo <o,p,v,r>, onde r representa uma referência da propriedade descrita que é definida em uma ontologia.

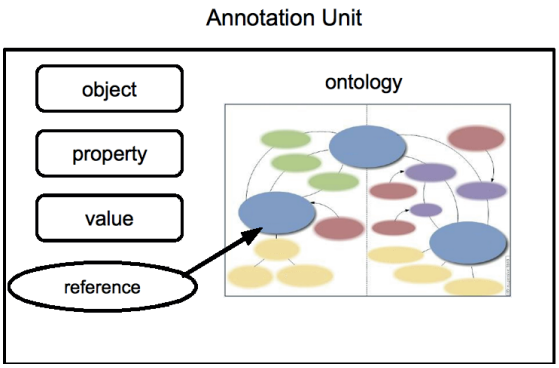


Figura 3.1: Unidade de Anotação

Definimos como objetos a serem anotados as seguintes entidades: workflow, atividade e conector. O atributo **o** representa o objeto anotado. O atributo **p** representa a propriedade definida e pode ser utilizada como parâmetro de busca em consultas por atividades ou workflows. O atributo **v** representa o valor atribuído à propriedade. Além do valor textual, é possível a associação da entrada a alguma ontologia já existente. O atributo **r** (*reference*) referencia uma ontologia, que define formalmente essa propriedade: a referência é um ponteiro para o termo correspondente da ontologia. A Figura 3.1 ilustra a estrutura descrita.

3.3 Anotação em Componentes

No primeiro nível de abstração, a construção de workflows consiste na especificação dos componentes básicos do workflow – as atividades, seus elementos de entrada e saída, e seu comportamento esperado. A Figura 3.2 mostra uma maneira de se representar os elementos básicos de uma atividade. As caixas rotuladas de **"C"** representam os conectores de entrada e saída. O comportamento esperado da atividade pode ser definido pela especificação das pré- e pós-condições.

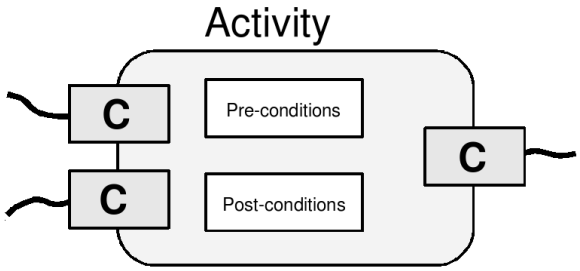


Figura 3.2: Atividade de um workflow

Esta seção define um modelo de anotação que permite aos usuários adicionar meta-informação às atividades, considerando os elementos (i) conectores e (ii) descrição da atividade.

3.3.1 Anotação nos Conectores

Os conectores representam as entradas e saídas de dados de uma atividade. Deste modo, a anotação em um conector deve permitir que o usuário possa descrever o(s) dado(s) correspondente(s), agregando informações que posteriormente poderão ser utilizadas para buscar e identificar o componente (atividade), e/ou as fontes de dados correspondentes.

Anotações fornecem detalhes sobre os conectores, adicionando semântica a eles. O modelo não foca um único domínio de conhecimento ou grupo de pesquisa. Desta maneira, é inviável pré-definir um conjunto fixo de anotações para os conectores. Assim, o modelo permite que usuários as definam de forma arbitrária. Como base para as anotações, são utilizadas as unidades de anotação, definidas na Seção 3.2. Existem tipos especiais de unidades de anotação para conectores, descritos a seguir.

Tipo de Dado

O **Tipo de dado** define qual formato é utilizado para a representação binária do dado. O modelo endereça os seguintes tipos: *integer, float, string, char, boolean* e *file*, onde *file* representa qualquer arquivo binário. Além disso, é possível cadastrar novos tipos de dados que estendem os tipos pré-definidos. Alternativamente, pode-se conceber a elaboração de uma ontologia de tipos. Com a criação de tal ontologia, utilizando dicionários de sinônimos [16], é possível verificar a compatibilidade entre tipos de dados

Como exemplo, consideramos um conector de entrada que deve receber um arquivo de imagem. A imagem pode ser considerada um arquivo; com isso, o tipo de dado *file* é

suficiente para representá-la. Entretanto, um usuário pode estender o tipo *file* e criar um tipo chamado *image*. Desta maneira, análogamente a conceitos de orientação a objetos, cria-se um subtipo para representar de forma mais precisa o tipo do conector (*image "extends" file*).

Natureza do Dado

Definimos aqui uma propriedade relevante de anotação para conectores, chamada **Natureza de Dado**, para que um usuario possa qualifica-los semânticamente. Uma seqüência de caracteres pode denotar o nome de uma região, em um modelo que calcula níveis de pluviosidade, ou um trecho de código de DNA, em um modelo de bioinformática.

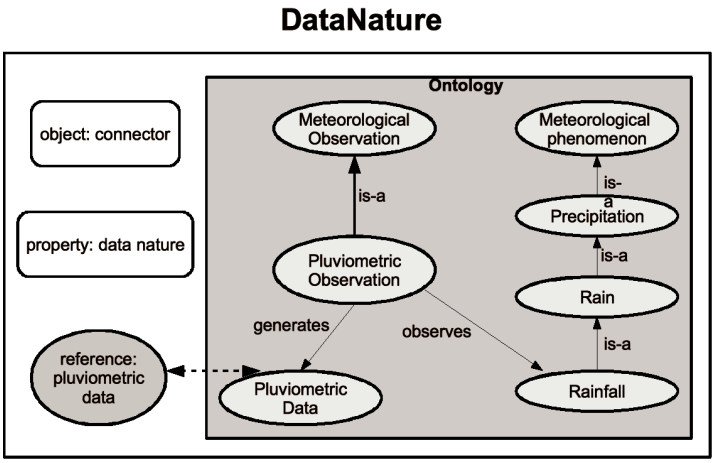


Figura 3.3: Natureza de dado de um conector

A Figura 3.3 ilustra este mecanismo. Considerada uma unidade de anotação, a natureza do dado utilizada no conector é definida em uma ontologia. No exemplo da figura, o conector anotado representa uma entrada de dados pluviométricos. Dados desta natureza são triviais para cientistas da área de meteorologia, mas para cientistas de outras áreas podem não ser. A ontologia relacionada fornece, assim, um melhor entendimento a respeito do conector – no exemplo, o significado semântico do dado que será utilizado no conector.

Um conector pode então reunir todas as anotações descritas na Seção 3.3.1. Como exemplo tomemos uma atividade que processa imagens de vegetação nativa, obtidas por satélites. A Tabela 3.1 mostra um possível conjunto de anotações do conector que recebe a imagem como entrada. O nome do conector é utilizado no exemplo para identificá-lo;

o **tipo de dado** é definido como *image*; a **natureza do dado** é definida como uma imagem de uma área de cobertura vegetal, e existe uma ontologia identificada pela URI, que possui a definição de "images:vegetation"; a anotação genérica **image_format** possui o valor "raster"; e por fim, a anotação **band** possui o valor "infra-red", referenciado na ontologia, "band.owl", como "band:infra-red". Vale notar que todas as unidades de anotação podem referenciar ontologias.

Object	**Property**	**Value**	**Reference**
connector_1	type	image	http://somesite.org/types.owl#file:image
connector_1	data_nature	vegetation	http://somesite.org/images.owl#vegetation
connector_1	image_format	raster	None
connector_1	creator	INPE	http://somesite.org/institutions.owl#INPE
connector_1	quality	good	http://somesite.org/qual.owl#quality:good'
connector_1	band	infra-red	http://somesite.org/band.owl#infra-red

Tabela 3.1: Exemplo de anotações de um conector

3.3.2 Anotação nas Atividades

As anotações nos conectores são focadas na descrição dos dados de entrada e saída das atividades. As anotações nas atividades devem, como principal objetivo, descrever o seu comportamento esperado. Elas devem fornecer informações para facilitar a descoberta de componentes no repositório.

Atividades podem ser atômicas ou complexas. Atividades complexas encapsulam workflows e suas anotações estão descritas na seção 3.4. Atividades atômicas devem possuir, no mínimo, as seguntes unidades de anotação: (i) palavras-chave, e (ii) classes de atividade. Palavras-chave definem, textualmente, termos relevantes à atividade e geralmente são utilizadas para indexação em sistemas de busca.

Atividades podem ser divididas de acordo com várias características (e.g. área de conhecimento, tipo de entrada de dados, tipo de serviço oferecido, etc). Esta classificação auxilia a busca por atividades, pois restringe o número de itens que são possivelmente relevantes em uma busca. Utilizamos unidades de anotação para caracterizar classes de atividades, ou seja, mostrar por meio de anotação, características da operação realizada pela atividade. A Figura 3.4 ilustra esta unidade de anotação. No exemplo da figura, a atividade realiza uma operação de *blast*, do campo da bioinformática. A ontologia associada facilita a descrição semântica do termo, correlacionando a atividade com outras atividades do mesmo domínio.

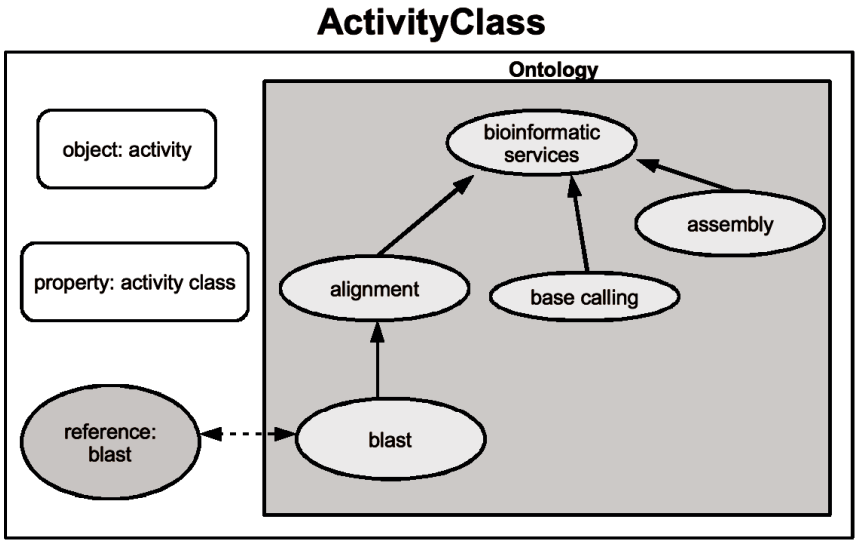


Figura 3.4: Classe de atividade - Modificado de [18]

3.4 Anotação em Workflows Abstratos

Em workflows abstratos, as atividades de um workflow já são definidas e estão conectadas, ou seja, já existe o conceito de transição, que representa a conexão de um conector de saída de uma atividade com um conector de entrada de outra.

Neste nível, o modelo de anotação descreve e caracteriza o workflow como um todo. Para isto, se adota nas unidades de anotação campos de metadados descritivos, baseados no padrão *Dublin Core* [27]. Cada entrada de metadados é modelada, como nas anotações em componentes, como uma unidade de anotação. A Tabela 3.2 mostra os campos propostos. Vale ressaltar que um workflow pode ser encapsulado como uma atividade composta. Neste caso, o workflow acumula os dois tipos de anotações: de componentes e de workflow.

Neste nível, as anotações reúnem informações básicas sobre o workflow, para serem utilizadas na busca, recuperação e indexação, com as seguintes propriedades:

- **Palavras-chave**
  É a forma mais tradicional de casamento de palavras em uma busca. As palavras-chave indicam os termos mais relevantes do workflow.

- **Autor**, **Organização** e **Contribuidor**
  Identificam a origem do workflow.

Atributo	Descrição
Palavras-chave	Uma lista de palavras-chave.
Autor	Pessoa responsável pela elaboração do workflow.
Organização	Organização à qual o criador do workflow pertence.
Contribuidor	Uma pessoa ou organização que fez contribuições na elaboração do workflow.
Domínio	Domínio de conhecimento da tarefa realizada pelo workflow (e.g. Biologia, Matemática, Geologia).
Subdomínio	Sub-domínio de conhecimento, uma área mais específica dentro de um domínio (e.g. Biologia Molecular, Estatística, Geologia Marinha).
Data de criação	Data de criação do workflow.
Data de atualização	Data da última modificação no modelo.
Descrição de atualização	Descrição textual sobre a última modificação no modelo

Tabela 3.2: Metadados - Workflows Abstratos

- **Domínio** e **Subdomínio**
  São utilizados para indexação e catalogação dos workflows. São úteis no caso de usuários que precisam encontrar modelos disponíveis na sua área de conhecimento. Domínio é a grande área de estudo como por exemplo, Biologia, Geologia, Matemática, Astronomia. Subdomínio é a especialização de uma grande área, como por exemplo Genômica (Biologia), Estudo de Composição de Solos (Geologia), Álgebra Linear (Matemática), etc. Neste caso, pressupõe-se uma ontologia específica, por exemplo, a Tabela de Áreas do Conhecimento do CNPq [12].

- **Data de criação**, **Data de atualização** e **Descrição da atualização**
  Registram a evolução da construção do workflow, funcionando como um mecanismo primitivo de versionamento. Uma possível extensão deste trabalho seria a implementação de um versionamento do modelo em si.

A Figura 3.5 mostra um exemplo de anotação de workflows e das atividades encapsuladas. A atividade “Classify Region” classifica uma imagem e “Weighted Overlay” realiza uma sobreposição das imagens classificadas. Apenas algumas unidades de anotação foram indicadas à guisa de ilustração. Como se verá no Capítulo 4, essas e outras anotações são preenchidas manualmente pelo usuário.

3.5 Perspectivas de Busca

A adoção de anotações semânticas abre novas possibilidades de busca e organização de workflows científicos.

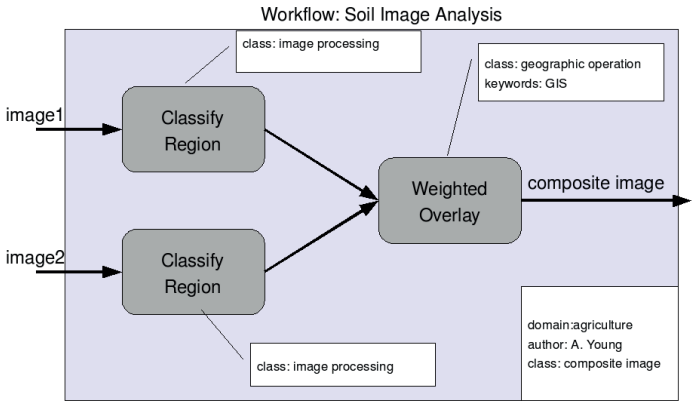


Figura 3.5: Workflow com anotação

3.5.1 Busca Textual

A busca textual é a busca tradicional baseada em casamento de palavras chave. Neste caso, um sistema de buscas pode recuperar workflows baseando-se nos termos das anotações.

Isto é implementável de forma padrão, por exemplo a partir de consultas a algum repositório onde estejam armazenadas as anotações (ver Seção 4.1).

3.5.2 Busca Semântica

Tendo em vista que unidadese anotação referenciam ontologias, torna-se agora possível buscar workflows e seus componentes a partir de operações sobre ontologias. Por exemplo, dado o conjunto de ontologias SWEET [40] da NASA, de fenômenos meteorológicos e outros, é possível descobrir, usando alinhamento de ontologias [16], workflows que se ocupam de um certo sub-domínio.

Para isto, a dissertação considera usar o Aondê [16], um serviço Web que realiza busca, emparelhamento, diferença, construção de visões, ranking e gerenciamento de ontologias. Estas podem ser recuperadas na Web ou construídas de forma *ad hoc* por usuários, sendo todas armazenadas em um repositório próprio do Aondê, para facilitar seu gerenciamento.

Com o Aondê há dois cenários de construção de anotações semânticas: (i) as ontologias são referenciadas diretamente pelo especialista que insere anotações; ou (ii) o especialista usa o Aondê para buscar ontologias apropriadas e recebe a referência como retorno da busca.

Da mesma forma, as operações providas pelo Aondê permitem uma grande gama de tipos de busca por workflows. A seguir são destacadas algumas possibilidades de busca a

partir de algumas destas operações.

Operação de busca. A busca permite recuperar termos em ontologias, ancestrais ou descententes de termos. Assim, pode-se procurar não apenas componentes anotados com um certo termo, mas também com ancestrais ou descendentes daquele termo. Inicialmente, o sistema retorna a subárvore de interesse e, a seguir, busca-se nas unidades de anotação do repositório de workflows por coincidência de referências.

Operação de alinhamento. Dado um workflow/componente do repositório, deseja-se encontrar que outros workflows/componentes abordam conceitos relacionados. Este tipo de busca pode ser implementado utilizando a operação de alinhamento do Aondê, que estabelece relacionamentos de equivalência entre termos de ontologias diferentes, gerando uma nova ontologia. Para isto, é necessário informar as duas ontologias a serem alinhadas – aquela usada na anotação do workflow de interesse e alguma outra referente ao mesmo domínio. Achadas as equivalências, pode ser aplicada a operação de busca descrita anteriormente, para identificar os componentes associados – de outros workflows ou até do próprio.

3.6 Comparação com Outras Propostas

A Seção 2.7 apresenta alguns sistemas gerenciadores de workflows existentes. Nesta seção analisamos as principais características de cada um deles e quais destas melhor se adequam ao nosso modelo. A Tabela 3.6 mostra as propriedades comparadas, por exemplo, a arquitetura do sistema Vistrails é do tipo desktop, e permite a edição de workflows. Sua principal característica é o mecanismo de versionamento de workflows. Além disso, o sistema Vistrais oferece mecanismos de anotação por metadados descritivos. A grande maioria dos sistemas possui uma arquitetura cliente/desktop, ou seja, é necessário ter instalado e configurado o sistema em cada máquina em que se vá utilizá-lo. Isso minimiza as chances de compartilhamento, pois o repositório de workflows fica restrito à máquina cliente. Diferente desta arquitetura, existem o MS WWF, que é um framework para desenvolvimento Web, e o myExperiment, que se trata de uma rede social, em que usuários compartilham seus workflows. Porém, o myExperiment não permite a edição *online* de workflows, pois oferece somente o serviço de compartilhamento de workflows já existentes, criados por algum sistema de arquitetura cliente. No aspecto de anotação, estes sistemas oferecem a utilização de metadados puramente descritivos, com exceção do sistema Kepler, no qual há a tentativa de uso de ontologias para anotações [5].

Dos sistemas descritos acima, utilizamos as características que melhor se adequam a sistemas para colaboração online, destacados em cinza na Tabela 3.6. Utilizamos a arquitetura Web, por ser transparente ao usuário, indepentende de sistema operacional, e por permitir a interação de usuários em um mesmo sistema, mesmo que estes usuários se

	Arquitetura	Edição de Workflows	Características	Tipos de Anotação
Taverna	Desktop	sim	Serviços de Bioinformática e Serviços Web	metadados descritivos
Kepler	Desktop	sim	Biblioteca de componentes de diferentes domínios	ontologias
Triana	Desktop	sim	Coordenação de coleções de serviços built-in	tags
VisTrails	Desktop	sim	Versionamento de Workflows	metadados descritivos
MS WWF	Framework	sim	Integração de workflows à plataforma .Net	metadados descritivos
Swift	Desktop	sim	Execução de larga escala e distribuída	metadados descritivos
myExperiment	Web	não	Repositório compartilhado de workflows	tags
WOODSS Web	**Web**	**sim**	**Edição online de Workflows, repositório compartilhado**	**tags, metadados descritivos, ontologias**

Figura 3.6: Comparativo entre sistemas de Workflows

encontrem em diferentes locais. Um diferencial do nosso sistema é a possibilidade de edição *online* de workflows. Dos sistemas descritos, somente aplicações que utilizam o MS WWF oferecem edição *online*. No aspecto de anotação, por utilizarmos o conceito de unidades de anotação, oferecemos metadados tanto descritivos, como metadados relacionados a ontologias, abrangendo a grande maioria dos tipos utilizados para anotação de workflows.

3.7 Conclusões

Este capítulo apresentou nosso modelo de anotação. O modelo está focado em dois níveis de abstração de modelagem de workflows: (i) componentes, e (ii) workflows abstratos. No nível de componentes, definimos uma forma de adicionar meta-informações a conectores e atividades. É possível associar ontologias aos componentes, que descrevem quais tipos de dados cada componente aceita como entrada e fornece como saída. O comportamento de uma atividade, assim como o contexto ao qual ela pertence, também podem ser mapeados pelo associação com ontologias. Aumentando, desta forma, a semântica agregada, os componentes tornam-se auto-explicativos, facilitando a identificação de modelos relevantes em uma busca.

Além da anotação nos componentes, o modelo disponibiliza um conjunto de metadados descritivos associados aos workflows, no nível *workflows abstratos*. A principal função destes metadados é fornecer informações básicas sobre o workflow, em que área de conhecimento ele se insere, e histórico de evolução. Utilizamos um conjunto bem restrito de metadados, porque a nossa proposta é bastante ampla, abrangendo vários domínios. Caso o conjunto fosse maior, mais restrito teria que ser o domínio de aplicação do modelo.

O capítulo mostra também novos cenários de busca por workflows científicos face às anotações científicas. Consideramos o uso do sistema Aondê, que manipula ontologias, para buscas, comparações e identificações de workflows e atividades.

Finalmente, fazemos uma comparação com outros sistemas, justificando a escolha de algumas opções e mostrando as características do nosso modelo.

Capítulo 4

Aspectos de Implementação

Este capítulo apresenta os aspectos de implementação da dissertação. A arquitetura é apresentada na seção 4.1. A seção 4.2 apresenta as tecnologias utilizadas no desenvolvimento do protótipo. A seção 4.3 mostra o modelo de classes. A seção 4.4 mostra um caso de uso de construção e busca de workflows. Por fim, a seção 4.5 apresenta as conclusões.

4.1 Arquitetura

A Figura 4.1 ilustra os componentes da arquitetura proposta, em três camadas. A interface do usuário com o sistema é via Web. O núcleo do sistema é dividido em 2 módulos, o Gerenciador de Workflows, e o Gerenciador de Anotações. Ambos acessam o mesmo repositório, onde são armazenados tanto workflows como suas anotações.

Gerenciador de Workflows
Aqui se encontra o modelo de workflows do WOODSS (ver diagrama na Figura 2.2). O módulo é responsável pela criação, armazenamento, e recuperação direta de workflows. Utilizando a interface Web, um usuário pode construir um workflow abstrato a partir de seus componentes. São oferecidas as seguintes funcionalidades:

- criação de workflow
- criação das atividades do workflow
- criação dos conectores das atividades
- criação de transições entre atividades, ligando os conectores

Além da manipulação dos componentes de um workflow, a busca por workflows é realizada diretamente, ou seja, buscando atividades ou workflows pelo nome ou parte dele. É o mecanismo de busca mais trivial.

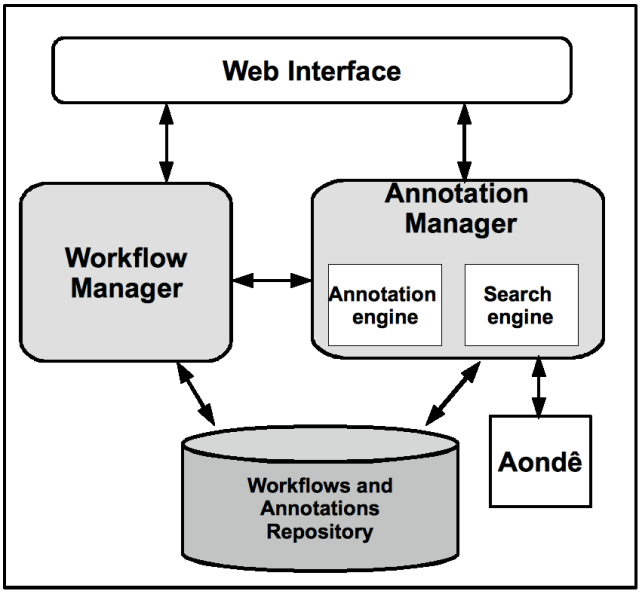


Figura 4.1: Arquitetura proposta

Gerenciador de Anotações

O Gerenciador de Anotações é responsável por criar e armazenar as anotações semânticas, e por relacioná-las com os componentes e workflows no repositório. Sao oferecidas as funcionalidades de criação, alteração e eliminação de anotações para atividades, conectores e workflows.

Ele também é responsável pelo Processador de Buscas. O Processador realiza buscas ao repositório de workflows a partir das anotações fornecidas como parâmetro. Como descrito na Seção 3.5, estas buscas podem ser padrão ou baseadas em ontologias. Neste caso, pode ser usado o sistema Aondê para apoiar o processamento. O processador de buscas precisa interagir com o Aondê para realizar as buscas semânticas.

4.2 Tecnologias utilizadas

O WOODSS, até o início desta dissertação, combinava vários modelos de experimentos, mas não estava disponível na Web. Inicialmente as alternativas consideradas para disponibilizar o sistema WOODSS juntamente com os mecanismos de anotação na Web foram: (i) o uso de serviços Web e (ii) disponibilização do sistema como aplicação Web. A proposta inicial da dissertação era utilizar Serviços Web para disponibilizar e anotar work-

flows. Para que a inclusão das anotações fosse possível, algum tipo de cliente (e.g. uma aplicação *standalone*, ou uma aplicação Web) deveria enviar as mensagens SOAP para o serviço Web. O uso de serviços Web foi descartado, pois, além do serviço de anotação, disponibilizamos a edição online de workflows. Este último serviço é implementado de maneira mais simples em uma arquitetura de aplicação web.

Optamos, portanto, pela forma de aplicação Web pois, além de disponibilizar modelos já existentes para busca, o uso de uma interface Web permite a construção e anotação interativa de componentes e workflows. Assim, o protótipo oferece a possibilidade de construção interativa de workflows, além das funcionalidades de anotação semântica. Uma vez decidido o modelo de implementação, foi definida qual tecnologia a ser utilizada no desenvolvimento. Optamos em utilizar um framework chamado **Django** para o desenvolvimento Web.

4.2.1 O Framework Django

Django [19] é um framework Web de alto nível, desenvolvido sobre a linguagem de programação Python [46], que estimula o desenvolvimento de aplicações Web de maneira rápida, limpa e pragmática. Uma das vantagens do framework é o baixo volume de código exigido para se obter resultados satisfatórios em uma aplicação Web, ou seja, processamento de dados em um servidor, apresentação formatada de informações e interação com usuários via um navegador Web. Para isso, o framework oferece algumas ferramentas e APIs que auxiliam no desenvolvimento da aplicação Web. A seguir, estão listadas as características oferecidas pelo framework que foram utilizadas no desenvolvimento do protótipo.

Mapeador Objeto-Relacional

O mapeador objeto-relacional permite que um conjunto de objetos e relacionamentos seja definido inteiramente utilizando a liguagem Python. O mapeador gera os esquemas de Banco de Dados de maneira automatizada. Este esquema reflete, de forma integral, os objetos escritos em Python. Um exemplo de um modelo definido em Django:

```
class Workflow(models.Model):
    name = models.CharField(max_length=30)
    creation_date = models.DateTimeField()

class Activity(models.Model):
    name = models.CharField(max_length=30)
    pre_condition = models.TextField(blank=True)
    post_condition = models.TextField(blank=True)
    workflow = models.ForeignKey(Workflow)
```

O código define uma classe chamada "Workflow" que possui um atributo do tipo texto (*models.CharField*) de tamanho 30 (*max_length=30*), e atributo do tipo data (*models.DateTimeField*). A classe "Activity" possui três atributos do tipo texto, e possui um atributo que referencia a classe "Workflow"(*models.ForeignKey*). Esta referência indica o relacionamento entre as duas entidades do tipo "um para N", ou seja, Um Workflow pode se relacionar com N Atividades. Desta maneira, as classes criadas no modelo Django são mapeadas automaticamente para o esquema relacional a seguir:

```
CREATE TABLE workflow (
    "id" serial NOT NULL PRIMARY KEY,
    "name" varchar(30) NOT NULL,
    "creation_time" date NOT NULL,
);

CREATE TABLE address (
    "id" serial NOT NULL PRIMARY KEY,
    "name" varchar(30) NOT NULL,
    "pre_condition" varchar(100) NULL,
    "post_condition" varchar(100) NULL,
    "workflow_id" integer NOT NULL REFERENCES "workflow"("id")
);
```

Vale notar que um id sequencial foi criado automaticamente, e utilizado como chave primária das tabelas. Isso acontece quando a chave primária não é explicitamente criada. Além disso, a restrição de chave estrangeira entre as tabelas também foi criada. Todos os tipos de relacionamentos em um modelo Entidade-Relacionamento podem ser mapeados no modelo de classes Django.

Outra característica do mapeador é a fácil manipulação dos objetos, em operações de inserção, atualização, remoção e seleção no banco de dados. Com simples chamadas de métodos dos objetos, tais operações são feitas de maneira transparente e automática no banco de dados. O trecho de código a seguir mostra como criar e inserir objetos no banco de dados. As chamadas dos métodos *wkf.save()* e *activity.save()* inserem os objetos no banco.

```
from mysite.myapp.models import Workflow, Activity

wkf = new Workflow(name="My Workflow", creation_date = "2009-01-01")
wkf.save()
activity = new Activity(name="Any Activity", workflow=wkf)
activity.save()
```

Estes recursos aumentam a rapidez com que um esquema de banco de dados é criado e a maneira como ele evolui, pois uma alteração em classes do sistema é refletida auto-

maticamente no esquema de dados. Portanto, foi somente necessária a definição de um modelo de classes, que se encontra disponível no Apêndice A.

4.2.2 Armazenamento e Visualização dos Workflows

Como a tecnologia Django fornece a criação automática de um esquema relacional, além de métodos de manipulação de objetos no banco de dados, utilizamos um Sistema de Banco de Dados Relacional para o armazenamento dos workflows. O SGBD escolhido para armazenar os repositórios de workflows e anotações foi o PostgreSQL [45], que não só atende plenamente as necessidades para o desenvolvimento do protótipo, como sua interface com o Django é trivial.

Para oferecer uma interface mais amigável para os usuários, foi desenvolvida uma interface interativa para representação visual dos workflows, sob forma de grafos. O uso de grafos como representação é uma boa maneira de se apresentar informações estruturadas, com dependências temporais, como é o caso de workflows. Para esta tarefa, utilizamos uma biblioteca chamada Graphviz [20], que gera imagens representando grafos, partindo de uma especificação textual, chamada *dot*, de relacionamentos (arestas) entre entidades (vértices). A Figura 4.2 mostra a representação de um workflow.

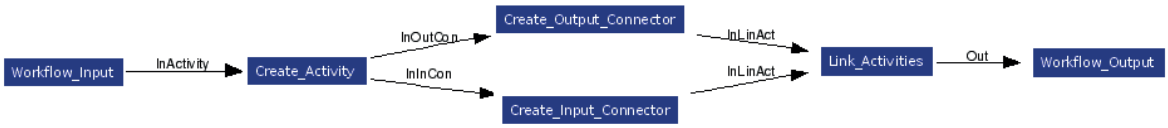


Figura 4.2: Grafo gerado pelo Graphviz [22]

Dessa forma, à medida que o workflow é construído, o usuário tem, em tempo real, a sua representação visual.

4.3 Diagrama de Classes

A Figura 4.3 mostra o diagrama de classes que implementa o modelo de anotações descrito no capítulo 3.

A classe *AnnotationUnit* representa o conceito de Unidade de Anotação. Um conjunto de unidades de anotação forma as anotações semânticas das entidades do modelo de

workflow. É um tipo de anotação genérica, que pode ser utilizada para anotar conectores, atividades e workflows (i.e. Natureza de Dado, Classe de Atividade).

4.4 Exemplo de Uso

Nesta seção apresentamos um exemplo de uso de construção de workflow e insclusão de anotações. São apresentadas telas do protótipo como ilustração da construção.

4.4.1 Criação de Workflow

Uma vez autenticado no sistema, o usuário requisita a criação de um workflow. O primeiro passo é preencher o nome do workflow, como mostra a Figura 4.4. Ao salvar o workflow, sua estrutura básica é criada, e apresentada na tela. São criadas duas "atividades" que representam os pontos de entrada e saída do workflow (*WorkflowInput* e *WorkflowOutput*). A Figura 4.5 mostra a tela de edição do workflow com os pontos de entrada/saída.

A seguir, o usuário cria as atividades (Figura 4.6) do workflow e seus conectores (mostrados na Figura 4.7). Vale notar que, neste ponto, o usuário define se o conector é input/output do workflow ou é um conector interno do workflow. Essa escolha é refletida na visualização gráfica do workflow, ligando ou não a atividade em uma das entidades *WorkflowInput* e *WorkflowOutput*.

Definidos os conectores, o usuário cria as transições entre as atividades. Para isso, é necessário escolher um conector (output) como *source* e um conector (input) como *target* da transição. A Figura 4.8 mostra a criação de uma transição.

As anotações semânticas são adicionadas a conectores, atividades e workflows. A Figura 4.9 apresenta a criação de uma unidade de anotação. São preenchidos os campos (i) propriedade anotada, (ii) valor atribuído, e caso exista, (iii) a referência a uma ontologia que define formalmente o termo anotado.

As Figuras 4.10 e 4.11 mostram, respectivamente, um conector e uma atividade com suas anotações. Por fim, realizadas estas etapas, a Figura 4.12 mostra um workflow completo, incluindo suas anotações.

4.5 Conclusões

Neste capítulo, apresentamos a implementação do protótipo, assim como os detalhes de implementação e casos de uso do protótipo. É possível realizar operações básicas de inclusão, remoção, atualização e visualização de atividades e workflows abstratos, além da manipulação de anotações semânticas para componentes e workflows. Para tal, foi utilizado o framework Django, escrito utilizando a linguagem Python, que trouxe ganhos de

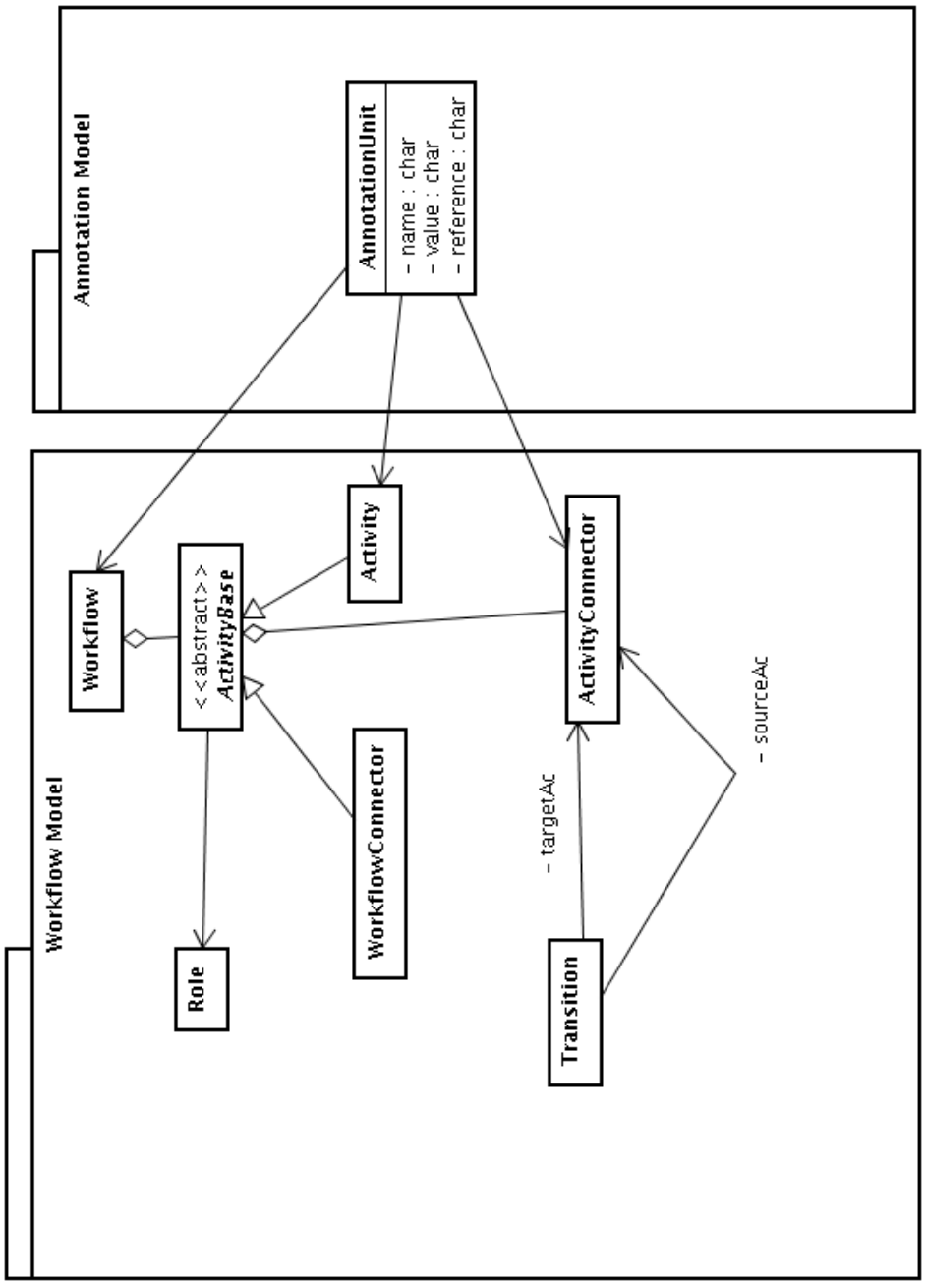


Figura 4.3: Modelo de Anotação - Diagrama de Classes

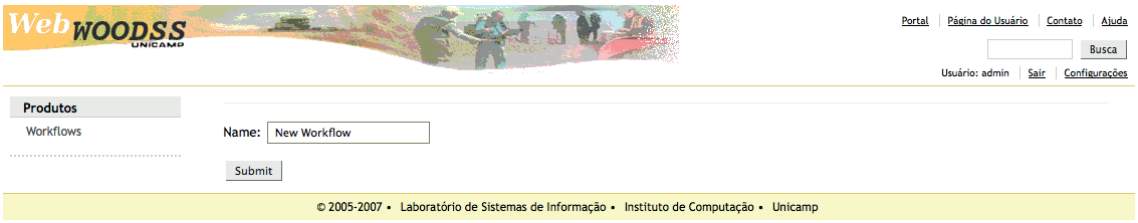


Figura 4.4: Protótipo - Novo Workflow

Figura 4.5: Protótipo - Estrutura básica de um Workflow

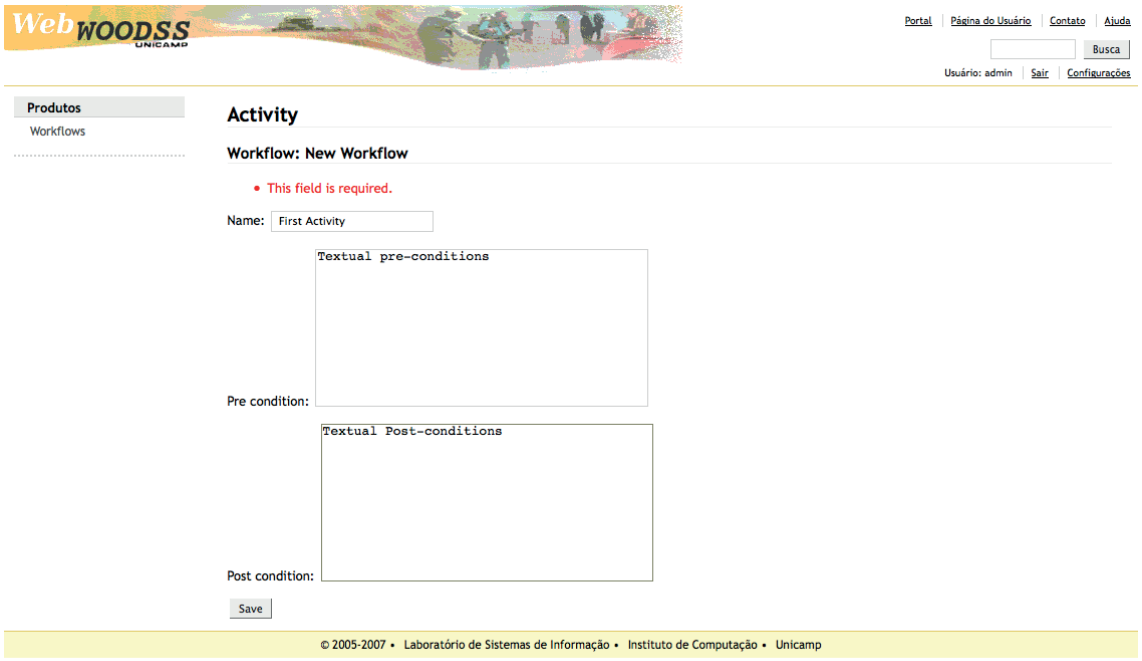


Figura 4.6: Protótipo - Criação de uma atividade

Figura 4.7: Protótipo - Criação de um conector

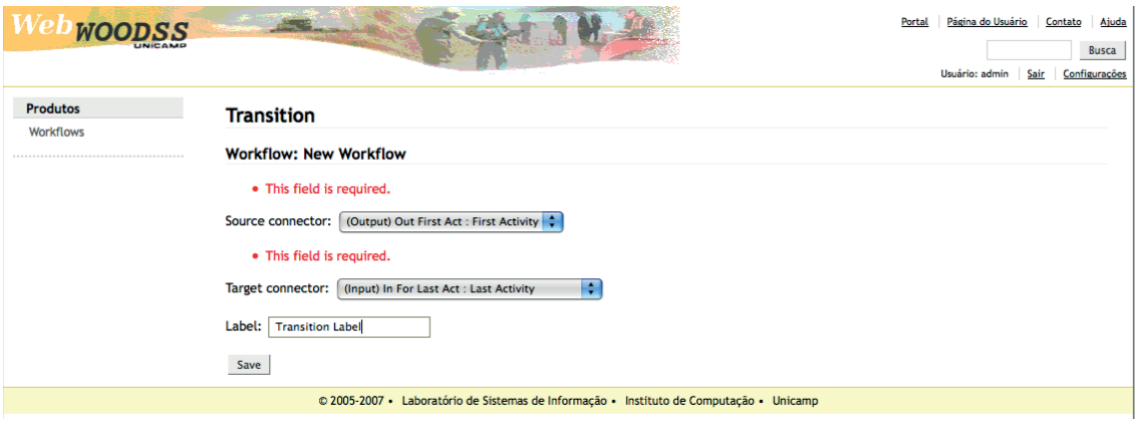


Figura 4.8: Protótipo - Criação de uma transição

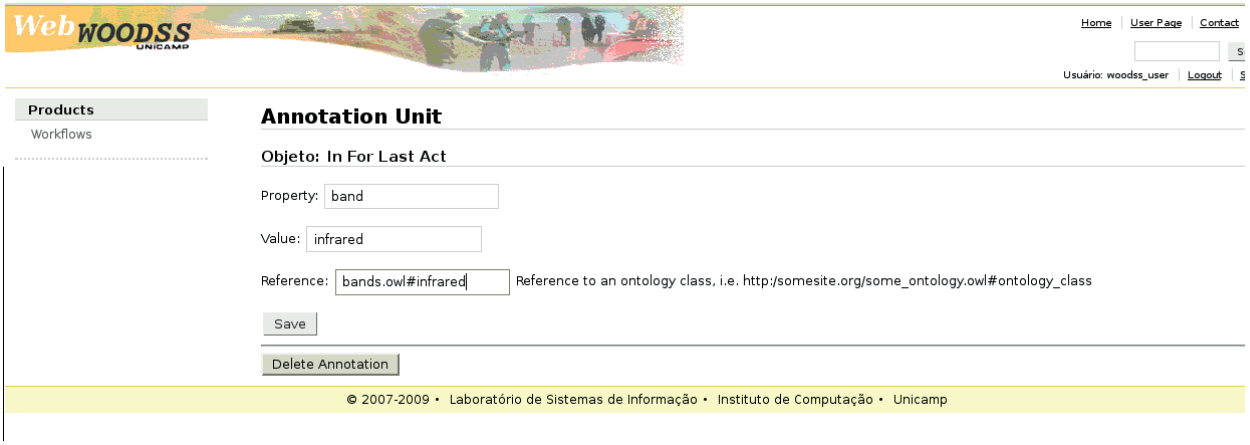


Figura 4.9: Protótipo - Criação de uma unidade de anotação

produtividade, por possuir uma maneira simples de se manipular objetos, e apresentá-los. Utilizando a biblioteca Graphviz, construímos uma representação visual dos workflows.

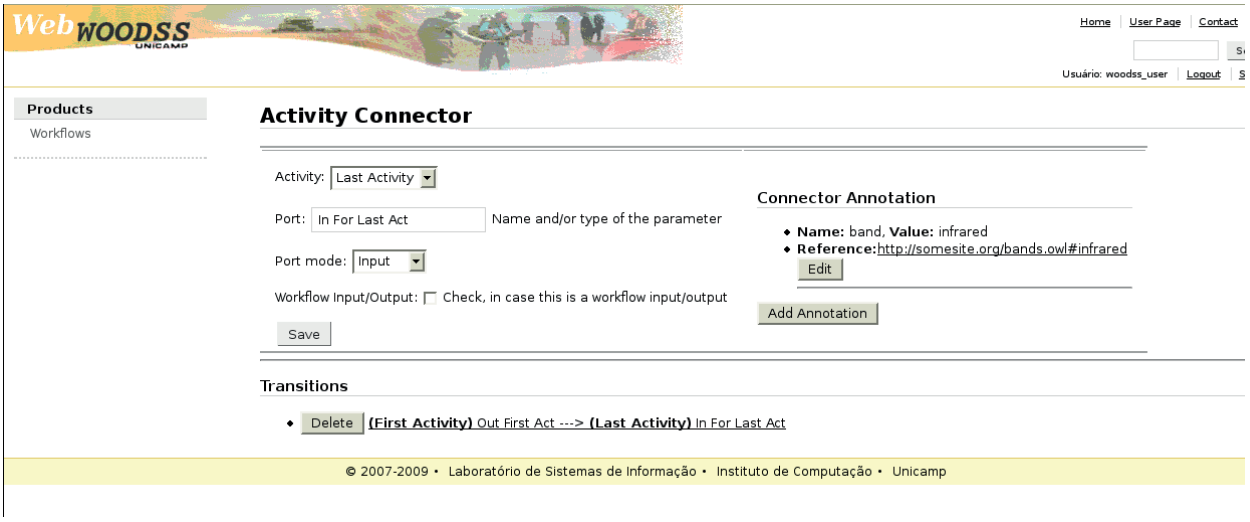


Figura 4.10: Protótipo - Conector com anotação

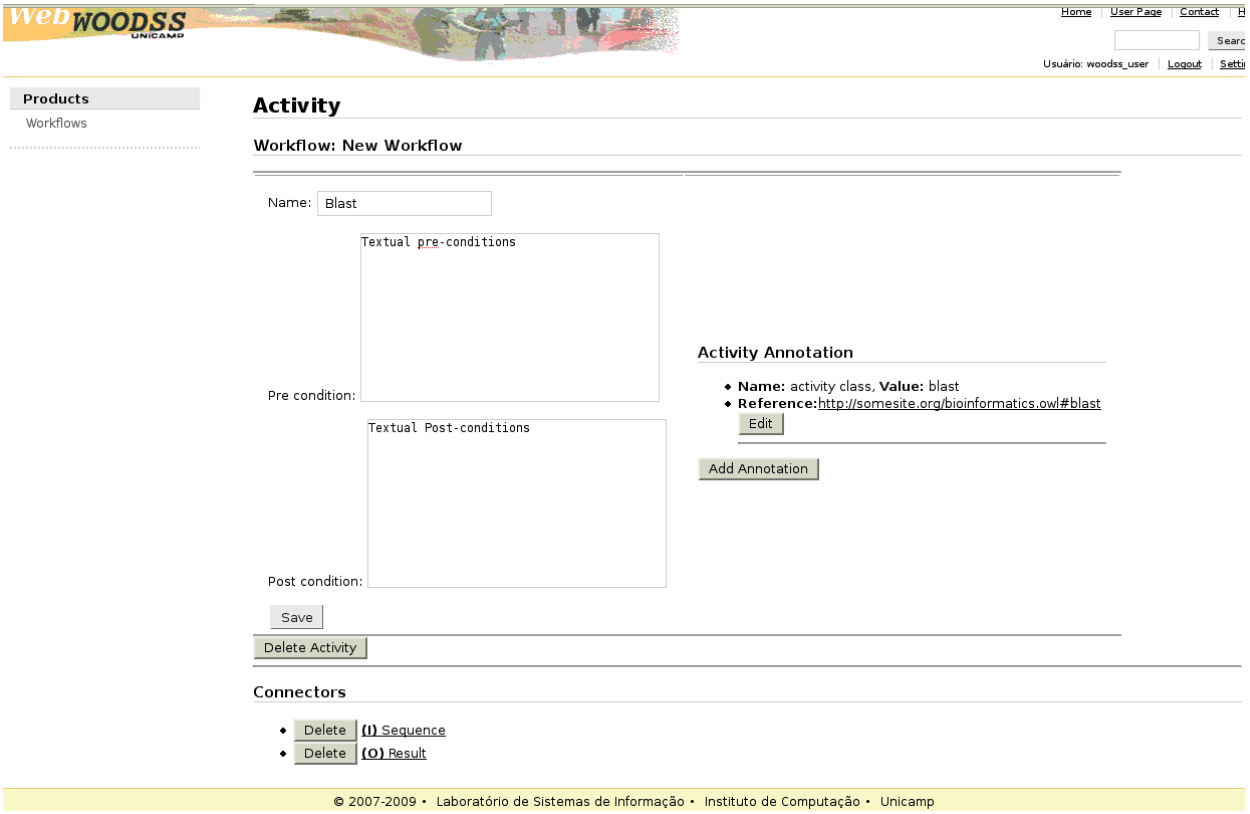


Figura 4.11: Protótipo - Atividade com anotação

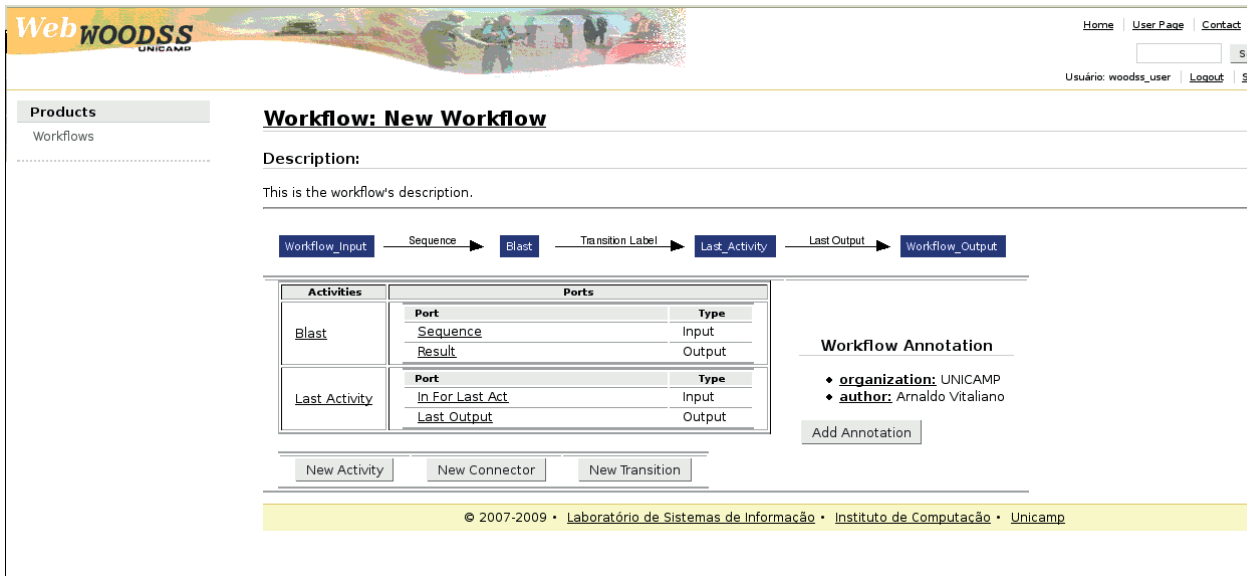


Figura 4.12: Protótipo - Workflow completo

Capítulo 5

Conclusões e Trabalhos Futuros

Uma das premissas do trabalho cooperativo é a possibilidade de compartilhamento e reuso de documentos, processos e modelos. Quando este trabalho ocorre na Web, a cooperação, o compartilhamento e o reuso exigem mecanismos que levem em conta aspectos específicos a ambientes distribuídos. Considerando que modelos podem ser descritos como workflows científicos, esta dissertação se preocupou em fornecer mecanismos que facilitassem o compartilhamento de workflows na Web. Especificamos, então, um modelo de anotação semântica para workflows científicos, que combina o uso de metadados e ontologias para fornecer informações sobre componentes e workflows.

5.1 Contribuições

Nosso trabalho partiu do pressuposto que experimentos científicos são modelados como workflows científicos. Como base, utilizamos o modelo de workflows definido por Pastorello [29]. Fizemos um levantamento de sistemas Web que gerenciam workflows, e de como alguns deles utilizam metadados e anotações em seus workflows. Junto a este levantamento, estudamos conceitos de Web Semântica e ontologias, que foram utilizados na especificação do modelo de anotação.

O próximo passo foi a elaboração do modelo de anotação para workflows científicos. Combinando o uso de metadados descritivos com o uso de ontologias como unidades de anotação semântica, desenvolvemos um modelo que permite anotar diferentes níveis de abstração de workflows (e.g. componentes de workflows – atividades e conectores – e workflows abstratos). O modelo é baseado em uma estrutura básica, chamada unidade de anotação, na qual ontologias são usadas para definir alguma característica do objeto anotado.

Um fator relevante relacionado à nossa pesquisa foi a escassez de trabalhos que correlacionam sistemas de workflows com aspectos de semântica. Em vários casos [51, 6, 8], essa

correlação é geralmente voltada a uma área específica. Isso mostra que, apesar do estágio inicial, as idéias aqui apresentadas agregam muito à área de pesquisa. Apresentamos um modelo simples, mas que permite que usuários de diferentes domínios possam colaborar, compartilhando seus modelos.

Em paralelo, desenvolvemos um protótipo, disponibilizado como uma aplicação Web. Na aplicação é possível construir componentes e workflows, incluir anotações, e realizar buscas por componentes e workflows disponíveis no repositório. A implementação do protótipo tomou um tempo considerável da pesquisa, mas nos possibilitou apresentar um sistema completamente funcional. Como visto na Seção 2.7, a maioria dos sistemas gerenciadores de workflows possuem arquitetura Desktop, e oferecem poucos meios de se agregar semântica aos workflows, na maioria dos casos, pelo uso de metadados descritivos. Nosso sistema possuí arquitetura Web e oferece a edição online de workflows. Além disso, nosso sistema oferece um modelo de anotação genérico o suficiente para possibilitar diferentes maneiras de se agregar metainformações aos workflows.

As principais contribuições deste trabalho são, portanto:

- Análise de sistemas gerenciadores de workflows e maneiras de se agregar semântica a workflows. Coletamos informações sobre alguns dos mais utilizados gerenciadores de workflows científicos, e sobre o modo que cada um utiliza para anotar seus workflows. Utilizamos as características de anotação destes sistemas que consideramos mais relevantes no desenvolvimento do nosso modelo;

- Um modelo de anotação semântica que permite anotar workflows científicos em diferentes níveis de abstração, não se restringindo a uma única área do conhecimento. Especialistas podem buscar experimentos, mesmo quando construídos por usuários de diferentes domínios de conhecimento e, além disso, o modelo cria a base para descoberta, reuso e compartilhamento de workflows científicos;

- Implementação de um protótipo que possibilita a construção, anotação e busca de workflows na Web, seguindo padrões do WfMC.

5.2 Extensões

Trabalhos futuros podem envolver extensões diretas de implementação ou do ponto de vista conceitual.

Como extensões conceituais destacamos:

- **Sistema de Busca.** Desenvolvimento de um sistema de busca semântica, ou seja, que utilize as anotações com ontologias. O sistema AONDÊ [16] manipula ontologias de biodiversidade, realizando buscas em repositórios. Como mostrado no

capítulo 3, uma possibilidade seria o uso do Aondê para buscar ontologias utilizadas nas anotações de workflows do WOODSS. É possível integrar ambos sistemas para prover o serviço de busca.

- **Versionamento.** Desenvolvimento de um mecanismo de versionamento para atividades e anotações. Tal mecanismo facilita o reuso de componentes. Um componente reutilizado pode gerar uma nova versão do componente original, possibilitando a comparação entre versões, o que permite uma análise de evolução dos componentes e workflows.

Dentre as extensões na implementação, podemos destacar:

- **Importação e Exportação de Workflows.** Implementação de um módulo responsável por traduzir o modelo de workflows de e para algum documento estruturado (baseado em XML). Destacam-se linguagens como XPDL [13] e WS-BPEL [42]. Isto permitiria maior interoperabilidade entre nossa implementação e outros sistemas disponíveis na Web.

- **Implementação do Reuso de Componentes.** Extensão da implementação para a construção de atividades, independente de estaremelacionadas a workflows. O modelo de workflows já prevê reutilização de componentes, mas nosso protótipo não contempla esta característica. Para reutilizar um componente é necessária a cópia deste para o workflow desejado.

- **Melhorias de Interface.** Hoje em dia, com o uso bem difundido da tecnologia AJAX, é possível criar interfaces Web bem interativas. Existem bibliotecas disponíveis para uso [26], que facilitam muito a criação de interfaces Web. É possível a criação de uma interface que utiliza formas geométricas na edição de workflows, facilitando a compreensão do experimento modelado.

- **Criação e Anotação de Workflows Concretos.** Há possibilidade de relacionar os workflows definidos no sistema com serviços que implementam operações descritas nos workflows. O uso de Serviços Web é uma forma de se disponibilizar a implementaçao de serviços. O sistema poderia relacionar uma atividade de um workflow com uma operação de um serviço Web descrita em um arquivo WSDL, e realizar uma chamada de execução deste serviço. Além do mais, tendo em vista que conectores e atividades são anotados com ontologias, é possível validar a criação dos workflows concretos.

- **Visualização de Ontologias.** Assim como as melhorias de interface Web, uma extensão seria implementar a visualização e edição das ontologias utilizadas nas anotações, com uso de tecnologias Web (javascript, AJAX, JAVA).

Apêndice A

Modelo de Classes

```
from django.db import models

#########################################################
#                   Workflow Model                      #
#########################################################

class Workflow(models.Model):
    name = models.CharField(max_length=50, db_index=True)
    creation_date = models.DateTimeField(auto_now_add=True)
    complete = models.BooleanField(blank=True, editable=False)
    description = models.TextField(blank=True)

    def __unicode__(self):
        return '%s'%(self.name)


class Role(models.Model):
    name =  models.CharField(max_length=50, db_index=True, unique=True)
    description = models.TextField(max_length=255)


class ActivityBase(models.Model):
    name = models.CharField(max_length=50, db_index=True)
    workflow = models.ForeignKey(Workflow, editable=False)

    class Meta:
        abstract = True
```

```
        unique_together = (('workflow','name'),)
        verbose_name_plural = 'Activities'

class WorkflowConnector(ActivityBase):
    WF_CONNECTOR_CHOICES = (('I','Input'), ('O','Output'))
    wc_input = models.CharField(max_length=1, choices=WF_CONNECTOR_CHOICES, default='I

class Activity(ActivityBase):
    CONTROL_MODE_CHOICES = (('A','AND'),('O','OR'),('X','XOR'))
    pre_condition = models.TextField(blank=True)
    post_condition = models.TextField(blank=True)
    control_mode_in = models.CharField(max_length=1,
                                       choices=CONTROL_MODE_CHOICES,
                                       default='A', blank=True,editable=False)
    control_mode_out = models.CharField(max_length=1,
                                        choices=CONTROL_MODE_CHOICES,
                                        default='A',blank=True,editable=False)
    role = models.ForeignKey(Role, null=True, blank=True,editable=False)

    def __unicode__(self):
        return '%s'%(self.name)

class ActivityConnector(models.Model):
    PORT_MODE_CHOICES = (('I','Input'),('O','Output'))
    CONTROL_MODE_CHOICES = (('M','Multiple'),('D','Discriminator'))
    activity = models.ForeignKey(Activity)
    port = models.CharField(max_length=50,
                            help_text="Name and/or type of the parameter")
    port_mode = models.CharField(max_length=1,
                                 choices=PORT_MODE_CHOICES, default='I')
    control_mode = models.CharField(max_length=1,
                                    choices=CONTROL_MODE_CHOICES, default='D',
                                    blank=True,editable=False)
    workflow_input = models.BooleanField(blank=True,
                    help_text='Check, in case this is a workflow input/output',
                    verbose_name='Workflow Input/Output')

    def __unicode__(self):
        if (self.port_mode == 'I'):
            return '(Input) %s : %s'%(self.activity,self.port)
```

```
        else:
            return '(Output) %s : %s'%(self.activity,self.port)

    def is_connected(self):
        if self.port_mode == 'I':
            if len(Transition.objects.filter(target_connector=self)) == 0:
                return True
            else:
                return False
        else:
            if len(Transition.objects.filter(source_connector=self)) == 0:
                return True
            else:
                return False

    def get_workflow(self):
        return self.activity.workflow


    class Meta:
        ordering = ('port_mode', 'activity')


class Transition(models.Model):
    source_connector = models.ForeignKey(ActivityConnector,
                                         related_name='source_transitions')
    target_connector = models.ForeignKey(ActivityConnector,
                                         related_name='target_transitions')
    activation_condition = models.TextField(blank=True,editable=False)
    workflow = models.ForeignKey(Workflow, editable=False)
    label = models.CharField(max_length=30, blank=True)

    def __unicode__(self):
        return '%s ---> %s'%(self.source_connector.activity,
                             self.target_connector.activity)

    class Meta:
        unique_together = (('source_connector','target_connector'),)


##########################################################
```

```
#                Annotation Model                      #
########################################################

class AnnotationUnit(models.Model):
    property = models.CharField(max_length=15)
    value = models.CharField(max_length=30)
    reference = models.URLField(verify_exists=False, blank=True,
                                    help_text='Reference to an ontology class,
                                    i.e. http://somesite.org/some_ontology.owl#ref')
    parent_id = models.IntegerField(editable=False)
    PARENT_CLASSES = (('C','Connector'),('A','Activity'),('W','Workflow'))
    parent_class = models.CharField(max_length=1,choices=PARENT_CLASSES, editable=False

    def __unicode__(self):
        return '%s(%s:%s)'%(self.parent_class,self.property,self.value)
```

Referências Bibliográficas

[1] W. Aalst, A. van der, B. Hofstede, and A. Kiepuszewski. Advanced workflow patterns. In O. Etzion en P. Scheuermann, editor, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18–29. Springer-Verlag, Berlin, 2000.

[2] R. Allen. *Workflow Handbook 2001*, chapter Workflows: An Introduction. Workflow Management Coalition (C), 2001.

[3] R. Bellinger E. Anderson E. Santos J. Freire B. Howe1, P. Lawson, C. Scheidegger, A. Baptista, and C. Silva. End-to-end escience: Integrating workflow, query, visualization, and provenance at an ocean observatory. In *Fourth IEEE International Conference on eScience*, pages 127–134, 2008.

[4] P. Barthelmess and J. Wainer. Workflow systems: a few definitions and a few suggestions. In *In Proc. of Conference on Organizational Computing Systems*, 1995.

[5] C. Berkley, S. Bowers, M. Jones, B. Ludäscher, M. Schildhauer, and J. Tao. Incorporating semantics in scientific workflow authoring. In *SSDBM'2005: Proceedings of the 17th international conference on Scientific and statistical database management*, pages 75–78, Berkeley, CA, US, 2005. Lawrence Berkeley Laboratory.

[6] C. Berkley, S. Bowers, M. B. Jones, B. Ludäscher, M. Schildhauer, and J. Tao. Incorporating semantics in scientific workflow authoring. In *In Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM'05*, 2005.

[7] BPMI.org. Business process management iniciative. http://www.bpmi.org. (Acessado em 03/04/2007).

[8] D. Kaster C. B. Medeiros and H. Rocha. Supporting modeling and problem solving from precedent experiences: The role of workflows and case-based reasoning. *Environmental Modeling and Software*, 20:689–704, 2005.

[9] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. Managing the evolution of dataflows with vistrails. In *ICDEW '06: Proceedings of the 22nd International Conference on Data Engineering Workshops*, page 71, Washington, DC, USA, 2006. IEEE Computer Society.

[10] M. C. Cavalcanti, R. Targino, F. Baião, S. C. Rössle, P. M. Bisch, P. F. Pires, M. L. M. Campos, and M. Mattoso. Managing structural genomic workflows using web services. *Data Knowl. Eng.*, 53(1):45–74, 2005.

[11] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang. Programming Scientific and Distributed Workflow with Triana Services. *Concurrency and Computation: Practice and Experience (Special Issue: Workflow in Grid Systems)*, 18(10):1021–1037, 2006.

[12] CNPq. Tabela de Áreas de conhecimento. http://www.cnpq.br/areasconhecimento/. (Acessado em: 31/03/2009).

[13] WfMC – Workflow Management Coalition. Interface – xml process definition language (xpdl). Technical report, Workflow Management Coalition, 2002.

[14] Federal Geographic Data Committee. Content standard for digital geospatial metadata fgdc-std-001-1998. Technical report, Federal Geographic Data Committee, 1998.

[15] J. Bhagat D. Cruickshank A. Goderis D. Michaelides D. Roure1, C. Goble and D. Newman. myexperiment: Defining the social virtual research environment. In *Fourth IEEE International Conference on eScience*, pages 182–189, 2008.

[16] J. Daltio and C. B. Medeiros. Aondê: An ontology web service for interoperability across biodiversity applications. *Inf. Syst.*, 33(7-8):724–753, 2008.

[17] E. Deelman, G. Singh, M. H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Sci. Program.*, 13(3):219–237, 2005.

[18] L. A. Digiampietri. *Management of Bioinformatics Scientific Workflows (partially in portuguese).* PhD thesis, Instituto de Computação - Unicamp, August 2007.

[19] Django. Django framework. http://www.djangoproject.com/. (Acessado em: 09/02/2008).

[20] E. R. Gansner and S. C. North. An open graph visualization system and its applications to software engineering. *Software — Practice and Experience*, 30(11):1203–1233, 2000.

[21] V. Getov. e-science: The added value for modern discovery. *Computer*, 41(11):30–31, 2008.

[22] Graphviz. Graph visualization software. http://www.graphviz.org/. (Acessado em: 09/02/2008).

[23] T. R. Gruber. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies, 1995.

[24] N. Guarino. Formal ontology and information systems. In *Proceedings of FOIS-98*, pages 3–15, 1998.

[25] D. Hollingsworth. The workflow reference model. technical report tc-1003. Technical report, Workflow Management Coalition, 1995.

[26] ILOG. Ilog jview. http://www.ilog.com/products/jviews/demos/index.cfm.

[27] Dublin Core Metadata Initiative. Dublin core. http://dublincore.org/. (Acessado em: 11/09/2007).

[28] G. Vossen J. Wainer, M. Weske and C. B. Medeiros. Scientific workflow systems. In *In Proc. of the NSF Workshop on Workflow and Process Automation Information Systems*, 1996.

[29] G. Z. Pastorello Jr. Publicação e integração de workflows científicos na web. Master's thesis, IC-UNICAMP, 2005.

[30] G. Z. Pastorello Jr. *Managing the lifecycle of sensor data: from production to consumption*. PhD thesis, Instituto de Computação - Unicamp, December 2008.

[31] D. S. Kaster. Combinando bancos de dados e raciocínio baseado em casos para apoio a decisão em planejamento ambiental. Master's thesis, IC-UNICAMP, Campinas-SP, 2001.

[32] J. Rocha L. Seffino, C. B. Medeiros and B. Yi. Woodss - a spatial decision support system based on workflows. *Decision Support Systems*, 27(1-2):105–123, 1999.

[33] Ontotext Lab. The kim platform: Semantic annotation. http://www.ontotext.com/kim/semanticannotation.html. Ontotext, 2007.

[34] B. Ludaescher and C. Goble. Special section on scientific workflows. *SIGMOD Record*, 34(3):3–4, 2005.

[35] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice & Experience*, 18(10):1039–1065, 2006.

[36] M. L. M. Campos M. C. R. Cavalcanti, M. L. Q. Mattoso, F. Llirbat, and E. Simon. Sharing scientific models in environmental applications. In *In Proc. of ACM Symposium on Applied Computing*, 2002.

[37] M. L. Q. Mattoso M. C. R. Cavalcanti and M. L. M. Campos. *Scientific Resources Management: Towards An In Silico Laboratory*. PhD thesis, UFRJ, Rio de Janeiro–RJ, 2003.

[38] C. C. Marshall. Toward an ecology of hypertext annotation. In *UK Conference on Hypertext*, pages 40–49, 1998.

[39] C. B. Medeiros, J. Pérez-Alcazar, L. Digiampietri, G. Z. Pastorello Jr., A. Santanchè, R. S. Torres, E. Madeira, and E. Bacarin. Woodss and the web: Annotating and reusing scientific workflows. *SIGMOD Record*, 34(3):18–23, 2005.

[40] NASA. Sweet - semantic web for earth and environmental terminology. http://sweet.jpl.nasa.gov/ontology/. (Acessado em: 29/05/2009).

[41] NISO. Understanding metadata, 2007. NISO Press, USA.

[42] OASIS. Web Services Business Process Execution Language (WSBPEL) Technical Committee. URL:http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, 2005. (Acessado em: 29/03/2007).

[43] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, November 2004.

[44] C. Plesums. *Workflow Handbook 2002*, chapter An Introduction to Workflow. Workflow Management Coalition (C), 2002.

[45] PostgreSQL. Postgresql database management system. http://httpd.postgresql.org/. (Acessado em: 09/02/2008).

[46] Python. Python programming language. http://www.python.org. (Acessado em: 09/02/2008).

[47] G. Stumme R. Studer, R. Volz and Andreas Hotho. Semantic web - state of the art and future directions. *KI Heft, Special Issue on the Semantic Web*, (3):5–9, 2003.

[48] R. Benjamins R. Studer and D. Fensel. Knowledge engineering: principles and methods. *Data and knowledge engineering*, 25:161–197, 1998.

[49] W3C Recommendation. Resource description framework (rdf): Model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1999. (Acessado em: 09/07/2007).

[50] H. A. Rocha. Metadados para workflows científicos no apoio ao planejamento ambiental. Master's thesis, IC-UNICAMP, Campinas-SP, 2003.

[51] C. Wroe S. Miles, J. Papay, P. Lord, C. Goble, and L. Moreau. Semantic description, publication and discovery of workflows in mygrid. Technical report, University of Southampton, 2004.

[52] A. Santanchè and C. B. Medeiros. Self describing components: Searching for digital artifacts on the web. In *SBBD*, pages 10–24, 2005.

[53] D. Shukla and B. Schmidt. *Essential Windows Workflow Foundation (Microsoft .Net Development Series)*. Addison-Wesley Professional, 2006.

[54] S. Thakkar, J. Ambite, and C. Knoblock. Composing, Optimizing and Executing Plans for Bioinformatics Web services. *VLDB Journal*, 14(3):330–353, 2005.

[55] World Wide Web Consortium W3C. W3C web site. URL: http://www.w3.org/, 2005. (Acessado em: 29/03/2007).

[56] WfMC. Workflow management coalition. http://www.wfmc.org. (Acessado em 03/04/2007).

[57] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, V. Nefedova, I. Raicu, T. Stef-Praun, and M. Wilde. Swift: Fast, reliable, loosely coupled parallel computation. In *Services, 2007 IEEE Congress on*, pages 199–206, 2007.