

Composicionalidade e Reuso em Workflows Científicos com Propriedades Não-Funcionais

Vívian Medeiros, Antônio Tadeu Azevedo Gomes

Laboratório Nacional de Computação Científica (LNCC) – Petrópolis, RJ – Brasil
{vivian, atagomes}@lncc.br

Abstract. *In this paper we propose and illustrate the use of connectors as first-class constructs in workflow description languages. The aim is to demonstrate that it is possible to improve the reuse and compositionality of configurations involving non-functional properties related with the communication between workflow tasks.*

Resumo. *Neste artigo propomos e ilustramos o uso de conectores como construções de primeira classe em linguagens de descrição de workflows. O objetivo é demonstrar ser possível melhorar o reuso e a composicionalidade das configurações dos mecanismos que lidam com propriedades não-funcionais ligadas à comunicação entre tarefas de um workflow.*

1. Introdução

O aprimoramento da qualidade na execução de workflows tem sido objeto de vários trabalhos relacionados a sistemas gerenciadores de workflows científicos [Ludäscher et al., 2006; Fahringer et al., 2007; Zhao et al., 2008]. Em geral, tais trabalhos advogam a introdução, nesses sistemas, de mecanismos que lidam com propriedades não-funcionais de interesse típico para workflows científicos, como tolerância a falhas, rastreamento da proveniência de dados, desempenho e interoperabilidade.

A hipótese que pretendemos confirmar neste trabalho é que os sistemas gerenciadores existentes, como apresentado na Seção 2, são limitados no que se refere à expressividade das linguagens de descrição de workflows por eles oferecidas. Em particular, a configuração dos mecanismos responsáveis pelo tratamento das propriedades não-funcionais de interesse em um workflow científico ou inexistem nesses sistemas (que tratam essas propriedades de forma transparente ao usuário), ou é concentrado somente na descrição das tarefas (componentes computacionais) do workflow.¹ A transparência “não-seletiva” ao usuário torna mais simples a especificação de um workflow, porém oferece menor flexibilidade na configuração desses mecanismos. O foco da configuração desses mecanismos em tarefas, por sua vez, compromete o reuso e a composicionalidade dessas configurações, pois dificulta o encapsulamento das estratégias de comunicação e coordenação empregadas *entre* tarefas, e é sobre essas estratégias que vários desses mecanismos tipicamente atuam.

Nesse contexto, propomos o uso de *conectores* como construções de primeira classe em linguagens de descrição de workflows. O conceito de conector é bastante explorado na área de arquitetura de software para modelar tanto tipos quanto instâncias de interações entre componentes computacionais e regras que governam essas interações [Clements et al., 2007]. Na Seção 3, apresentamos uma linguagem de descrição de workflows com suporte à especificação

¹Tomamos como condição para nossa hipótese que um workflow em si também é visto pelo sistema gerenciador como uma tarefa composta por outras tarefas aninhadas.

de conectores com o objetivo de demonstrar que é possível melhorar o reuso e a composicionalidade das configurações dos mecanismos que lidam com propriedades não-funcionais ligadas à comunicação entre tarefas de um workflow científico. Na Seção 4 discutimos o estado atual de nosso trabalho e as perspectivas de desenvolvimento futuro no mesmo.

2. Trabalhos Relacionados

Vários trabalhos abordam o tratamento de propriedades não-funcionais em workflows científicos na literatura. Yu e Buyya [2006] expõem uma coletânea ampla – embora não exaustiva – de sistemas gerenciadores de workflows científicos. Porém, por questões de espaço, apresentamos somente alguns daqueles que trazem claramente algum suporte à configuração de mecanismos de tratamento dessas propriedades através de linguagens de descrição de workflows.

Os sistemas Gridbus [Yu e Buyya, 2004], QWE [Brandic et al., 2006] e ASKALON [Fahringer et al., 2007] definem linguagens de descrição de workflows com construções para a especificação de parâmetros de desempenho e de tolerância a falhas. Os sistemas Triana [Taylor et al., 2004], Kepler [Ludäscher et al., 2006] e Taverna [Oinn et al., 2006] oferecem suporte à configuração de mecanismos de rastreamento de proveniência de dados e de tolerância a falhas. Zhao et al. [2008] integram o sistema Swift [Zhao et al., 2007] com mecanismos para escalonamento eficiente de tarefas e rastreamento de proveniência de dados.

Em linhas gerais, percebe-se que os trabalhos apresentados oferecem algum suporte à composicionalidade de configuração dos mecanismos de tratamento de propriedades não-funcionais. Contudo, esse suporte é provido de forma *ad hoc* e sem a preocupação com o desenvolvimento *para* reuso de configurações. Nesse sentido, é importante destacar alguns trabalhos que ressaltam a importância de se oferecer primitivas para reuso de (partes de) workflows em sistemas gerenciadores de workflows [Qin et al., 2008; Gil et al., 2009]. Porém, nesses trabalhos o suporte para reuso é associado unicamente à configuração das tarefas, não havendo suporte para reuso de configuração dos mecanismos de tratamento de propriedades não-funcionais nas interações entre tarefas.

O uso de conectores e sua associação a propriedades não-funcionais em workflows organizacionais já foi explorado na literatura. Por exemplo, Han e Shim [2000] utilizam conectores para compartilhamento de dados entre tarefas, bem como para o registro de eventos e a geração de estatísticas relacionadas a esses dados. Contudo, conectores não são tratados como entidades de primeira classe, limitando a possibilidade de desenvolvimento para reuso.

3. Conectores em Workflows Científicos

A abordagem apresentada neste trabalho introduz *conectores* como construções de primeira classe na linguagem de descrição de workflows baseada em XML oferecida pelo framework CSBase [de Lima et al., 2006].² Nesta proposta, tarefas comunicam-se entre si obrigatoriamente por meio de conectores. Tarefas e conectores possuem interfaces denominadas, respectivamente, de *portas* e *papéis*. A composição de um workflow envolve a ligação de portas de tarefas a papéis de conectores.

A linguagem considera a existência de dois tipos de usuários no processo de especificação de workflows científicos: *projetistas* e *cientistas*.

O *projetista* define tipos de tarefas e de conectores (desenvolvimento *para* reuso). Tipos de tarefas são usados para associar o conceito abstrato de tarefa no workflow com sua

²Esse framework oferece também uma ferramenta gráfica que auxilia na edição de descrições de workflows.

implementação concreta (um programa ou ferramenta, por exemplo). Tipos de conectores são usados para associar o conceito abstrato de conector no workflow com os mecanismos concretos que implementam as estratégias de comunicação e coordenação entre tarefas (p.ex. um *pipe* de caracteres, o transporte de um arquivo ou a automação de um fluxo). Por questões de espaço, não expomos neste artigo como tais tipos podem ser definidos em nossa linguagem.

O *cientista* descreve workflows a partir de relações entre instâncias de tipos de tarefas e conectores (desenvolvimento *com reuso*). Como exemplo, a Figura 1 ilustra uma simplificação do workflow MHOLine [Cavalcanti et al., 2003] descrita através do uso de nossa linguagem.

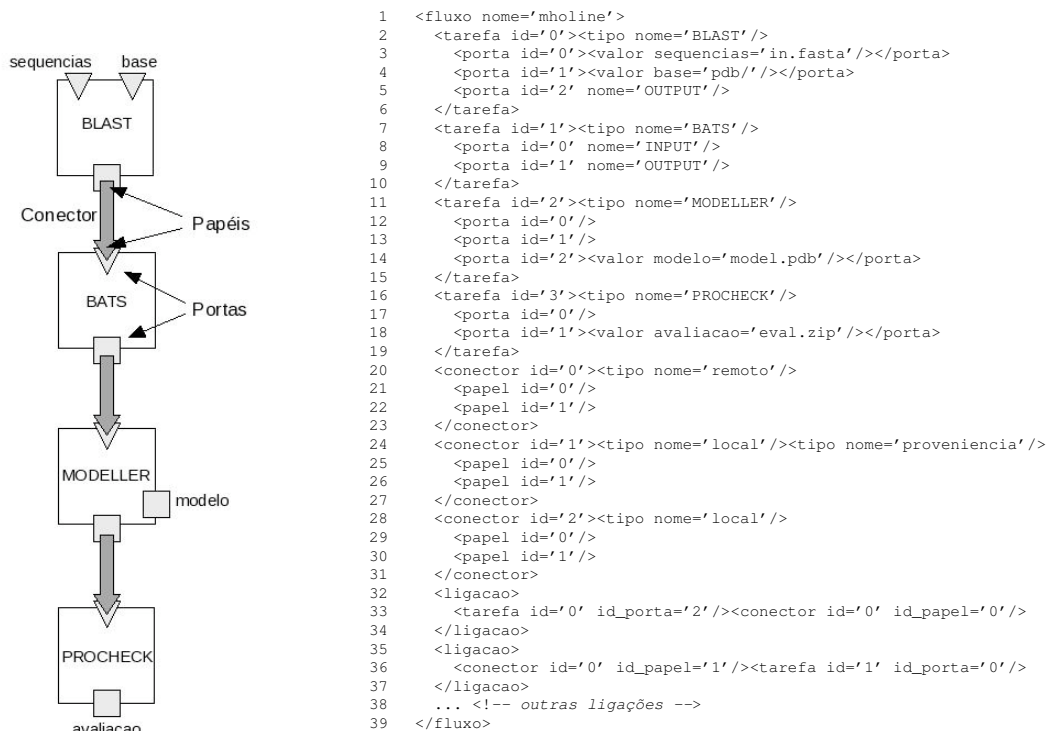


Figura 1. Exemplo de workflow com conectores.

A partir da Figura 1 podemos observar duas características de nossa abordagem. Em primeiro lugar, a configuração do conector 1 (linhas 24-27) compõe dois tipos de mecanismos distintos: de distribuição de tarefas (tipo `local`) e de rastreamento de proveniência (tipo `proveniencia`). A composição entre esses tipos de mecanismos é independente da definição das tarefas. Em segundo lugar, os tipos de conectores `local` e `remoto` (linhas 20, 24 e 28) definem estratégias de distribuição de tarefas nos recursos computacionais disponíveis para a execução do workflow. A informação provida por esses dois tipos é útil aos mecanismos de escalonamento de tarefas usados pelo sistema gerenciador de workflows. Nesse caso, a identificação da *afinidade* de uma tarefa com um recurso [Brandic et al., 2006] está relacionada às características das tarefas, mas o desejo do usuário que as mesmas executem ou não em um mesmo recurso não pode ser declarado, de forma reusável e sem quebrar a composicionalidade do workflow, nem nas propriedades das tarefas nem nas do workflow como um todo.

4. Estado atual e Perspectivas

Neste trabalho apresentamos a hipótese de que os sistemas gerenciadores de workflows científicos existentes têm expressividade limitada no que se refere à configuração e composição

dos mecanismos responsáveis pelo tratamento das propriedades não-funcionais de interesse nesses workflows. Nesse contexto, propusemos e ilustramos o uso de conectores em linguagens de descrição de workflows, objetivando demonstrar que é possível melhorar o reuso e a composicionalidade das configurações desses mecanismos quando as propriedades não-funcionais correspondentes estão ligadas à comunicação entre as tarefas de um workflow.

Este trabalho, ainda em estágio preliminar, será aprofundado em diferentes frentes. Em primeiro lugar, precisamos definir mais claramente quais mecanismos devem poder ser definidos e configurados por meio de conectores. Em segundo lugar, precisamos testar nossa abordagem com outros workflows, identificando novas oportunidades e desafios no uso de conectores. Por fim, pretendemos explorar o conceito de *estilos arquiteturais* [Clements et al., 2007] em complemento a conectores para melhorar o nível de reuso de especificações de workflows.

Referências

- Brandic, I., Pllana, S., e Benkner, S. (2006). An approach for the high-level specification of QoS-aware grid workflows considering location affinity. *Scientific Programming*, 14:231–250.
- Cavalcanti, M. C., Baião, F., Rössle, S. C., Bisch, P. M., Targino, R., Pires, P. F., Campos, M. L., e Mattoso, M. (2003). Structural genomic workflows supported by web services. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pages 45 – 49.
- Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., e Stafford, J. (2007). *Documenting Software Architectures: Views and Beyond*. Addison Wesley.
- de Lima, M. J., Ururahy, C., Lucia de Moura, A., Melcop, T., Cassino, C., dos Santos, M. N., Silvestre, B., Reis, V., e Cerqueira, R. (2006). CSBase: A framework for building customized grid environments. In *Proc. 15th IEEE WETICE*, pages 187–194, Washington, DC, USA. IEEE ComSoc.
- Fahringer, T., Prodan, R., Duan, R., Hofer, J., Nadeem, F., Nerieri, F., Podlipnig, S., Qin, J., Siddiqui, M., Truong, H.-L., Villazon, A., e Wiczorek, M. (2007). ASKALON: A development and grid computing environment for scientific workflows. In *Workflows for e-Science*, pages 450–471. Springer.
- Gil, Y., Groth, P., Ratnakar, V., e Fritz, C. (2009). Expressive reusable workflow templates. In *Proc. 5th IEEE E-SCIENCE*, pages 344–351, Washington, DC, USA. IEEE ComSoc.
- Han, D. e Shim, J. (2000). Connector oriented workflow system for the support of structured ad hoc workflow. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 6, page 6029, Los Alamitos, CA, USA. IEEE Computer Society.
- Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E. A., Tao, J., e Zhao, Y. (2006). Scientific workflow management and the Kepler system. *CCPE*, 18(10):1039–1065.
- Oinn, T., Greenwood, M., Addis, M., Alpdemir, N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M. R., Senger, M., Stevens, R., Wipat, A., e Wroe, C. (2006). Taverna: lessons in creating a workflow environment for the life sciences. *CCPE*, 18(10):1067–1100.
- Qin, J., Fahringer, T., e Berger, M. (2008). Towards workflow sharing and reuse in the ASKALON grid environment. In *Proc. CGW*, pages 111–119, Cracow, Poland.
- Taylor, I., Shields, M., e Wang, I. (2004). Resource management for the Triana peer-to-peer services. In *Grid resource management*, pages 451–462. Kluwer, Norwell, MA, USA.
- Yu, J. e Buyya, R. (2004). A novel architecture for realizing grid workflow using tuple spaces. In *Proc. 5th IEEE/ACM GRID*, pages 119–128, Washington, DC, USA. IEEE ComSoc.
- Yu, J. e Buyya, R. (2006). A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*, 3:171–200.
- Zhao, Y., Hategan, M., Clifford, B., e Foster, I. (2007). Swift: Fast, reliable, loosely coupled parallel computation. In *Proc. IEEE SCW*, pages 199–206.
- Zhao, Y., Raicu, I., Foster, I., Hategan, M., Nefedova, V., e Wilde, M. (2008). Realizing fast, scalable and reliable scientific computations in grid environments. In *Grid Computing Research Progress*, pages 1–40. Nova Science, Commack, NY, USA.