

# Linhas de Experimento: Reutilização e Gerência de Configuração em Workflows Científicos

Eduardo Ogasawara<sup>1</sup>, Leonardo Murta<sup>2</sup>, Cláudia Werner<sup>1</sup>, Marta Mattoso<sup>1</sup>

<sup>1</sup>Programa de Engenharia de Sistemas e Computação - COPPE / UFRJ  
Caixa Postal 68511, Rio de Janeiro, RJ, 21945-970  
{ogasawara,werner,marta}@cos.ufrj.br

<sup>2</sup>Instituto de Computação – Universidade Federal Fluminense  
Rua Passo da Pátria 156, Niterói, RJ, 24210-240  
leomurta@ic.uff.br

**Abstract.** *Over the last years, scientists have been using scientific workflows to build computer simulations to support the development of new theories. Nevertheless, due to their increasing usage, the scientific workflows conception and usage activities can no longer be performed in an ad-hoc manner. These activities can probably be improved if we apply some existing software engineering techniques, adapted from the software context to the scientific workflow context. Thus, we introduce the concept of experiment lines, which applies two traditional software engineering techniques to support the conception and usage of scientific workflows: reuse and configuration management.*

**Resumo.** *Nos últimos anos, cientistas vêm usando workflows científicos para apoiar o desenvolvimento de novas teorias via simulações computacionais. Entretanto, devido aos seus crescentes usos, as atividades de concepção e utilização de workflows científicos não podem ser mais feitas de modo ad-hoc. Estas atividades podem ser possivelmente melhoradas se forem aplicadas técnicas de engenharia de software existentes, adaptadas do contexto de software para o contexto de workflows científicos. Desta forma, este trabalho apresenta o conceito de linhas de experimento, que aplica duas técnicas tradicionais de engenharia de software na concepção e utilização de workflows científicos: reutilização e gerência de configuração.*

## 1. Introdução

Com a evolução da informática e o advento da internet, a ciência passou a explorar novas possibilidades de experimentação científica. Particularmente, permitiu-se que experimentos computacionais pudessem ser compartilhados entre as comunidades científicas. Visando ao apoio computacional desses experimentos, surgiram os denominados Sistemas de Gerência de Workflows Científicos (SGWfC), cujo objetivo é propiciar a orquestração de vários algoritmos e processamentos computacionais, fazendo-se valer de processamento paralelo e distribuído, bancos de dados, inteligência artificial, dentre outros, construindo, assim, um arcabouço para experimentação através de simulação [Taylor et al., 2006].

Apesar dos grandes avanços presentes nos SGWfC, estes ainda apresentam limitações no apoio à concepção de workflows científicos para experimentação e na utilização de workflows previamente elaborados. Desta forma, os cientistas tendem a redefinir workflows previamente construídos por outros cientistas e incorrer nos mesmos erros, por falta de uma sistemática na elaboração desses workflows. Além disto, os próprios SGWfC estão em constante evolução e workflows científicos, que outrora funcionavam perfeitamente, deixam de funcionar com a evolução destas ferramentas juntamente com suas bibliotecas de apoio.

Por outro lado, nas últimas quatro décadas, a engenharia de software tem estudado formas sistemáticas para apoiar na concepção e utilização de software. Técnicas como reutilização [Frakes et al., 2005] e gerência de configuração [Estublier, 2000] têm sido aplicadas com sucesso em diversas empresas, apoiando na diminuição do retrabalho e conseqüente aumento de produtividade e qualidade.

Desta forma, este trabalho tem por objetivo propor uma abordagem para apoiar as etapas de concepção e utilização de workflows científicos baseada em técnicas de engenharia de software, mais precisamente reutilização e gerência de configuração. Para isso, é proposto o conceito de linhas de experimento, fazendo-se uma analogia com o conceito de linhas de produto [Northrop, 2002], oriundo da disciplina de reutilização de software. Ademais, é levado em consideração o problema de evolução das linhas de experimento, com o apoio da disciplina de gerência de configuração.

Na seção 2, é feita uma análise sobre as limitações no apoio à concepção e utilização de workflows científicos para experimentação; na seção 3, é apresentada a proposta de projeto e desenvolvimento desta solução, aliada a um estudo de caso; e, na seção 4, são apresentadas as considerações finais e atividades futuras deste trabalho.

## **2. Análise dos SGWfC com uma perspectiva de engenharia de software**

Reutilização é uma disciplina da engenharia de software que prega a criação de sistemas de software a partir de software preexistente [Krueger, 1992]. Desta forma, ativos reutilizáveis podem ser desde componentes, passando pelo domínio de conhecimento, até se chegar a conceitos como ontologias [Frakes et al., 2005]. O emprego destas técnicas está relacionado tanto a ter primitivas para a construção de ativos de reutilização, na qual o foco está nos produtores, quanto à exploração destes ativos de reutilização, na qual o foco está nos consumidores.

Em Engenharia de Software, a necessidade de gerência de configuração<sup>1</sup> é um consenso [Estublier, 2000], tanto que diversos modelos de qualidade, como MPS.BR e CMMI, incorporam estes conceitos e os requerem em seus níveis mais baixos de maturidade, ou seja, trata-se de um passo básico para o sucesso de desenvolvimento de software.

O apoio à reutilização e gerência de configuração é um item importante a ser explorado em sistemas de workflows científicos. Fazendo um paralelo com estes conceitos e aplicando estas idéias a workflows científicos, podem-se classificar os esforços

---

<sup>1</sup> Gerência de Configuração é a disciplina responsável pelo controle da evolução de software [Dart, 1991].

atuais em workflows científicos no apoio à concepção (i) de workflows e (ii) de componentes. Estes esforços serão mais bem detalhados nas seções a seguir.

## 2.1. Apoio à concepção de workflows

A experimentação por simulação é uma das formas adotadas no método científico no apoio à criação de novas teorias. Um workflow científico representa o encadeamento de processos que transformam dados visando uma experimentação por simulação. A Figura 1 mostra exemplos de workflows de bioinformática, segundo a notação do SGWfC Taverna [Oinn et al., 2004]. Nesta notação, processos estão representados por um pequeno quadrado ao redor, dados de entrada dos processos estão representados por um triângulo voltado para cima e os de saída, por um triângulo voltado para baixo. Para trazer uma uniformidade à nomenclatura, denominar-se-á atividade a cada processo. Em geral, as atividades de um workflow representam pacotes ou programas de software específicos do domínio da aplicação. Estas atividades são ligadas direcionalmente entre si, representando um encadeamento de ações e, ao mesmo tempo, uma relação de dependência. Desta forma, atividades possuem portas de entrada e portas de saída que podem ser utilizadas para se conectar com outras atividades. Há trechos do workflow que permitem execuções em paralelo e outros que funcionam como pontos de concentração ou consolidação dos resultados intermediários para dar continuidade à execução.

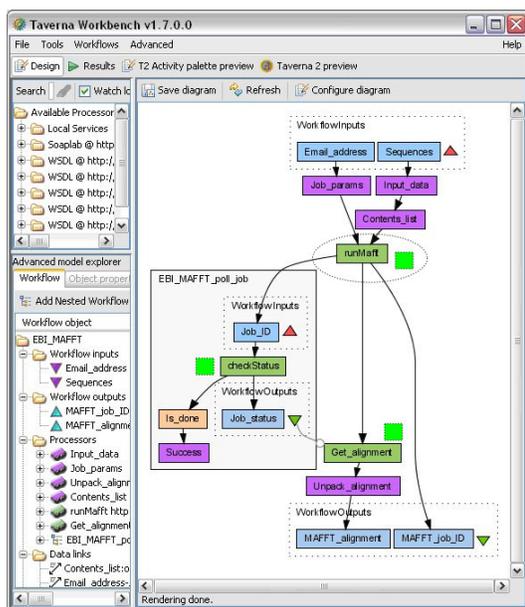


Figura 1 – Workflow extraído do site [myExperiment](#) para análise de seqüências genéticas com o software *MAFFT*

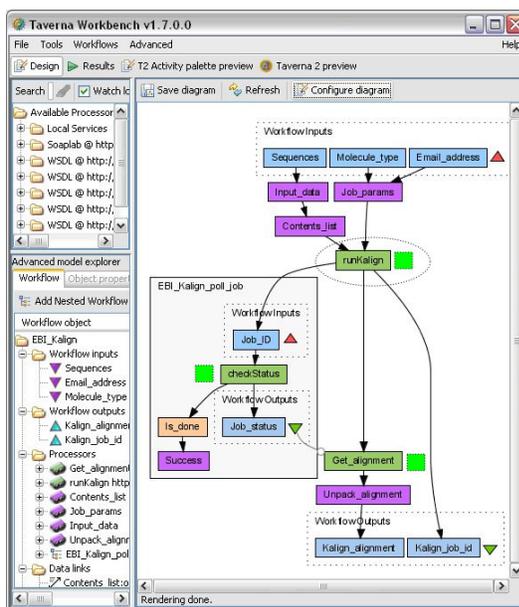


Figura 2 - Workflow extraído do site [myExperiment](#) para análise de seqüências genéticas com o software *KAlign*

No que tange à concepção de workflows, pode-se perceber que o desenvolvimento de workflows é, normalmente, sob demanda. As principais dificuldades durante a composição de um workflow estão na seleção das atividades e nos seus encadeamentos corretos. Algumas ferramentas oferecem mecanismos de buscas, como, por exemplo, ontologias, para facilitar a escolha de atividades [Oliveira et al., 2008].

O processo de encadeamento apresenta ainda alguns pontos em aberto. O conhecimento de quais atividades podem ser ligadas a outras é ainda tácito. É necessário exercitar vários exemplos para ter experiência no encadeamento de atividades e na escolha de adaptadores, que são elementos usados para uniformizar a entrada de dados requerida por uma atividade. Se a definição de quais atividades faz sentido serem encadeadas entre si não estiver registrada em documentos, esta informação acaba sendo perdida, visto que não há nenhum outro mecanismo que transmita esta informação entre as pessoas responsáveis pela composição do workflow.

Existem trabalhos que concentram esforços em apoiar a busca de workflows previamente elaborados por meio de um repositório, como o <sup>my</sup>Experiment [Roure et al., 2007][Goderis et al., 2008], que facilita a publicação e o compartilhamento de workflows. Eles trazem resultados no que se refere a entender como os cientistas atuam na busca de workflows. Entretanto, apesar da definição de um ambiente ou repositório para busca de workflows, é somente possível recuperar workflows completos, visto que não há nenhum apoio no sentido de se reaproveitar partes de workflows previamente elaborados, tendo a restrição adicional de somente apoiar workflows científicos desenvolvidos para o sistema Taverna. Além disso, o referido trabalho não propõe o estabelecimento formal de uma ligação entre workflows semelhantes, ou seja, não é difícil encontrar no site do <sup>my</sup>Experiment exemplos de workflows praticamente iguais, nos quais há apenas a variação de uma atividade, não havendo ligação formal entre elas.

Durante a experimentação, pode ser necessário realizar alterações no workflow. Estas alterações podem ter duas origens. A primeira seria através da variação de parâmetros [Goderis et al., 2008] (por exemplo, a seleção de uma outra base de dados para a experimentação, ou a definição de um outro local para a chamada de serviço remoto da atividade). A segunda seria pela necessidade de se usar outro algoritmo para uma mesma atividade. Isto pode implicar a inclusão ou exclusão de atividades. Normalmente, estas alterações são feitas usando uma cópia da configuração do workflow e edição dos seus parâmetros. Este tipo de abordagem traz limitações no que se refere à escalabilidade e usabilidade [Taylor et al., 2006]. Para tornar a compreensão deste problema mais nítida, a Figura 2 apresenta um exemplo muito semelhante ao da Figura 1, no qual o objetivo é realizar alinhamento genético, onde há apenas a variação da atividade (envolvida por uma elipse tracejada) responsável pela operação de alinhamento genético. Nos SGWfC atuais, qualquer mudança representa um novo workflow ou uma versão como, por exemplo, no Vistrails [Callahan et al., 2006]. Neste sentido, embora somente haja uma variação de atividade para desempenhar uma mesma função, não há nenhuma ligação entre estas definições, podendo gerar retrabalho na manutenção das mesmas e dificuldades na busca destas variações.

Outra deficiência dos SGWfC no apoio à concepção de workflows está relacionada à ausência de gerência de configuração. Alguns SGWfC, como o Vistrails [Callahan et al., 2006], possuem o registro de alterações realizadas no workflow, mas este tipo de facilidade não resolve os problemas relacionados à gerência de configuração, pois não há a separação clara entre uma versão de produção (que pode ser reutilizada por outros cientistas) e uma versão em desenvolvimento (ainda em elaboração, sem garantias de qualidade). Além disto, não é registrada uma relação de dependência entre as versões das atividades que compõem um determinado workflow, ou seja, o workflow X na

versão 1.0 pode necessitar das atividades  $Y_1$ ,  $Y_2$  e  $Y_3$  em suas versões 1.2, 2.5 e 4.0, respectivamente.

Para se apoiar a gerência de configuração [Estublier, 2000], seria desejável:

- Repositório com acesso controlado, onde os workflows possam ser armazenados, e onde se possam registrar quais versões do workflow são estáveis e quais estão em fase de desenvolvimento;
- Mecanismos para representar ou armazenar as versões das atividades utilizadas durante a composição de um determinado workflow;
- Adição de características ao conceito de espaço de trabalho (*workspace*) que apoiem tanto o desenvolvimento de um workflow como também a sua posterior publicação junto ao repositório;
- Gestão de uso de atividades, ou seja, quais atividades estão sendo mais utilizadas e quais workflows seriam afetados durante uma possível evolução de uma atividade.

Além da gerência de configuração, outro ponto importante é apoiar, também, a presença de uma biblioteca de componentes [Werner et al., 2007]. Algumas ferramentas disponibilizam repositórios globais de workflows, como o <sup>my</sup>Experiment. Entretanto, pretende-se, neste trabalho, armazenar workflows que possam ser vistos como um componente, ou seja, workflows que sejam usados como subworkflows, e as linhas de experimento propriamente ditas. Somente os subworkflows e linhas de experimento, devidamente testados e aprovados dentro de uma organização, deveriam fazer parte da biblioteca. Documentações e casos de uso, tanto para subworkflows como também para linhas de experimento, fariam parte da biblioteca. É importante notar que este tipo de problema é o mesmo que existe para software, ou seja, existem diversos componentes presentes na Internet, mas é aconselhável que se trabalhe com os componentes que passaram por algum critério de certificação da organização, antes de serem incluídos na biblioteca.

## 2.2. Apoio à concepção de componentes

Um componente é uma unidade de composição que contratualmente especifica uma interface e explicita suas dependências. Um componente de software pode ser distribuído independentemente e pode ser usado numa composição por componentes de terceiros [Szyperski, 1997]. No contexto do desenvolvimento baseado em componentes, estes são elementos fundamentais, que servem como unidade de encapsulamento e possuem interfaces bem definidas, podendo ser reutilizados ou substituídos, o que eleva a produtividade e qualidade [Sametinger, 2001].

Existem diversos trabalhos [Gary et al., 1999][Fusaro et al., 1998] que comparam uma atividade do workflow a um componente dentro do workflow. Neste contexto, pode-se imaginar que um componente do workflow pode ser uma atividade direta do sistema, como um objeto que executa uma determinada funcionalidade, ou ser responsável por uma chamada a um subworkflow, através de estratégias de composição de workflows. Neste sentido, os SGWfC atualmente procuram atender suficientemente bem estas definições, tanto no que tange ao apoio a componentes como sendo atividades

de execução de programas ou de chamadas a serviços remotos, como por meio da composição de workflows em subworkflows. Em [Bowers et al., 2006], há o emprego de *frameworks* que permitem a composição de workflows sobre subworkflows. Não há relação formal entre os workflows derivados e tampouco há a preocupação de garantir a integridade do workflow padrão e suas partes.

Quanto ao apoio a componentes num workflow, vale ressaltar que os SGWfC fazem uso de mecanismos que apóiam o encadeamento de diferentes atividades, de modo que os SGWfC possam focar na orquestração das atividades sem impor limitações sobre o conjunto de elementos que estão sendo orquestrados. Além disso, outra característica chave é a capacidade de inclusão de componentes granulares, também conhecidos como *plug-ins* [Taylor et al., 2006]. Essa característica consiste na capacidade de aumentar os recursos dos SGWfC, permitindo que desenvolvedores de workflows possam expandir funcionalidades ou programar novos algoritmos ou rotinas. Neste sentido, há uma preocupação grande em fornecer componentes básicos [Altintas et al., 2004][Oinn et al., 2004], como, por exemplo, alguns pacotes estatísticos, atividades para cópias de arquivos, objetos que permitam chamadas de *Serviços Web*, adaptadores para troca de informações em XML, apoio ao desenvolvimento de *plug-ins* e apoio à composição de workflows.

Uma deficiência dos SGWfC neste apoio, entretanto, está no fato de não haver uma relação de dependência entre os workflows e as versões das atividades utilizadas, ou seja, é relativamente comum que um workflow deixe de funcionar a partir do momento em que uma nova versão do SGWfC é lançada, contendo novas versões para seus componentes básicos.

Desta forma, em termos gerais, qualquer proposição para apoiar as etapas de concepção e utilização de workflows científicos não pode perder as características que permitem a composição de atividades tanto como uma unidade de processamento individual quanto como um subworkflow. Além disto, precisa também tratar a questão da gerência de configuração.

### **3. Linhas de experimento**

Conforme dito anteriormente, o workflow científico serve como uma abstração para representar o encadeamento de atividades para experimentação. Entretanto, em diversos casos, pode-se separar o procedimento básico, ou processo básico, da escolha de um determinado algoritmo ou técnica. De acordo com a escolha de um determinado componente, algumas etapas do processo sofreriam variações, seja por variabilidade de parâmetros, seja pela definição de atividades especializadas para executar uma determinada manipulação diferenciada, de acordo com alguma característica do parâmetro em questão, como observado na Figura 1 e na Figura 2. Vale notar que o objetivo era realizar alinhamento genético, mas, dada a presença de diferentes algoritmos, foram desenvolvidos dois workflows diferentes para se executar o que seria uma mesma atividade do workflow.

É conveniente que se tenha a definição do workflow padrão, ou seja, a apresentação do que precisa ser feito em cada etapa do workflow. Neste sentido, está se falando num modelo de atividade, dentro deste modelo de atividades, quais possíveis atividades reais ou subworkflows reais poderiam ser utilizados para desempenhar aquela determi-

nada operação. Para cada atividade, deve-se indicar a ação que esta deve executar e apenas definir a interface de entrada e de saída de dados. Neste momento, estar-se-á indicando um componente, ou seja, um contrato de entrada e de saída para as informações que, uma vez respeitado, produza uma ação ou resultado processado [Szyperski, 1997]. De acordo com a indicação no workflow padrão, estas atividades têm a possibilidade de apresentar pontos de variações para acomodar diferentes escolhas de algoritmos ou configurações diferenciadas. Este workflow padrão pode ser obtido através de uma modelagem do processo [Barreto et al., 2007].

### 3.1. Analogia a linhas de produto

A abordagem de linhas de experimento, conforme apresentado a seguir, foi inspirada no conceito de linhas de produto em Engenharia de Software [Northrop, 2002][Frakes et al., 2005]. Uma linha de produto representa um conjunto de sistemas que compartilham estruturas comuns em termos de aplicabilidade ou em termos de segmentos de mercado. Para se obter sucesso nesta abordagem, é necessário esforço de planejamento, bem como a estruturação dos elementos a sofrer técnicas de reutilização.

Trazendo esta forma de pensar para o domínio dos workflows científicos, pode-se estabelecer uma arquitetura para a composição de workflows segundo a definição de linhas de produto, na qual se tem a possibilidade de formalizar o workflow padrão para um determinado tipo de experimentação.

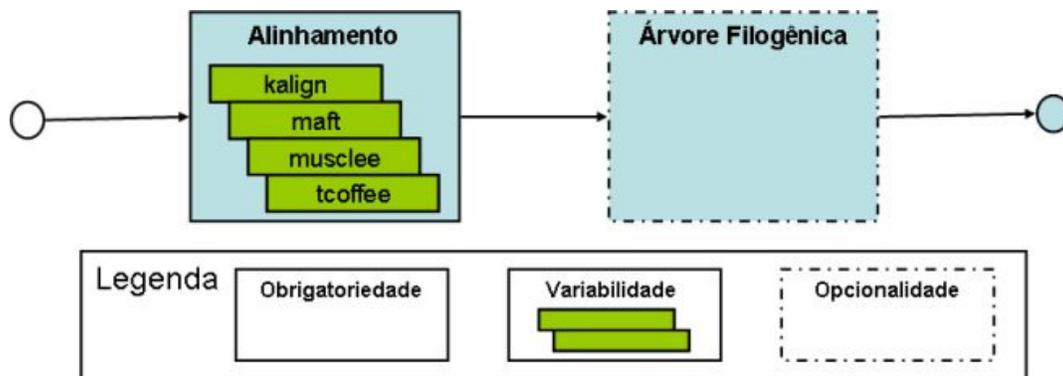
Uma linha de experimento poderia ser entendida como sendo o estabelecimento de um workflow padrão que contém um conjunto de modelos de atividades encadeadas entre si. Nestes modelos de atividades são feitas formalizações das portas de entradas e saídas, estabelecendo um comportamento de componente. Como também é definido numa linha de produto, para cada modelo de atividade devem-se indicar especificamente quais são as atividades que podem ser utilizadas na experimentação. Estas atividades podem ser tanto algoritmos e componentes pré-existentes no sistema gerenciador de workflow científico quanto subworkflows, sendo que esta última possibilidade permite os ajustes nas entradas e saídas através de adaptadores [Bosch, 2000].

A partir da linha de experimento, pode-se criar um workflow real para experimentação, que será montado via um recorte do workflow padrão com todas as opções de componentes definidas e a inclusão ou não de trechos de opcionalidade. A definição destes pontos de opcionalidade requer uma validação formal [Taylor et al., 2006], visto que a não inclusão de um ponto de opcionalidade precisa ainda garantir a estabilidade do workflow a ser montado a partir da linha de experimento.

A Figura 3 apresenta a definição de um workflow padrão para o alinhamento de seqüência genética. Na primeira etapa, tem-se uma atividade obrigatória de variabilidade, na qual se realiza a análise de seqüência genética, que pode ser desempenhada por um dos programas (atividades) *kalign*, *maft*, *muscle* ou *tcoffee*. A segunda etapa seria uma atividade opcional, onde teríamos a geração de uma árvore filogenética. Os exemplos das Figuras 1 e 2 seriam recortes deste workflow padrão.

É importante notar as diferenças entre esta abordagem e as abordagens detalhadas na Seção 2. Naquelas, a reutilização acontece ao se recuperar um workflow previamente elaborado e modificar os parâmetros de uma ou mais atividades. Na abordagem

de linhas de experimento, além de se poder realizar esta possibilidade, a reutilização acontece no nível do workflow como um todo, possibilitando que, a partir de um conjunto de critérios, seja possível realizar um recorte sobre o workflow padrão (i.e., linha de experimento), gerando um workflow específico (i.e., experimento específico). Esta definição de linhas de experimento está para linhas de produto assim como o experimento específico derivado de uma linha de experimento está para um produto específico, derivado de uma linha de produto.



**Figura 3 – Workflow padrão para seqüência genética**

No que tange ao controle da evolução, uma linha de experimento completa e seus respectivos recortes devem estar depositados num repositório controlado. Para se evoluir uma linha de experimento ou um de seus recortes, deve-se proceder à criação de ramos (*branches*). A criação do ramo permite que a evolução seja identificada unicamente e depois migrada para as demais instâncias do workflow em questão (recortes ou linha de experimentos em si).

### 3.2. Desenvolvimento para reutilização e com reutilização

Em qualquer abordagem de reutilização, deve-se analisar o tema tanto a partir das ações para a geração de ativos de reutilização (i.e., desenvolvimento *para* reutilização) quanto das ações realizadas no sentido de se fazer uso dos ativos de reutilização (i.e., desenvolvimento *com* reutilização).

No que tange ao desenvolvimento *para* reutilização, referente à etapa de concepção da linha de experimento, é necessária uma organização maior da definição da experimentação. Há um trabalho inicial maior de se compor o workflow padrão da linha de experimento, com a indicação de todas as opções possíveis para cada etapa, até que seja armazenado no repositório.

No que tange ao desenvolvimento *com* reutilização, não se tem mais a necessidade de se procurar por diversas atividades na ferramenta e na Internet; faz-se apenas um recorte da linha de experimento para derivar uma determinada experimentação. Tais recortes possibilitam o registro de quais atividades são mais utilizadas dentro de atividades de variabilidade da linha de experimento. Desta forma, pode-se ter o controle de quais atividades são pouco úteis, indicando candidatas a serem descartadas da biblioteca, o que seria um raciocínio semelhante à gestão de componentes em uma biblioteca de componentes [Werner et al., 2007]. Por outro lado, ao tentar gerar um workflow a partir de uma linha de experimento, é possível identificar atividades que ainda não estão sendo

contempladas pela linha de experimento. Este cenário pode motivar a evolução da linha de experimento, adicionando essas atividades como opcionais ou variáveis.

#### 4. Conclusão

Os workflows científicos vieram trazer organização para as experimentações baseadas em simulações. Embora ainda existam diversos assuntos em aberto nesta área, houve grande progresso. O conceito de workflow científico trouxe certo grau de reutilização às experimentações, possibilitando a reexecução dos experimentos. Entretanto, apesar de ser possível a estruturação e distribuição dos experimentos, faltam mecanismos que impeçam a desorganização de diferentes experimentos em inúmeros workflows de experimentação que, rapidamente, quando misturados, podem perder a semântica. Ademais, falta formalismo para a separação entre uma configuração específica e um processo padrão de experimentação, dentre outras limitações.

A partir de uma sinergia com a Engenharia de Software e seus conceitos, como reutilização e gerência de configuração, pode-se entender que há a necessidade de um segundo nível de reutilização para se apoiar sistemas de workflows científicos. É preciso formalizar melhor os ambientes de experimentação, separando a definição do workflow padrão de suas variações ou configurações específicas.

Este trabalho faz parte do projeto de Gerência de Experimentos Científicos em Larga Escala (GEXP)<sup>2</sup> e contribui por discutir a possibilidade de se abordar uma solução para este problema inspirada em técnicas orientadas a linhas de produto, através de linhas de experimento, para apoiar a reutilização em workflows científicos e comparar esta solução a outros trabalhos relacionados.

Contudo, este trabalho se limitou a atuar em controle de versões no que tange à gerência de configuração. A gerência de configuração fornece mecanismos adicionais, como controle de solicitações (*issue tracking*) e gerenciamento de construção (*build management*), que possivelmente seriam de grande valia para apoiar no controle da evolução de workflows científicos.

Atualmente estamos desenvolvendo um protótipo em Java para gerenciamento de linhas de experimento, de modo a permitir uma posterior avaliação dos conceitos apresentados. Além disso, pretendemos adaptar técnicas de mineração de dados desenvolvidas no sistema de recomendação de processos científicos sobre o SGWf Vistrails [Oliveira et al., 2008], de modo a reaproveitar os workflows já elaborados para propor a criação semi-automática de linhas de experimento por construção *bottom-up*.

#### 5. Agradecimentos

Os autores agradecem ao CNPq pelo apoio financeiro.

#### 6. Referências

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., & Mock, S. (2004). Kepler: an extensible system for design and execution of scientific workflows. Em *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*.

---

<sup>2</sup> <http://gexp.nacad.ufjf.br>

- Barreto, A., Rocha, A., & Murta, L. (2007). Uma Abordagem de Definição de Processos de Software Baseada em Reutilização. *ProQuality*, vol. 3.
- Bosch, J. (2000). *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. Addison-Wesley Professional.
- Bowers, S., Ludascher, B., Ngu, A. H. H., & Critchlow, T. (2006). Enabling Scientific Workflow Reuse through Structured Composition of Dataflow and Control-Flow. Em *Proceedings of the 22nd International Conference on Data Engineering Workshops*. IEEE Computer Society.
- Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., & Vo, H. T. (2006). Vis-Trails: visualization meets data management. Em *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. Chicago, IL, USA: ACM.
- Dart, S. (1991). Concepts in configuration management systems. Em *Proceedings of the 3rd international workshop on Software configuration management*. Trondheim, Norway: ACM.
- Estublier, J. (2000). Software configuration management: a roadmap. Em *Proceedings of the Conference on The Future of Software Engineering*. Limerick, Ireland: ACM.
- Frakes, W., & Kyo Kang. (2005). Software reuse research: status and future. *Software Engineering, IEEE Transactions on*, 31(7).
- Fusaro, P., Visaggio, G., & Tortorella, M. (1998). REP - ChaRacterizing and Exploiting Process Components: Results of Experimentation. Em *Proceedings of the Working Conference on Reverse Engineering (WCRE'98)*. IEEE Computer Society.
- Gary, K., & Lindquist, T. (1999). Cooperating process components. Em *Computer Software and Applications Conference, 1999. COMPSAC '99. Proceedings. The Twenty-Third Annual International*.
- Goderis, A., De Roure, D., Goble, C., Bhagat, J., Cruickshank, D., Fisher, P., et al. (2008). Discovering Scientific Workflows: The myExperiment Benchmarks. .
- Krueger, C. W. (1992). Software reuse. *ACM Comput. Surv.*, 24(2).
- Northrop, L. (2002). SEI's software product line tenets. *Software, IEEE*, 19(4).
- Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., et al. (2004). *Taverna: a tool for the composition and enactment of bioinformatics workflows* (Vol. 20). Oxford Univ Press.
- Oliveira, F., Murta, L., Werner, C., & Mattoso, M. (2008). Using Provenance to Improve Workflow Design. *Second International Provenance and Annotation Workshop - IPAW* .
- Roure, D. D., Goble, C., & Stevens, R. (2007). Designing the myExperiment Virtual Research Environment for the Social Sharing of Workflows. Em *Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*. IEEE Computer Society.
- Sametinger, J. (2001). *Software Engineering with Reusable Components* (1st ed.). Springer.
- Szyperski, C. (1997). *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Professional.
- Taylor, I. J., Deelman, E., Gannon, D. B., & Shields, M. (2006). *Workflows for e-Science: Scientific Workflows for Grids* (1st ed.). Springer.
- Werner, C., Murta, L., Lopes, M., Dantas, A., Lopes, L., Fernandes, P., et al. (2007). Brechó: Catálogo de Componentes e Serviços de Software. *XXI SBES*, (XIV Sessão de Ferramentas).