Uma abordagem para viabilizar experimentos *in silico* colaborativos

Eduardo Jandre[1], Bruna Diirr[2], Vanessa Braganholo[1]

[1]Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói– RJ – Brasil

[2]Programa de Pós-Graduação em Informática – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Rio de Janeiro, RJ, Brasil

eduardojandre@id.uff.br, bruna.diirr@uniriotec.br, vanessa@ic.uff.br

Abstract. *By definition, science is a collaborative activity. Scientific research is usually performed by several scientists working together, and this behavior has intensified in the last decades. In addition, experiments are increasingly performed in silico, which requires adequate support tools. Although some approaches deal with the elaboration and analysis of experiments in a collaborative way, they require scientists to use workflow management systems, ignoring the fact that many scientists use scripts in their research. In fact, tools for running script-based experiments typically have very limited support for collaboration, therefore some scientists end up using versioning tools to collaborate in this context. Although the versioning tools deal with several collaborative aspects of script building, they are software development tools that are only concerned with the script's composition phase, not dealing with specific problems of scientific research. In this work we present a research project that aims at supporting scientists in the conduction of in silico experiments based on scripts in a collaborative way.*

Resumo. *A ciência é uma atividade colaborativa por definição. A pesquisa científica é geralmente realizada por vários cientistas trabalhando juntos, e esse comportamento tem sido intensificado nas últimas décadas. Além disso, os experimentos são cada vez mais realizados in silico, o que exige ferramentas de suporte adequadas. Apesar de algumas abordagens tratarem da elaboração e análise de experimentos de forma colaborativa, elas exigem que os cientistas usem sistemas de gerência de workflow, ignorando o fato de muitos cientistas utilizarem scripts nas suas pesquisas. As ferramentas para execução de experimentos baseados em scripts normalmente apresentam suporte muito limitado a colaboração. Sendo assim, alguns cientistas acabam utilizando ferramentas de versionamento para colaborar nesse contexto. Apesar das ferramentas de versionamento tratarem diversos aspectos colaborativos da construção de scripts, elas são ferramentas voltadas para o desenvolvimento de software e estão preocupadas somente com a fase de composição do script, não tratando problemas específicos da pesquisa científica. Nesse trabalho apresentamos um projeto de pesquisa que visa a apoiar o cientista na condução de experimentos in silico baseados em script de forma colaborativa.*

1. Introdução

O conhecimento científico é construído de forma incremental e cumulativa. Um ponto importante no processo científico é a comunicação dos resultados obtidos, que permite a revisão do trabalho realizado e também que outros cientistas utilizem o conhecimento adquirido para descobertas futuras [Roy Caldwell and David Lindberg 2018]. Nesse contexto, fica claro que a colaboração é parte presente e fundamental da ciência, e que os avanços científicos são altamente dependentes desse processo colaborativo.

Também é muito comum que as pesquisas sejam realizadas em conjunto por diversos pesquisadores. Essa colaboração pode ocorrer em diversas intensidades, o que pode refletir em publicações com múltiplos autores ou menções de agradecimento a outros pesquisadores. Wuchty, Jones e Uzzi [2007] analisaram quase 20 milhões de publicações e concluíram que a produção de artigos em times tem crescido ao longo do tempo e que esses times têm aumentado de tamanho. Além disso, os autores concluíram que artigos produzidos em grupo costumam receber mais citações na média que artigos feitos por um único autor, mesmo quando são desconsideradas autocitações. Essa colaboração é frequentemente estimulada e até mesmo exigida pelas instituições que financiam as pesquisas [Sonnenwald 2007].

Somado a isso, a tecnologia avançou bastante, com computadores cada vez mais baratos e acessíveis, e redes de computadores espalhadas por todo o mundo. Esse movimento produziu dois efeitos diretos: (i) permitiu que a colaboração ocorra não apenas entre as pessoas próximas, mas também entre pessoas localizadas em todo o mundo; e (ii) também aumentou o número de experimentos científicos realizados *in silico*. Experimentos *in silico* são caracterizados por serem experimentos totalmente executados no computador, onde tanto o ambiente de execução como o de observação são totalmente compostos por modelos numéricos para os quais nenhuma interação humana é prevista [Travassos and Barros 2003].

Experimentos *in silico* normalmente exigem muito mais suporte de ferramentas de gerência de dados e engenharia de software quando comparados a experimentos tradicionais [Travassos and Barros 2003]. Durante a execução de experimentos *in silico* são produzidos, além do resultado do experimento, outros dados relacionados que precisam ser gerenciados e armazenados, tais como: sequência lógica de atividades realizadas; parâmetros utilizados; resultados intermediários das atividades executadas; informações sobre o ambiente de execução; dentre outras informações. Tudo isso constitui o que chamamos de proveniência do experimento [Herschel, Diestelkämper, and Lahmar 2017]. A coleta de proveniência é uma característica comum em muitos sistemas de gerência de *workflows* [Freire et al. 2006; J. Zhang 2010] e sistemas baseados em script [Murta et al. 2014; Davison 2012; Miao, Chavan, and Deshpande 2017], que têm sido uma maneira popular de executar experimentos *in silico*. No entanto, a colaboração ainda é um dos desafios na área [Davidson and Freire 2008; Herschel, Diestelkämper, and Lahmar 2017], pois questões como (i) como coletar proveniência em um experimento colaborativo e (ii) como a proveniência pode ser usada para facilitar a colaboração neste ambiente, emergem quando falamos de experimentos colaborativos.

Apesar de alguns autores [Freire et al. 2006; Jia Zhang, Kuc, and Lu 2012; Ellqvist et al. 2009] tratarem da elaboração e análise de experimentos de forma colaborativa, eles estão baseados em sistemas de gerência de *workflow*, ignorando o fato de muitos cientistas utilizarem scripts nos seus experimentos [Pimentel et al. 2019]. Quando

olhamos para as soluções baseadas em scripts, o suporte a colaboração ainda é bastante limitado. Assim, com o objetivo de cobrir essa lacuna, nesse artigo propomos um desenho de pesquisa visando a construção de uma abordagem que auxilie cientistas na condução de experimentos colaborativos *in silico*.

Esse artigo está organizado da seguinte maneira: na seção 2 discutimos os trabalhos relacionados; na seção 3 a proposta de pesquisa é explicada, assim como a forma de avaliação pretendida é apresentada; na seção 4 são feitas algumas considerações finais.

2. Trabalhos Relacionados

Apesar das ferramentas de versionamento (ex.: Git [Spinellis 2012] e Mercurial ["Mercurial SCM" 2019]) tratarem diversos aspectos colaborativos da construção de *scripts*, elas são ferramentas voltadas para o desenvolvimento de *software* e estão preocupadas somente com a fase de composição do *script*. Problemas específicos da pesquisa científica, como a reprodutibilidade do experimento e a consequente necessidade da coleta de proveniência na fase de execução do experimento [Herschel, Diestelkämper, and Lahmar 2017], não podem ser tratados usando apenas ferramentas de versionamento.

Além disso, as soluções mais tradicionais de sistemas de gerência de experimentos baseados em *script* [Murta et al. 2014; Davison 2012] estão focadas somente na coleta da proveniência e não tratam a realização do experimento de forma colaborativa por múltiplos usuários. Mesmo soluções mais recentes que abordam aspectos colaborativos do experimento, como o ProvDB [Miao, Chavan, and Deshpande 2017], ainda apresentam algumas limitações: dependência de ferramenta externa (Git), funcionamento apenas em sistema operacional específico (UNIX) e foco em um tipo específico de experimento (*data science analysis*). Dessa forma, existe uma lacuna na literatura em relação ao suporte para colaboração em experimentos *in silico* executados em forma de *script*, que pretendemos cobrir com esse desenho de pesquisa.

Para atacar essa lacuna, pretendemos aproveitar pesquisas desenvolvidas anteriormente por outros pesquisadores. A ideia é estender uma ferramenta já existente para a condução de experimentos em *script*, adicionando capacidades colaborativas a tal solução. Dentre as abordagens existentes, escolhemos o noWorkflow [Murta et al. 2014], uma ferramenta que permite a captura da proveniência de *scripts* Python. Escolhemos o noWorkflow por ser uma ferramenta de código aberto bem documentada e que quando comparada ao Sumatra [Davison 2012], que seria uma outra opção de código de aberto, possui a capacidade de capturar a proveniência em um grão mais fino e também apresenta uma camada de visualização mais elaborada que facilita a percepção da evolução da pesquisa e o relacionamento entre as execuções de um experimento.

Para coletar a proveniência usando o noWorkflow, o pesquisador deve passar a incluir o comando *"now run"* como prefixo do *script* Python a ser executado. O noWorkflow armazena as informações capturadas em uma pasta chamada '.noWorkflow' no diretório em que foi invocada a execução do script. Essa pasta é chamada de 'base de proveniência' e cada execução de *script* é chamada de *trial*. A informação é separada em uma base de dados SQLite e uma base de conteúdo. A base de conteúdo é responsável por armazenar todos os arquivos utilizados durante a execução do *script*. Já a base de dados SQLite é responsável por armazenar outras informações referentes à execução do *script*, como variáveis de ambiente, parâmetros de execução, chamadas de função,

resultados e outras informações dependendo do nível de granularidade escolhido na captura. Dessa forma, o sistema é capaz de capturar todas as informações necessárias para garantir a reprodutibilidade do *trial*. Para consultar tais informações, o pesquisador pode utilizar consultas via linha de comando, a interface *web* do noWorkflow, consultas via Prolog ou até mesmo executar consultas SQL na base de conteúdo.

3. Proposta de Pesquisa

3.1. Contribuição Proposta

Nesse trabalho, propomos a construção de uma solução que permita a realização colaborativa de experimentos *in silico* baseados em *script*. Como já dito anteriormente, para isso vamos estender a ferramenta noWorkflow [Murta et al. 2014].

No contexto do noWorkflow, todos os dados do experimento são capturados e armazenados na base proveniência. Para apoiar a realização de experimentos colaborativos, visamos a criação de um portal que permita agregar, sincronizar e analisar as informações do experimento realizado por diversos cientistas que estão colaborando entre si. Assim, é necessário construir um mecanismo capaz de consolidar a base de proveniência dos diversos cientistas que estão envolvidos no experimento.

Usando alguns conceitos de DVCS (*Distributed Concurrent Versions System*) [Spinellis 2012; "Mercurial SCM" 2019] já consolidados na engenharia de software, cada cientista teria sua base de proveniência local contendo as informações do experimento. Esse repositório poderia ser sincronizado com um repositório central, onde estaria disponível para outros interessados. Outra possibilidade seria a aplicação de um modelo *peer-to-peer*, onde os cientistas compartilhariam as informações diretamente entre si, sem a necessidade de um nó centralizador. Para atingir esses objetivos, a ideia é criar dois comandos (*pull* e *push*) no noWorkflow que permitam aos cientistas envolvidos o envio e recebimento de dados de uma outra base de proveniência. Ambos os comandos receberiam por parâmetro uma *url*, que indica com qual base a operação seria feita.

Além da integração das bases de proveniência dos cientistas, pretendemos também trabalhar na visualização e busca em dados da pesquisa. É importante que essa informação seja apresentada para o cientista de uma forma que melhore a percepção de como a pesquisa tem evoluído, o que foi feito por outros pesquisadores, que pesquisador executou determinado experimento etc. Para isso pretendemos estender a camada de visualização do noWorkflow para contemplar a percepção nesse ambiente colaborativo.

3.2. Métodos de Investigação e Avaliação

Para avaliar se a abordagem proposta de fato facilita a colaboração entre os pesquisadores, planejamos realizar experimentos com usuários voluntários.

Pretendemos elaborar uma série de tarefas e perguntas, que exijam a colaboração dos envolvidos para serem concluídas. Pretendemos dividir os usuários em dois grupos, de forma que um dos grupos utilize a abordagem tradicional do noWorkflow enquanto o outro grupo utilize a abordagem proposta nesse projeto. Em seguida repetiríamos o experimento com um novo conjunto de tarefas, invertendo a abordagem utilizada por cada grupo. Dessa forma os dois grupos teriam acesso a ambas abordagens.

No final seria feita uma análise qualitativa e quantitativa dos resultados obtidos. Para a análise quantitativa seria considerada a corretude das respostas e o tempo gasto para a conclusão das tarefas. Para análise qualitativa será elaborado um questionário avaliando alguns aspectos de cada uma das abordagens, que seria respondido pelos usuários voluntários após a realização das tarefas.

Nesse momento ainda não definimos a série de tarefas e perguntas, e o questionário de avalição qualitativa a serem utilizados no processo de avalição. A ideia é que esses itens sejam definidos durante o andamento da pesquisa descrita nessa proposta.

4. Considerações Finais

Nesse trabalho apresentamos um desenho de pesquisa para a construção de uma ferramenta colaborativa que auxilie cientistas no desenvolvimento de experimentos *in silico* baseados em *script*. Entendemos que as abordagens existentes de gerência de experimentos baseados em *script* apresentam muitas limitações em ambientes colaborativos. Isso somando ao fato de que cada vez mais experimentos são realizados de forma colaborativa, é essencial que os cientistas possuam ferramentas com suporte para esse tipo de ambiente. Abordagens desse tipo, além de se tornarem importantes no próprio desenvolvimento da pesquisa, se tornam peças relevantes para a comunidade científica como um todo, fornecendo informações que podem ser usadas no processo de *peer-review* e para a reprodutibilidade do experimento.

Até o momento, já evoluímos na camada de agregação e sincronização dos dados de proveniência do experimento, tendo os comandos *pull* e *push* já implementados. Isso já possibilita que os cientistas compartilhem e consolidem informações relativas ao experimento com colaboradores de uma maneira prática. Como próxima etapa, pretendemos trabalhar na camada de visualização e análise dos dados de proveniência. Nesse caso, o intuito está em melhorar a experiência de colaboração entre os envolvidos e também possibilitar a obtenção de informação de valor das iterações entre os pesquisadores (*awareness*). Por último, pretendemos executar os experimentos descritos na seção 3.2, verificando a efetividade da abordagem proposta e identificando oportunidades para novas pesquisas.

Referências

Davidson, Susan B., and Juliana Freire. 2008. “Provenance and Scientific Workflows: Challenges and Opportunities.” In *ACM SIGMOD*, 1345–1350. New York, NY, USA: ACM. https://doi.org/10.1145/1376616.1376772.

Davison, A.P. 2012. “Automated Capture of Experiment Context for Easier Reproducibility in Computational Research.” *Computing in Science & Engineering* 14 [4]: 48–56. https://doi.org/10.1109/MCSE.2012.41.

Ellqvist, Tommy, David Koop, Juliana Freire, Cláudio Silva, and Lena Strömbäck. 2009. “Using Mediation to Achieve Provenance Interoperability.” In *2009 World Conference on Services - I*, 291–98. IEEE. https://doi.org/10.1109/SERVICES-I.2009.68.

Freire, Juliana, Cláudio T. Silva, Steven P. Callahan, Emanuele Santos, Carlos E. Scheidegger, and Huy T. Vo. 2006. “Managing Rapidly-Evolving Scientific Workflows.” In *Provenance and Annotation of Data*, edited by Luc Moreau and

Ian Foster, 10–18. Lecture Notes in Computer Science 4145. Springer Berlin Heidelberg.

Herschel, Melanie, Ralf Diestelkämper, and Houssem Ben Lahmar. 2017. “A Survey on Provenance: What for? What Form? What From?” *The VLDB Journal* 26 [6]: 881–906. https://doi.org/10.1007/s00778-017-0486-1.

“Mercurial SCM.” 2019. April 23, 2019. https://www.mercurial-scm.org/.

Miao, Hui, Amit Chavan, and Amol Deshpande. 2017. “ProvDB: Lifecycle Management of Collaborative Analysis Workflows.” In *Proceedings of the 2Nd Workshop on Human-In-the-Loop Data Analytics*, 7:1–7:6. HILDA’17. New York, NY, USA: ACM. https://doi.org/10.1145/3077257.3077267.

Murta, Leonardo, Vanessa Braganholo, Fernando Chirigati, David Koop, and Juliana Freire. 2014. “NoWorkflow: Capturing and Analyzing Provenance of Scripts.” In *International Workshop on Provenance Annotation (IPAW)*, 1–12.

Pimentel, João Felipe, Juliana Freire, Leonardo Murta, and Vanessa Braganholo. 2019. “A Survey on Collecting, Managing, and Analyzing Provenance from Scripts.” *ACM Comput. Surv.* 52 [3]: 47:1–47:38. https://doi.org/10.1145/3311955.

Roy Caldwell, and David Lindberg. 2018. “Participants in Science Behave Scientifically.” Understanding Science. 2018. https://undsci.berkeley.edu/article/0_0_0/whatisscience_09.

Sonnenwald, Diane H. 2007. “Scientific Collaboration.” *Annual Rev. Info. Sci & Technol.* 41 [1]: 643–681. https://doi.org/10.1002/aris.144.v41:1.

Spinellis, D. 2012. “Git.” *IEEE Software* 29 [3]: 100–101. https://doi.org/10.1109/MS.2012.61.

Travassos, G. H., and M. O. Barros. 2003. “Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering.” In *2nd Workshop on Empirical Software Engineering the Future of Empirical Studies in Software Engineering*, 117–30. Rome, Italy.

Wuchty, Stefan, Benjamin F. Jones, and Brian Uzzi. 2007. “The Increasing Dominance of Teams in Production of Knowledge.” *Science* 316 [5827]: 1036–1039.

Zhang, J. 2010. “Co-Taverna: A Tool Supporting Collaborative Scientific Workflows.” In *2010 IEEE International Conference on Services Computing*, 41–48. https://doi.org/10.1109/SCC.2010.99.

Zhang, Jia, Daniel Kuc, and Shiyong Lu. 2012. “Confucius: A Tool Supporting Collaborative Scientific Workflow Composition.” *IEEE Transactions on Services Computing*, no. 1.