MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

FABRÍCIO NOGUEIRA DA SILVA

IN SERVICES: UM SISTEMA PARA GERENCIAMENTO DE DADOS INTERMEDIÁRIOS EM WORKFLOWS CIENTÍFICOS NA BIOINFORMÁTICA

Rio de Janeiro
2006

INSTITUTO MILITAR DE ENGENHARIA

FABRÍCIO NOGUEIRA DA SILVA

IN SERVICES: UM SISTEMA PARA GERENCIAMENTO DE DADOS INTERMEDIÁRIOS EM WORKFLOWS CIENTÍFICOS NA BIOINFORMÁTICA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Profª. Maria Cláudia Reis Cavalcanti, D.Sc

Rio de Janeiro
2006

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

S568i Silva, Fabrício Nogueira da
In Services: Um Sistema para Gerenciamento de Dados Intermediários em Workflows Científicos na Bioinformática, Fabrício Nogueira da Silva.
– Rio de Janeiro: Instituto Militar de Engenharia, 2006.
119 p.:il, graf., tab.

Dissertação: (mestrado) – Instituto Militar de Engenharia, Rio de Janeiro, 2006.

1. Bioinformática, workflows científicos. 2. Gerência de dados, banco de dados. I. Instituto Militar de Engenharia. II. Título.

CDD 005.74

INSTITUTO MILITAR DE ENGENHARIA

FABRÍCIO NOGUEIRA DA SILVA

IN SERVICES: UM SISTEMA PARA GERENCIAMENTO DE DADOS INTERMEDIÁRIOS EM WORKFLOWS CIENTÍFICOS NA BIOINFORMÁTICA

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Profª. Maria Cláudia Reis Cavalcanti, D.Sc

Aprovada em 28 de julho de 2006 pela seguinte Banca Examinadora:

Profª. Maria Cláudia Reis Cavalcanti, D.Sc do IME - Presidente

---

Prof. Alberto Martín Rivera Dávila, D.Sc da FIOCRUZ

Profª. Vanessa de Paula Braganholo, D.Sc do IM-DCC/UFRJ

Rio de Janeiro
2006

Aos meus pais José Nogueira da Silva e Maria de Jesus Pitombeira da Silva.

AGRADECIMENTOS

À minha orientadora, Drª. Maria Cláudia Reis Cavalcanti, por toda a atenção dada durante o desenvolvimento deste trabalho. Pelas orientações fornecidas a mim para a correta condução do mesmo. Pela paciência, confiança e amizade.

Aos professores Alberto Martín Rivera Dávila, Marta Lima de Queiroz Mattoso e Vanessa de Paula Braganholo pela presença em minha banca de avaliação.

Aos professores Alberto Martín Rivera Dávila, Marta Lima de Queiroz Mattoso, Maria Luiza de Machado Campos e Ricardo Choren Noya, pelas orientações e conhecimentos repassados.

Ao coordenador do curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, Dr. Paulo Fernando Ferreira Rosa, por ter confiado nos meus trabalhos e fornecido as informações necessárias ao cumprimento das minhas atividades como aluno.

Aos demais professores do curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia pelo conhecimento repassado durante o meu período como aluno.

Aos meus pais, José Nogueira da Silva e Maria de Jesus Pitombeira da Silva, por tudo que representam na minha vida e que sem eles, este trabalho seria apenas um desejo inalcançável.

Aos meus irmãos, Anna Karlla Pitombeira Silva e José Nogueira da Silva Júnior, pela confiança depostitada em mim e eterno companheirismo.

À minha namorada, Alexandra Kolontai de Sousa Oliveira, por ter suportado toda distância, todas as dificuldades, e sempre estar apta, com suas doces palavras, a me confortar nos momentos difíceis.

Aos amigos, Arthur Henrique, Carlos André, Diogo Munhoz, Eduardo Albuquerque, Francisco Hanrick e Marcelo Loutfi, pela convivência diária e pelos momentos de descontração na república.

Aos amigos, Alexandre Tadeu, Ana Carolina Brito de Almeida, Fábio Vidal e Vitor Guerra, pela companhia no curso, pelas ajudas nas dificuldades, pelas atividades realizadas em conjunto e que certamente contribuíram bastante à realização deste trabalho.

A todos os amigos que fiz durante a caminhada de graduação e pós.

Aos demais colegas do mestrado, que em proveitosas conversas técnicas contribuíram direta e indiretamente no trabalho.

Aos amigos e colegas do Serviço Federal de Processamento de Dados de Fortaleza que me ensinaram muitas coisas e também sempre demonstraram apoio.

A todos os funcionários do Departamento de Engenharia de Sistemas (SE/8) do IME.

À Capes por financiar parcialmente este trabalho, o tornando viável.

"É melhor tentar e falhar, que preocupar-se e ver a vida passar; é melhor tentar, ainda que em vão, que sentar-se fazendo nada até o final. Eu prefiro na chuva caminhar, que em dias tristes em casa me esconder. Prefiro ser feliz, embora louco, que em conformidade viver ..."

Martin Luther King

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SÍMBOLOS

ABREVIATURAS

BPEL4WS	-	Business Process Execution Language for Web Services
HTTP	-	Hypertext Transfer Protocol
PC	-	Personal Computer
SOAP	-	Simple Object Access Protocol
VDL	-	Virtual Data Language
XML	-	Extensible Markup Language
WSDD	-	Web Service Deployment Description
WSDL	-	Web Service Description Language
SGBD	-	Sistema Gerenciador de Banco de Dados

RESUMO

Workflows científicos têm sido utilizados para a gerência de experimentos *in silico*. Esses experimentos são caracterizados por serem executados e analisados através de computadores. Em suas execuções, uma seqüência de programas computacionais é processada e os dados de saída de um programa são compostos como dados de entrada no programa seguinte. Na área de bioinformática, workflows científicos são comumente definidos utilizando-se linguagens de script. Porém, essa abordagem de definição, apesar de permitir a automatização da execução do experimento, possui algumas deficiências com relação a flexibilidade para atender diferentes necessidades dos cientistas sobre o workflow. Além disso, conta ainda com a deficiência de interoperabilidade entre os passos do workflow ou entre outros workflows. Para superar essas deficiências, a tecnologia de serviços Web vem sendo adotada pela comunidade científica como um facilitador para a disponibilização e acesso a programas científicos. Isso possibilitou que programas científicos utilizados em ambientes distintos pudessem ser integrados para comporem workflows científicos. Porém, a natureza distribuída dos serviços Web resgatou algumas questões referentes à gerência dos dados gerados e processados durante a execução dos experimentos - *dados intermediários*. Onde e como disponibilizar esses dados de modo que fiquem acessíveis para análises e reutilizações em futuras execuções dos experimentos foram algumas dessas questões levantadas com o uso de serviços Web na composição de workflows científicos. Uma outra questão diz respeito a como realizar a automatização de filtragem de dados durante as execuções dos workflows. Neste trabalho, foi proposto um sistema cuja arquitetura visa solucionar essas questões de gerência de dados intermediários em workflows científicos na bioinformática compostos por serviços Web - sistema *In Services*.

Um workflow real chamado GARSA, em uso por um grupo de pesquisa (BioWebDB) da Fundação Oswaldo Cruz, foi utilizado como estudo de caso para validar o sistema proposto. Um prtótipo desse sistema foi implementado facilitando a geração de filtros a serem aplicados sobre dados intermediários de workflows científicos de bioinformática.

ABSTRACT

Scientific workflows have been used to describe *in silico* experiments. Typically, these experiments are performed and analyzed by computers. During their executions, a set of computational programs is processed and output data of a program are set as input data to the following program. In bioinformatics, scientific workflows are usually defined using script languages. However, this definition approach, in spite of providing an automatic way to perform the experiment, it fails regarding flexibility to keep up with distinct scientists needs in workflows. Besides, it fails with respect to interoperability between workflow steps or other workflows. To deal with these deficiencies, the scientific community has been adopting Web services technology as a facilitator to deploy and access scientific programs. Thus, scientific programs used in different scientific environments could be integrated in a workflow. However, considering the Web services distributed feature, some issues regarding data generated and processed during workflow executions - *intermediate data* - have to be revisited. Where and how to store these data to be available for analysis and reuse in future experiments executions, are some of these issues. Another issue is with respect to how to automate the data filtering during workflows executions. In this work, a system whose architecture aims to solve those issues, about intermediate data management in scientific workflows composed by Web services in bioinformatics, was proposed - *In Services* system. A real workflow called GARSA, in use by a research group (BioWebDB) from Fundação Oswaldo Cruz, was used as a case study to validate the proposed system.

1 INTRODUÇÃO

É crescente o uso de recursos computacionais em ambientes científicos para auxílio na realização de seus experimentos. Uma série de programas de computador são diariamente desenvolvidos pela comunidade científica no intuito de realizarem simulações, cálculos, comparações, e enfim, processamento sobre dados científicos. Esses tipos de programas computacionais são conhecidos como programas científicos. O uso desses programas auxiliou na realização dos experimentos costumeiramente realizados em ambientes de laboratórios especializados. Além da obtenção e análise dos dados gerados pelas técnicas tradicionais de experimentação científica, utiliza-se o ambiente computacional para o processamento dos dados e incremento das análises sobre os mesmos. Os experimentos realizados ou analisados através de computadores são conhecidos pela comunidade científica como experimentos *in silico*.

Workflows são definidos como uma série de tarefas (passos) executadas em uma determinada seqüência através de um conjunto de regras. A execução do workflow pode ser iniciada a partir de um conjunto de dados a serem utilizados pelos passos iniciais do workflow. Além disso, mais dados podem ser gerados em cada um dos passos e repassados aos passos subseqüentes do workflow. Após a execução de todos os passos, espera-se que o workflow gere um resultado válido para o propósito ao qual foi definido.

No meio científico, o uso de workflows tem sido adotado para descrever e realizar experimentos *in silico*. Nesses workflows, os experimentos são planejados através da composição de programas de computador em uma seqüência de execução. Cada programa, geralmente compõe um passo do workflow científico e executa uma atividade específica dentro do mesmo. Durante a execução desses workflows, os dados de saída de um passo são usados como dados de entrada do passo seguinte e ao final de toda a execução desse fluxo de programas, tem-se o experimento *in silico* realizado.

Alguns passos do experimento são custosos e de processamento pesado. Tal fato, algumas vezes requer o uso exclusivo de máquinas com alto poder de processamento. Essa necessidade leva a uma busca de paralelismo e distribuição dos passos que compõem os workflows científicos. Conseqüentemente, a comunidade científica tem buscado alternativas que possibilitem essa distribuição. Uma das alternativas bem aceitas foi o uso de

serviços Web (ALTINTAS et. al., 2003), (GOBLE et. al., 2003), (KÜNZL, 2002), (STEIN, 2002), (WILKINSON, 2002). Serviços Web servem como provedores de acesso aos programas científicos, pois fornecem uma interface padrão de acesso que independe de linguagem de programação e/ou plataforma. Nesse contexto, uma série de laboratórios científicos já disponibilizam serviços Web para acesso a programas que compõem seus experimentos e para então serem acessados e utilizados por outras comunidades científicas (ECKART et. al., 2003), (FINN et. al., 2006), (LIEFELD et. al., 2005), (STANDLEY, 2005). O uso de serviços Web aumentou o nível de compartilhamento de workflows científicos e de interoperabilidade entre passos de workflows. Por conta disso, alguns sistemas que permitem modelar e executar workflows passaram a fazer uso dos serviços Web na composição de seus passos, facilitando assim a atividade de definição dos workflows (CAVALCANTI et. al., 2005), (TARGINO, 2004), (TARGINO et. al., 2005).

Na área de bioinformática, workflows científicos são freqüentemente modelados utilizando-se linguagens de script como *Perl*. Em sua programação, são feitas as invocações aos programas científicos como passos no workflow, bem como transformações de dados para serem processados no decorrer da execução do mesmo. Porém, essa abordagem além de ser dependente do ambiente onde é executado (sistema operacional, localização dos programas científicos); fornece meios pouco adequados que permitam a execução de programas residentes em diferentes plataformas. Além disso, essa abordagem não facilita o reuso de passos do workflow que poderiam ser utilizados na composição de outros workflows. Um outro problema é sobre a execução distribuída de passos que necessitem de alto poder de processamento, pois essa abordagem geralmente define e executa workflows num contexto de máquina local.

No intuito de resolver o problema de interoperabilidade entre os passos dos workflows científicos de bioinformática, a comunidade científica dessa área vem adotando também o uso de serviços Web para composição de workflows (CHANDRASEKARAN et. al., 2002), (DAVIES, 2002), (SLIDEL, 1999). Tornando padrão a interface de acesso aos passos (serviços Web) dos workflows.

1.1 MOTIVAÇÃO

Na execução de workflows científicos, uma série de dados é gerada e processada em cada passo do workflow. Nos ambientes científicos, não somente os dados finais resultantes de um workflow são importantes para a validação de um experimento. Os *dados inter-*

mediários, dados gerados por cada passo de um workflow durante sua execução, também são importantes, pois permitem validar cada passo do workflow, contribuindo também para a análise da sua execução como um todo. Soma-se a isso, a possibilidade de re-utilização desses dados para a execução de novos experimentos, permitindo assim, re-execuções parciais de um workflow. Ou seja, a partir de um conjunto de dados científicos, previamente obtidos, evita-se que alguns passos do workflow sejam re-executados. Por conta disso, é interessante se pensar em mecanismos que dêem suporte à gerência de dados intermediários. Isso envolveria dispor os dados de maneira adequada para que fossem favorecidas suas manutenções, acessos e análises. Além disso, dados irrelevantes ao experimento são constantemente gerados durante as execuções de workflows. Esses dados, caso não sejam eliminados adequadamente, podem levar a análises inconsistentes dos resultados. Por conta disso, filtros de dados são geralmente necessários nesse contexto. Entretanto, diferentes cientistas podem ter visões e necessidades distintas sobre os dados de um mesmo workflow. Conseqüentemente, podem necessitar de filtros de dados diferentes para tratamento de dados de seus interesses.

1.2 CARACTERIZAÇÃO DO PROBLEMA

A composição de workflows científicos através de serviços Web também necessita tratar questões de gerenciamento de dados. Questões essas referentes a como e onde manterem-se os dados gerados nas execuçoes dos passos (serviços Web) distribuídos, de modo que fiquem acessíveis para uso por outros passos, novos workflows e também facilitem as atividades de análises eficientes dos workflows já executados. Além dessas, há a questão de como gerar e utilizar filtros de dados para os passos desses workflows científicos.

1.3 VISÃO GERAL DA PROPOSTA

Muitas das ferramentas de gerência de workflows (ALONSO, 2001), (BAUSCH et. al., 2002), (DURHAM et. al., 2005), (EUROGRID), (FOSTER et. al., 1995), (HOPPE, 2002), (KEPLER PROJECT), (LUDÄSCHER et. al., 2005), (MYGRID PROJECT), (STEVENS, 2003), (TARGINO, 2004), (TARGINO et. al., 2005) científicos existentes provêem poucas funcionalidades para a gerência de dados intermediários, ou não as provêem. Deixando sob a responsabilidade do executor do experimento, as atividades de coleta, agrupamento e manutenção desses dados.

Neste trabalho, propõe-se um sistema cuja arquitetura fornece mecanismos para gerenciamento de dados intermediários em workflows científicos compostos por serviços Web na área de bioinformática. Baseado na especificação desse sistema, foi desenvolvido um protótipo - *In Services* - que além de permitir a gerência de dados intermediários de modo a padronizar e facilitar o acesso aos mesmos, facilita também a (re)definição e execução de workflows. O sistema *In Services* é baseado no uso de serviços Web para modelagem de workflows, execução dos experimentos *in silico* e gerenciamento dos dados gerados durante a realização dos mesmos. A possibilidade de um armazenamento organizado e eficiente desses dados permite ainda aos cientistas de bioinformática uma melhor visualização e acesso aos mesmos. Soma-se a isso, a capacidade de re-execuções parciais de workflows através da recuperação dos dados armazenados e disponíveis para uso como entrada em outros workflows.

Na especificação do sistema proposto neste trabalho, a gerência de workflows inclui um módulo específico para gerência de dados intermediários. Tal módulo provê um conjunto pré-definido de serviços Web especializados no tratamento de dados gerados pelos passos dos workflows científicos - *serviços de dados*. Esse mesmo módulo efetua uma geração "semi-automática" dos serviços de dados para serem inseridos na composição original de workflows que necessitem de tratamento de dados intermediários. Os serviços de dados inicialmente propostos, fornecem funcionalidades de inserção de dados intermediários em um banco de dados, recuperação de dados do banco para serem utilizados em workflows e filtragem de dados que estejam fora de parâmetros necessários ao experimento. Através de uma interface amigável o sistema facilita aos usuários a geração desses serviços de dados e uso dos mesmos nos workflows científicos.

Para o armazenamento estruturado dos dados intermediários propõe-se o uso do Genomics Unified Schema - GUS - (DAVIDSON et. al., 2001), (BRETT TYLER'S LAB) que é uma plataforma apropriada para a gerência de dados genômicos e já adotada por alguns projetos de pesquisa de bioinformática (BAHL et. al., 2002), (HERTZ-FOWLER et. al., 2004), (LUCHTAN et. al., 2004). Essa plataforma possui um vasto esquema relacional que permite armazenar diversos dados de experimentos científicos ligados à pesquisa genômica. Além de sua funcionalidade para armazenamento de dados, a plataforma GUS possui um conjunto de ferramentas próprias para extração e análise de informação a partir dos dados armazenados.

1.4 CONTRIBUIÇÕES

Dentre as contribuições deste trabalho podemos citar:

a) Especificação de um sistema para gerenciamento de workflows científicos incluindo o tratamento de dados intermediários. Sistema esse que visa facilitar a redefinição de workflows para manutenção e integração de dados intermediários dos experimentos *in silico*, além de prover meios mais adequados para a realização de análises dos dados;

b) Identificação de serviços básicos para tratamento de dados em workflows científicos;

c) Incentivo à adoção de serviços Web como solução hábil para interoperabilidade de programas computacionais científicos;

d) Disponibilização de um protótipo funcional sobre a base de dados genérica GUS, podendo ser reutilizado pela comunidade de bioinformática;

e) Extensão do esquema do GUS para atender aos requisitos do sistema proposto;

1.5 ORGANIZAÇÃO DO TRABALHO

O presente trabalho está organizado da seguinte maneira. O capítulo 2 fornece uma visão geral sobre o conceito de Workflows; em seguida expõe os conceitos relativos a Workflows científicos, mostrando alguns exemplos na comunidade científica; a seção seguinte desse mesmo capítulo apresenta o uso de serviços Web na comunidade científica de bioinformática e por fim apresenta uma discussão sobre o uso de workflows científicos compostos por serviços Web dentro da área de bioinformática. O capítulo 3 faz uma breve análise sobre alguns sistemas para gerenciamento de workflows científicos procurando-se analisar as funcionalidades de gerenciamento de dados que esses sistemas oferecem. O capítulo 4 descreve a arquitetura do sistema *In Services* e o capítulo 5 expõe em detalhes o processo de redefinição de workflows para uso dos serviços para gerenciamento de dados. Ainda nesse capítulo, descreve-se um pouco as características de implementação do sistema *In Services* e algumas simplificações adotadas na especificação do mesmo. O capítulo 6 apresenta um exemplo de uso do protótipo funcional do sistema *In Services* e por fim, o capítulo 7 apresenta as conclusões do trabalho, as contribuições obtidas e os possíveis trabalhos futuros.

2 WORKFLOWS CIENTÍFICOS: FUNDAMENTOS

2.1 WORKFLOWS: UMA VISÃO GERAL

Workflow é a automação de um processo de negócio em parte ou como um todo onde documentos, informações e tarefas são passadas de um participante ao outro de acordo com um conjunto de regras definidas (HOLLINGSWORTH, 2005). É um conceito amplamente conhecido no meio empresarial onde as tarefas de produção ou os processos de negócio são executados em uma determinada ordem, havendo uma troca de dados entre eles para no fim de todo o fluxo de execução, obter-se um resultado final. São comumente conhecidos como workflows de negócio.

Um exemplo de workflow pode ser extraído de uma loja de vendas pela internet e ilustrado na FIG. 2.1. Tal workflow pode ser detalhado como segue: Inicialmente o usuário ao encontrar um produto que seja de seu interesse solicita efetuar a compra do mesmo (FIG. 2.1.a). Nesse momento o sistema da loja gera o número do pedido (FIG. 2.1.b) e solicita que o usuário escolha a forma de pagamento (FIG. 2.1.c). Dependendo da escolha do cliente o sistema irá efetuar diferentes processos como:

- aguardar o pagamento de boleto bancário (FIG. 2.1.c.1);
- solicitar o número do cartão de crédito (FIG. 2.1.c.2);
- emitir débito em conta corrente do cliente (FIG. 2.1.c.3);

Após a realização do pagamento por uma das opções, um novo processo será disparado que é emitir nota para o funcionário coletar os produtos que o cliente solicitou no seu pedido (FIG. 2.1.d). Concluído esse processo o sistema deverá encaminhar os itens do pedido para serem postados para o cliente (FIG. 2.1.e) e por fim emitir uma nota para o cliente afirmando que seu pedido já foi enviado (FIG. 2.1.f).

São notados, no workflow descrito, vários passos realizando atividades específicas. Cada passo pôde ser automatizado e a execução ordenada de cada um constituiu a execução do workflow. Nesse exemplo a obtenção do resultado de valor foi a venda e entrega do pedido ao cliente que só foi possível pela execução sincronizada desses passos distintos.

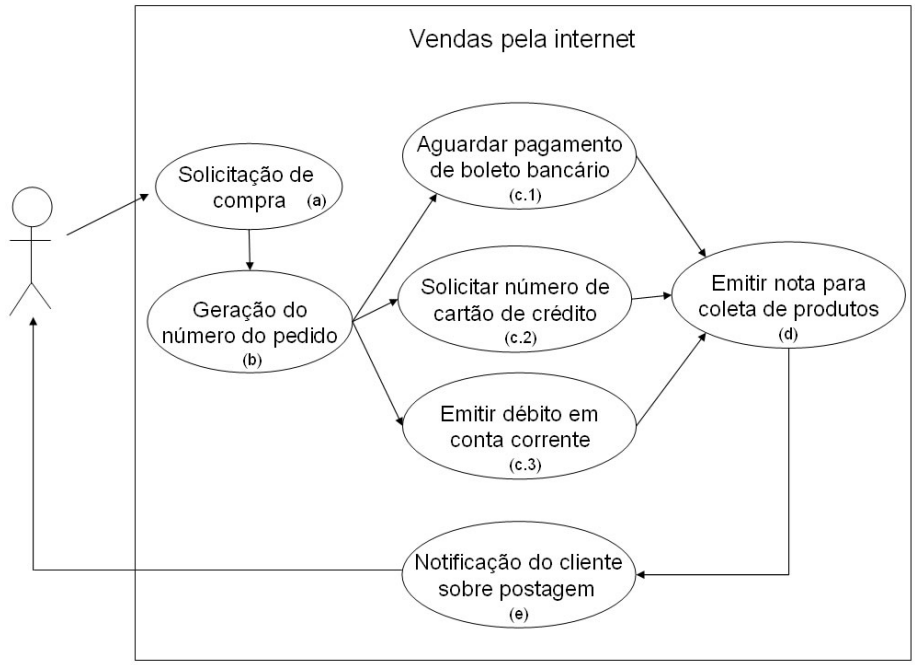

FIG. 2.1: Exemplo de um workflow em uma loja de vendas pela internet.

Com o uso da tecnologia de informação, foi possível, na execução de workflows, integrar e organizar a realização de atividades interdependentes permitindo uma série de benefícios organizacionais. Maior produtividade por funcionário, melhor atendimento aos clientes, melhor controle do andamento das atividades e menores tempos para a conclusão de processos são alguns desses benefícios que surgiram com o uso de wokflows.

Para o desenvolvimento de workflows, duas atividades são destacadas: a atividade de definição (ou modelagem) do workflow e a atividade de execução do mesmo (AALST et. al., 2002), (TARGINO, 2004), (TARGINO et. al., 2005).

Na atividade de **definição do workflow** é especificado como ocorrerá o fluxo de execução, indicando-se os passos envolvidos, as restrições existentes, a ordem de execução dos passos do workflow, o tratamento dos dados envolvidos, a existência de possíveis desvios no fluxo normal da execução e possivelmente o tratamento de erros.

Em (VAN DER AALST, 1999), o autor define formalmente um workflow como sendo uma coleção de tarefas (passos) organizadas para a realização de algum processo específico. A coleção de **tarefas** de um workflow é um conjunto de componentes de software independentes que implementam alguma funcionalidade. Exemplos de tarefas incluem executar um programa, transformar um arquivo ou atualizar um banco de dados. Um workflow

define também a **ordem de execução** dessas tarefas ou as **condições** em que serão executadas e sua eventual sincronização. Os **dados** de entrada e saída das tarefas (variáveis) são definidos como o fluxo de dados do workflow. No mesmo trabalho (TARGINO, 2004), é exposta uma definição mais formal de workflow *W* como uma quádrupla (*T, V, Sf, Cf*) onde:

- *T* é um conjunto t_1, t_2, ..., t_n de tarefas de *W*;
- *V* é um conjunto de variáveis v_1, v_2, ...,v_n de *W* definindo um fluxo de dados;
- *Sf* é uma função sucessora associada a cada tarefa $t \in T$, e
- *Cf* é uma função de condição associada a cada tarefa $t \in T$.

No trabalho desta dissertação, convencionou-se que tarefas e passos do workflow são sinônimos. Mas procurou-se utilizar o termo passo do workflow com maior freqüência, devido ao fato de ser um termo mais representativo para a implementação de alguma funcionalidade específica dentro do workflow e que em conjunto com outras compõem o worklfow como um todo.

Na atividade de **execução do workflow** os passos especificados na definição do workflow são executados considerando-se os quatro elementos que formalmente compõem o workflow (*T, V, Sf, Cf*). É necessário seguir a ordem de execução dos passos, os dados gerados possivelmente necessitarão de algum tipo de tratamento para serem trocados entre os mesmos, a invocação poderá ser feita de modo automático ou manual e pode haver uma ferramenta que monitore a execução.

Para auxílio na realização dessas duas atividades que envolvem o desenvolvimento de workflows existem os chamados Sistemas para gerência de Workflows (WfMS - *Workflow Management Systems*) que fornecem as funcionalidades necessárias à definição, execução e monitoramento. Algumas empresas de informática (ORACLE, Microsoft, IBM) disponibilizam softwares para o gerenciamento de workflows (IBM, 2005), (MICROSOFT, 2005), (ORACLE, 2005).

2.2 WORKFLOWS CIENTÍFICOS

Em (SINGH, 2005), workflows científicos são definidos como resolução de problemas científicos através de técnicas tradicionais de workflows. Ou seja, as idéias de execução de

um conjunto de tarefas em uma determinada seqüência foram aproveitadas na área científica para a realização de experimentos e estudos. Nos workflows científicos os passos são na maioria das vezes compostos por programas computacionais (programas científicos) que recebem, processam e geram um conjunto de dados científicos que podem ser repassados aos demais passos do workflow. O resultado de um workflow científico são os dados gerados pelo seu último passo que possivelmente fez uso de dados manipulados pelos passos anteriores a ele. Em muitas áreas da ciência e engenharia, o uso de recursos computacionais não é apenas intenso, mas também complexo e estruturado para fins de realização de estudos e experimentos. Para a execução de atividades que utilizam recursos computacionais, tem-se um ambiente onde cada passo do workflow é bem definido para executar uma funcionalidade específica. A execução em conjunto e em uma determinada seqüência desses passos auxilia na realização de um estudo ou experimento científico.

Embora o objetivo existente nos workflows tradicionais e científicos seja similar, ambos diferem em muitas características e necessidades:

- Ao contrário dos workflows tradicionais, os científicos podem ter suas definições mudadas constantemente, pois um mesmo experimento pode ser executado inúmeras vezes sobre uma seqüência de passos, dados e resultados diferentes;
- Os resultados de workflows científicos são sempre importantes, independente de serem positivos ou negativos, pois refletem o andamento de um experimento;
- Há a necessidade de um registro de como os dados gerados e manipulados pelos passos do workflow foram obtidos, caso seja necessário reproduzir o experimento. A esse registro dá-se o nome de proveniência de dados. Através das informações de proveniência, é possível adquirir uma rastreabilidade de como e em que contexto os dados foram gerados e o workflow executado;
- Os programas científicos que compõem o workflow podem estar em ambientes bem mais heterogêneos e distintos, como em grupos com escopo científico diferente, mas que fornecem alguma funcionalidade útil a um outro.

2.3 WORKFLOWS CIENTÍFICOS NA BIOINFORMÁTICA

No campo da bioinformática, workflows científicos são utilizados em experimentos biológicos que requerem análise e processamento computacional. Os workflows dessa área uti-

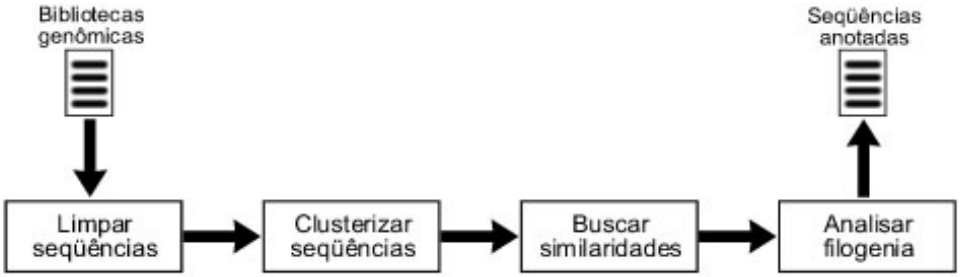

FIG. 2.2: Visão geral do Workflow GARSA para anotação de seqüências.

lizam informações dependentes de programas computacionais distintos. Cada programa computacional compõe uma parte específica do workflow, geralmente um passo específico do mesmo. Cabe ao workflow aplicado a essa área organizar a execução desses passos além de gerenciar os dados que serão processados por cada um deles dentro do workflow. Em (LEMOS, 2004) alguns workflows de bioinformática são descritos, o workflow MHOLline (MEYER et. al., 2004) é um deles.

O workflow MHOLline é um exemplo típico de workflow composto por uma seqüência de passos onde cada um deles realiza uma funcionalidade específica para a modelagem estrutural de proteínas. Um outro workflow (GARSA - (DÁVILA et. al., 2005)) tem como finalidade produzir um conjunto de seqüências anotadas FIG. 2.2. Seqüências anotadas podem ser definidas como a identificação de uma lista de segmentos de seqüência com algum significado biológico (GIBAS). A identificação de genes e de suas funções no genoma são um exemplo típico do processo de anotação. A partir da FIG. 2.2, percebe-se que para o experimento de anotação de seqüências ser efetuado, uma série de passos foi executada havendo-se troca de dados no fluxo da execução.

A execução de workflows no campo da bioinformática é feita, na maioria das vezes, em máquina local do biólogo através de sripts escritos, principalmente, em linguagem *Perl*. Os dados de cada programa computacional que compõe um passo do workflow são geralmente armazenados de maneira não estruturada como arquivos texto agrupados em diretórios ou mantidos como arquivos proprietários dos programas que os geraram.

O uso dessa linguagem de script para a modelagem de workflows, é de fácil assimilação pela comunidade de bioinformática e supre grande parte das necessidades de implementação dessa área. Em contrapartida, devido à necessidade de se desenvolver workflows

mais robustos nesse meio científico, esse tipo de linguagem não cobre satisfatoriamente as funcionalidades requeridas para tal desenvolvimento. Por exemplo, pelo fato de geralmente executarem localmente em máquinas onde os experimentos científicos de bioinformática são realizados, não há um ambiente favorável para disponibilização desses workflows para acessos remotos e fácil integração com outros ambientes científicos. Muitos dos processamentos de experimentos científicos requerem grande poder computacional devido ao fato de serem custosos e complexos. Isso leva à necessidade de uso exclusivo de computadores para processamento de certas tarefas. As linguagens de script em que workflows científicos na bioinformática são tradicionalmente modelados, não fornecem meios adequados para a execução de passos distribuídos. Por conta disso, a comunidade científica de bioinformática tem buscado soluções que provejam facilidade de acesso a aplicações científicas distribuídas. Uma das soluções adotadas vem sendo o uso de serviços Web para disponibilização e acesso dessas aplicações.

2.4 SERVIÇOS WEB DE BIOINFORMÁTICA

Serviços Web provêem um meio padrão de interoperabilidade entre diferentes aplicações de software, executando em plataformas e/ou *frameworks* heterogêneos (W3C, 2003). Um serviço Web é um sistema de software projetado para suportar interações entre aplicações de plataformas diferentes sobre uma rede de computadores; possui uma interface que descreve o serviço em um formato de arquivo processável por máquina - WSDL (Web Services Description Language). A interação de outros sistemas com o serviço Web ocorre via troca de mensagens SOAP (Simple Object Access Protocol) que podem ser transportadas via protocolo HTTP. Um programa como um serviço Web pode se comunicar com qualquer aplicação que utilize a mesma interface padrão para comunicação de serviços Web não importando a linguagem que as aplicações foram desenvolvidas ou a plataforma onde cada uma está sendo executada. Assim, é possível disponibilizar e acessar funcionalidades de qualquer serviço Web via sua interface padrão de acesso.

Cada serviço Web disponível para uso possui um artefato escrito e formatado em uma linguagem específica para descrever algumas características do serviço - WSDL - FIG. 2.3.A. Nesse artefato, são descritos os mecanismos de troca de mensagens com o serviço, ou seja, são definidos os formatos de mensagens, os tipos de dados que o serviço manipula, os protocolos de transporte das mensagens de requisição ao serviço e de resposta do mesmo. Uma aplicação cliente que interprete esse artefato descritor do serviço Web,

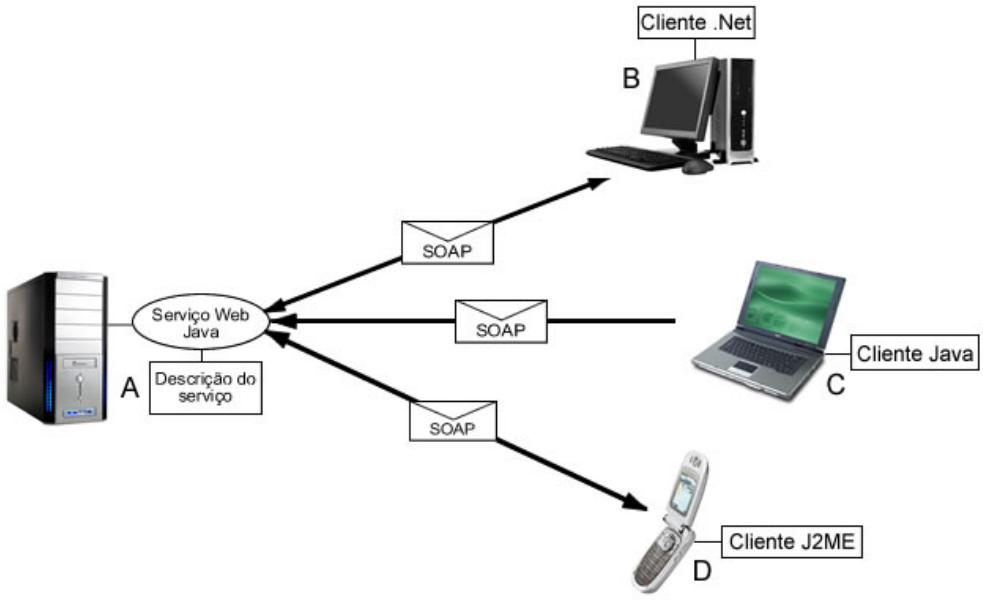

FIG. 2.3: Serviço Web sendo acessado por clientes de plataformas diferentes.

consegue enviar requisições ao serviço para que realize alguma atividade. As mensagens trocadas com os serviços Web seguem um padrão definido (SOAP) e podem ocorrer via o protocolo HTTP. Assim, aplicações que saibam interpretar os artefatos que descrevem um serviço Web e tenham a capacidade de trocar mensagens SOAP podem interoperar com os serviços Web, embora estejam implementadas em plataformas e/ou linguagens diferentes FIG. 2.3.B, C e D.

A tecnologia de serviços Web permitiu incorporar aos programas de bioinformática as características de facilidade para comunicação entre aplicações. Isso vem sendo feito construindo-se serviços Web que disponibilizem o acesso aos programas científicos de bioinformática. Alguns laboratórios científicos de bioinformática já disponibilizam alguns serviços Web que permitem executar parte de seus experimentos (ECKART et. al., 2003), (FINN et. al., 2006), (LIEFELD et. al., 2005), (STANDLEY, 2005). A arquitetura tradicional de workflows científicos como um conjunto de programas científicos executados em uma determinada seqüência e seguindo um fluxo lógico de execução pôde ser estendida para um conjunto de serviços Web dispostos, ou não, de maneira distribuída, arquiteturas heterogêneas ou mesmo podendo estar implementados em linguagens de programação distintas. Desse modo, pode ser obtido ganho na interação de aplicações participantes de um workflow de bioinformática.

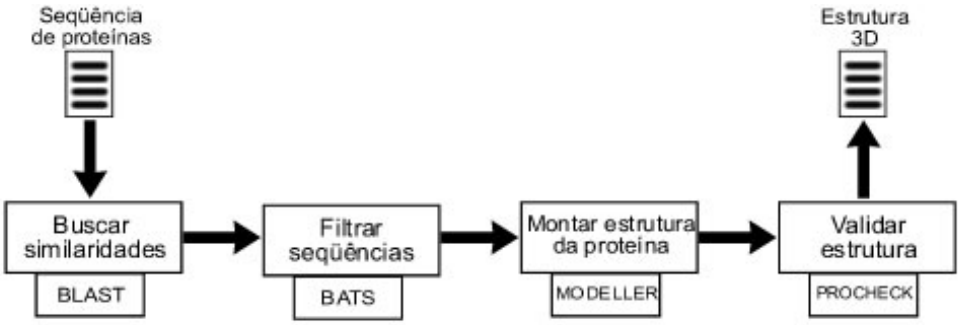

FIG. 2.4: Workflow MHOLline.

Disponibilizar os programas envolvidos em um workflow científico de bioinformática como serviços Web pode conferir a esses programas uma melhor capacidade de reutilização para diferentes experimentos, tendo-se um único programa (serviço) disponibilizando suas funcionalidades para outros experimentos de bioinformática que necessitem utilizá-lo. Essa característica também facilita o processo de definição de um workflow uma vez que é possível desenvolver-se uma arquitetura que permita o registro de serviços Web de bioinformática para posterior montagem de workflows através da seleção e ordenação de execução desses serviços Web, como desenvolvido em (TARGINO, 2004), (TARGINO et. al., 2005) e por outras ferramentas para suporte a workflows de bioinformática (GOBLE et. al., 2003), (LUDÄSCHER et. al., 2005).

Em (TARGINO, 2004), o autor expõe o workflow MHOLline fazendo uso de quatro programas que foram disponibilizados como serviços Web. Assim os mesmos podem interagir através da interface padrão que a tecnologia de serviços Web confere, podendo, cada um estar distribuído em máquinas distintas bem como implementados sobre plataformas e/ou linguagens de programação diferentes. A FIG. 2.4 ilustra o workflow (MHOLline) referido.

Nessa figura, nota-se, como dito anteriormente, que o workflow faz uso de quatro serviços Web, a saber: um que disponibiliza o acesso à execução do programa Blast (ALTSCHULA et. al., 1990), outro para o Bats (RÖSSLE, 2003), um para o Modeller (SALI) e um quarto para o programa Procheck (LASKOWSKI et. al., 1993). O primeiro serviço executado nesse workflow é o Blast juntamente com um arquivo de entrada contendo uma seqüência de proteínas. Através de comparações com um banco de dados, seleciona-se um conjunto de seqüências similares à fornecida, sendo que aquelas que

possuem maior semelhança serão utilizadas como molde para a construção da estrutura tridimensional. O serviço Web do BATS executado após o BLAST efetua uma filtragem sobre as seqüências similares geradas no primeiro passo para que somente aquelas que atendam alguns pré-requisitos sirvam como molde para a geração da estrutura tridimensional. O serviço seguinte (Modeller) recebe as seqüências dos passos anteriores e cria a estrutura 3D da proteína. Em seguida o Prochek avalia a estrutura gerada pelo Modeller e faz uma validação da mesma, ou seja, se a estrutura resultante pode ocorrer na natureza. Percebe-se ainda nesse workflow a troca de dados entre os passos envolvidos para a execução do experimento e obtenção do resultado final.

2.5 GERÊNCIA DE DADOS EM WORKFLOWS CIENTÍFICOS

Foram apresentadas nas seções anteriores algumas características dos workflows científicos. Nesta seção serão abordadas algumas questões referentes ao gerenciamento de dados nesses workflows e como propomos tratá-las.

A característica de modelagem *ad hoc* de workflows científicos na bioinformática através de linguagens de script, como *Perl*, gera uma dificuldade de redefinição desses workflows segundo a necessidade de diferentes cientistas. Isso porque, desse modo, esses workflows são modelados para atender necessidades individuais dos cientistas em um experimento científico dificultando assim o compartilhamento do workflow por um grupo de cientistas. Um exemplo disso é que um mesmo experimento científico pode gerar dados que possuem requisitos diferentes para os cientistas que executam o workflow. Isso pode levar à necessidade de que dados do workflow sejam filtrados para atender aos requisitos dos cientistas. A filtragem de dados pode ser feita após a execução do workflow, onde o cientista iria coletar o conjunto de dados gerados pelo workflow e então eliminar aqueles que estejam fora de parâmetros desejados. Ou, pode-se haver uma filtragem de dados durante a execução do workflow. Esse segundo método de filtragem pode ocorrer com a interferência do cientista após a execução de cada passo do workflow, onde os dados seriam analisados e selecionados apenas os de interesse do experimento. Em seguida, o executor do experimento dá continuidade ao mesmo passando os dados para os passos seguintes do workflow. Há ainda a possibilidade de que a filtragem seja automatizada, através da inserção de passos no workflow que analisam os dados gerados pelos programas científicos envolvidos no experimento e então efetuam a filtragem baseada nas necessidades do cientista. Essa última abordagem seria a mais desejável, pois livraria o cientista

da atividade de seleção manual dos dados o que poderia acarretar em possíveis erros de interpretação dos dados e conseqüentemente em uma filtragem de dados ineficiente. Porém, essa abordagem implica no desenvolvimento de filtros por parte dos cientistas e inserção dos mesmos no workflow.

Outra questão também apresentada nas seções anteriores sobre os workflows científicos na bioinformática é com relação à gerência de dados intermediários nesse ambiente. Esses dados são importantes para que possam validar como os passos dos workflows científicos foram executados. Além disso, esses dados podem ser utilizados para atender consultas analíticas baseadas em resultados de execuções de workflows passados. Ou seja, pode haver a necessidade de que as análises sobre os dados científicos requeiram comparações e/ou relacionamentos com dados intermediários ou com dados resultantes de execuções passadas de workflows. Os workflows científicos na bioinformática modelados sobre linguagens de script geralmente mantêm esses dados nos formatos proprietários em que foram gerados, ou em arquivos texto. Essa característica dificulta ainda mais o relacionamento entre os dados durante as atividades de análise e conseqüentemente a obtenção de informação. A gerência de dados intermediários é de importância também para permitir re-execuções parcias de workflows científicos. Ou seja, algumas vezes é necessário que experimentos científicos sejam re-executados parcialmente com novos parâmetros para que sejam analisados sobre diversos cenários. Assim, dados gerados por passos de workflows executados previamente podem ser reutilizados em futuras execuções do workflow, não havendo a necessidade de que todo o workflow seja re-executado.

Alguns projetos científicos de bioinformática propõem arquiteturas para gerência de dados no contexto de seus experimentos (DÁVILA et. al., 2005). Mais recentemente, iniciativas de esquemas de dados genéricos têm surgido no intuito de uniformizar o armazenamento de dados genômicos nos ambientes científicos de bioinformática. Dentre essas iniciativas, pode-se citar o Chado (CHADO, 2005) e o GUS (DAVIDSON et. al., 2001), (BRETT TYLER'S LAB) que consistem em um conjunto de módulos de um esquema de banco de dados biológico relacional. Além disso, também existem propostas de modelos genéricos de representação do conhecimento da área de bioinformática. Um exemplo disso são as ontologias (GRUBER et. al., 1995), (GUARINO, 1998), (STOFFEL, 1997) como Gene Ontology (GENE ONTOLOGY CONSORTIUM, 2006), TAO (Tambis Ontology) (BACKER et. al., 1999) e EcoCyc (IBM, 2005) que identificam e associam termos de Biologia molecular; e a ontologia definida no sistema myGrid (STEVENS,

2003),(WROE et. al., 2003) que define conceitos dos programas de análise de dados da biologia molecular. Com relação aos esquemas de dados, o Chado disponibiliza uma estrutura capaz de suportar dados genômicos relativos aos mais diversos experimentos. Os módulos presentes no esquema do Chado são relacionados a conceitos biológicos específicos e existem ainda outros módulos que permitem adicionar mais semântica aos dados biológicos com o uso de ontologias e vocabulários controlados. Neste trabalho, a gerência de dados científicos foi proposta utilizando-se a arquitetura do GUS. Essa escolha deveu-se inicialmente pelo fato de o GUS apresentar-se como um esquema de dados mais tipado, completo e maduro conceitualmente (NASCIMENTO, 2004). Soma-se a isso o fato de o projeto ao qual este trabalho esteve vinculado, BioWebDB (BIOWEBDB), passar a adotar o GUS como plataforma para armazenamento de seus dados científicos.

O Genomics Unified Schema - GUS, além de prover um extenso esquema, contém um conjunto de ferramentas (plug-ins) que possibilitam extrair dados de fontes heterogêneas e inseri-los nas tabelas do esquema. Sendo possível integrar dados de diversos bancos de dados genômicos públicos amplamente utilizados pela comunidade científica de bioinformática; contendo seqüências de DNA, RNA e proteínas. Uma característica interessante sobre o esquema do GUS é que o mesmo foi modelado tendo como base o dogma central da biologia molecular que descreve que o DNA pode se replicar e dar origem a novas moléculas de DNA, pode ainda ser transcrito em RNA, e este por sua vez traduz o código genético em proteínas. Essa característica contribui para o entendimento mais intuitivo de como as informações, nele contidas, se relacionam. Há ainda, a possibilidade de se adicionar mais semântica aos dados presentes no esquema. Vocabulários controlados, Ontologias, proveniência dos dados, publicações científicas podem ser vinculados aos dados genômicos armazenados, conferindo a esses últimos a possibilidade de se obter informações analíticas mais precisas. Para as atividades analíticas, o GUS conta com um módulo que permite montar um ambiente Web para consulta dos dados armazenados. Esse módulo denominado de WDK (Web Development kit) possibilita uma rápida modelagem de interfaces para consultas e análises sobre os dados genômicos armazenados.

O esquema do GUS, bem como o do Chado, possui um conjunto de módulos que agrupam tabelas relativas a informações distintas, mas que se relacionam no contexto de todo o esquema. Os módulos presentes no esquema do GUS são *Core*, *Dots*, *Tess*, *Prot*, *Rad*, *Study* e *Sres*. O conjunto de tabelas de cada módulo permite formar categorias de dados. Um exemplo de categoria seria aquela que mantém a proveniência de dados armazenados

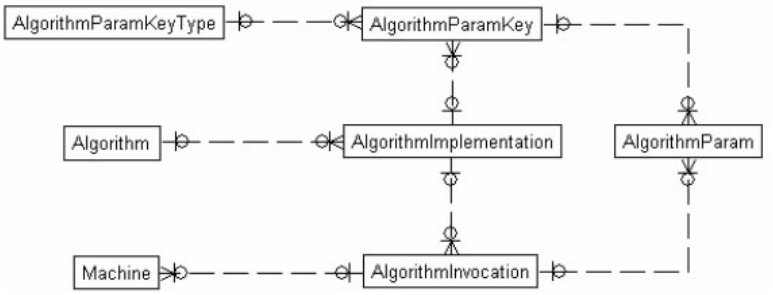

FIG. 2.5: Representação de algoritmos através do módulo Core (NASCIMENTO, 2004)

e é formada pelas tabelas do módulo *Core* que registram a execução de algoritmos que manipularam dados armazenados - FIG. 2.5. Os algoritmos e suas implementações são representados respectivamente pelas tabelas *Algorithm* e *AlgorithmImplementation*. A primeira permite manter o nome e a descrição dos algoritmos enquanto a segunda mantém os dados relacionados às implementações do algoritmo. As invocações dos algoritmos são registradas pela tabela *AlgorithmInvocation*. As tabelas *AlgorithmParam*, *AlgorithmParamKey* e *AlgorithmParamKeyType* representam os parâmetros que os algoritmos e suas invocações podem utilizar. Há ainda a possibilidade de se representar dados técnicos da máquina onde os algoritmos foram invocados através da tabela *Machine*.

Dentro da categoria de proveniência de dados tem-se, além dos dados referentes a execuções dos algoritmos, informações sobre bancos de dados externos que podem conter dados armazenados no GUS e serem então referenciados. Há também a categoria de dados de administração do GUS contendo informações de usuários, grupos e projetos que fazem uso do esquema além de metadados contendo informações sobre as tabelas do esquema e do banco de dados como um todo.

Por fim, há as categorias referentes às informações genômicas. Um exemplo seriam as categorias de Seqüências e Características; Funções Genômicas; Transcrição; e Experimentos. Essas categorias relacionam tabelas dos módulos *Dots*, *Tess* e *RAD*; com conteúdo mais específico para dados genômicos.

A arquitetura do GUS faz com que as aplicações ao inserirem dados no esquema o façam de maneira consistente. Isso é garantido pelas restrições de integridade modeladas no esquema, pelo fato de a maioria dos dados serem carregados com o uso dos plug-ins e

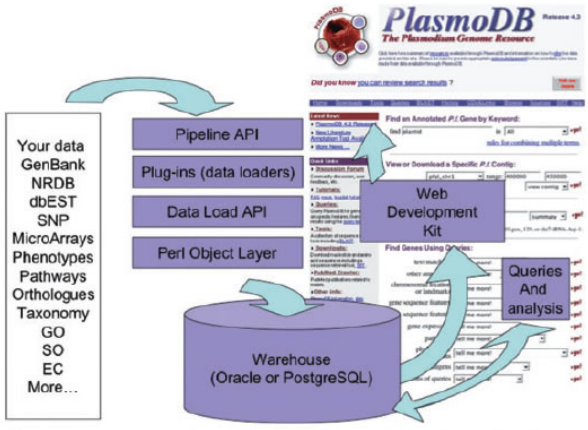

FIG. 2.6: Visão geral da arquitetura do GUS (STOECKERT, 2005)

pelo registro de cada execução dos plug-ins. Sempre que uma carga de dados é efetuada no GUS via algum plug-in, é feito o registro em tabelas do esquema sobre a execução do mesmo. Essa característica permite ter ainda a proveniência de dados em nível de item de dado armazenado. Assim há possibilidade de se mapear quando, quais e por quem os dados foram carregados.

Na FIG. 2.6 temos a arquitetura do GUS (Genomics Unified Schema) ilustrando o banco de dados e as aplicações que fazem parte da ferramenta. Dados de diferentes fontes são carregados usando a camada API Pipeline que serve como uma interface para acesso às funcionalidades de carga de dados. Essas funcionalidades são implementadas pelos plug-ins que se comunicam com o banco de dados via uma camada de objetos Perl. O banco de dados armazena os dados e pode ser diretamente consultado ou utilizado para fornecer dados a projetos científicos. Um exemplo seria o projeto PlasmoDB (BAHL et. al., 2002), (PLASMODB) que possui sua estrutura de dados genômicos montada sobre a arquitetura do GUS.

2.6 CONSIDERAÇÕES FINAIS

No contexto de workflows científicos na bioinformática compostos por serviços Web, novas perspectivas para a gerência de workflows científicos surgiram. O uso de serviços Web possibilitou a execução distribuída de passos dos workflows científicos na bioinformática.

Essa tecnologia permitiu ainda que cada passo pudesse interoperar com os demais sem a preocupação de homogeneidade de linguagens de programação ou plataforma de execução. Porém, a natureza de distribuição das execuções dos passos de workflows nesse novo contexto passou a tratar diferentes questões sobre a gerência de dados intermediários. Questões de como e onde disponibilizar esses dados nesse ambiente distribuído são ainda não solucionadas eficientemente no meio científico. Além disso, ainda existe a questão de como atender às necessidades de realização de filtragem de dados pelos cientistas nesse novo contexto de workflows.

Neste trabalho, baseado na importância e carência de gerência dos dados intermediários em workflows científicos compostos por serviços Web, foi proposta uma arquitetura que objetiva prover mecanismos para a manutenção e disponibilização desses dados científicos. Com isso, pretende-se atender às necessidades de filtragem de dados nos workflows científicos, re-execuções parciais de workflows e análises mais eficientes sobre os dados científicos gerados a partir das execuções dos workflows. A arquitetura é apresentada a partir do protótipo desenvolvido - *In Services* - que faz uso do GUS como meio de armazenamento dos dados. A escolha do GUS como meio de armazenamento dos dados se deveu ao fato de ter sido a inciativa que mais se encaixou ao nosso escopo de bioinformática.

Alguns trabalhos relacionados à gerência de workflows científicos serão analisados no capítulo seguinte, para então detalhar-se a arquitetura do sistema *In Services*, proposto neste trabalho.

3 SISTEMAS PARA GERÊNCIA DE WORKFLOWS CIENTÍFICOS

O uso de serviços Web para disponibilização de programas de bioinformática facilitou a composição de workflows científicos, como já citado anteriormente. O compartilhamento mais adequado desses programas, conferido pela tecnologia de serviços Web, aumentou a possibilidade de uso de um único serviço por diferentes grupos de cientistas, uma vez que passa-se a ter uma interface padronizada de acesso ao serviço (SOAP) e disponível em um meio de acesso global, internet, via protocolo HTTP. Desta forma, a comunidade de bioinformática pôde fazer uso dos serviços Web para realização de uma atividade específica a ser definida como um passo de um workflow no contexto de experimentos científicos. Através de chamadas remotas aos serviços do workflow, o mesmo pode ser executado. Essa característica traz a vantagem de que os passos do workflow não necessitem executar localmente na máquina onde o experimento deve ser realizado, além de disponibilizar novos serviços desenvolvidos por outras comunidades científicas (reuso de serviços). Assim um grupo de estudo científico pode desenvolver um serviço específico ligado ao seu foco de estudo e ao mesmo tempo deixar esse serviço acessível pela Web para qualquer outro que possa ter interesse em utilizá-lo.

Devido a essa característica de distribuição, inerente aos serviços Web, atividades de localização, chamada, e gerenciamento de execução de cada serviço pode ser de certa complexidade para um cientista que apenas necessite executá-los. Baseado nisso, alguns estudos têm sido direcionados para o desenvolvimento de sistemas que permitam definir, gerenciar e executar workflows científicos que utilizam serviços Web em alguns, ou todos, passos dos workflows. Tais sistemas visam facilitar as tarefas de criação e execução de workflows científicos, pois permitem integrar, junto ao sistema, serviços Web destinados a tarefas científicas para uso nos workflows definidos. Neste capítulo são demonstrados alguns sistemas propostos para dar suporte à modelagem e execução de workflows científicos, e por fim um breve comparativo entre eles é mostrado. Inicialmente, são descritos alguns sistemas que trabalham com workflows científicos; em seguida, é dada uma ênfase maior para aqueles que gerenciam workflows científicos permitindo o uso de serviços Web na composição de seus passos, uma vez que foi para esse tipo de workflow que este trabalho foi desenvolvido.

O primeiro deles, EGene (DURHAM et. al., 2005), é um sistema que permite definir workflows integrando programas computacionais biológicos distintos. A execução integrada desses programas representa a execução do workflow. O gerenciamento de dados nos workflows definidos no EGene é de reponsabilidade do usuário. Cada passo do workflow é configurado para gerar um arquivo XML como resultado. Um outro sistema, LabBase (GOODMAN et. al., 1998), foi proposto para modelar e integrar dados em ambientes científicos. Esse sistema provê suporte para o gerenciamento de workflows contendo operações para armazenamento e recuperação de dados em e a partir de bancos de dados. O desenvolvimento do LabBase foi feito na linguagem *Perl* o que pode dificultar a obtenção das características de fácil interoperabilidade e execução remota nos workflows definidos.

Voltado para aplicações de bioinformática o BioOpera (ALONSO, 2001), (BAUSCH et. al., 2002) oferece uma arquitetura de cluster de PCs, permitindo a definição de workflows científicos para a realização de experimentos. Cada aplicação envolvida poderá ser executada de maneira distribuída nos PCs do cluster objetivando uma otimização na execução dos algoritmos envolvidos. O ambiente permite que os usuários definam o workflow de maneira amigável, através da seleção de um conjunto de aplicações pertencentes a uma biblioteca de aplicações do BioOpera, além de possibilitar um monitoramento das execuções do workflow.

O Chimera (FOSTER et. al., 1995) é um workflow como solução para geração de dados por demanda implementando, com isso, um sistema de dados virtuais. O objetivo do Chimera é transformar os dados científicos de uma maneira que possa ser mais facilmente utilizada pela comunidade de pesquisadores. O mesmo é um sistema onde é possível ser especificado, através de uma Linguagem Virtual de Dados (VDL), como as transformações sobre os dados devem ser feitas. Inclui um interpretador para VDL que permite aos usuários submeterem requisições para construção ou reconstrução de conjunto de dados. Reconstrução é útil para localizações remotas nos casos em que a transferência de dados é um fator de custo elevado. Através da VDL, é possível povoar um banco de dados do Chimera com definições de dados além de possibilitar consultas ao mesmo.

Nos conceitos que envolvem o sistema Chimera uma distinção entre transformação e derivação de dados é feita; distinção essa que é importante para um sistema de dados virtuais. Transformação de dados contém informações sobre operações a serem executadas sobre os dados, o programa a ser executado, argumentos, etc. Uma derivação especifica

os argumentos para execução de uma transformação. A VDL do Chimera permite que usuários especifiquem as transformações e derivações necessárias à geração do conjunto de dados. As definições de transformações e derivações feitas sobre os dados nesse ambiente correspondem a um workflow. O ambiente para execução de workflows tem seu foco voltado para tratamento de dados a serem transformados e processados por experimentos científicos em meio computacional.

O Eurogrid (EUROGRID), (HOPPE, 2002) demonstra o uso de Grids para comunidades científicas e indústria. Tem como objetivo estabelecer um grande ambiente de Grid na Europa para formação de centros computacionais de alto desempenho; efetuar simulações distribuídas para aplicações de algumas áreas (simulações biomoleculares, meteorologia, aplicações de engenharia); e tornar essa plataforma de Grid disponível também para outras plataformas semelhantes.

A proposta do Eurogrid é fornecer uma infra-estrutura capaz de prover alto desempenho para a execução de experimentos científicos, definidos via workflows. Usuários membros do projeto podem utilizar a arquitetura para a realização de experimentos que requerem um grande volume de recursos computacionais.

Os sistemas de gerência de workflows apresentados até aqui são sistemas que não fazem proveito do uso de serviços Web científicos para definição e execução de workflows. Os próximos sistemas a serem apresentados dão suporte ao uso de serviços Web para comporem passos dos workflows definidos. Por esse motivo terão uma consideração maior em suas apresentações e serão comparados entre si segundo alguns critérios que o sistema *In Services* propõe como funcionalidade.

3.1 MYGRID TOOLKIT

O myGrid Toolkit (MYGRID PROJECT), (STEVENS, 2003) é um sistema desenvolvido pelo projeto myGrid (MYGRID) e consiste em uma coleção de serviços que objetivam oferecer um alto nível de integração para dados e aplicações biológicas. Os componentes de sua arquitetura servem a dois propósitos distintos. O primeiro é prover um ambiente onde a definição e execução de experimentos *in silico* possam ser estruturadas através de workflows e processamento de consultas distribuídas. O segundo é de se ter um sistema onde a realização de experimentos *in silico* possa ser bem executada através do registro da origem dos dados e de mecanismos para configuração de experimentos.

Os componentes que fazem parte do myGrid realizam atividades específicas para dar

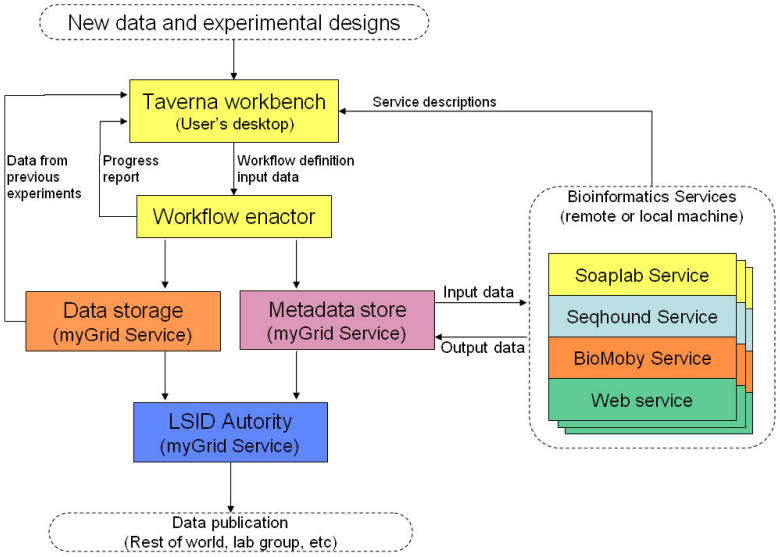

FIG. 3.1: Componentes do myGrid Toolkit (MYGRID PROJECT)

suporte à definição e execução de workflows, descoberta de recursos, gerenciamento de execução dos processos envolvidos no experimento, armazenamento de dados intermediários e metadados, além da exploração dos dados gerados. A FIG. 3.1 ilustra como esses componentes interagem no sistema.

O componente Taverna Workbench é o que permite a interação do usuário com a sistema. A partir dele é possível realizar as atividades de definição e execução de workflows científicos, bem como explorar e analisar dados resultantes de execuções anteriores. O componente Workflow Enactor é o que faz o gerenciamento da execução de cada passo do workflow. O mesmo controla o fluxo de dados entre os passos envolvidos na execução do workflow. O myGrid implementa sua própria linguagem de definição de workflows (Scufl (TAVERNA PROJECT, 2005)) que consegue cobrir as atividades de definição de seqüência de passos do experimento, bem como a definição dos dados utilizados e tratamento de exceções que possam ocorrer durante a execução do workflow. Os componentes Data Storage e Metadata Store fazem o armazenamento dos dados resultantes da execução do workflow e de metadados envolvidos na execução do experimento científico. Para implementar essa funcionalidade, o myGrid Toolkit faz uso do SGBD mySQL para manter esses dados em um esquema definido e usado pelo sistema. O componente LSID Authority é o que faz a identificação única de cada dado gerado na execução. Este componente faz uso de Life Science Identifiers-LSID (OMG) que permite uma identificação única e

padronizada para tipos de dados provenientes de qualquer aplicação.

Esse sistema oferece um mecanismo de definição de workflows bem simplificado. A partir de uma lista de programas e serviços Web de bioinformática disponibilizada no sistema, o usuário facilmente monta a seqüência de passos necessária para a execução do workflow. Além de especificar como ocorrerá o fluxo de dados de um processo para outro, o myGrid permite ainda que quaisquer serviços de bioinformática possam ser adicionados à sua lista de serviços, disponibilizando-os para posterior uso em novos experimentos. Enfim, o myGrid Toolkit apresenta-se como um sistema de simples uso para a realização de experimentos científicos de bioinformática. Uma característica que vale ser mencionada sobre o myGrid é a capacidade de descrição semântica dos serviços que compõem o sistema. Essa descrição é feita com o uso de ontologias que permitem fornecer informações mais precisas sobre os processos e dados envolvidos no workflow definido. Há ainda o suporte a proveniência de dados. Mantendo-se informações sobre por meio de quais processos os dados foram gerados permitindo uma melhor análise dos mesmos. Durante o desenvolvimento deste trabalho, o myGrid não apresentava mecanismos que possibilitassem a seleção de dados de maneira automática durante a execução do workflow.

3.2 KEPLER

O propósito do Kepler (LUDÄSCHER et. al., 2005), (KEPLER PROJECT) é bem semelhante ao do myGrid, porém este não pretende ser restrito ao ambiente científico de bioinformática como é o myGrid. O Kepler pretende também servir a outros experimentos científicos como acompanhamento de dados ecológicos (SEEK PROJECT), estudos geológicos (GEON), etc. Também é um sistema com a finalidade de permitir a definição e execução de workflows científicos usando a tecnologia de serviços Web e de computação distribuída. Faz uso de uma interface gráfica para interação com o usuário, facilitando as atividades envolvidas na realização do experimento *in silico*. A FIG. 3.2 ilustra um workflow definido no Kepler.

A linguagem de definição do workflow no Kepler é a Modelling Markup Language-MoML (LEE et. al., 2000) e também cobre as funcionalidades propostas pelo sistema de gerência na execução dos processos que compõem os workflows. De modo similar ao myGrid, é possível no Kepler adicionar e executar novos serviços necessários à realização de um experimento através da importação de serviços Web e/ou programas escritos em linguagens suportadas pelo sistema como Java, Python e scripts de Matlab.

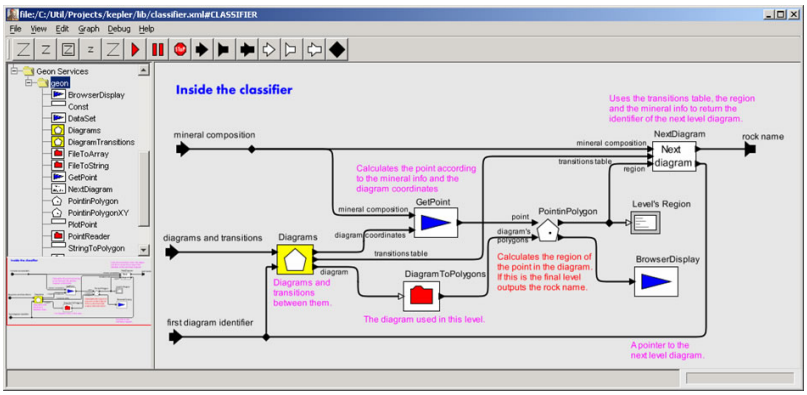

FIG. 3.2: Workflow definido no sistema Kepler

No momento da escrita deste trabalho, o sistema ainda não apresentava um meio consistente para o registro de origem de dados e metadados. Assim, o acompanhamento da proveniência de dados gerados na execução de workflows para melhores análises é um pouco comprometido. Quanto à persistência de dados intermediários e gerenciamento de fluxo de dados no workflow, o Kepler propôs que os dados gerados durante a execução do workflow sejam mantidos na máquina onde foi executado o passo que os gerou. Caso um outro passo do workflow necessite desse dado para processamento, o mesmo não é transferido pela rede; apenas será transferido um "ponteiro lógico para o dado" que informa onde o dado se encontra e qual protocolo permite acessar o mesmo. Dessa forma, tem-se reduzido o volume de dados que possam trafegar pela rede para alimentar os passos do workflow.

Para lidar com a heterogeneidade de formatos de dados oriundos dos passos distintos do workflow, o Kepler faz uso de um conjunto de serviços para tratamento de dados, acessos a bancos de dados e execução de consultas distribuídas. Dentre os serviços de tratamento de dados, o Kepler contém um conjunto de transformadores que permitem converter dados em formatos diferentes. Com relação à descrição semântica dos processos e dados no ambiente Kepler, essa característica ainda estava sob estudo para fazer parte das funcionalidades do sistema.

3.3 10+C

10+C é um sistema proposto e desenvolvido por Targino (TARGINO, 2004), (TARGINO et. al., 2005) para dar suporte ao desenvolvimento e execução de workflows científicos

na bioinformática com serviços Web. É baseado em uma interface Web que permite o registro de serviços, definição e execução de workflows científicos. Sua implementação foi feita em linguagem Java para Web (Jsp e Servlets) juntamente com um SGBD mySQL para armazenamento de dados de controle dos workflows executados. Para se definir um workflow faz-se o registro dos serviços Web no sistema para então ser possível organizar o fluxo de execução de cada serviço no workflow. A linguagem BPEL4WS - Business Process Execution Language for Web Services (CURBERA et. al., 2003) foi utilizada para a coordenação da execução dos serviços do workflow, assim o processo de execução do workflow definido no 10+C é traduzido para a linguagem BPEL4WS que então é interpretada pela máquina de execução de workflows no momento que esses devem ser executados. Os próprios workflows desenvolvidos no ambiente 10+C são disponibilizados como serviços Web, favorecendo que sejam reutilizados como parte de outros experimentos.

Para validar o desenvolvimento do ambiente 10+C, Targino utilizou o experimento MHOLline, apresentado anteriormente, onde cada programa requerido pelo experimento foi disponibilizado como um serviço Web e então registrado no sistema. Com isso, pôde-se definir o fluxo de execução, compondo cada serviço Web como um passo do workflow, definindo-se também os parâmetros de entrada necessários para a execução de cada um deles, bem como o fluxo de dados que deve ocorrer para alimentar os requisitos desse experimento, ou seja, quais dados provenientes de quais serviços devem alimentar os parâmetros de entrada de passos seguintes. Em sua validação, Targino constatou que o workflow MHOLline definido no sistema 10+C possuiu um método de definição mais simples, pois não ficou vinculado à linguagem de script antes utilizada para sua definição. Quanto à execução do workflow, ficou constatado que o uso inicial em linguagem de script (*Perl*) era dependente da máquina em que iria ser executado e que os programas que fazem parte do mesmo, necessitavam estar funcionalmente disponíveis em uma mesma maquina; fato que não ocorre na execução desse workflow com serviços Web, uma vez que o ponto de início do mesmo é através do sistema que invoca os serviços para a execução da tarefa específica de cada um e coordena tais execuções até a conclusão do experimento. Sendo assim, se mais de um cientista necessitar executar esse mesmo experimento, pode fazê-lo utilizando o workflow já definido no ambiente 10+C. Não estando vinculado a uma máquina específica ou tendo que redefinir o mesmo workflow para seu uso individual.

A arquitetura do 10+C apresentada na FIG. 3.3 apresenta os componentes que dão suporte à execução de suas funcionalidades. Na FIG. 3.3.a as aplicações científicas são

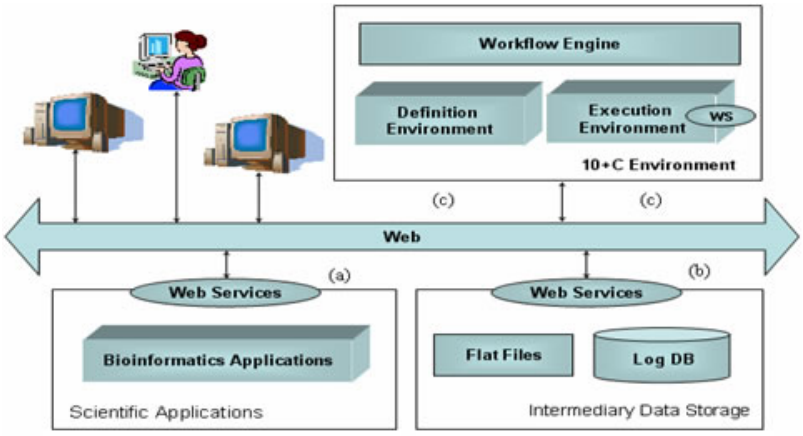

FIG. 3.3: Arquitetura do Ambiente 10+C

registradas no sistema para serem então disponibilizadas para uso em workflows. Na seção (c) da FIG. 3.3, têm-se os módulos de Definição de workflows onde os cientistas podem modelar o workflow. Essa modelagem ocorre selecionando-se os serviços Web previamente registrados compondo-os como passos do workflow. Além disso, organiza-se o fluxo de execução desses serviços e o fluxo de dados entre os passos do workflow. A modelagem do workflow no ambiente 10+C gera, no fim de seu processo, um arquivo em linguagem BPEL4WS. Tal arquivo deve então ser utilizado para fazer a disponibilização do workflow junto ao módulo Workflow Engine, essa disponibilização é que permite à máquina de execução de workflows entender como o workflow deve ser processado, como deverá ocorrer a seqüência de execuções dos passos do workflow e as interações entre esses. O módulo de Execução de workflows promove a execução do workflow fazendo as devidas invocações aos serviços Web que compõem o mesmo, passando os parâmetros de entrada necessários e controlando o fluxo de dados de um serviço para o outro. Na FIG. 3.3.b é apresentado o módulo onde o sistema efetua o armazenamento dos dados gerados durante a execução do workflow. Como dito anteriormente, o 10+C faz uso do SGBD mySQL para manter os dados sobre serviços registrados no sistema, workflows modelados e executados, variáveis dos serviços, etc... Desse modo, além de facilitar a composição de novos workflows, workflows já modelados podem ser executados diversas vezes mudando-se apenas parâmetros de entrada, caso seja necessário; favorecendo a re-execuções de experimentos além de manter um registro de proveniência dos dados sobre experimentos

executados. Quanto aos dados gerados na execução dos serviços Web que fazem parte dos passos de execução do workflow, esse sistema os mantêm em arquivos de texto comum no mesmo sistema de arquivo onde o ambiente 10+C é executado. Essa característica permite a re-execuções parciais de workflows executados previamente, ou seja, é possível realizar um experimento a partir de um determinado ponto aproveitando-se resultados obtidos anteriormente e disponíveis nesse meio de armazenamento.

O modo de armazenamento de dados intermediários implementado pelo sistema 10+C satisfaz às necessidades de recuperação de dados para re-execuções parciais, mas não é um modo favorável a análises desejáveis pelos cientistas que queiram obter mais informações sobre execuções de seus experimentos, uma vez que arquivos de texto comum não são um meio bem estruturado de serem mantidas informações, dificultam uma integração entre informações diferentes presentes em arquivos distintos, e não fornecem um meio adequado para manipulação de dados. O fato de o sistema 10+C possuir seu mecanismo próprio de armazenamento de dados, pode gerar um isolamento evitando o reuso dos dados por outros sistemas de execução de workflows científicos compostos por serviços Web.

3.4 COMPARAÇÃO ENTRE OS SISTEMAS

Os sistemas para gerência de workflows científicos mencionados nesse capítulo serão aqui analisados com base em algumas funcionalidades que julgamos interessantes que fossem fornecidas por esses tipos de sistemas. Ao fim da seção, uma tabela com um resumo comparativo é apresentada.

A primeira das funcionalidades desejáveis para a gerência na execução de workflows científicos seria a presença de um meio estruturado que permitisse armazenar e disponibilizar os dados gerados na execução do workflow - *dados intermediários*. Como meio estruturado para armazenamento e disponibilização de dados entende-se como um mecanismo que garantisse a consistência e segurança dos dados armazenados e fornecesse acesso de maneira precisa aos dados que se tem interesse. Dos sistemas aqui mencionadas, o myGrid possui um componente, *Data Storage*, que, montado sobre um SGBD, permite o armazenamento desses dados. O uso de um SGBD para essa finalidade pode tomar proveito das funcionalidades de gerência que os SGBDs garantem sobre os dados; tais como consistência, segurança, consultas eficientes e etc. O Kepler mantém os dados gerados nos passos do workflow no local onde foram gerados; para acesso a esse dados, um ponteiro lógico é fornecido indicando onde o dado está e qual protocolo é necessário para

acessá-lo. Dessa maneira, pode-se afirmar que o mecanismo de armazenamento de dados intermediários do Kepler não é estruturado, pois os dados gerados nos passos do workflow permanecem onde foram gerados e em seus formatos proprietários. Tal característica não facilita a extração de informações relacionadas entre os dados gerados pelos passos que compõem o workflow, devido ao fato de ser necessário saber interpretar diferentes formatos extraindo as informações de interesse e relacioná-las. O sistema 10+C provê um mecanismo de armazenamento de dados intermediários utilizando arquivos texto e mantendo-os na máquina onde o sistema é executada. Assim, a busca dos dados gerados pelas execuções de workflows fica centralizada e a obtenção desses dados é facilitada. Porém, o fato de os dados serem mantidos como arquivos texto dificulta as análises realizadas sobre os mesmos.

A segunda funcionalidade desejável nos sistemas de gerência de workflows científicos é a capacidade de se realizar execuções parciais dos workflows. Isso seria desejável quando um cientista já possui uma certa quantidade de dados que o permitiria executar o workflow de um certo passo que não o inicial. Essa funcionalidade seria de certo modo dependende da funcionalidade de armazenamento e disponibilização de dados intermediários, citada no parágrafo anterior. Isso devido ao fato de se usar os dados gerados em execuções prévias dos workflows para re-executar parcialmente novos workflows. Os três sistemas aqui discutidas provêem essa funcionalidade. Embora esses sistemas forneçam mecanismos diferentes de disponibilização dos dados intermediários, é possível obtê-los e assim alimentar o passo desejado de um workflow.

Uma outra característica interessante de ser fornecida por esses sistemas seria a manutenção da proveniência de dados. A proveniência de dados é de grande importância principalmente para o meio científico uma vez que os dados envolvidos nas pesquisas precisam de uma garantia que são corretos e que foram gerados dentro dos padrões esperados. A proveniência de dados tem o objetivo de manter informações sobre os procedimentos que os geraram garantindo a rastreabilidade de como os mesmos foram gerados e manipulados. Dos sistemas aqui discutidos o myGrid e 10+C possuem estruturas que dão suporte à proveniência de dados dos workflows. Ambas mantêm registros sobre os procedimentos e pessoas que executaram workflows científicos. O Kepler, no momento da escrita deste trabalho ainda não tinha um mecanismo para proveniência de dados, mas já havia a preocupação de se ter suporte a essa funcionalidade, como relatado em (LUDÄSCHER et. al., 2005).

TAB. 3.1: Comparação entre os sistemas apresentados, segundo as funcionalidades desejáveis.

	myGrid	Kepler	10+C
Meio estruturado para armazenamento e disponibilização de dados gerados nas execuções nas execuções de workflows	X	-	-
Ambiente que facilita análises sobre os dados gerados	X	-	-
Capacidade de execuções parciais a partir de dados gerados por execuções prévias dos workflows	X	X	X
Proveniência de dados	X	-	X
Filtragem de dados automática durante a execução de workflows	-	-	-

Geralmente, na execução de workflows, uma grande quantidade de dados é processada e gerada. Dependendo do experimento, alguns desses dados podem ser descartados por conterem valores que estejam fora do contexto do experimento. No processo tradicional de execução de workflows científicos essa filtragem é geralmente feita no momento em que as análises sobre os dados são efetuadas. Essa característica favorece a uma possível análise falha, pois há a possibilidade de que dados irrelevantes ao experimento científico sejam interpretados levando-se a um estudo inconsistente. Uma abordagem onde os dados intermediários pudessem ser filtrados no momento em que são gerados seria interessante pois não haveria a necessidade dessas filtragens serem executadas no momento do estudo dos dados. Outra vantagem da aplicação dessa abordagem de filtragem de dados, é que poder-se-ia reduzir o volume de dados a serem processados no fluxo do processo científico. Dos sistemas aqui discutidos nenhuma delas fornece tal suporte para realização de filtros. O que poderia ser feito é utilizar passos no workflow que efetuassem a filtragem de dados gerados pelo passo anterior do workflow. Porém, essa definição de filtros estaria fora do momento de definição do workflow no sistema. Ou seja, os filtros seriam implementados e disponibilizados em um momento anterior à modelagem do workflow e então utilizados como passos do mesmo. Uma abordagem de geração automática de filtros no momento da definição de workflows pelo próprio sistema livraria o usuário das atividades de implementação ou descoberta de filtros.

A TAB. 3.1 expõe um resumo do que cada sistema apresentado neste capítulo disponibiliza ou não. Aqueles sistemas que possuem a funcionalidade citada na primeira coluna são marcadas com (X) enquanto aqueles que não apresentam, são marcadas com (-).

4 SISTEMA *IN SERVICES* - ARQUITETURA

O sistema *In Services* tem por objetivo servir como um sistema para gerenciamento e redefinição de workflows científicos. Como o foco deste trabalho é em workflows compostos por serviços Web, a redefinição permite o acréscimo de serviços responsáveis pelo tratamento de dados gerados durante o fluxo de execução do workflow, dados intermediários. Esses serviços, aqui chamados de serviços de dados, são adicionados ao workflow para realizarem três tipos de tratamentos de dados: filtragem de dados gerados por algum passo do workflow, inserção de dados no banco de dados GUS e recuperação de dados do GUS para um passo do workflow científico. O uso dos serviços de dados é uma proposta para a gerência de dados em workflows científicos compostos por serviços Web.

A característica de workflows científicos baseados em serviços Web direcionou o desenvolvimento do sistema *In Services* para trabalhar com tecnologias que dão suporte a esse tipo de workflow. Por conta disso, a linguagem que descreve os workflows definidos e executados na ferramenta provê o uso de serviços Web como passos do workflow. Além disso, a máquina de execução de workflows acoplada à ferramenta deve ser capaz de interpretar essa linguagem e gerenciar a execução do mesmo, efetuando as devidas invocações aos serviços Web de cada passo, bem como repassar apropriadamente os dados necessários entre esses passos do workflow.

Este capítulo descreve o sistema *In Services*, como o mesmo foi desenvolvido e como executa as funcionalidades para prover o gerenciamento de dados intermediários proposto no trabalho.

4.1 ARQUITETURA DO SISTEMA *IN SERVICES*

A arquitetura do sistema *In Services* para gerência de workflows científicos e uso de serviços de tratamento de dados é mostrada na FIG. 4.1. A arquitetura é subdividida nos seguintes componentes:

- A - Gerência de workflows
  - A.1 - Construção do workflow

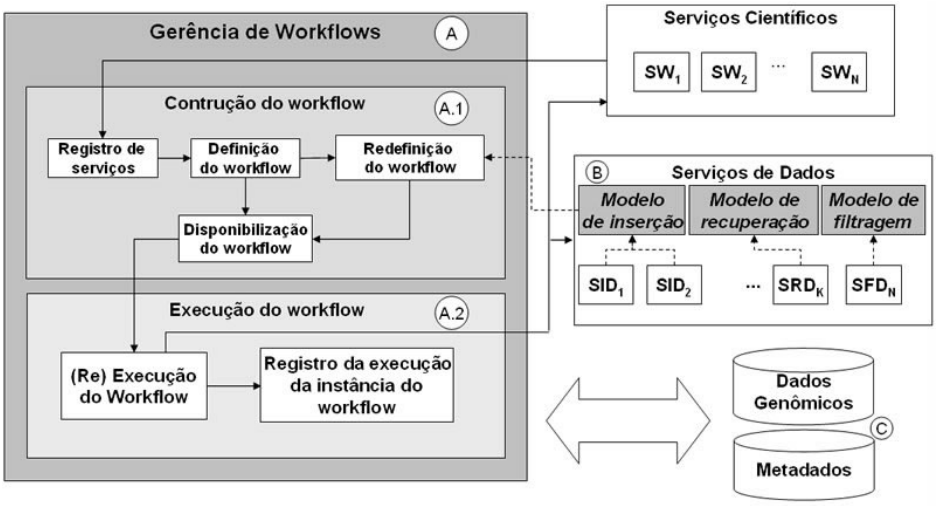

FIG. 4.1: Arquitetura do sistema *In Services*

– A.2 - Execução do workflow

- B - Serviços de dados
- C - Banco de dados e de metadados

O primeiro componente (FIG. 4.1.A) é o de *Gerência de workflows*. Esse componente representa toda a parte do sistema *In Services* responsável pela construção, execução e controle das execuções de workflows. Esse componente é compostos de outros dois componentes: o de *Construção do workflow* (FIG. 4.1.A.1), e de *Execução do workflow* (FIG. 4.1.A.2). O primeiro permite cadastrar serviços Web científicos disponíveis para que possam compor passos de algum workflow científico. Esses serviços são desenvolvidos pela comunidade científica de bioinformática e disponibilizados para uso. Geralmente, encapsulam uma funcionalidade específica de algum programa científico. Alguns laboratórios científicos de bioinformática disponibilizam serviços Web que permitem executar alguns programas relacionados a seus experimentos (LIEFELD et. al., 2005), (STANDLEY, 2005), (FINN et. al., 2006), (ECKART et. al., 2003).

Esse componente ainda é responsável pela definição, redefinição e disponibilização do workflow científico. A definição de workflows é feita a partir dos serviços Web registrados no sistema que podem ser compostos como os passos do workflow em uma determinada seqüência. A partir de workflows já definidos, é possível, através do módulo de *Redefinição do workflow*, redefinir os workflows científicos para que façam uso do serviços de dados.

Essa redefinição objetiva inserir ao workflow os serviços de dados como novos passos do mesmo. Assim, é possível acrescentar os três tipos de tratamento de dados fornecidos pelos serviços de dados propostos neste trabalho: inserção, recuperação e filtragem. Vale ressaltar que a redefinição de workflows é feita apenas para a inclusão de serviços de dados como novos passos de workflows. Não sendo possível redefinir workflows para inclusão de novos serviços Web científicos. Workflows definidos ou redefinidos podem então serem disponibilizados para futuras execuções. A disponibilização do workflow significa passar o arquivo que o descreve - gerado no momento de suas definições - para a máquina de execução de workflows. A máquina de execução então interpreta esse arquivo fazendo as devidas invocações aos serviços Web que compõem seus passos e conseqüentemente executando o workflow científico como um todo.

O componente de *Serviços de Dados* (FIG. 4.1.B) é o que fornece os serviços Web responsáveis pelo tratamento dos dados em um wokflow redefinido. No momento da redefinição do workflow, o usuário pode inserir esses serviços como passos adicionais do workflow para que executem o tratamento dos dados intermediários. Os três tipos de serviços de dados propostos são: serviço de inserção de dados de um passo do workflow para o banco de dados, serviço de recuperação de dados do banco para um passo do workflow, e serviço de filtragem de dados.

Cada serviço de dado irá operar sobre os dados gerados de um passo específico do workflow científico. Para atender a essa característica de serem os serviços de dados capazes de operar sobre os diferentes tipos de dados específicos de cada passo do workflow, foi proposto um modelo de serviço de dados para cada um dos três tipos: *Modelo de inserção, Modelo de recuperação e Modelo de filtragem.*

Cada modelo possui as atividades básicas sobre como realizar cada uma de suas funcionalidades, mas não são operacionais, pois não têm conhecimento prévio de como são estruturados os tipos de dados que irão manipular. Ou seja, o modelo do serviço de inserção de dados contém operações que efetuam as inserções dos dados, gerados num passo do workflow, no banco de dados; mas não tem, em sua implementação, a estrutura dos dados que o permitiria manipulá-los. O modelo do serviço de recuperação de dados contém as operações que recuperam dados do banco e os enviam para um passo do workflow. Porém, assim como o modelo de inserção, o modelo de recuperação não tem, em sua implementação, a estrutura dos dados a serem enviados ao passo seguinte do wokflow e nem quais dados devem ser obtidos do banco. E por fim, o serviço de filtragem de dados

contém operações que o permite comparar valores de dados e assim descartar aqueles que não obedeçam a algum critério de comparação, mas sua implementação depende de como as comparações para filtragem de dados devem ser feitas e quais os dados que devem ser filtrados.

Essas informações ausentes só podem ser obtidas quando o usuário interage com o sistema no momento da redefinição do workflow. Por conta disso, os serviços de dados são gerados somente quando o workflow é redefinido. Na FIG. 4.1.B o componente de serviços de dados possui os três modelos de serviços de dados (Inserção, Recuperação e Filtragem) e a partir desses modelos são gerados os serviços de dados, que na figura estão representados pelos retângulos SID_1 e SID_2 como serviços de inserção de dados, SRD_K para o serviço de recuperação de dados e SFD_N como serviço de filtragem de dados. Em seções posteriores cada um dos serviços de dados será descrito em detalhes.

O próximo componente da arquitetura é o de *Execução de workflows* (FIG. 4.1.A.3). Nesse componente é onde a descrição do workflow definido ou redefinido será interpretada e a execução do workflow efetuada. O mesmo faz uso diretamente de uma máquina de execução de workflows. Seus dois módulos - *(Re)execução do workflow* e *Registro da execução da instância do workflow* - são responsáveis respectivamente por permitir a execução e re-execução do workflow definido e por registrar no banco de metadados os dados referentes à execução do workflow e de seus passos executados.

O último componente da arquitetura (FIG. 4.1.C) é o banco de dados usado pelo sistema para armazenar os dados provenientes das execuções do workflow. A tarefa de armazenamento de dados é garantida pelo uso do serviço de inserção de dados. O uso do banco de dados da arquitetura também possibilita que os dados lá armazenados possam ser obtidos para servir como dado de entrada para algum passo de um workflow a ser executado, visando-se a re-execuções parciais de workflows. A realização da tarefa de recuperação de dados do banco de dados para um passo do workflow é garantida com o uso do serviço de recuperação de dados. Para o ambiente de bioinformática, o esquema de banco de dados genômico usado no sistema *In Services* foi o GUS. A característica genérica desse esquema foi o fator determinante para sua escolha, pois assim pode-se abranger o armazenamento e gerenciamento para um amplo conjunto de dados genômicos.

O banco de metadados é utilizado pela arquitetura do *In Services* para manter dados sobre as instâncias de execução dos workflows e seus passos (serviços Web). Além disso, informações de proveniência sobre os dados armazenados, recuperados ou filtrados pelos

serviços de dados são registradas, procurando-se garantir uma manutenção de informações que possam auxiliar análises científicas como, por exemplo, em quais contextos o workflow foi executado, que dados manipulou e quais serviços foram executados. O esquema que dá suporte ao armazenamento dos metadados do *In Services* foi modelado como uma extensão do esquema GUS.

4.2 PAPÉIS DE USUÁRIOS NO SISTEMA *IN SERVICES*

Dentro da arquitetura do *In Services* existem basicamente 4 papéis de usuários distintos: o usuário que define o workflow científico, o que redefine o workflow, o que executa e o que analisa os dados referentes às execuções dos workflows científicos. Um mesmo usuário pode atuar com todos os papéis, mas cada papel pode ser atribuído para usuários distintos.

O usuário que define o workflow consulta os serviços Web científicos registrados para compor passos dos workflows em definição. Como citado anteriormente, cada serviço é responsável por uma atividade científica específica. Assim, esse tipo de usuário tem o objetivo de realizar um determinado experimento científico e por tanto, sabe compor um conjunto de serviços Web que o auxiliem na realização do seu experimento científico.

O papel do usuário que redefine o workflow, assim como o usuário que definiu o workflow, possui domínio sobre o experimento a ser realizado. Além disso, tem conhecimento sobre os passos que compõem o workflow e, opcionalmente, sobre o banco de dados onde os dados gerados no experimento podem ser armazenados. Esse usuário sente necessidade de alterar o workflow original para tratar os dados intermediários gerados durante sua execução. Para isso, é possível que utilize algumas das três funcionalidades de tratamento de dados fornecidas pelos serviços de dados: Filtragem de dados entre passos do workflow, inserção de dados de um passo do workflow para um banco de dados e recuperação de dados do banco para alimentar determinado passo do workflow. Esse usuário redefine o workflow inserindo os serviços de dados entre os passos do workflow onde julgar necessário que ocorra um dos três tipos de tratamento de dados citados.

O usuário que executa o workflow obtém o mesmo já modelado e solicita sua execução, passando os dados de entrada e obtendo os dados finais da execução do workflow, efetuando as devidas análises sobre o experimento realizado.

O usuário que analisa os dados referentes a execuções do workflow, embora não o faça diretamente pelo sistema *In Services*, analisa os dados gerados pelo sistema. Esse usuário obtém, via acesso ao banco de dados e de metadados, as informações sobre os

workflows científicos executados, os serviços Web invocados no workflow, e sobre os dados manipulados durante a execução do workflow. Com base nessas informações esse usuário pode validar ou invalidar determinadas execuções do experimento, analisar os resultados e inferir informações.

4.3 CENÁRIOS DE USO

Agora que a arquitetura do *In Services* e os papéis dos usuários estão descritos, podemos citar alguns cenários de uso do sistema. O cenário de modelagem de um workflow não será descrito, pois já existem diversas ferramentas que realizam essa atividade. Por conta disso, este trabalho parte do pressuposto que o workflow já foi modelado, sendo apenas necessário registrá-lo no *In Services*. Esta seção irá descrever quatro cenários distintos, o primeiro deles descreve como ocorre a redefinição de um workflow para fazer uso de serviços de filtragem e inserção de dados. O segundo cenário descreve a execução de um workflow redefinido com os serviços adicionados no primeiro cenário. O terceiro cenário descreve a redefinição de um workflow para fazer uso de um serviço de recuperação de dados e o quarto cenário descreve a execução desse workflow.

4.3.1 PRIMEIRO CENÁRIO: REDEFINIÇÃO DO WORKFLOW PARA USO DE SERVIÇOS DE INSERÇÃO E FILTRAGEM DE DADOS

Neste primeiro cenário, o usuário de posse de um workflow já modelado, obtém o arquivo que o descreve e no módulo de redefinição de workflows adiciona o(s) serviço(s) de dados que deseja - FIG. 4.2.a e FIG. 4.2.b. Durante a redefinição, o usuário especifica em qual passo do workflow pretende utilizar um serviço de dados. Esse poderá ser um serviço de inserção, recuperação ou filtragem de dados. Os serviços de dados que devem atender aos passos especificados pelo usuário são gerados a partir dos modelos de serviços de dados.

No cenário da FIG. 4.2, foi utilizado um serviço de inserção de dados SID_1 e um serviço de filtragem de dados SFD_N. Dessa forma, o usuário indicou em qual passo do workflow pretende ter seus dados armazenados bem como quais dados resultantes desse passo devem ser armazenados. Além disso, indica o serviço Web que compõe o passo onde pretende aplicar um filtro sobre os dados gerados na execução desse passo e quais dados resultantes desse serviço devem ser filtrados.

A redefinição do workflow dá origem a um novo - FIG. 4.2.c. Esse novo workflow pode

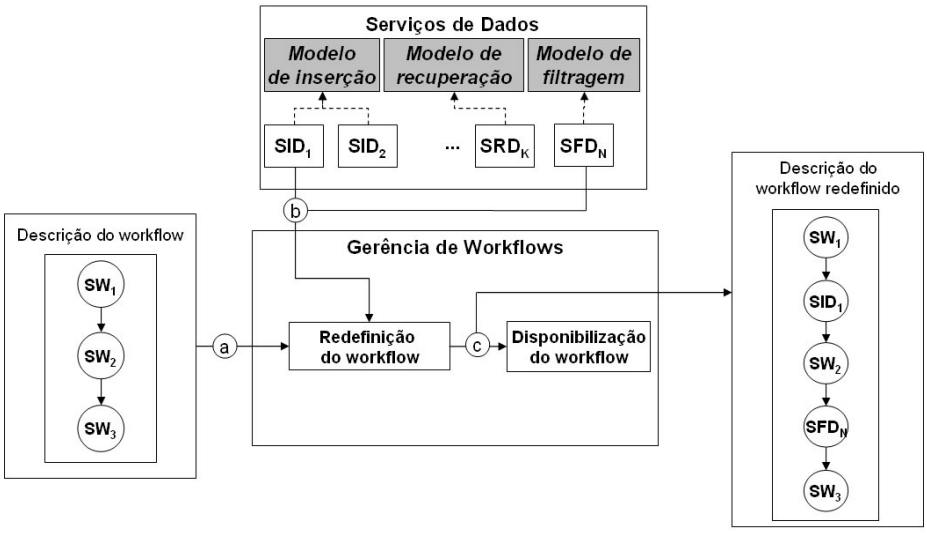

FIG. 4.2: Cenário de uso do sistema *In Services* para redefinição de workflows para uso de serviço de inserção e filtragem de dados.

então ser submetido ao processo de disponibilização que irá torná-lo apto para posterior execução.

4.3.2 SEGUNDO CENÁRIO: EXECUÇÃO DO WORKFLOW REDEFINIDO NO PRIMEIRO CENÁRIO

Nesse segundo cenário, o usuário de posse do workflow redefinido submete sua execução ao módulo de execução de workflows - FIG. 4.3.a. A máquina de execução de workflows efetua as invocações dos serviços Web que compõem os passos do workflow. Os serviços de dados que também fazem parte do workflow são chamados durante a execução - FIG. 4.3.b.

O serviço de inserção realiza o armazenamento dos dados que provieram do passo anterior a ele - FIG. 4.3.c. O serviço de filtragem também obtém os dados do passo anterior a ele e descarta aqueles que não obedecem a algum parâmetro de filtragem.

Após a execução, o sistema *In Services* utiliza o log gerado pela máquina de execução de workflows e efetua o registro referente à instância de execução do workflow. Soma-se a isso, o registro da execução de cada passo do workflow, incluindo os serviços de dados - FIG. 4.3.d e FIG. 4.3.e.

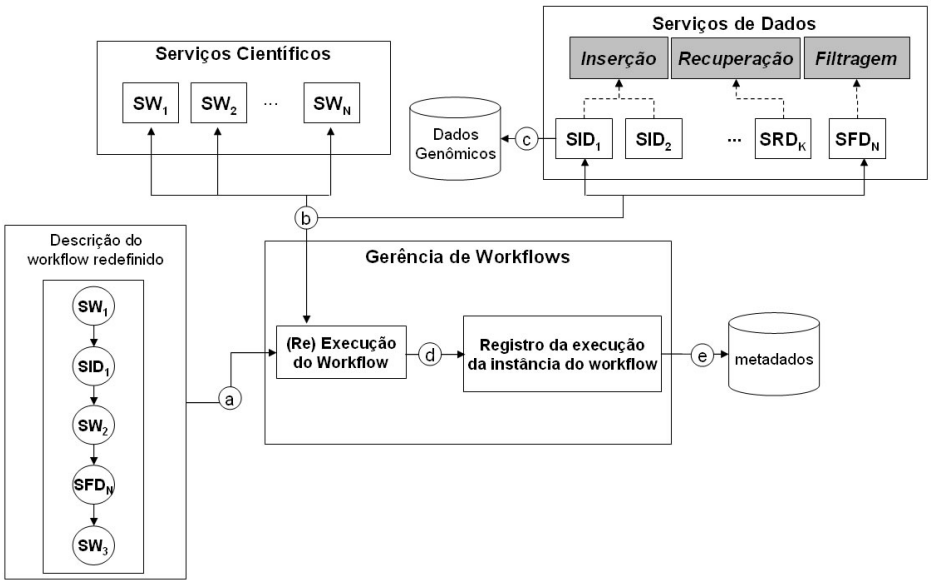

FIG. 4.3: Cenário de uso do sistema *In Services* para execução de workflow redefinido.

4.3.3 TERCEIRO CENÁRIO: REDEFINIÇÃO DE WORKFLOW PARA USO DE SERVIÇO DE RECUPERAÇÃO DE DADOS

Este cenário possui os procedimentos de redefinição de workflows similares aos que ocorrem no primeiro cenário, porém, neste, um workflow é redefinido para fazer uso de um serviço de recuperação de dados. O usuário de posse do arquivo que descreve um workflow já modelado efetua sua redefinição no intuito de usar um serviço de recuperação. Assim como no primeiro cenário o usuário deve especificar de qual passo do workflow pretende-se recuperar dados previamente armazenados - FIG. 4.4.a.

Nesse mesmo momento, o usuário especifica quais dados para aquele passo devem ser recuperados. Isso é feito a partir dos registros de execuções prévias desse workflow que tiveram armazenamento de dados. Os dados armazenados nas execuções anteriores serão escolhidos para alimentar o passo do workflow. Nesse exemplo, o serviço de recuperação de dados SRD_K é utilizado para compor um novo passo do workflow - FIG. 4.4.b - e os dados recuperados serão utilizados pelo serviço Web que compõe o segundo passo do workflow. Após a redefinição do workflow com a inclusão do serviço de recuperação de dados, o workflow remodelado é então disponibilizado para posterior execução - FIG. 4.4.c.

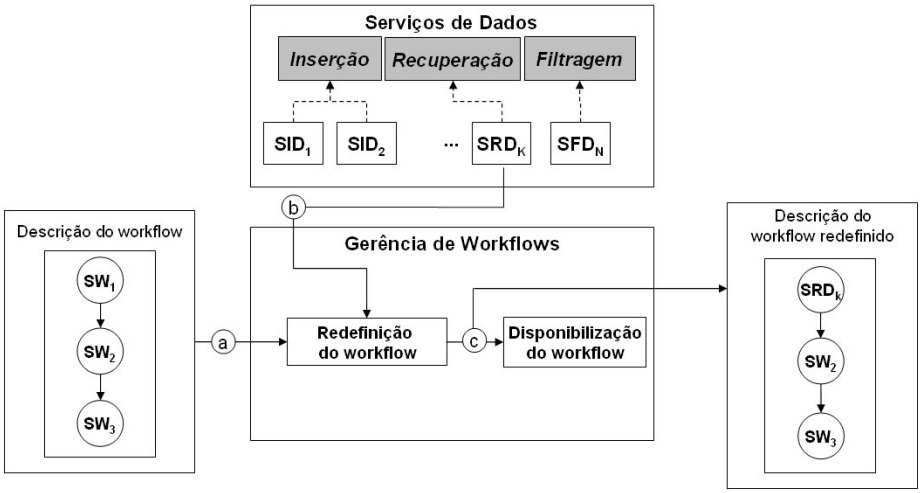

FIG. 4.4: Cenário de uso do sistema *In Services* para redefinição de workflows para uso de serviço de recuperação de dados.

4.3.4 QUARTO CENÁRIO: EXECUÇÃO DO WORKFLOW REDEFINIDO NO TERCEIRO CENÁRIO

No quarto cenário, o usuário de posse do workflow redefinido dá início à execução do mesmo - FIG. 4.5.a. Neste caso tem-se um workflow redefinido com um serviço de recuperação de dados. O início da execução do workflow se dá com a execução do serviço de recuperação, o qual obtém os dados a partir do banco de dados.

Após a execução do serviço de recuperação de dados e obtenção dos dados do banco, os mesmos são repassados ao passo seguinte do fluxo de execuções do workflow - FIG. 4.5.b. Tomando-se proveito de dados armazenados no banco de dados, pode-se executar parcialmente um workflow, não havendo a necessidade de que certos passos do mesmo sejam executados.

As informações sobre a execução do workflow e de seus serviços são então registradas no banco de metadados - FIG. 4.5.c e FIG. 4.5.d.

4.4 MODELO DE DADOS

Como mencionado anteriormente, o banco de dados em uso no *In Services* utiliza o esquema de dados do GUS. O modelo de dados desse esquema permite utilizá-lo para

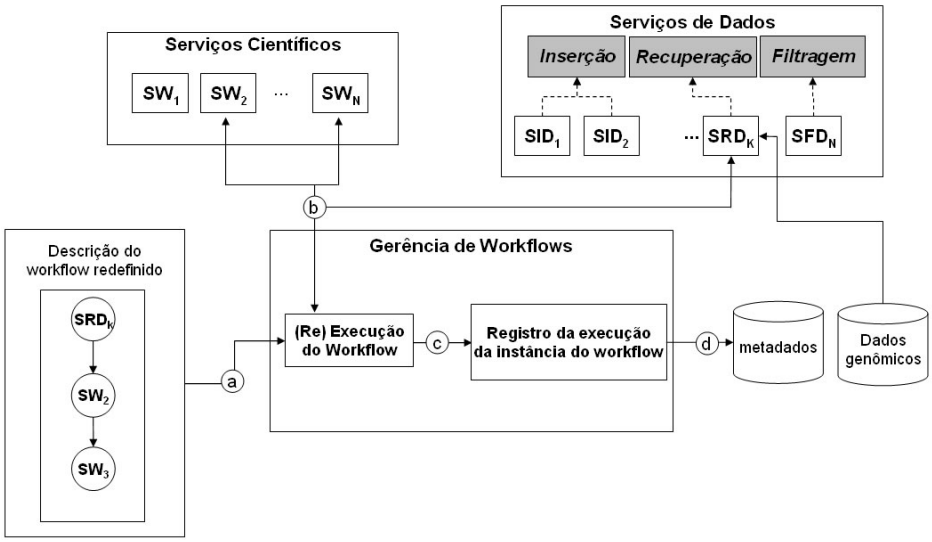

FIG. 4.5: Cenário de uso do sistema *In Services* para execução de workflow redefinido com serviço de recuperação de dados.

armazenar os dados genômicos dos workflows científicos executados no *In Services*. Tais dados são inseridos no banco através do serviço de inserção de dados. Pode-se utilizar o serviço de recuperação de dados para obter os dados armazenados e então utilizá-los para alimentar algum passo do workflow.

Entretanto, para o sistema *In Services*, foi proposta uma extensão do esquema GUS (GUS+) para dar suporte ao armazenamento dos dados referentes aos workflows definidos, redefinidos e executados no sistema. Além dessas informações, a extensão do esquema permite manter dados sobre a definição e execução de cada passo do workflow, dos serviços Web que compõem esses passos, e dos serviços de dados utilizados em workflows redefinidos.

Para os serviços de dados armazenados, são mantidas as informações sobre as atividades executadas por eles. Ou seja, através do esquema do GUS+ podem-se obter informações sobre quais dados foram inseridos pelos serviços de inserção de dados, quais os que foram consultados pelos serviços de recuperação de dados e como os serviços de filtragem de dados foram definidos. Essas informações revelam o comportamento das instâncias de execução de cada workflow executado no *In Services*. Comportamento esse, referente a quando e por qual usuário os workflows foram submetidos à execução, o estado final da execução, quais passos fizeram parte da execução do workflow, quais dados foram

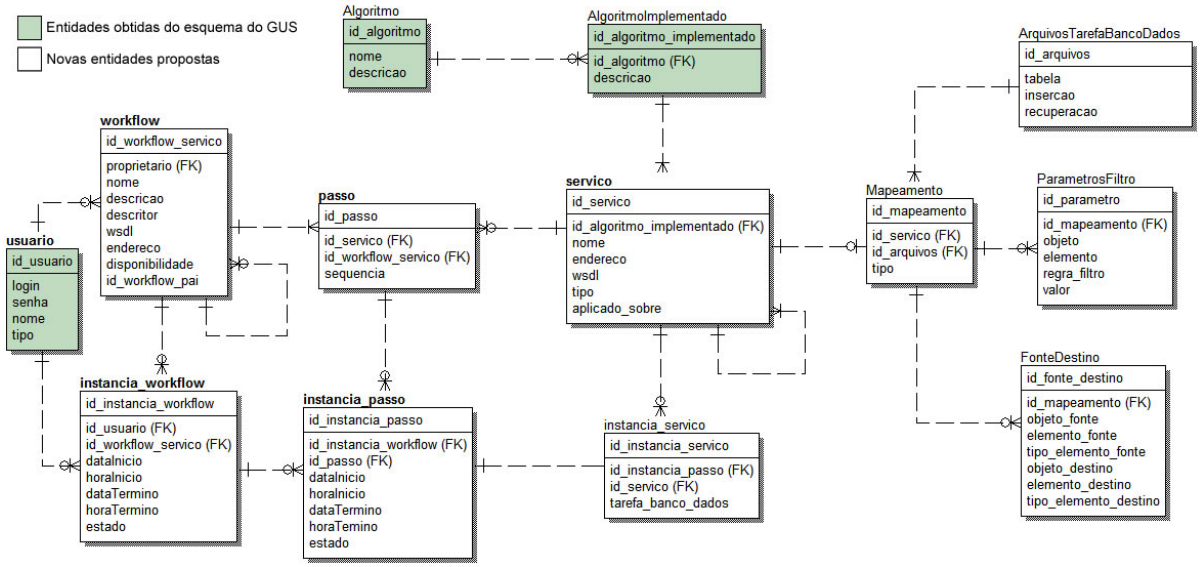

FIG. 4.6: Modelo de dados do esquema estendido do GUS.

manipulados por esses passos e como ocorreu o uso dos serviços do dados dentro dos workflows redefinidos. Emfim, é possível manter-se um registro sobre a proveniência dos dados referentes aos workflows científicos redefinidos e executados no *In Services*. A FIG. 4.6 contém o modelo de dados do esquema GUS+ e uma explicação sobre esse modelo é dada em seguida.

Das entidades do esquema, três são diretamente obtidas do esquema do GUS. São elas: *Algoritmo, AlgoritmoImplementado e Usuario*. As demais são propostas como parte da extensão para a gerência de workflows científicos compostos por serviços Web, dando suporte às funcionalidades do sistema *In Services*. Apesar de algumas entidades terem sido herdadas diretamente do esquema original do GUS, alguns de seus atributos foram dispensados. Um exemplo disso foi feito na entidade *AlgoritmoImplementado*. Nessa entidade os atributos que continham dados referentes aos executáveis dos algoritmos foram dispensados para serem melhor representados pela entidade *servico*.

A entidade *Algoritmo* representa uma descrição abstrata dos algoritmos disponíveis para realizar alguma atividade que pode compor um passo do workflow. As implementações dos algoritmos têm suas informações representadas pela entidade *AlgoritmoImplementado*. A entidade *servico* mantém as informações técnicas sobre os serviços Web

registrados no sistema e que disponibilizam a execução da implementação dos algoritmos. Nela, há o atributo *tipo* cujos valores especificam se o serviço Web é científico ou é um serviço de dados. Um serviço Web do tipo científico, é um serviço Web que provê uma interface de acesso à execução de algum algoritmo científico. Caso seja um serviço de dados, ocorrerá um auto-relacionamento na entidade *servico* via o atributo *aplicado_sobre* indicando que aquele serviço foi aplicado para tratamento de dados de outros serviços ao qual se relaciona.

A entidade *passo* descreve os passos que compõem um workflow. Cada passo do workflow corresponde um serviço Web, seja ele científico ou um serviço de dados. Por isso, há o relacionamento da entidade *passo* com a entidade *servico*. O atributo *sequencia* especifica qual é a posição daquele passo com relação a todos os outros passos que compõem o workflow.

A entidade *workflow* mantém os dados dos workflows registrados no sistema. Um workflow deve ser registrado com um nome que o caracterize e uma descrição textual da atividade que realiza. Além disso, atributos referentes a workflows compostos por serviços Web devem ser fornecidos. Por exemplo, o arquivo que descreve a definição do workflow, o arquivo WSDL desse workflow e um endereço onde o workflow pode ser invocado. Como os workflows registrados serão submetidos à execução pelo módulo de execução de workflows do *In Services*, o atributo *endereco* que representa onde o workflow pode ser executado, será atribuído pelo próprio sistema no momento em que ocorrer a disponibilização do workflow na máquina de execução de workflows. Na entidade *workflow*, o atributo *proprietario* mantém o relacionamento dessa entidade com o usuário que o registrou cujos dados são mantidos na entidade *usuario*. O atributo *disponibilidade* indica se o workflow é público ou privado. No primeiro caso, qualquer usuário do sistema pode acessá-lo, redefini-lo para uso dos serviços de dados e executá-lo; já o workflow com disponibilidade privada somente estará disponível para o usuário que o registrou. A atividade de redefinição de workflows origina novos workflows que incluem, além dos passos dos serviços Web científicos, aqueles que compõem os passos dos serviços de dados adicionados. O esquema GUS+ mantém, pelo atributo *id_workflow_pai*, o relacionamento entre um workflow e aqueles que foram redefinidos a partir dele.

As entidades *instancia_workflow, instancia_passo* e *instancia_servico* representam respectivamente o registro de quando ocorreu a execução do workflow e dos serviços Web a que correspondem seus passos. O atributo *estado* indica o estado dessas execuções,

dado importante para se ter conhecimento se o workflow foi concluído com êxito ou se ocorreram falhas que impediram o término de sua execução.

As entidades *Mapeamento, FonteDestino, ParametrosFiltro* e *ArquivosTarefaBancoDados* foram modeladas para manter dados referentes aos serviços de dados. Sempre que um serviço de dados é utilizado em algum workflow, ocorrerá um dos três tipos de mapeamento: inserção, recuperação ou filtragem.

Um mapeamento de inserção especifica quais dados de saída de um passo do workflow devem ser armazenados em quais tabelas do banco de dados e em quais atributos dessas tabelas. Esse tipo de mapeamento é definido pelo usuário no momento da redefinição de um workflow que fará uso de um serviço de inserção de dados. Um mapeamento de recuperação de dados já descreve o contrário, quais dados de atributos das tabelas deverão ser mapeados para entradas de um passo do workflow. Esse mapeamento é definido automaticamente pelo sistema quando o workflow é redefinido para usar um serviço de recuperação de dados previamente armazenados por um serviço de inserção. A entidade *mapeamento* indicará se o mapeamento especificado é do tipo inserção, recuperação ou filtragem. Para mapeamentos de inserção e recuperação, essa entidade se relaciona com *ArquivosTarefasBancoDados* e *FonteDestino*. A entidade *FonteDestino* especifica o mapeamento dos dados de origem para os dados destinos. Para um relacionamento de inserção de dados, os dados de origem serão aqueles dados de saída de um passo do workflow e o destino serão os atributos das tabelas que irão armazenar aqueles dados. Em um relacionamento de recuperação de dados, os dados de origem serão aqueles armazenados no banco e o destino serão os elementos do tipo de dado de entrada do passo do workflow. A entidade *ArquivosTarefasBancoDados* manterá os arquivos que contêm os comandos necessários para a realização das atividades de inserção e recuperação de cada uma das tabelas do banco de dados a ser utilizado - GUS no caso. Esses arquivos são aqui chamados de *Arquivos de Tarefas de Banco de Dados*. Para cada tabela do esquema de banco de dados em uso, é pré-definido um arquivo com os comandos de carga e um com os comandos de recuperação de dados para a mesma. Esses arquivos serão utilizados pelo serviço de inserção de dados quando, de posse dos dados, a inserção for feita e também serão utilizados pelo serviço de recuperação de dados para a obtenção dos dados armazenados no banco. Neste trabalho não foi considerado inserções e recuperações de dados que envolvam mais de uma tabela simultaneamente, isso necessitaria analisar adequadamente as junções envolvidas para que fossem respeitadas as regras de integridade dos dados. Durante a execução desses serviços

TAB. 4.1: Representação de Arquivo de Tarefa de Banco de Dados.

```
<sql[Update | Query]Statement name="nome_da_tabela">
        <expression>
        EXPRESSAO SQL PARA INSERÇAO | ATUALIZAÇAO | RECUPERAÇAO DE DADOS
        </expression>
</sqlUpdateStatement>
```

de dados, esses arquivos são interpretados para que ocorram as inserções ou recuperações de dados. Os Arquivos de Tarefas de Banco de Dados são arquivos XML que possuem em sua estrutura, comandos SQL para inserção e recuperação de dados nas tabelas do banco de dados em uso. O uso desses arquivos nos serviços de inserção e recuperação de dados foi baseado na tecnologia OGSA-DAI (ANJOMSHOAA et. al.) para acesso a fontes de dados. Uma representação de um Arquivo de Tarefa de Banco de Dados é dada na TAB. 4.1.

Um mapeamento de filtragem utiliza apenas as entidades *Mapeamento* e *Parametros-Filtro*. A primeira especifica o tipo de mapeamento como do tipo filtragem. Já a entidade *ParametrosFiltro* especifica os parâmetros de filtragem que foram definidos no momento da inserção do serviço de filtragem de dados no workflow. Esses parâmetros especificam os valores aos quais os dados de saída de um passo do workflow devem atender para serem considerados válidos. Aqueles dados cujos valores divergirem daqueles especificados nos parâmetros de filtragem devem ser descartados no momento que o serviço de filtragem de dados for executado.

4.5 CONSIDERAÇÕES FINAIS

A proposta do sistema *In Services* apresentada neste trabalho é fornecer uma iniciativa para gerenciamento de dados intermediários em workflows científicos na bioinformática compostos por serviços Web. A arquitetura apresentada visa fornecer um meio para a redefinição de workflows científicos no intuito de serem adicionados serviços Web especiais para tratamento de dados - serviços de dados. Esses serviços, quando inseridos nos workflows científicos, permitem que os dados gerados nos passos dos workflows possam ser filtrados, inseridos em um banco de dados ou recuperados do mesmo. Além disso, a arquitetura provê uma estrutura que registra a definição e execução dos workflows além dos dados manipulados durante suas execuções. Tal fato possibilita o acompanhamento das situações em que os workflows científicos foram executados e quais dados foram ma-

nipulados. O registro dessas informações confere análises mais detalhadas nos resultados de experimentos realizados pelos workflows científicos executados no sistema *In Services*.

5 SISTEMA *IN SERVICES* - DETALHAMENTO

Este capítulo apresenta um estudo de caso sobre a redefinição de um workflow científico no sistema *In Services*. O objetivo desse estudo de caso é detalhar o funcionamento do módulo de redefinição de workflows através de uma descrição passo a passo do uso de cada tipo de serviço de dados.

O workflow GARSA (DÁVILA et. al., 2005) é utilizado para anotação de alguns organismos em estudo pelo projeto BioWebDB (BIOWEBDB) como por exemplo o *Trypanosoma vivax, Trypanosoma rageli* entre outros, realização do experimento de anotação de seqüências. Esse workflow é composto por uma série de programas científicos que realizam uma atividade específica e consiste basicamente em obter um conjunto de arquivos (cromatogramas) a partir de um seqüenciador genômico. Desses arquivos, com o uso do programa Phred (EWING et. al., 1998), são extraídas as seqüências genômicas. Além disso, ocorre o levantamento da qualidade dessas seqüências e dependendo desses valores de qualidade, o próprio programa pode descartar aquelas que não fornecem dados confiáveis. O segundo passo do workflow, faz uso do programa Cap3 (WANG, 2004) para montar as seqüências geradas pelo passo anterior em um ou mais conjuntos de segmentos de seqüências genômicas sobrepostas - tarefa essa conhecida como clusterização de seqüências. Em seguida, esse conjunto de seqüências agrupadas é submetido a uma busca de similaridade com outras seqüências genômicas já conhecidas e armazenadas. Esse passo é executado no workflow GARSA pelos programas Blast (**?**) e Interpro (MULDER et. al., 2005). A partir dos resultados da busca de similaridades executa-se o programa Phylip (FELSENSTEIN) para a realização do passo de análise filogenética. Após a execução dessa seqüência de passos, obtém-se o resultado de anotação de seqüências do workflow. O workflow GARSA é definido através da linguagem *Perl* que contém as invocações dos programas utilizados. A FIG. 5.1 ilustra em detalhes como o workflow GARSA está definido. É intenção do projeto BioWebDB, onde esse workflow é utilizado, migrar gradualmente os passos do workflow para a plataforma de serviços Web. Este trabalho faz parte dos esforços neste sentido.

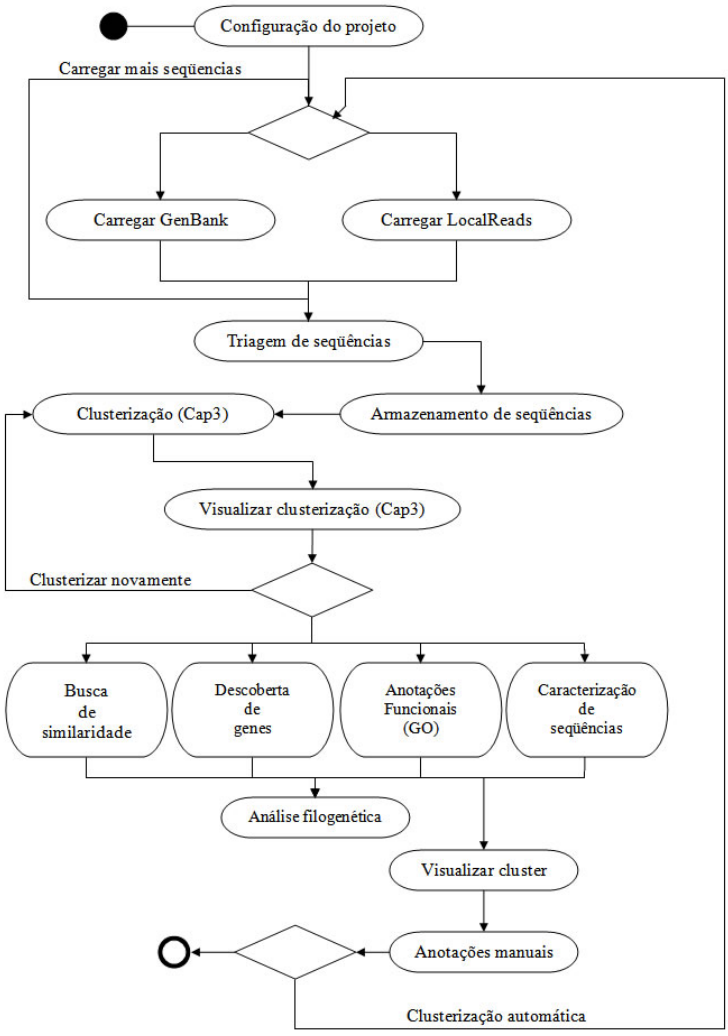

FIG. 5.1: Workflow GARSA (GARSA).

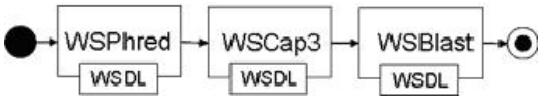

FIG. 5.2: Workflow SUBGARSA.

5.1 WORKFLOW SUBGARSA

No estudo de caso apresentado neste capítulo, o workflow utilizado constitui-se dos passos iniciais do workflow GARSA. Esse workflow foi aqui denominado de workflow SUBGARSA e utiliza os três primeiros passos definidos no workflow GARSA. Ou seja, o workflow é constituído pela seqüência de programas: Phred, Cap3 e Blast. Porém, no workflow SUBGARSA, esses três programas foram dispostos como serviços Web e o workflow foi definido através da linguagem BPEL4WS. Apesar de já existirem versões de serviços Web para esses três programas, os mesmos tiveram que ser implementados para atender a alguns requisitos. O primeiro deles foi tornar as definições de dados explícitas nos serviços Web. A definição explícita dos tipos de dados dos programas possibilita que pela descrição do serviço (WSDL) seja possível identificar o formato do tipo de dado manipulado. Característica essa que geralmente não é disponibilizada por esses tipos de serviços Web, mas sim apenas um tipo de dado genérico, como texto, que é utilizado na troca de mensagens entre o serviço e o cliente. Essa característica esconde os detalhes de formatação de dados, impedindo que a interpretação e manipulação dos mesmos seja facilitada. Outro requisito, foi atender à compatibilidade de tipos de dados entre os serviços Web do workflow. Ou seja, para os dados de saída do Phred poderem ser utilizados como dados de entrada no serviço do Cap3, e os dados de saída do serviço do Cap3 serem utilizados como dados de entrada do serviço do Blast houve a necessidade de torná-los compatíveis entre si. A FIG. 5.2 mostra como o workflow SUBGARSA foi estruturado com os três serviços Web que o compõem.

De maneira geral, no sistema *In Services* a inserção dos serviços de dados é feita indicando-se em qual passo do workflow pretende-se adicionar um desses serviços. No caso dos serviços de filtragem e inserção de dados, os mesmos serão adicionados após um serviço Web (passo) escolhido no workflow e receberão os dados gerados na execução desse serviço. Assim, nas seções seguintes, será detalhado como esses tipos de serviços de dados

são inseridos no workflow SUBGARSA. O serviço de recuperação de dados é adicionado escolhendo-se um passo do workflow cujos dados já foram previamente armazenados por um serviço de inserção de dados. O detalhamento de seu uso no workflow SUBGARSA é apresentado mais adiante.

5.2 INSERINDO SERVIÇO DE FILTRAGEM DE DADOS PARA O BLAST

O serviço de filtragem de dados é definido a partir do modelo de serviço de filtragem de dados disponibilizado no *In Services* e apresentado no capítulo anterior. O usuário, no momento de redefinição do workflow, indica em qual passo pretende inserir um serviço de filtragem de dados. O módulo de redefinição de workflows gera então o serviço de filtragem depois de obter os tipos de dados que esse deve manipular. Além disso, o usuário especifica quais são os parâmetros de filtragem a serem aplicados pelo serviço de dados no momento em que for executado.

Esse serviço de dados será aplicado em uma redefinição do workflow SUBGARSA. Nele, pretende-se realizar uma filtragem dos dados gerados pelo passo que executa o serviço Web do Blast. Assim, no módulo de redefinição de workflows, o usuário indica que pretende inserir o serviço de filtragem de dados sobre a saída do terceiro passo do workkflow - serviço Web do Blast (WSBlast).

Depois disso, o módulo de redefinição do *In Services* efetua a leitura do WSDL do serviço do Blast para obter seu formato de dados de saída. Na definição do serviço de filtragem de dados essa leitura tem duas finalidades. A primeira delas é para criar-se o serviço de filtragem de dados que manipule os mesmos tipos de dados do serviço Web (WSBlast, no caso). Isso é feito criando-se o serviço de filtragem a partir do modelo de serviço de filtragem de dados e dos tipos de dados que devem ser manipulados pelo serviço. A segunda finalidade da leitura é montar a interface onde os usuários irão definir os parâmetros de filtragem a serem adotados no momento da execução do serviço de filtragem de dados. Esses parâmetros farão parte do código do serviço de filtragem e serão consultados quando o mesmo for executado para a realização da filtragem. A FIG. 5.3 ilustra como ocorre a definição do serviço de filtragem de dados para o serviço Web Blast do workflow SUBGARSA.

Na FIG. 5.3, a elipse 1 representa o processo de obtenção dos tipos de dados que estão definidos no serviço Web que disponibiliza a execução do programa Blast. Essa obtenção é feita a partir de uma leitura no WSDL do serviço pelo componente de redefinição de

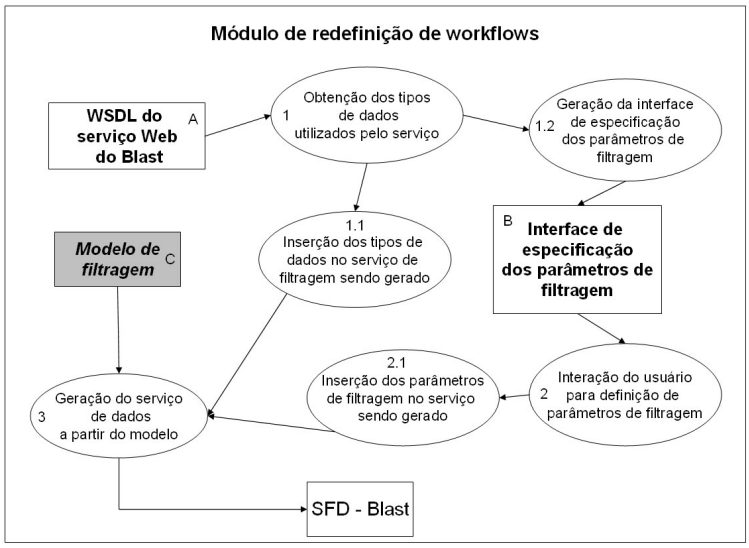

FIG. 5.3: Definição do serviço de filtragem de dados para o serviço Web do Blast no workflow SUBGARSA.

workflows do *In Services*. A partir desse processo, outros dois são iniciados. Um para geração da interface onde o usuário irá definir os parâmetros de filtragem para os dados de saída do serviço Web do Blast e outro que insere os tipos de dados no serviço de filtragem de dados que está sendo gerado - elipses 1.1 e 1.2. Através da interface de definição de parâmetros de filtragem, o usuário irá especificar as regras que o serviço de filtragem irá utilizar para realizar o filtro de dados - elipse 2. Esses parâmetros, após serem definidos, também irão fazer parte do código do serviço de filtragem de dados - elipse 2.1. Como exemplo, temos o código XML contido na TAB. 5.1 como parte do WSDL do serviço Web do Blast cujos dados de saída serão processados pelo serviço de filtragem de dados sendo definido e inserido no workflow. No momento em que ocorre a leitura desse WSDL, o tipo de dado de saída do serviço é localizado (TAB. 5.1.C). A descrição do tipo de saída está localizada em outra parte do mesmo WSDL (TAB. 5.1.A). Nesse exemplo tem-se que o tipo de saída é uma seqüência de itens (Array) do tipo *BlastResult* e a estrutura desse tipo de dado é definida na TAB. 5.1.B.

Em resumo, os dados da TAB. 5.1.A, TAB. 5.1.B e TAB. 5.1.C especificam que o serviço Web do Blast produz um tipo de dado que é uma seqüência de itens de dados, onde cada item dessa seqüência é um elemento *BlastResult*. A estrutura do elemento *BlastResult* é descrita no mesmo documento, TAB. 5.1.B. A partir dessa estrutura o

TAB. 5.1: Parte do WSDL do serviço Web do programa Blast.

```
<definitions>
    <types>
       <xsd:schema>
            ...
<!—A -->
          <complexType name="ArrayOfBlastResult">
           <complexContent>
                <restriction base="soapenc:Array">
                <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:BlastResult[]"/>
                </restriction>
           </complexContent>
          </complexType>
<!-- B -->
          <complexType name="BlastResult">
            <sequence>
                <element name="EValue"          type="xsd:double"/>
                <element name="QEnd"            type="xsd:int"/>
                <element name="QStart"          type="xsd:int"/>
                <element name="SEnd"            type="xsd:int"/>
                <element name="SStart"          type="xsd:int"/>
                <element name="alignmentLength" type="xsd:float"/>
                <element name="bitScore"        type="xsd:float"/>
                <element name="gapOpenings"     type="xsd:int"/>
                <element name="identity"        type="xsd:float"/>
                <element name="mismatches"      type="xsd:int"/>
                <element name="queryId"         nillable="true" type="soapenc:string"/>
                <element name="subjectId"       nillable="true" type="soapenc:string"/>
            </sequence>
           </complexType>
        </xsd:schema>
    </types>
...
<!-- C -->
<message name="runBlastNResponse">
    <part name="runBlastNReturn" type="impl:ArrayOfBlastResult"/>
</message>

<portType name="WSBlast">
    <operation name="runBlastN" parameterOrder="in0">
      <input message="impl:runBlastNRequest" name="runBlastNRequest"/>
      <output message="impl:runBlastNResponse" name="runBlastNResponse"/>
    </operation>
</portType>
...
</definitions>
```

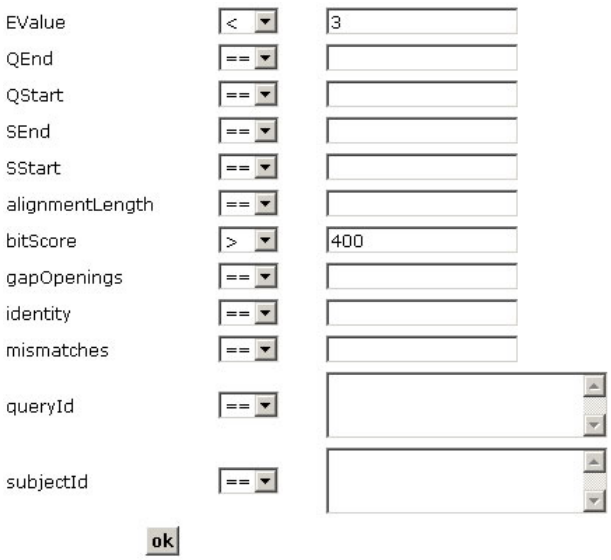

FIG. 5.4: Interface para definição dos parâmetros de filtragem para o serviço Web do Blast.

módulo de redefinição monta a interface para a especificação dos parâmetros de filtragem. A FIG. 5.4 mostra a interface gerada pelo *In Services* a partir do WSDL da TAB. 5.1 para que o usuário insira os parâmetros de filtragem a serem aplicados nos dados de saída do serviço Web do Blast. Por exemplo, no formulário da FIG. 5.4 o usuário define que somente os resultados do Blast com o campo *bitScore* acima de 400 e o campo *EValue* menor que 3 são relevantes para o experimento. Esses parâmetros de filtragem farão parte do código do serviço de filtragem de dados que está sendo definido. Assim, no momento em que o serviço de filtragem de dados for executado, será feito uma iteração sobre todos os itens *BlastResult* da seqüência resultante do serviço Web do Blast. Aqueles itens que não possuírem os elementos *bitScore* e *EValue* de acordo com os parâmetros que o usuário definiu serão descartados.

Após o serviço de filtragem está completamente definido, ocorre o processo de disponibilização do serviço. Isso permitirá que o serviço de filtragem seja utilizado na execução do workflow. Para isso ocorrer, um arquivo que descreve o workflow é gerado contendo o serviço de filtragem de dados após o passo que é composto pelo serviço Web do Blast. O módulo de redefinição de workflows efetua a geração automática desse arquivo. O workflow resultante é mostrado na FIG. 5.5 contendo o serviço de filtragem de dados.

Quando um workflow é redefinido no sistema *In Services* com serviços de filtragem

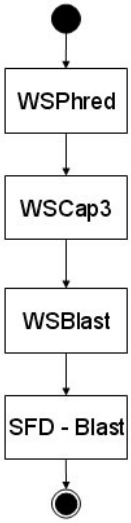

FIG. 5.5: Workflow SUBGARSA redefinido com serviço de filtragem de dados para o Blast - SFD-Blast.

de dados, informações referentes a essa redefinição são mantidas no banco de dados do sistema. Isso é relevante para se ter conhecimento sobre como os serviços de filtragem de dados foram aplicados nos workflows científicos.

5.3 INSERINDO SERVIÇO DE INSERÇÃO DE DADOS PARA O PHRED

O serviço para inserção de dados gerados na execução dos workflows científicos em um esquema de banco de dados é definido a partir de um modelo de serviço de inserção de dados e através de interações do usuário com o sistema *In Services*. O modelo de serviço de inserção de dados, como dito no capítulo anterior, contém os passos básicos para a realização das atividades de inserção. Porém, as mesmas só podem ser de fato executadas após o usuário indicar quais dados gerados de um serviço Web (passo) do workflow deverão ser armazenados. É nesse momento que o usuário deve efetuar a interação.

Assim como na definição do serviço de filtragem de dados, o módulo de redefinição de workflows obtém os tipos de dados manipulados pelo serviço Web que o usuário indicou e os enxerta numa nova instância do modelo do serviço de inserção de dados. Isso permite que o serviço de inserção sendo gerado, manipule o mesmo formato de dados do serviço Web.

Outra informação necessária ao serviço de inserção de dados é o mapeamento de quais dados serão armazenados no esquema de banco de dados. É necessário então que o usuário

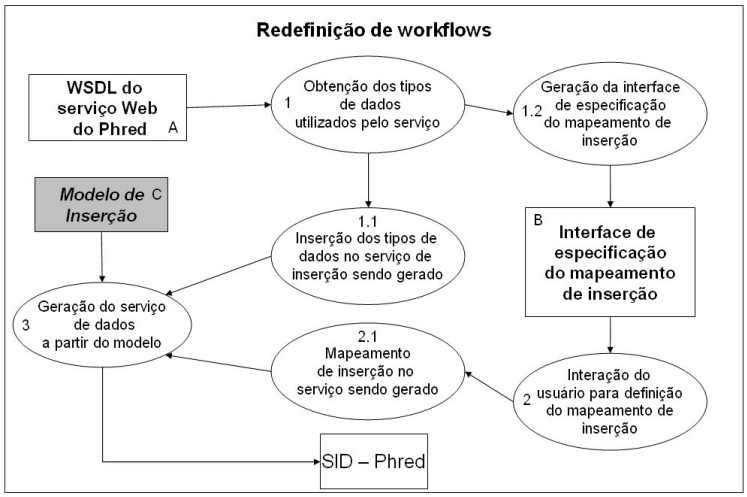

FIG. 5.6: Geração do serviço de inserção de dados para o serviço Web do Phred no workflow SUBGARSA.

também mapeie os dados de saída do serviço Web para os atributos das tabelas onde se pretende armazenar. Para isso, após ser selecionado o passo do workflow ao qual deve ter seus dados armazenados, o usuário é apresentado ao conjunto de tabelas que formam o banco de dados em uso, GUS no caso. Desse conjunto, o usuário deve selecionar aquelas tabelas que serão utilizadas para o armazenamento dos dados. Feito isso, de maneira similar ao que ocorre na definição do serviço de filtragem de dados, o usuário especifica o mapeamento de quais dados devem ser armazenados. As informações sobre o mapeamento de inserção permitirão ao serviço de inserção de dados efetuar o armazenamento somente daqueles dados de interesse do usuário.

As definições do mapeamento de dados da saída do serviço para o esquema de banco de dados juntamente com os tipos de dados do serviço Web permitem então que o serviço de inserção seja criado. A FIG. 5.6 ilustra as atividades executadas pelo módulo de redefinição de workflows quando a definição de um serviço de inserção de dados ocorre para que sejam armazenados os dados de saída do serviço Web do Phred no workflow SUBGARSA. Assim, no sistema *In Services*, o usuário deve indicar que nesse passo pretende-se realizar a inserção de dados no banco.

Na FIG. 5.6, as atividades para a geração de um serviço de inserção de dados para o serviço Web do Phred no workflow SUBGARSA são representadas pelas elipses. Inicial-

mente, a partir do WSDL do serviço do Phred, o componente de redefinição de workflows do sistema *In Services* obtém a descrição dos tipos de dados daquele serviço - elipse 1. Essa informação é inserida no serviço de inserção de dados que está sendo gerado a partir do modelo - elipse 1.1. Isso fará com que o serviço de dados sendo gerado, reconheça os mesmos tipos de dados de saída do serviço do Phred. Além disso, é gerada uma interface para que o usuário especifique o mapeamento dos dados que devem ser inseridos - elipses 1.2 e 2. O mapeamento de dados da saída do serviço do Phred para o banco de dados, após ser definido, entra na composição do serviço de inserção de dados - elipse 2.1.

Na TAB. 5.2 uma parte do WSDL do serviço Web do Phred que descreve a estrutura dos tipos de dados que esse serviço manipula é mostrada. A saída desse serviço TAB. 5.2.D especifica que o tipo de dado retornado é uma seqüência (Array) de *PHDInfo* descrito na TAB. 5.2.C. Para montar o serviço de inserção de dados o *In Services* precisa obter essa definição do tipo de dado de saída e utilizá-la como entrada para o serviço de inserção de dados. Por isso, as mesmas definições de dados mostradas na TAB. 5.2.A, B e C são inseridas no serviço de inserção de dados que está sendo gerado.

O mapeamento de inserção dos dados também é feito com base no WSDL mostrado na TAB. 5.2. Antes de o usuário montar o mapeamento de inserção, há a necessidade de se escolher qual tabela será utilizada para armazenamento dos dados. A partir do conjunto de tabelas disponíveis apresentadas no sistema, o usuário irá escolher aquela que deve armazenar os dados gerados pelo passo do workflow. Para o serviço do Phred, um mapeamento possível poderia ser dos elementos *strSequence* e *strChromatFile* pertencentes aos elementos *PHDInfo* gerados pelo serviço. Esses dois elementos contêm respectivamente o valor da seqüência genômica e uma descrição do arquivo (cromatograma) que contém tal seqüência. No banco de dados do GUS esses dados poderiam ser armazenados na tabela *Dots.NASequenceImpl*. Nessa tabela, os campos *sequence* e *description* representam a mesma semântica desses dados e poderiam ser os escolhidos para o armazenamento dos mesmos.

A FIG. 5.7 contém uma representação de como esse mapeamento pode ser efetuado. A partir do objeto fonte, ou seja, dos dados de saída do serviço Web, o usuário pode efetuar o mapeamento para o objeto destino que é representado pelos atributos escolhidos da tabela que possibilita o armazenamento dos dados. No exemplo ilustrado na FIG. 5.7 o usuário está mapeando os elementos *strSequence* e *strChromatFile* do serviço Web (WSPhred) para os atributos *sequence* e *description* da tabela Dots.NASequenceImp.

TAB. 5.2: Parte do WSDL do serviço Web do programa Phred.

```
<definitions>
    <types>
        <schema>
            ...
<!—A -->
        <complexType name="Dna">
            <sequence>
             <element name="strBase" nillable="true" type="soapenc:string"/>
             <element name="intQuality" type="xsd:int"/>
             <element name="intPeakLoc" type="xsd:int"/>
            </sequence>
        </complexType>

<!—B-->
        <complexType name="ArrayOfDna">
            <complexContent>
             <restriction base="soapenc:Array">
              <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:Dna[]"/>
             </restriction>
            </complexContent>
           </complexType>

<!— C-->
        <complexType name="PHDInfo">
            <sequence>
             <element name="strChromatFile" nillable="true" type="soapenc:string"/>
             <element name="intAbiThumbprint" type="xsd:int"/>
             <element name="strPhredVersion" nillable="true" type="soapenc:string"/>
             <element name="strCallMethod" nillable="true" type="soapenc:string"/>
             <element name="intQualityLevels" type="xsd:int"/>
             <element name="strTime" nillable="true" type="soapenc:string"/>
             <element name="intTraceArrayMinIndex" type="xsd:int"/>
             <element name="intTraceArrayMaxIndex" type="xsd:int"/>
             <element name="intTrimFirstBase" type="xsd:int"/>
             <element name="intTrimLastBase" type="xsd:int"/>
             <element name="intTrimErrorProb" type="xsd:int"/>
             <element name="strChem" nillable="true" type="soapenc:string"/>
             <element name="strDye" nillable="true" type="soapenc:string"/>
             <element name="strSequence" nillable="true" type="soapenc:string"/>
             <element name="dnaDataBlock" nillable="true" type="impl:ArrayOfDna"/>
            </sequence>
        </complexType>

        </schema>
    </types>
...
<!-- D -->
        <complexType name="ArrayOfPHDInfo">
            <complexContent>
              <restriction base="soapenc:Array">
                <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:PHDInfo[]"/>
              </restriction>
            </complexContent>
        </complexType>

        <message name="processPhredResponse">
            <part name="processPhredReturn" type="impl:ArrayOfPHDInfo"/>
        </message>
...
        <wsdl:portType name="Phred">
            <wsdl:operation name="processPhred" parameterOrder="in0 in1">
              <wsdl:input message="impl:processPhredRequest" name="processPhredRequest"/>
              <wsdl:output message="impl:processPhredResponse" name="processPhredResponse"/>
            </wsdl:operation>
          </wsdl:portType>
...
</definitions>
```

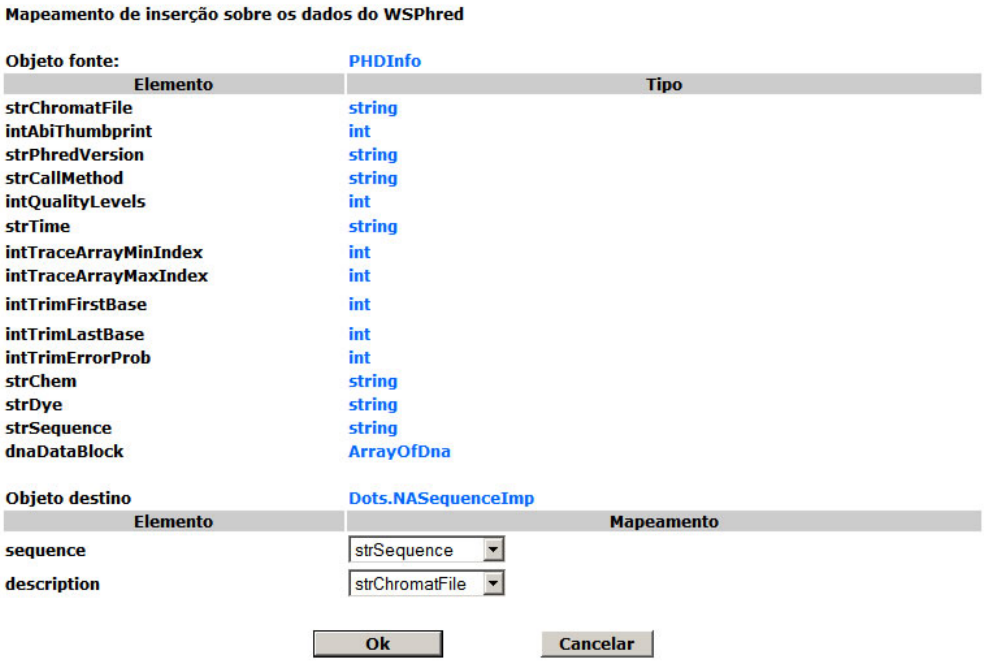

FIG. 5.7: Definição do mapeamento de inserção de dados.

Tendo-se criado o serviço de inserção de dados para o passo que o usuário deseja, o módulo de redefinição de workflows do *In Services* deve alterar o processo do workflow científico. Essa alteração visa adicionar ao fluxo de execução do workflow o serviço de dados criado. Um novo arquivo que descreve o workflow redefinido é gerado com o serviço de inserção de dados - FIG. 5.8.

Os serviços de inserção e recuperação de dados definidos no sistema *In Services* farão uso dos Arquivos de Tarefas de Banco de Dados para efetuarem o acesso à fonte de dados, seja ela pra inserção ou para recuperação de dados. Na TAB. 5.3 há um exemplo desse arquivo com o comando de inserção para a tabela *Dots.NASequenceImp* do esquema do GUS. No momento da execução do serviço de inserção, baseado no mapeamento especificado pelo usuário, o Arquivo de Tarefa de Banco de Dados da tabela utilizada no mapeamento é obtido. De posse desse arquivo, o serviço de inserção o interpreta e finaliza o comando para o armazenamento no banco passando os dados que devem ser inseridos. Dentre esses dados, há um que identifica a execução do serviço (*service_ execution*) e será atribuído pelo próprio sistema. Seu valor referencia-se ao identificador da instância de execução de um serviço, isto é, para cada tupla da tabela, referencia-se a instância do

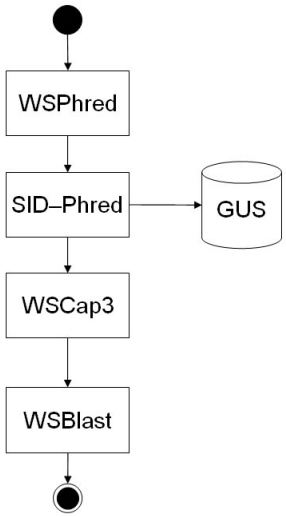

FIG. 5.8: Workflow SUBGARSA redefinido com serviço de inserção de dados para o Phred - SID-Phred.

TAB. 5.3: Parte de arquivo de tarefa de banco de dados para inserção na tabela Dots.NASequenceImp.

```
<sqlUpdateStatement name="dots_nasequenceimp_insert">
  <expression>
      INSERT INTO dots.nasequenceimp
      (sequence_version, subclass_view, ..., sequence,..., service_execution)
      VALUES
      (@sequence_version, '@subclass_view', ..., '@sequence',..., '@service_execution')
  </expression>
</sqlUpdateStatement>
```

serviço que a gerou. Uma instância de um Arquivo de Tarefa de Banco de Dados para a inserção é então gerada com os dados a serem armazenados. Essa instância é então interpretada e a inserção de fato efetuada.

Voltando ao estudo de caso, na execução do serviço de inserção de dados, os dados resultantes do passo que compõe a execução do serviço Web do Phred seguem o fluxo de execução sendo os dados de entrada do serviço de inserção de dados. Com o início da execução desse serviço, uma interpretação do mapeamento de inserção de dados é feita, elipse 1 da FIG. 5.9. Com isso, baseado nessa interpretação, o serviço de inserção de dados obtém o arquivo de tarefa de banco de dados (TAB. 5.3) que contém os comandos de inserção para as tabelas referenciadas no mapeamento, elipse 2. De posse dos dados a serem inseridos e do arquivo de tarefa de banco de dados, o serviço de inserção instancia um arquivo de tarefas de banco de dados que fará a inserção nas tabelas com os dados

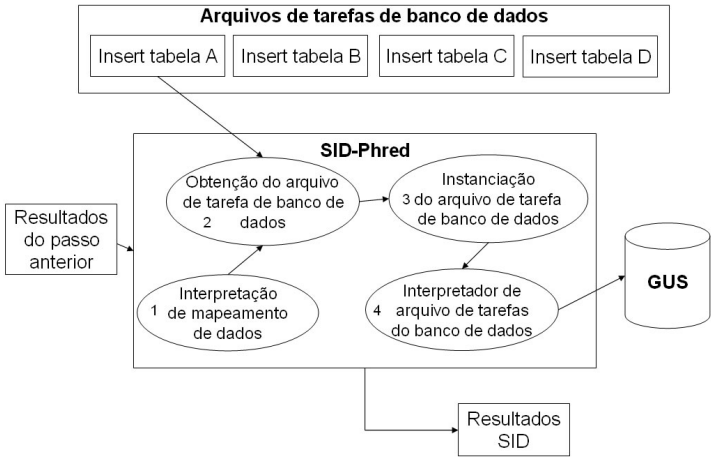

FIG. 5.9: Execução do serviço de inserção de dados.

de entrada do serviço, elipse 3 da FIG. 5.9. A TAB. 5.4 contém uma parte do arquivo de tarefas de banco de dados instanciado para inserção dos dados resultantes do serviço Web do Phred na tabela *Dots.NASequenceImp*. Dos dados resultantes do primeiro passo do workflow SUBGARSA, baseado no mapeamento de inserção, somente os elementos *strSequence* e *strChromatFile* serão armazenados, vide TAB. 5.4. Além desses dados, há o registro da execução do serviço através do atributo *service_ execution*. Por fim, o arquivo de tarefas de banco de dados é interpretado, a tarefa de carga de dados é executada com os valores contidos no arquivo, e a inserção é feita, elipse 4 da FIG. 5.9. Os dados que compuseram a entrada do serviço de inserção são então repassados ao passo seguinte do workflow.

As informações referentes à redefinição de workflows com serviços de inserção de dados também são mantidas no banco de dados do sistema. Essas informações irão possibilitar aos cientistas, que executam os workflows no *In Services*, a saber como os serviços de inserçao de dados foram utilizados no workflow além de quais tipos de dados foram inseridos e como foram gerados. O registro da execução dos serviços de inserção de dados também é importante para que se possa definir serviços de recuperação de dados. O registro de qual execução gerou os dados armazenados possibilita a recuperação desses. Dessa forma será possível redefinir o workflow original para permitir a recuperação de dados a partir do ponto de inserção de dados.

TAB. 5.4: Parte de arquivo de tarefa de banco de dados instanciado para inserção dos dados gerados no serviço Web do Phred na tabela *Dots.NASequenceImp*.

```
<sqlUpdateStatement name="dots_nasequenceimp_insert">
        <expression>
            INSERT INTO dots.nasequenceimp (sequence, description, service_execution)
            VALUES
            ('cattgtgttatcacactgtaaaattcattgtggtgtttgtaatctgtaagtatag...', 'A04.esd', 1)
        </expression>
        <expression>
            INSERT INTO dots.nasequenceimp (sequence, description, service_execution)
            VALUES
            ('atgcaggtcgactctagaggatcatcgagtatgctttcgtgtggtattaaccc...', 'A05.esd', 1)
        </expression>
        <expression>
            INSERT INTO dots.nasequenceimp (sequence, description, service_execution)
            VALUES
            ('cctataagatcgcgtcgcgtagtgccgctcgtgtagcgcgctgagcggttcg...', 'A06.esd', 1)
        </expression>
        <expression>
            INSERT INTO dots.nasequenceimp (sequence, description, service_execution)
            VALUES
            ('tcctcctgaagccctacgcgatttagagatcaccagggaataacgaggaga...', 'A07.esd', 1)
        </expression>
</sqlUpdateStatement>
```

5.4 INSERINDO UM SERVIÇO DE RECUPERAÇÃO PARA OBTENÇÃO DE DADOS DO PHRED

A redefinição de um workflow científico para incluir um serviço de recuperação de dados é normalmente feita quando o usuário pretende recuperar dados gerados por execuções prévias de um workflow redefinido com um serviço de inserção de dados. Isso devido ao fato do serviço de recuperação depender dos registros identificados por instâncias de execução do workflow científico que utilizou o serviço de inserção de dados. Os dados a serem recuperados serão aqueles que foram armazenados e identificados por alguma instância (execução) do serviço de inserção de dados aplicado ao passo escolhido. Ou seja, pode-se redefinir o workflow SUBGARSA (FIG. 5.2) (redefinido com um serviço para armazenamento dos dados gerados pelo serviço WSPhred), para inserir um serviço para recuperação de dados gerados pelo WSPhred. A geração do serviço de recuperação de dados se dá após esse momento, ou seja, o sistema verifica como o serviço de inserção de dados para aquele passo foi definido e então efetua a geração do serviço de recuperação de dados. Com relação à atribuição dos tipos de dados no serviço de recuperação compatíveis com os tipos de dados do serviço Web que deverá recebê-los, supõe-se que a descrição dos tipos de dados contidas no WSDL do serviço que irá receber os dados seja a mesma do serviço que os gerou no momento da inserção. Isso porque esses serviços são consecutivos na definição do workflow (WSPhred e WSCap3) e conseqüentemente as descrições dos tipos de dados de saída do primeiro são compatíveis com as descrições dos tipos de dados

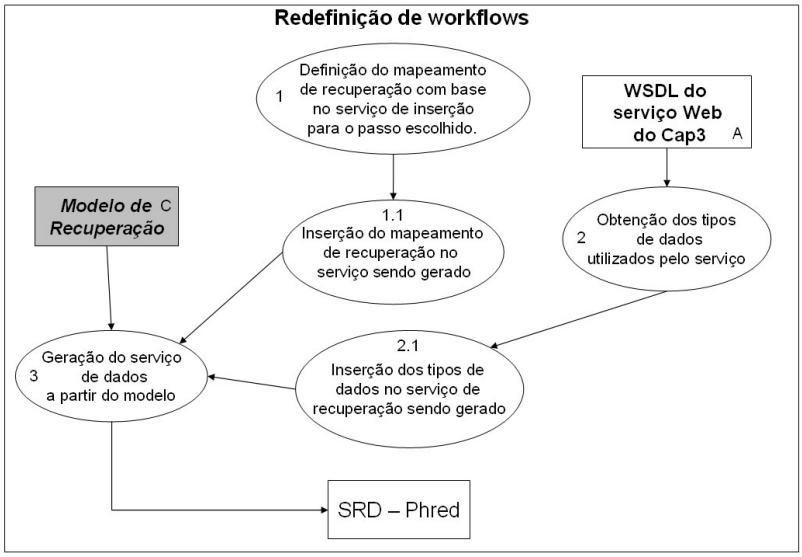

FIG. 5.10: Geração do serviço de recuperação de dados.

do segundo. Por esse motivo a descrição dos tipos de dados é obtida do WSDL do serviço que irá receber os dados do banco (WSCap3) e será então inserida no WSDL do serviço de recuperação de dados sendo gerado. Nesse momento não haverá a necessidade de que o usuário especifique manualmente um mapeamento do banco de dados para o serviço, como é feito de maneira inversa durante a geração do serviço de inserção. Isso porque esse mapeamento é baseado naquele que foi definido durante a geração do serviço de inserção de dados.

A FIG. 5.10 mostra como ocorre a redefinição do serviço de recuperação de dados pelo componente de redefinição de workflows do sistema *In Services*. A redefinição do serviço de recuperação de dados tem início após a escolha do passo que teve seus dados armazenados em um workflow redefinido com um serviço de inserção de dados. Baseado no mapeamento definido para o serviço inserção de dados para aquele, o sistema *In Services* montará um mapeamento inverso (do banco de dados para o serviço) para ser utilizado no serviço de recuperação de dados - elipses 1 e 1.1 da FIG. 5.10. Além desse mapeamento, há a necessidade que o serviço de recuperação de dados possua seus tipos de dados compatíveis com a definição dos tipos de dados do serviço que irá recebê-los. Por esse motivo, ocorre a leitura do WSDL desse serviço e atribuição dos tipos de dados na descrição do serviço de recuperação de dados que está sendo gerado - elipse 2 e 2.1 da FIG. 5.10. A partir dessas informações, o serviço de recuperação é gerado para obter os dados do serviço Web

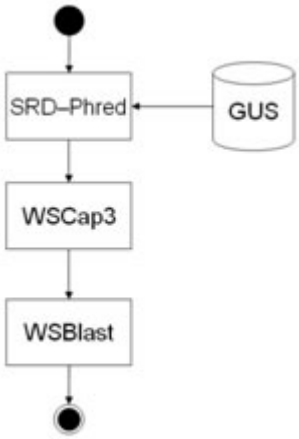

FIG. 5.11: Workflow SUBGARSA redefinido com serviço de recuperação de dados para o Phred.

do Phred e que foram previamente armazenados por um serviço de inserção de dados.

Após a geração do serviço de recuperação de dados, o mesmo deve ser inserido no workflow para compor um de seus passos. Um novo arquivo que descreve o workflow redefinido é gerado contendo então o serviço de recuperação de dados. Caracteristicamente, os serviços de recuperação de dados serão os primeiros passos do workflow, pois apenas realizarão a atividade de busca dos dados no banco e passagem desses dados para um determinado serviço Web do workflow. A FIG. 5.11 ilustra o workflow SUBGARSA redefinido com o serviço de recuperação de dados obtendo aqueles dados gerados previamente pelo serviço do Phred e armazenados no banco.

A execução do serviço de recuperação é semelhante a do serviço de inserção. A FIG. 5.12 ilustra como essa execução ocorre. No início da execução desse serviço, deve ser escolhida a instância de execução do serviço que armazenou os dados necessários à re-execução do workflow - FIG. 5.12.1. Isso deve-se ao fato de que, provavelmente, o serviço que armazenou os dados ter sido submetido a diversas execuções e conseqüentemente ter armazenado diversas versões dos dados em cada execução. A partir dos dados referentes às instâncias de execução dos workflows, seus passos e serviços, é possível escolher aquela que gerou os dados que pretende-se obter e, a partir dela, recuperá-los. Após a escolha da instância, o serviço de recuperação de dados efetua a interpretação do mapeamento de dados que especifica a correspondência dos dados a serem obtidos do banco para os tipos de entrada do serviço que irá utilizá-los - elipse 2 da FIG. 5.12. Com base no mapeamento, o serviço de recuperação de dados obtém o Arquivo de Tarefa de Banco de

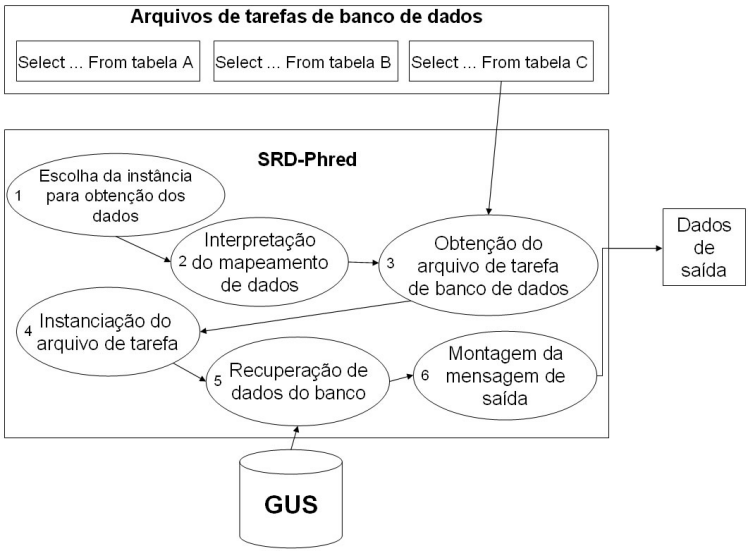

FIG. 5.12: Execução do serviço de recuperação de dados.

TAB. 5.5: Instância do arquivo de tarefa de banco de dados para recuperação de dados do Phred.

```
...
<sqlQueryStatement name=" dots_nasequenceimp_insert "/>
        <expression>
        SELECT (sequence, description) from dots.nasequenceimp where service_execution =1
        </expression>
  <resultStream name="statementOutput"/>
</sqlQueryStatement>
...
```

Dados que o permite executar o comando apropriado para a obtenção dos dados - elipse 3 da FIG. 5.12. Esse arquivo será instanciado com base no mapeamento de recuperação de dados e na instância de execução do servido de inserção de dados escolhida - elipse 4. A TAB. 5.5 contém uma representação desse arquivo com o comando para obter os dados referentes à instância de execução do serviço que armazenou os dados no banco. A partir da interpretação desse arquivo, os dados são obtidos do banco e formatados apropriadamente para o tipo de dado reconhecido pelo passo seguinte do workflow - elipse 5. Após isso, o serviço de recuperação monta a mensagem de saída (elipse 5) que é encaminhada pela máquina de execução de workflows para o passo seguinte. O fluxo de execução do workflow continua tendo como dados de entrada aqueles que foram obtidos pelo serviço de recuperação de dados.

Assim como nos serviços de filtragem e inserção de dados, as informações referentes à

redefinição de workflows para inclusão de serviços de recuperação também são mantidas no banco de dados do sistema. Isso possibilita a obtenção de conhecimento sobre como os serviços de recuperação de dados foram utilizados no workflow, além de quais dados foram recuperados no banco e utilizados nas execuções dos workflows.

5.5 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO DO SISTEMA *IN SERVICES*

O sistema *In Services* foi desenvolvido seguindo as propostas de desenvolvimento de software livre. Por conta disso, as tecnologias em uso no sistema foram escolhidas levando-se em consideração essa idéia. A primeira delas é o sistema gerenciador de banco de dados o qual foi utilizado o PostgreSQL. Além de ser um sistema gerenciador de banco de dados de distribuição livre, possui um ótimo conceito com relação a desempenho e robustez perante a comunidade de usuários, desenvolvedores e pesquisadores de banco de dados. Soma-se a isso o fato do esquema do GUS ser distribuído também para esse SGBD.

Outra característica da implementação do sistema *In Services* diz respeito à linguagem de programação escolhida. A linguagem Java foi utilizada para o desenvolvimento devido ao fato de possuir uma vasta quantidade de componentes reutilizáveis para se trabalhar com serviços Web o que favoreceu o desenvolvimento do módulo gerador de serviços de dados. A existência de ambientes para fácil disponibilização de serviços Web desenvolvidos em Java foi outro fator levado em consideração para a escolha dessa linguagem. Para o ambiente de acesso Web do sistema *In Services* foi usado o servidor TOMCAT e o ambiente AXIS foi utilizado para a disponibilização e execução dos serviços de dados gerados no sistema quando ocorrerem redefinições de workflows.

Com relação a execução dos workflows há a possibilidade de se usar qualquer ferramenta que execute workflows compostos por serviços Web. Isso porque essa camada é fracamente acoplada à ferramenta. Esse tipo de workflow pode ser descrito através da linguagem BPEL4WS ou, abreviadamente, BPEL. Essa linguagem é candidata para se tornar o padrão para descrição de workflows compostos por serviços Web (BPMI.ORG), (HARMON, 2005). Na versão do sistema *In Services* desenvolvida neste trabalho, foi utilizada a ferramenta Oracle BPEL Process Manager (ORACLE BPEL, 2005) que possui sua distribuição gratuita para a comunidade de desenvolvedores e interpreta workflows descritos em BPEL. A vantagem dessa ferramenta é possuir uma interface simples para execução de workflows, bem como permite verificar como ocorreu a invocação dos passos executados no workflow e a troca de dados entre eles. Funcionalidade essa, importante

FIG. 5.13: Arquitetura do sistema *In Services* orientada a componentes de software.

para a validação do trabalho. No sistema *In Services* a redefinição de workflows gera um conjunto de arquivos referentes à descrição do workflow redefinido e aos serviços Web que compõem seus passos. Isso permite que, através desses arquivos, o workflow seja disponibilizado em qualquer outra máquina de execução de workflows.

A FIG. 5.13 ilustra uma visão do sistema *In Services* orientada a componentes de software que compõem a ferramenta como um todo. Na camada de apresentação, o usuário tem acesso às funcionalidades do sistema. Por ela é possível executar as atividades da camada de aplicação. A camada de aplicação provê a execução das funcionalidades do sistema. Funcionalidades essas que envolvem desde o registro dos serviços Web no sistema, definição e redefinição dos workflows, e disponibilização e execução dos workflows. Por isso, essa camada envolve o uso de outras ferramentas. A primeira delas, o AXIS, é usada para dar suporte aos serviços de dados que forem gerados. Além dessa ferramenta, tem-se uma máquina de execução de workflows. Que irá fornecer a possibilidade de que os workflows sejam executados pelos usuários do sistema. A camada de serviços de dados é utilizada tanto na camada de apresentação quanto na camada de aplicação. Isso devido ao fato de a definição dos serviços de dados serem feitas com base nas necessidades do

usuário que interage com o sistema através da camada de apresentação. Já a geração, disponibilização e inserção desses serviços de dados no workflow é feita na camada de aplicação do sistema. Dando suporte a todas essas camadas está a camada de banco de dados que é responsável por manter os dados utilizados pelo sistema. Nessa camada estão os esquemas do GUS para o armazenamento dos dados científicos gerados a partir da execução dos workflows e a extensão do esquema do GUS (GUS+) para atender às funcionalidades do sistema *In Services*.

O protótipo do sistema *In Services* desenvolvido neste trabalho não possuiu todas as suas funcionalidades para atendimento à gerência de dados implementadas. Das dificuldades encontradas para a sua implementação as mudanças de versão do GUS e suas inconsistências de configuração foram relevantes no decorrer do trabalho. Das funcionalidades disponibilizadas na ferramenta, tem-se o registro de serviços Web para composição de workflows; o registro de workflows científicos previamente modelados; a alteração de workflows científicos para uso de serviços de filtragem de dados entre passos do workflow. Os serviços de inserção e recuperação de dados propostos na arquitetura tiveram suas funcionalidades apenas definidas na ferramenta. Essas definições atendem aos requisitos funcionais propostos na arquitetura do *In Services* e não foram desenvolvidas devido a limitações de tempo.

5.6 CONSIDERAÇÕES SOBRE SIMPLIFICAÇÕES DO SISTEMA *IN SERVICES*

A versão do sistema *In Services* desenvolvida neste trabalho adotou algumas simplificações em suas funcionalidades devido ao fato de já serem fornecidas por outras ferramentas e para atender a limitações de tempo para desenvolvimento do trabalho. A primeira delas diz respeito ao processo de modelagem de workflows compostos por serviços Web. No protótipo desenvolvido, essa funcionalidade não é fornecida, pois esse processo pode ser feito utilizando-se uma série de ferramentas livres (ACTIVEBPEL, 2006), (BPELPROJECT), (ECLARUS, 2006), (PAULUSBERGER, 2004) disponibilizadas para esse fim. No sistema *In Services*, é possível registrar os workflows previamente modelados e a partir de então efetuar a redefinição do workflow para utilizar os serviços de dados. Uma outra simplificação do sistema diz respeito ao sistema suportar apenas workflows sequenciais.

Uma segunda simplificação diz respeito ao registro das execuções dos workflows e de seus passos. O método proposto faz uso dos arquivos de log gerado pela máquina de execução de workflows quando um workflow científico é executado. Isso torna o sistema

In Services acoplado ao formato de arquivo de log específico da máquina de execução de workflows em uso. Fato esse que pode dificultar uma possível alteração do sistema para trabalhar com outras máquinas de execução. Ou até mesmo invalidar a funcionalidade de registro das execuções de workflows caso opte-se por usar uma máquina de execução de workflows que não gere arquivos de log.

Já com relação à especificação dos serviços de dados, as seguintes simplificações foram feitas:

- O serviço de filtragem de dados foi desenvolvido nesta versão inicial para trabalhar com tipos de dados compostos por uma seqüência (*Array*) de um tipo de dado complexo. O filtro é definido sobre os elementos de tipos simples que compõem o tipo de dado complexo da seqüência e a filtragem de dados será feita sobre os valores desses elementos. Um tipo de dado complexo é aqui referenciado com sendo um tipo de dado que contém em sua estrutura um conjunto de um ou mais tipos de dados. Analogamente, pode ser pensado como um objeto da linguagem *Java* que possui um conjunto de atributos ou como uma estrutura (*struct*) na linguagem *C*. Já os tipos simples são considerados como tipos de dados cujos valores são representados por tipos primitivos de dados como numérico ou textual. Durante o processo de filtragem ocorre uma iteração sobre todos os itens da seqüência de dados e os valores correspondentes àqueles tipos, que estão envolvidos nos critérios de filtragem especificados pelo usuário, são verificados;

- O mapeamento de inserção de dados proposto é especificado pelo usuário de maneira direta apontando-se quais dados de um serviço Web do workflow devem ser armazenados em quais campos de uma tabela do banco de dados. Para a recuperação de dados, o mapeamento é feito de maneira automatizada baseando-se naquele definido no momento da inserção de dados;

- Os serviços de inserção e de recuperação de dados, assim como o serviço de filtragem, foram especificados para manipular tipos de dados definidos como uma seqüência de um tipo complexo. Cada elemento do tipo simples que faz parte do elemento complexo pode ser mapeado para um campo de uma tabela quando o armazenamento está sendo mapeado. Para o mapeamento de recuperação de dados, isso é feito apontando-se cada dado de um campo da tabela para um elemento do tipo de dado de entrada do serviço Web;

- Os serviços de dados propostos foram especificados a tratarem inicialmente tipos de dados numéricos e textuais

- A especificação dos serviços de inserção e recuperação de dados foi feita sobre a tecnologia OGSA-DAI (ANJOMSHOAA et. al.), (ANTONIOLETTI et. al., 2003), (OGSA-DAI PROJECT) para acesso a fontes de dados via serviços Web;

O capítulo seguinte aborda um exemplo de uso do sistema *In Services* demonstrando a redefinição do workflow SUBGARSA para uso de um serviço de filtragem de dados. Além disso, é apresentado como a execução desse workflow redefinido ocorre na ferramenta.

6 EXEMPLO DE USO DO SISTEMA *IN SERVICES*

Neste capítulo é mostrado um exemplo de redefinição e execução de workflows através do sistema *In Services*. Esse exemplo envolve a utilização de um serviço de filtragem de dados no workflow SUBGARSA. A filtragem de dados abordada no exemplo é definida sobre a saída de dados do passo que executa o serviço Web do programa Blast - terceiro passo do workflow SUBGARSA. A partir de parâmetros de filtragem definidos pelo usuário, os dados resultantes desse passo serão verificados e aqueles que não atenderem a tais parâmetros serão desconsiderados na execução do workflow.

6.1 REGISTRANDO OS SERVIÇOS DO WORKFLOW SUBGARSA

A redefinição de workflows se dá sobre workflows científicos previamente registrados no sistema. Antes do registro de workflows ser feito, há a necessidade de se registrar os serviços Web científicos que compõem os seus passos. Os serviços Web registrados disponibilizam a implementação de algum algoritmo científico. No exemplo do workflow trabalhado, tem-se três algoritmos: um para obtenção de seqüências a partir de arquivos gerados de seqüenciadores genômicos (implementado pelo programa Phred), outro para clusterização de seqüências (implementado pelo programa Cap3) e um terceiro de busca de similaridades entre seqüências (implementado pelo programa Blast). Os três serviços Web que compõem o workflow SUBGARSA disponibilizam a implementação desses algoritmos. A execução desse workflow corresponde uma parte do workflow GARSA utilizado no experimento de anotações de seqüências genômicas realizado no projeto BioWebDB (BIOWEBDB).

Assim, o registro de workflows no sistema *In Services* envolve o registro de algoritmos implementados e disponibilizados pelos serviços Web que compõem os passos do workflow, o registro dos serviços Web e por fim o registro do workflow. Na FIG. 6.1 mostra-se o registro de algoritmos no sistema *In Services*. No exemplo da figura, o registro está sendo feito atribuindo-se o nome do algoritmo (Blast) e uma descrição do processamento desse algoritmo (Algoritmo para busca de similaridades entre seqüências).

O registro da implementação de cada um desses algoritmos é feito através do registro de serviços Web. Vale ressaltar que para um algoritmo pode-se registrar vários serviços

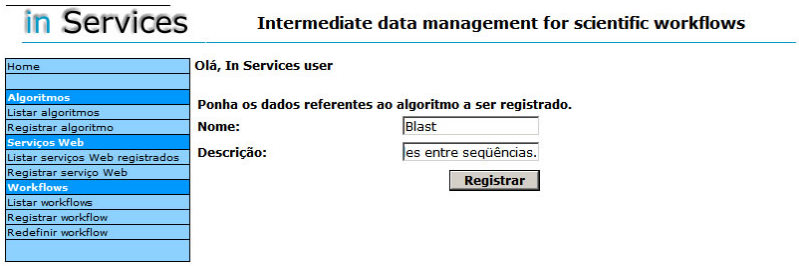

FIG. 6.1: Registro do algoritmo Blast no sistema *In Services*.

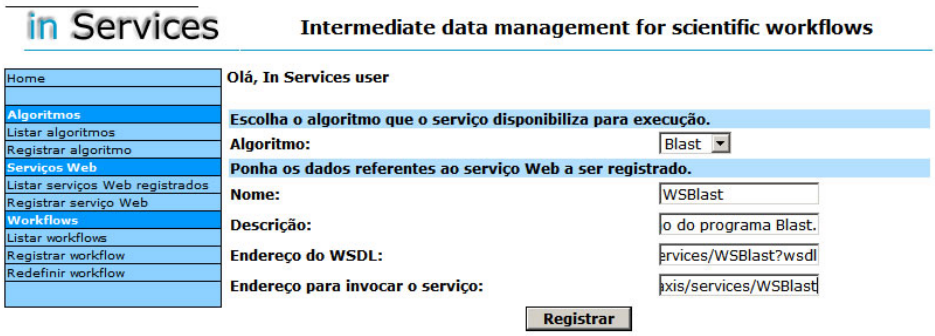

FIG. 6.2: Registro do serviço Web que disponibiliza o acesso à execução do algoritmo Blast.

Web que o disponibilizem. Os serviços Web registrados no sistema *In Services* comporão os passos de workflows também registrados na ferramenta. A FIG. 6.2 mostra o registro de um serviço Web que disponibiliza a execução do algoritmo Blast. Por conta disso, o usuário efetua a associação entre o algoritmo (Blast) e o serviço sendo registrado, vide FIG. 6.2. Além disso, é fornecido um nome e uma descrição caracterizando o serviço. Há a necessidade também de que seja fornecido o endereço de acesso do WSDL do serviço e de invocação do mesmo.

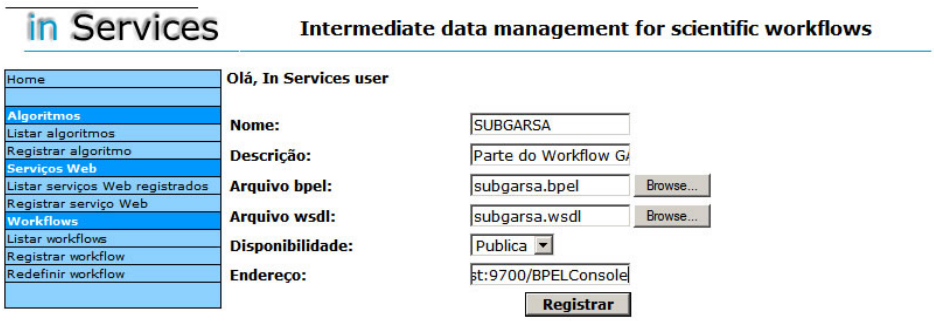

FIG. 6.3: Registro do workflow SUBGARSA.

6.2 REGISTRANDO O WORKFLOW SUBGARSA

Uma vez definido o workflow em uma ferramenta de modelagem de workflows qualquer, é feito o seu registro no sistema *In Services* - FIG. 6.3. O registro de workflows é feito fornecendo-se um nome e uma descrição do mesmo. Fornece-se também, o arquivo descritor de workflows - definido em linguagem BPEL - e o arquivo contendo o WSDL do workflow modelado. Completando-se o registro do workflow SUBGARSA, fornece-se a disponibilidade (pública ou privada) do mesmo e o endereço onde está acessível - como mostra a FIG. 6.3.

O registro da seqüência de passos que compõem o workflow é feito através da leitura do arquivo que descreve o workflow (Anexo 1). Nesse arquivo, há a seqüência de invocações aos serviços Web que compõem os passos do workflow. A partir desses dados, o registro dos passos do workflow é feito de maneira automática. Adotou-se a identificação única de cada serviço no arquivo que descreve o workflow como sendo o atributo nome. Ou seja, quando ocorrer a modelagem do workflow registrado, os serviços e suas invocações definidas no arquivo descritor do workflow têm no atributo nome, o mesmo valor para o campo nome do serviço registrado no *In Services*. Isso pelo fato de o sistema utilizar esses dados para consultar os serviços Web registrados no banco de dados do sistema *In Services* e então registrar os passos do workflow de maneira automatizada. Para exemplificar, tome-se o trecho de arquivo XML a seguir como parte do arquivo descritor de um workflow

TAB. 6.1: XML contendo parte de um arquivo descritor de workflow.

```
<!--A-->
<partnerLinks>
      <partnerLink name="WSPhred" partnerLinkType="ns1:WSPhred_PL"
      partnerRole="WSPhred_Role"/>
      <partnerLink name="WSCap3" partnerLinkType="ns2:Cap3If_PL"
      partnerRole="Cap3If_Role"/>
...
</partnerLinks>
...
      <scope name="Scope_1">
            <sequence name="Sequence_1">
...
<!--B-->
                  <invoke
                  inputVariable="WSPhred_ InputVariable"
                  name="Washed"
                  operation="processPhredReturningSequences"
                  outputVariable="WSPhred_OutputVariable"
                  partnerLink="WSPhred"
                  portType="ns1:WSPhred"/>
...
<!--C-->
                  <invoke
                  inputVariable="WSCap3_InputVariable"
                  name="WSCap3"
                  operation="runCap3"
                  outputVariable="WSCap3_OutputVariable"
                  partnerLink="WSCap3"
                  portType="ns2:Cap3If"/>
...
            </sequence>
      </scope>
```

modelado em linguagem BPEL.

Na TAB. 6.1, em (A) tem-se os elementos *partnerLinks* que indicam os serviços que compõem os passos do workflow. Em (B) e (C) há os elementos *invoke* que representam as invocações aos serviços representados pelos *partnerLinks*. Uma vez que os elementos *invoke* e *partnerLink* possuem no atributo *name* o mesmo valor correspondente ao campo nome dos serviços Web registrados no *In Services* (WSPhred e WSCap3), os passos do workflow podem ser identificados automaticamente a partir das seqüências de *invoke* presentes no arquivo.

6.3 REDEFINIÇÃO DO WORKFLOW SUBGARSA

Após o registro do workflow, é possível dar início ao processo de redefinição do mesmo para incluir serviços de dados. Neste exemplo, será criado um serviço de filtragem de dados sobre a saída do serviço Web que disponibiliza a execução do programa Blast (WSBlast). O primeiro passo para a redefinição de workflows ser feita é escolher essa opção no sistema - FIG. 6.4.A. Feito isso, escolhe-se qual workflow redefinir fornecendo-se um nome e uma

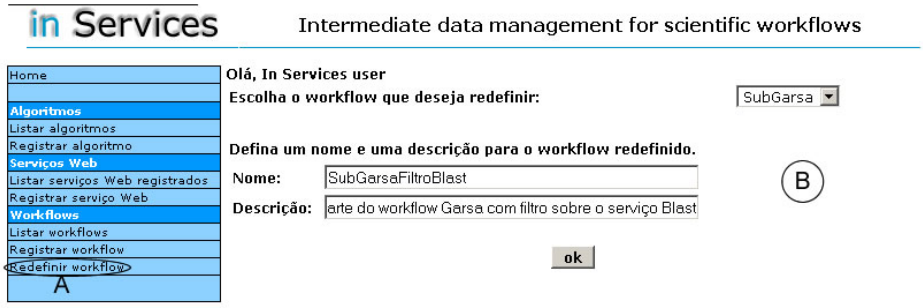

FIG. 6.4: Escolha do workflow a ser redefinido.

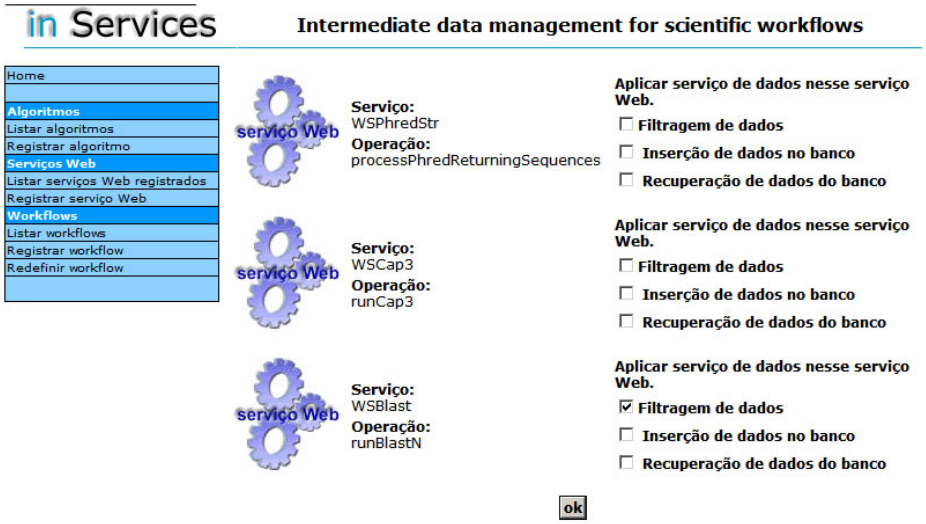

FIG. 6.5: Escolha do serviço de dados a ser aplicado sobre o passo do workflow.

descrição para o novo workflow - FIG. 6.4.B.

Em seguida, o sistema obtém os serviços Web que compõem os passos do workflow e solicita que o usuário selecione aquele sobre o qual pretende-se aplicar um serviço de dados. O exemplo da FIG. 6.5 ilustra a seqüência de passos que compõem o workflow SUBGARSA. Nesse momento, o usuário deve especificar qual serviço de dados pretende utilizar em qual passo do workflow. Neste exemplo, o serviço escolhido é o WSBlast e o serviço de dados pretendido é o de filtragem.

O próximo passo é montar os parâmetros de filtragem que devem ser levados em consideração no momento em que o workflow for executado. Esses parâmetros farão parte do serviço de filtragem de dados e serão aplicados sobre os dados resultantes do WSBlast - FIG. 6.6.

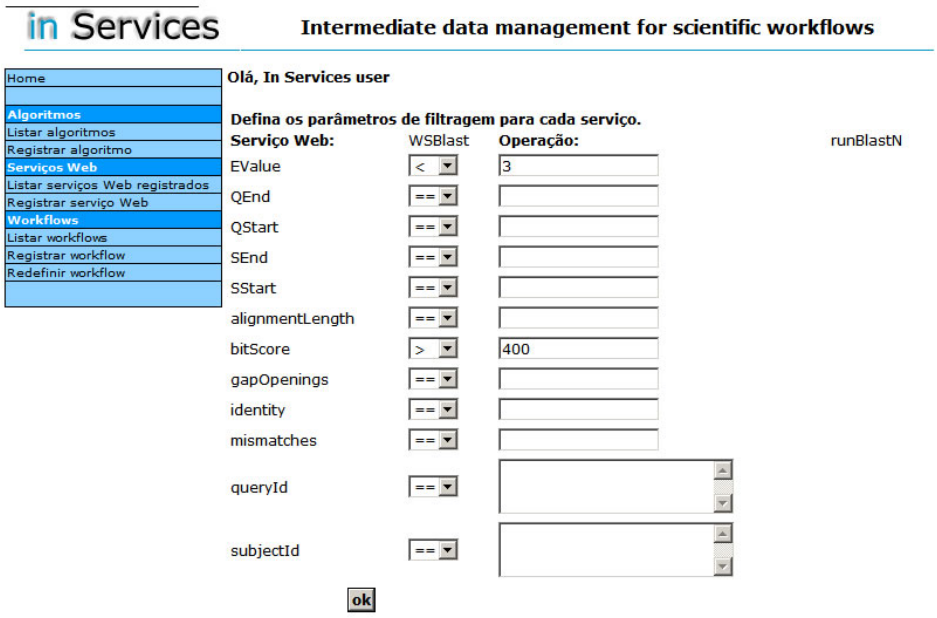

FIG. 6.6: Definição dos parâmetros de filtragem.

No exemplo de definição de parâmetros de filtragem da FIG. 6.6, somente os dados resultantes do serviço Web do Blast que tiverem o elemento *EValue* com valor menor que 3 e elemento *bitScore* com valor maior que 400 serão mantidos no fluxo de execução. Aqueles resultados com valores que não atendam a esses parâmetros serão descartados.

Por fim, um novo arquivo que descreve o workflow remodelado (Anexo 3) é gerado e fornecido pelo sistema juntamente com os arquivos WSDL do novo workflow (Anexo 4) e dos serviços Web (Anexo 5, 6, 7 e 8) que compõem os seus passos como na FIG. 6.7. Dentre os arquivos fornecidos, consta o arquivo WSDL do novo serviço de dados (FilterWSBlast) gerado para efetuar a filtragem sobre os dados do serviço Web do Blast.

6.4 EXECUÇÃO DO WORKFLOW SUBGARSA REDEFINIDO

De posse dos arquivos fornecidos pelo sistema é possível ao usuário efetuar a disponibilização do workflow em qualquer máquina de execução de workflows. Para este exemplo foi utilizada a máquina de execução de workflows Oracle BPEL Process Manager. Apesar de não ser uma ferramenta de código aberto, é uma ferramenta de distruição grátis e que possui uma interface bem intuitiva para verificação da troca de mensagens que ocorreram durante a execução do workflow. Característica essa importante para a validação

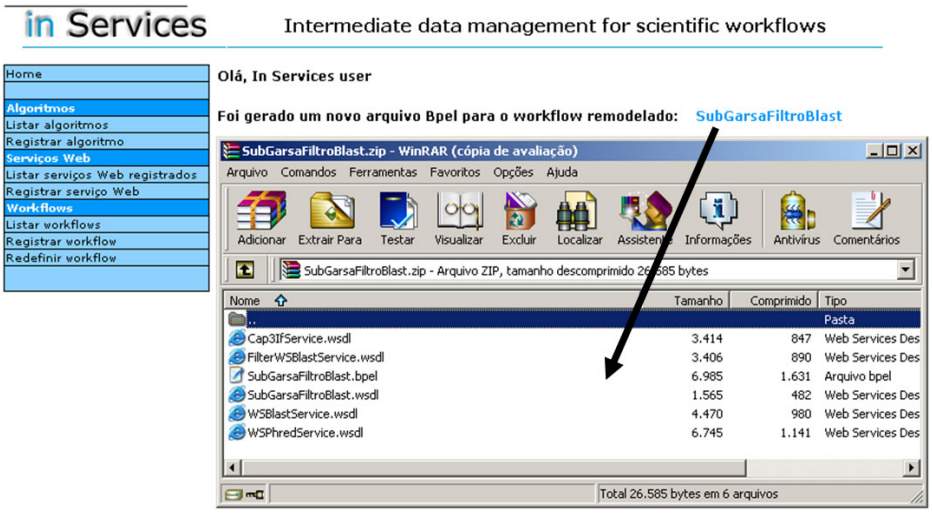

FIG. 6.7: Arquivos do workflow remodelado.

do trabalho.

Para executar o workflow nessa ferramenta, utilizou-se um arquivo gerado pelo seqüenciador genômico do projeto BioWebDB da Fundação Oswaldo Cruz. Esse arquivo contém uma série de dados referentes a seqüenciamentos para estudos genômicos de organismos dentro projeto. Neste trabalho, o arquivo foi tomado apenas como uma amostra para a validação do sistema.

Na execução do workflow SUBGARSA, o resultado do Blast (terceiro passo do workflow) resulta em uma seqüência (Array) contendo 23 itens, onde cada item contém os dados referentes à similaridade. A redefinição desse workflow inserindo-se um serviço de filtragem de dados após o passo que executa o programa Blast permite que o resultado desse passo seja apenas de dados que atendam aos requisitos de filtragem.

A FIG. 6.8 ilustra a instância de execução do workflow SUBGARSA (redefinido com o serviços de filtragem de dados) na máquina de execução de workflows utilizada. É possível visualizar na FIG. 6.8 a troca de dados entre os serviços (por exemplo *InputToPhredStr*, *PhredStrToCap3*, *Cap3ToBlast*) bem como a invocação de cada um deles *WSPhredStr*, *WSCap3*, *WSBlast* e *FilterWSBlastService*.

Ao clicar nos ícones que representam a transição de dados de um serviço para o outro (por exemplo *WSBlastToFilterName*), é possível visualizar uma parte dos dados de saída do serviço Web do Blast, destacando-se o volume de dados retornados - FIG. 6.9. Esses

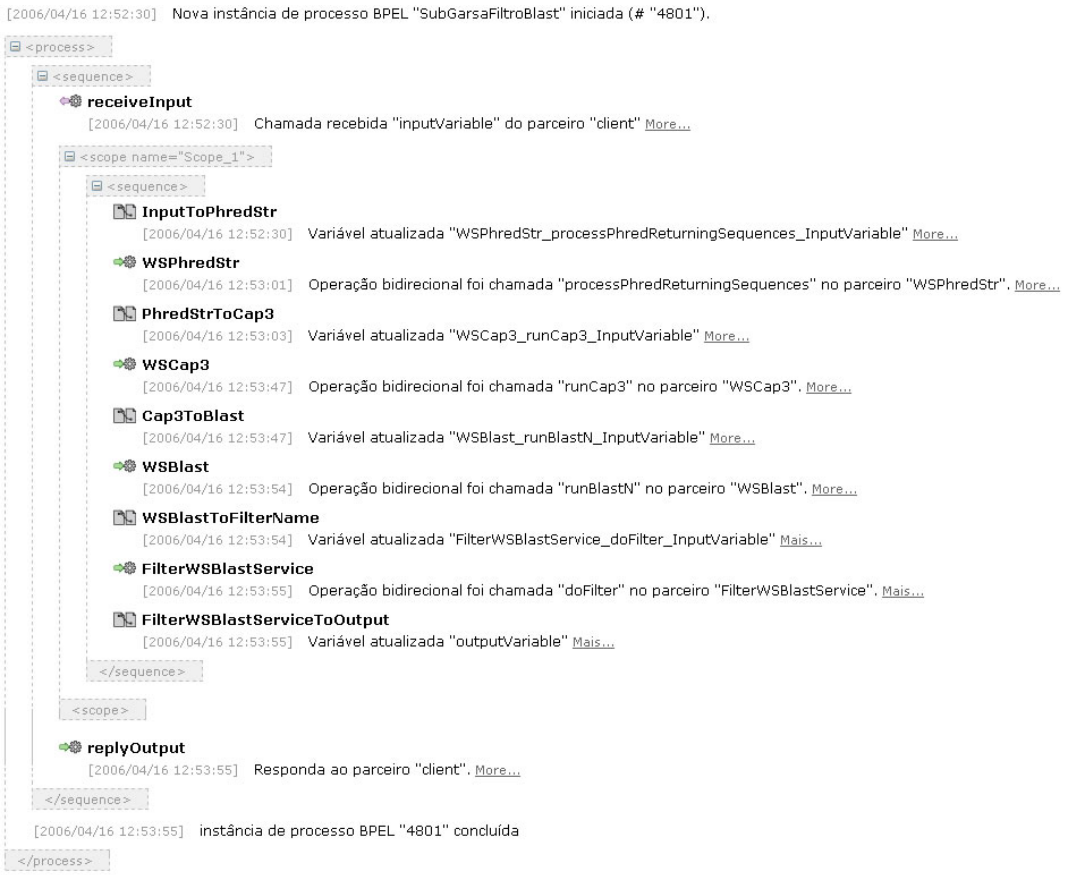

FIG. 6.8: Instância de execução do workflow SUBGARSA.

FIG. 6.9: Volume de dados resultantes do serviço Web do Blast.

foram os dados de entrada para o serviço de filtragem de dados - *FilterWSBlastService*.

A TAB. 6.2 contém o volume de dados resultantes do serviço de filtragem de dados; após aplicar o filtro, o volume de dados é reduzido para o número daqueles que atendem aos parâmetros de filtragem definidos pelo usuário ($EValue < 3$ e $bitScore > 400$). No exemplo apresentado, o resultado do workflow com o filtro de dados (TAB. 6.2) é de uma seqüência (Array) com apenas um item de dados, diferentemente do resultante do workflow sem a aplicação do filtro FIG. 6.9.

TAB. 6.2: Resultado final do workflow SUBGARSA com serviço de filtragem de dados sobre os resultados do Blast.

```
<result arrayType="ns1:BlastResult[1]" type="soapenc:Array" >
       <doFilterReturn type="ns1:BlastResult" >
              <EValue type="xsd:double" >2.0E-124</EValue>
              <QEnd type="xsd:int" >376</QEnd>
              <QStart type="xsd:int" >143</QStart>
              <SEnd type="xsd:int" >8928</SEnd>
              <SStart type="xsd:int" >8694</SStart>
              <alignmentLength type="xsd:float" >235.0</alignmentLength>
              <bitScore type="xsd:float" >442.0</bitScore>
              <gapOpenings type="xsd:int" >1</gapOpenings>
              <identity type="xsd:float" >99.15</identity>
              <mismatches type="xsd:int" >1</mismatches>
              <queryId type="soapenc:string" >Contig1</queryId>
              <subjected
              type="soapenc:string">gi|1786532|gb|AE000141.1|AE000141</subjectId>
       </doFilterReturn>
</result>
```

7 CONCLUSÕES

A realização de experimentos *in silico* com o auxílio do uso de workflows científicos compostos por serviços Web tem sido gradativamente adotada pela comunidade científica de bioinformática. A tecnologia de serviços Web permitiu definir experimentos científicos executados em ambientes computacionais heterogêneos a partir da composição de uma série de serviços disponibilizados pela comunidade científica. Esses serviços Web, geralmente, disponibilizam a execução de algum programa científico e podem ser acessados por uma interface padrão via Web. Assim, workflows compostos por serviços Web estão de fato executando uma série de programas científicos através de invocações desses serviços Web.

Na realização de experimentos científicos, a análise de dados envolvidos em tal atividade é de importância relevante para que os cientistas validem os resultados gerados. A característica distribuída dos serviços Web que compõem passos de workflows científicos levantou a questão de como acompanhar os dados gerados e manipulados durante a execução desses tipos de workflows. Atualmente, uma série de sistemas tem dado suporte à definição e execução de workflows compostos por serviços Web. Porém, pouco se tem trabalhado em soluções para o gerenciamento de dados gerados ao longo da execução de workflows - dados intermediários, de especial interesse para o acompanhamento e/ou registro do experimento *in silico*.

Neste trabalho, foi proposto um sistema cuja arquitetura possibilita a redefinição de workflows científicos compostos por serviços Web. Essa redefinição visa permitir o gerenciamento de dados envolvidos na execução de workflows, incluindo algumas funcionalidades necessárias para o acompanhamento de experimentos ao tratamento de dados em ambientes científicos. A primeira delas refere-se a ter-se um meio apropriado de serem mantidos os dados manipulados nas realizações de experimentos via workflows científicos. A arquitetura do sistema *In Services*, proposta aqui, oferece um meio de armazenamento centralizado para os dados envolvidos nos workflows científicos de bioinformática. O esquema utilizado deve ser capaz de armazenar dados genômicos resultantes de execuções de vários passos dos workflows científicos. Para esse armazenamento é utilizado um sistema

gerenciador de banco de dados.

No que diz respeito à redefinição dos workflows científicos, quando de interesse do cientista, há a possibilidade de serem adicionados serviços Web apropriados para inserção de dados intermediários no banco de dados. Esses serviços aqui denominados de serviços de inserção de dados, são disponibilizados pelo sistema *In Services* e obtêm, durante a execução do worklow, os dados gerados pelo passo anterior efetuando seu armazenamento no banco de dados. Além desse serviço, há o serviço de recuperação de dados que pode ser adicionado aos workflows para que dados previamente armazenados pelos serviços de inserção sejam utilizados como entrada para passos dos workflows. Essa característica possibilita que workflows sejam re-executados parcialmente, ou seja, de posse dos dados já armazenados, não há a necessidade de que todos os passos do workflow sejam re-executados. Poder-se-á executá-lo a partir de um passo que receba os dados armazenados e continue a execução do workflow a partir daquele ponto.

Além dos serviços de inserção e recuperação de dados, no *In Services*, há a possibilidade ainda de serem adicionados aos workflows científicos, serviços de filtragem de dados. Esses serviços, baseados em parâmetros de filtragem definidos pelo usuário, possibilitam a eliminação de dados que estejam fora do escopo do experimento durante a execução do workflow.

O protõtipo desenvolvido baseou-se no uso de tecnologias de distribuição livre. Essas tecnologias envolvem desde o banco de dados utilizado no sistema, passando pela linguagem na qual o mesmo foi implementado, o uso (geração e disponibilização) dos serviços de dados além da máquina de execução de workflows. Para atender a limitações de tempo para a conclusão do trabalho, foram adotadas algumas simplificações no desenvolvimento do protótipo. Essas simplificações permitiram validar algumas das propostas do trabalho.

O estudo de caso elaborado permitiu validar a idéia de redefinição de workflows. Analisou-se, através dele, os cenários de uso dos serviços de dados para os tipos de gerenciamento propostos. Além disso, o estudo de caso possibilitou a validação da execução dos workflows redefinidos.

7.1 CONTRIBUIÇÕES

A proposta da arquitetura do sistema *In Services* para gerenciamento de workflows científicos de bioinformática com suporte a gerenciamento de dados intermediários é a principal

contribuição deste trabalho. O funcionamento dessa arquitetura é detalhado neste trabalho através de sua aplicação a um workflow científico real (GARSA) em uso por um grupo de pesquisa (BioWebDB) na Fundação Oswaldo Cruz.

Deriva dessa proposta a identificação de alguns serviços básicos de gerência desses dados: *inserção, recuperação* e *filtragem*. Através destes serviços facilita-se a gerência dos dados intermediários em um ambiente estruturado, mantendo-se somente os dados relevantes e permitindo a realização de análises e a reutilização dos mesmos por diferentes workflows ou por re-execuções parciais do workflow que os gerou.

Outra contribuição do trabalho é o incentivo à escolha de serviços Web como uma maneira adequada para a composição de programas científicos dispostos de maneira distribuída, na medida em que o sistema oferece facilidades para a geração de serviços de dados.

Uma outra contribuição do trabalho foi a disponibilização do protótipo funcional do sistema sobre o esquema de dados GUS, podendo ser reutilizado pela comunidade científica de bioinformática. Por ter sido baseado em tecnologias de distribuição livre, o protótipo disponível pode ser utilizado em ambientes científicos que necessitem dos gerenciamentos de dados propostos neste trabalho. Além disso, o protótipo contitui-se como um passo inicial para a identificação de serviços de tratamento de dados em workflows científicos compostos por serviços Web. Sendo assim, o mesmo pode futuramente ser incrementado provendo novas funcionalidades e serviços para o gerenciamento de dados intermediários.

Nesse protótipo, a extensão do esquema GUS para atender à proveniência de dados relativos à redefinição e execução de workflows científicos no sistema *In Services* também é uma outra contribuição. A extensão proposta permitiu o registro de proveniência de dados também sobre os serviços de dados que manipulam os dados intermediários a serem filtrados ou inseridos e recuperados do banco. Com isso, pode-se ter conhecimento de como os workflows científicos foram redefinidos além de como os dados foram manipulados nas execuções dos workflows.

7.2 MELHORIAS NO SISTEMA *IN SERVICES* E TRABALHOS FUTUROS

Como melhorias a serem desenvolvidas no sistema *In Services* pode-se apontar inicialmente a extensão dos tipos de dados manipulados pelos serviços de dados. Como apresentado anteriormente, os serviços de dados estão limitados a trabalharem com um conjunto limitado de tipos de dados. É interessante torná-los mais abrangentes para poderem ser

utilizados sobre uma diversidade maior de serviços Web científicos. Além disso, seria interessante propor uma maneira mais intuitiva de se efetuar o mapeamento de inserção de dados. Como foi apresentado, esse mapeamento atualmente é proposto com o usuário especificando a correspondência dos dados dos serviços Web para tabelas no banco de dados. Uma maneira interessante que poderia ser trabalhada, seria a identificação automática de quais tabelas do banco de dados são relacionadas com os dados de saída de um passo do workflow. Assim, quando o usuário escolhesse, no sistema, um serviço Web (passo do workflow) a ter seus dados inseridos no banco, o sistema poderia ser capaz de relacionar automaticamente os dados de saída desse passo com as tabelas que possibilitam seus armazenamentos. O relacionamento entre tabelas e dados de saída de um serviço Web poderia ser feito um úncia vez pelo usuário no momento do registro de serviços Web no sistema. Assim, esse relacionamento seria consultado no momento em que o usuário fosse utilizar um serviço de inserção de dados em algum passo do workflow e o mapeamento seria feito de maneira automática nesse mesmo momento. Uma outra melhoria a ser adotada no sistema seria a possibilidade de se trabalhar workflows com passos paralelos ou alternativos, além de workflows seqüenciais.

Como trabalho futuro, pode-se apontar o levantamento de outros serviços de dados a serem providos pelo sistema *In Services*. Alguns desses novos serviços poderiam efetuar transformações entre serviços Web com formatação de tipos de dados diferentes. Assim seria possível, através do uso desse serviço de dados, compor workflows com serviços Web mesmo que esses últimos não possuam uma formatação idêntica de tipos de dados. Além desse serviço, há a possibilidade de que serviços de atualização e manutenção de dados sejam utilizados em workflows científicos. O primeiro deles poderia ser utilizado para manter os dados armazenados no banco de dados utilizado pelo sistema *In Services* sincronizados com outras bases de dados. Como exemplo, tem-se que as seqüências armazenadas no GUS necessitassem ser sincronizadas com as seqûëncias recentemente dispostas no banco de dados do GenBank. Nesse cenário, o serviço de atualização de dados poderia comparar os dados do banco de dados local do sistema *In Services* atualizando-os com os dados do banco externo (GenBank). O outro serviço de dados, serviço de manutenção, poderia ser proposto para eliminar dados armazenados no banco que não fossem mais relevantes no contexto do experimento científico. Isso evitaria a manutenção desnecessária de dados armazenados que não estejam mais dentro do escopo do experimento.

Um outro trabalho futuro que poderia ser desenvolvido seria trabalhar com níveis mais

abstratos de descrição de workflows no sistema *In Services*. Atualmente, um workflow é modelado através da composição de um conjunto de serviços Web que realizam uma atividade específica. Com uma maior abstração para definição de workflows, pode-se compor um workflow científico através de funcionalidades desejáveis em um passo e cada funcionalidade pode ser disponibilizada por um, ou mais de um, serviço Web. Assim, durante a execução de um workflow, caso um serviço Web que atenda à funcionalidade de um passo estivesse indisponível, o sistema *In Services* poderia automaticamente executar aquele passo com um outro serviço Web que também atenda àquela funcionalidade. Esse novo modo de definição de workflows facilitaria a navegação e reuso dos mesmos e de seus passos.

Por fim, outro trabalho poderia ser o de aplicar a arquitetura do sistema *In Services* a outros ambientes científicos que não só a bioinformática, precisando para isso realizar levantamentos e estudos de caso em laboratórios científicos dessas outras áreas. Desse trabalho pode-se ainda propor o uso integrado de diversas instâncias do sistema *In Services* em ambientes científicos distintos. Assim, a integração de suas bases de dados pode permitir uma análise de dados sobre resultados de experimentos de diversos laboratórios.

8 REFERÊNCIAS BIBLIOGRÁFICAS

AALST, W. e HEE, K. **Workflow Management: Models, Methods, and Systems**. *MIT Press, January 2002.*

ActiveBPEL Designer *http://active-endpoints.com/products/activebpeldes/index.html. [Consultado em 15/02/2006].*

ALONSO, G. **BioOpera: Grid Computing in Virtual Laboratories**. *Ercim News online edition. N. 45. April 2001.*

ALTINTAS, I., BHAGWANANI, S., BUTTLER, D., et. al. **A Modeling and Execution Environment for Distributed Scientific Workflows**. *Demonstration track, 15th Intl. Conference on Scientific and Statistical Database Management (SSDBM), Boston, Massachussets, 2003.*

ltintasAltschul97 ALTSCHUL, S. F., et. al. **Gapped BLAST and PSI-BLAST: a new generation of protein database search programs**. *Nucleic Acids Research, 1997, Vol. 25, N. 17, pp. 3389-3402.*

ALTSCHULA, S., GISHA, W., MILLERB, W., MEYERSC, E., LIPMANA, D. **Basic local alignment search tool**. *Journal of Molecular Biology, 215(3), 1990.*

ANJOMSHOAA, A., et. al. **The Design and Implementation of Grid Database Services in OGSA-DAI**. *Concurrency and Computation: Practice and Experience. Vol. 17, Issue 2-4 , pp. 357 - 376.*

ANTONIOLETTI, M., et. al. **Experiences of Designing and Implementing Grid Database Services in the OGSA-DAI project**. *Global Grid Forum Workshop on Designing and Building Grid Services. 2003..*

BACKER, P. G., et. al. **An Ontology for Bioinformatics Applications**. *Bioinformatics, Vol. 15, N. 6, pp. 510-520, 1999*

BAHL, A., et. al. **PlasmoDB: The** *Plasmodium* **genome resource. An integrated database providing tools for accessing, analyzing and mapping expression and sequence data (both finished and unfinished)**. *Nucleic Acids Research, 2002, Vol. 30, N. 1 pp. 87-90*

BAUSCH, W., PAUTASSO, C., SCHAEPPI, R. e ALONSO, G., **BioOpera: Cluster-aware Computing**. *Proceedings of the 4th IEEE International Conference on Cluster Computing, 2002.*

BENSON,D.A., KARSCH-MIZRACHI,I., LIPMAN,D.J., OSTELL,J. WHEELER,D.L. **GenBank: update**. *Nucleic Acids Res., 32, pp. D27-D30, 2004.*

BioWebDB Consortium *http://www.biowebdb.org/index.html.*

BPEL Project *http://www.eclipse.org/bpel/. [Consultado em 15/02/2006].*

BPMI.ORG **BPML-BPEL4WS. A Convergence Path toward a Standard BPM Stack**. *Position Paper. August 15, 2002.*

Brett Tyler's Lab. **Installing GUS at VBI**. *Disponível em: http://www.gusdb.org/documentation/vbidoc.pdf.*

CAVALCANTI, M. C., TARGINO, R., BAIÃO, F., RÖSSLE, S., BISCH, P., PIRES, P. F., CAMPOS, M. L. e MATTOSO, M. L. Q. **Managing Structural Genomic Workflows Using Web Services**. *Data & knowledge engineering, Elsevier, Vol. 53, N. 1, pp. 45-74, 2005.*

Chado. GMOD Modular Schema **http://www.gmod.org/schema/index.shtml [Consultado em 15/09/2005]**.

CHANDRASEKARAN, S., et. al. **Composition, Performance Analysis and Simulation of Web Services**. *Distributed and Parallel Database (DPDB), September 2002.*

CURBERA, F., GOLAND, Y., ANDREWS, T., et. al. **Business Process Execution Language for Web Services v1.1**. *Microsoft, BEA, IBM, May-2003. Disponível em: http://www.ibm.com/developerworks/library/ws-bpel/.*

DÁVILA, A. M. R. et. al. **Garsa: genomic analysis resources for sequence annotation**. *Bioinformatics. pp. 4302-4303. 2005.*

DAVIDSON, S. B. R., CRABTREE, J. B. B., et. al. **K2, Kleisli and GUS: experiments in integrated access to genomic data sources**. *IBM Syst J 2001; 40 (2): pp. 512-31.*

DAVIES, K. **Combating Creative Chaos in Bioinformatics**. *Bio-IT World, 2002..*

DURHAM, A., et. al. **EGene: a configurable pipeline generation system for automated sequence analysis**. *Bioinformatics, 2005. Vol 21 N. 12 2005, pp. 2812 - 2813.*

ECKART, J. D., et. al. **A Life Scientist's Gateway to Distributed Data Management and Computing: The PathPort/ToolBus Framework**. *OMICS: A Journal of Integrative Biology, 2003, Vol. 7, N. 1: pp. 79-88.*

eClarus *http://www.eclarus.com/ [Consultado em 15/02/2006].*

EuroGrid. **http://www.eurogrid.org**.

EWING, B., et. al. **Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy Assessment**. *Genome Research. Vol. 8, Issue 3, pp. 175-185, March 1998.*

FINN, R. D., et. al. **Pfam: clans, web tools and services**. *Nucleic Acids Research, 2006, Vol. 34, pp. D247-D251.*

FELSENSTEIN, J. **PHYLIP - phylogeny inference package - version 3.5c**. *Technical report. Department of Genetics, University of Washington, Seattle.*

FOSTER, I., VOECKLER, J., WILDE, M. e ZHAO, Y. **Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation**. *Proceedings of Global and Peer-toPeer Computing on Large Scale Distributed Systems Workshop, May 1995.*

Garsa: genomic analysis resources for sequence annotation *http://www.biowebdb.org/garsa.*

NCBI. **GenBank Statistics**. *http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html.*

Geon. **GEON: Cyberinfrastructure for the Geosciences**. *http://www.geongrid.org/.*

Gene Ontology Consortium *http://www.gusdb.org/documentation/vbidoc.pdf . [Consultado em 09/08/2006].*

GIBAS, C., JAMBECK, P. **Desenvolvendo Bioinformática**. *Campus, 1st ed..*

GOBLE, C., WROE, C., STEVENS, R. e myGrid Consortium. **The myGrid Project: Services, Architecture and Demonstrator**. *In Proceedings UK e-Science All Hands Meeting 2003 Editors - Simon J Cox, (2003) pp. 595-603.*

GOODMAN, N., et. al. **The LabBase System for Data Management in Large Scale Biology Research Laboratories**. *Bioinformatics, Vol. 14, N.7, (1998) pp. 562-574.*

GRUBER, T. R. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**. *Int., Journal of Human-Computer Studies, Vol. 43, 1995.*

GUARINO, N. **Formal Ontology and Information Systems, Formal Ontology in Information Systems**. *Ed. Amsterdam, Netherlands: IOS Press, 1998.*

GUS. **Genomics Unified Schema**. *http://www.gusdb.org.*

HARMON, P. **BPEL and BPM**. *Business Process Trends. March, 2005.*

HERTZ-FOWLER, C. et. al. **GeneDB: a resource for prokaryotic and eukaryotic organisms**. *Nucleic Acids Research, 2004, Vol. 32.*

HOLLINGSWORTH, D. **Workflow Management Coalition. The Workflow Reference Model: WfMc**. *In http://www.wfmc.org/standards/docs/tc003v11.pdf. [Consultado em 20/02/2005].*

HOPPE, H., et. al. **EUROGRID - European Testbed for GRID Applications**. *GRIDSTART Technical Bulletin. 1. October, 2002.*

IBM. **Lotus Workflow**. *Disponível em: http://www.lotus.com/products/product3.nsf/wdocs/wfhome. [Consultado em 02/03/2005].*

KARP, P., et. al. **The EcoCyc Database**. *Nucleic Acids Research, Vol. 30, N. 1, pp. 56-58, 2002.*

Kepler Project. **Kepler: An Extensible System for Design and Execution of Scientific Workflows. User's Guide**. *Disponível em: http://cvs.ecoinformatics.org/cvs/cvsweb.cgi/ checkout /kepler/docs/user/KeplerEndUserDoc.pdf.*

KÜNZL, J.. **Development of a Workflow-based Infrastructure for Managing and Executing Web Services**. *Universität Stuttgart, Fakultät Informatik, Diplomarbeit N.1997, 2002.*

LASKOWSKI, R. A., MACARTHUR, M. W., MOSS, D. S. e THORNTON, J. M. **PROCHECK: a Program to Check the Stereochemical Quality of Protein Structures**. *Journal of Appl. Cryst., Vol. 26, pp. 283-291, 1993.*

LEE, E. e NEUENDORFFER, S. **MoML - A Modeling Markup Language in XML - Version 4.0**. *Technical Memorandum ERL/UCB M 00/12. March 2000.*

LEMOS, M. **Workflow para Bioinformática**. *Tese de Doutorado. Defendida, Pontifícia Universidade Católica - PUC-Rio, Rio de Janeiro, 2004.*

LIEFELD, T., et. al. **GeneCruiser: a web service for the annotation of microarray data**. *Bioinformatics, 2005 21(18): pp. 3681-3682.*

LUCHTAN, M., et. al. **TcruziDB: an integrated Trypanosoma cruzi genome resource**. *Nucleic Acids Research, 2004, Vol. 32.*

LUDÄSCHER, B., ALTINTAS, I. , et. al. **Scientific Workflow Management and the Kepler System**. *September 2004; revised March 2005.*

MEYER, L. A. V., ROSSLE, S. C. , et. al. **Parallelism in Bioinformatics Workflows**. *VECPAR'2004: 6th International Conference, Valencia, Spain, Revised Selected and Invited Papers, 2005, Valencia. Lecture Notes in Computer Science. Vol. 3402. pp. 583-597.*

Microsoft. **Microsoft Message Queuing (MSMQ) Center**. *http://www.microsoft.com/windows2000/technologies/communications/msmq/ [Consultado em 02/03/2005].*

MULDER, N. J., ATTWOOD, T. K., et. al. **InterPro, progress and status in 2005**. *Nucleic Acids Res. 33 (Database Issue): pp. D201-5.*

myGrid Project. **The myGrid User Guide**. *Disponível em: http://prdownloads.sourceforge.net/mygrid-uk/mygrid-docs-0.6-beta1.zip.*

http://www.mygrid.org.uk.

NASCIMENTO, T. **Estudo sobre Esquemas de Banco de Dados Biológicos**. *Trabalho de conclusão de curso. DCC/UFRJ. Dezembro, 2004.*

NCBI. **GenBank Overview**. *Disponível em: http://www.ncbi.nlm.nih.gov/Genbank/index.html [Consultado em: 15/03/2005].*

OGSA-DAI Project *http://www.ogsadai.org.uk/.*

OMG. **Life Science Identifiers Specification**. *OMG Final Adopted Specification. May 2004.*

Oracle. **Oracle Workflow: Feature overview**. *Disponível em: http://www.oracle.com/technology/products/ias/workflow/.* [Consultado em 02/03/2005].

Oracle BPEL Process Manager. *Disponível em: http://www.oracle.com/technology/products/ias/bpel/in*

PAULUSBERGER, G. E. **BPEL-Editor**. *Department of Computer Science. University of Salzburg. July, 2004.*

PlasmoDB: The Plasmodium Genome Resource *http://plasmodb.org/.*

RÖSSLE, S., CARVALHO, P., DARDENNE, L. e BISCH, P., **Development of a Computational Environment for Protein Structure Prediction and Functional Analysis**. *In Proccedings of 2nd Brazilian Workshop on Bioinformatics (WOB), Rio de Janeiro, 2003.*

SALI, A. Seek Project. **MODELLER: A Program for Protein Structure Modeling Release 6**. *Rockefeller University.*

Seek Project. **SEEK: Science Environment for Ecological Knowledge**. *http://seek.ecoinformatics.org/.*

SINGH, M. P., VOUK, M. A., **Scientific Workflows: Scientific Computing Meets Transactional Workflows**. *In http://www.csc.ncsu.edu/faculty/mpsingh/papers/databases/workflows/ [Consultado em 20/02/2005].*

STANDLEY, D. M., et. al. **GASH: An improved algorithm for maximizing the number of equivalent residues between two protein structures**. *BMC Bioinformatics 2005, pp. 6:221.*

STEIN, L.. **Creating a Bioinformatics Nation**. *Nature, Vol. 417, 2002, pp. 119-120.*

STEVENS, R. e GOBLE, C. **myGrid: personalised bioinformatics on the information grid**. *Bioinformatics. Vol. 19, 2003, pp. i302-i304.*

SLIDEL, T. **Distributed Computing in Life Sciences**. *BioInformer, 1999.*

STOECKERT, C. J. **Functional genomics databases on the web**. *Cellular Microbiology. 2005, 7. pp. 1053 - 1059.*

STOFFEL, K., TAYLOR, M., HENDLER, J.. **Efficient Management of Very Large Ontologies**. *Proc. 14th Nat'l Conf. AI, MIT-AAAI Press, Menlo Park, Calif., 1997.*

TARGINO, R. S. **O Ambiente 10+C para definição e execução de workflows in silico através de serviços Web**. *Tese de Mestrado. COPPE/UFRJ. Novembro, 2004.*

TARGINO, R. S., CAVALCANTI, M. C. , MATTOSO, M. **An Environment to Define and Execute In-Silico Workflows Using Web Services**. *International Workshop on Data Integration in the Life Sciences - DILS 2005, pp. 288-291.*

Taverna Project. **XScufl Language Reference**. *Disponível em: http://taverna.sourceforge.net/api/org/embl/ebi/escience/scufl/XScufl.html [Consultado em: 08/05/2005].*

TEIXEIRA, F., CAVALCANTI, M. C. , BAIÃO, F., MEYER, L. A. V. C., RÖSSLE, S. C., BISCH, P. M. e MATTOSO, M. **Data Management via Web Services in Bioinformatics Workflows**. *Workshop on Bioinformatics, 2003, Macaé. Second Brazilian Workshop on Bioinformatics, 2003.*

VAN DER AALST, W.M.P., **Formalization and Verification of Event-driven Process Chains**. *Information and Software Technology 41 (1999) pp. 639-650.*

WANG, J. Z., et. al. **EST clustering error evaluation and correction**. *Bioinformatics 2004. Vol. 20 Issue 17. pp. 2973-2984.*

W3C. **Web Services Architecture. W3C Working Group**. *Note 11 February 2004. Disponível em: http://www.w3.org/TR/ws-arch/. [Consultado em: 20/02/2005].*

WILKINSON, M. D., LINKS, M. **BioMOBY: an Opensource Biological Web Services Proposal**. *Bioinformatics, Vol. 3, N. 4, 2002, pp. 331-341.*

WROE, C., et. al. **A Suite of DAM+OIL Ontologies to Describe Bioinformatics Web services and Data**. *International Journal of Cooperative Information Systems, Vol. 12, N. 2, 2003.*

9 ANEXOS

9.1 ANEXO 1: ARQUIVO DESCRITOR DO WORKFLOW SUBGARSA

a

a

9.2 ANEXO 2: ARQUIVO WSDL DO WORKFLOW SUBGARSA

9.3 ANEXO 3: ARQUIVO WSDL DO SERVIÇO WEB WSPHRED

a

a

9.4 ANEXO 4: ARQUIVO WSDL DO SERVIÇO WEB WSCAP3

a

9.5 ANEXO 5: ARQUIVO WSDL DO SERVIÇO WEB WSBLAST

a

9.6 ANEXO 6: ARQUIVO DESCRITOR DO WORKFLOW SUBGARSAFILTROBLAST

a

9.7 ANEXO 7: ARQUIVO WSDL DO WORKFLOW SUBGARSAFILTROBLAST

9.8 ANEXO 8: ARQUIVO WSDL DO SERVIÇO DE FILTRAGEM DE DADOS

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo