

Desafios de monitoração de requisitos não funcionais: avaliação em Transparência de Software

André Luiz de Castro Leal^{1,2}, Henrique Prado Sousa¹, Julio Cesar Sampaio do Prado Leite¹

¹Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rio de Janeiro, RJ, Brasil

²Departamento de Matemática
Universidade Federal Rural do Rio de Janeiro (UFRRJ)
BR-465, Km 7, Seropédica, Rio de Janeiro - CEP 23890-000
andrecastr@gmail.com, hsousa@inf.puc-rio.br, www.inf.puc-rio.br/~julio

Resumo. Com a demanda da sociedade pela disponibilização da informação aberta, a Transparência passa a ser reconhecida como um requisito de potencial aplicação a software. Como requisito não funcional, a Transparência desafia engenheiros de software com suas características peculiares e ainda pouco pesquisadas. A partir dessas demandas, surge o questionamento referente às formas de avaliação da Transparência em software, tanto em sua arquitetura como em seu ambiente de execução e entrega da informação. O presente trabalho fomenta a discussão sob o olhar da avaliação de Transparência de software.

Keywords: Transparência de software, requisito não funcional, monitoração, sistemas multi-agentes.

1 Introdução

Uma sociedade madura, fortemente envolvida em suas rotinas com o uso de sistemas computacionais, almeja por acesso a todo e qualquer tipo de informação. Nesse sentido, a informação aberta é um tema central que evidencia um requisito não funcional, a Transparência [1].

Uma recente demonstração do movimento em torno informação explícita é a criação da Lei de Acesso, Lei nº 12.527, de 18 de novembro de 2011 [8]. A lei estabelece obrigações para os órgãos e entidades do poder público quanto à gestão da informação; define os tipos de informação que podem ser solicitadas; estabelece obrigações de divulgação espontânea de informações pelos órgãos da Administração Pública e medidas que devem ser adotadas para assegurar o acesso às informações.

É sabido que sistemas de informação estão implantados e dão suporte atualmente em quase todo domínio onde a informação é consumida. Diante do desafio da Transparência e da importância do software no suporte ao uso da informação, Engenheiros de Software passam a ter o desafio de lidar com dados abertos, usabilidade orientada ao cidadão, acessibilidade, apresentação de informação,

proveniência da informação e confiança na informação, que são características inseridas no contexto da Transparência a fim de prover o chamado Sistemas de Software Transparentes [6]. E esse desafio não ocorre apenas no uso do sistema para prover Transparência, mas também em sua própria arquitetura que pode ser transparente.

O Grupo de Pesquisa em Engenharia de Requisitos da PUC-Rio (Grupo ER - PUC-Rio) [1] desenvolve pesquisa sobre o tema Transparência a partir da definição de conceitos, estruturas, símbolos, termos, catálogos, sua aplicação em software [6] e modelos de negócio [7], criados de forma colaborativa, com o objetivo de sistematizar, a partir de critérios técnicos, ações, modelos, sistemas e definições sobre o tema. Dentro dessa realidade, há um recorte que particularmente nos interessa: a avaliação semi-automática ou automática da Transparência [2]. Ou seja, como conceber e implementar sistemas que possam tratar a monitoração de atributos pré-definidos em catálogo de Transparência de Software [1].

O artigo apresentará a perspectiva de pesquisa sobre o tema e será dividido em seções como segue: na seção dois serão apresentadas os objetivos da pesquisa; na seção 3, as contribuições científicas; a seção 4 apresentará as conclusões; e por fim, na seção 5 serão apresentadas as perspectivas de trabalhos futuros e o estado atual da pesquisa.

2 Objetivos da Pesquisa

O principal objetivo da pesquisa é focar a questão da avaliação de Transparência no contexto de software, que é uma questão importante a ser tratada uma vez que a avaliação de requisitos não funcionais não é algo que aceite valores exatos. Tornar software transparente requer inicialmente avaliações para se entender quais pontos são passíveis de suportar tal qualidade [6].

O escopo inicial restringe-se em avaliar software em execução. Ou seja, avaliar traços da execução de um software a fim de compará-los com catálogo pré-definido para extrair pontos de conformidade e não conformidade entre o software e atributos pré-definidos de Transparência.

Inicialmente, o interesse é desenvolver e evoluir um projeto baseado em políticas de monitoração utilizando a arquitetura de sistemas multi-agentes (SMA). Para isso é feito um aprofundamento dos conceitos relacionados com SMA a fim de compreender como agentes de software atuando como monitores seriam capazes de efetuar coletas de dados confiáveis e em tempo real sobre o processo de software, para posteriormente divulgar os resultados acerca das ações do usuário e do software monitorado.

Serão criados modelos de avaliação da Transparência para estabelecer políticas de monitoração. Tais políticas serão sugeridas com base no *framework* i* [3]. Outra forma de representação de intencionalidades dos agentes será abordada a partir do modelo do Painel de Intencionalidades proposto por [4]. O objetivo do uso do Painel de Intencionalidade é motivado por seus benefícios de representação exclusiva de metas concretas e flexíveis, e suas inter-relações representadas através de elos de dependência, correlação, contribuição e equivalência no mesmo diagrama, considerando os níveis de relevância. A importância de criação de modelos é tornar

transparente também a arquitetura do SMA. Esse ponto toca diretamente no ponto da Transparência da Informação.

Os estudos tem nos revelado que outras características podem ser acrescentadas às políticas de monitoração para potencializar os atributos de Transparência. Confiança e Proveniência possuem características que parecem estar intimamente relacionadas nesse contexto. Em [5] são descritas relações de potencialização de influência bilateral entre Confiança e Transparência. Já em [9, 10] encontra-se registros da relação entre Confiança e Proveniência.

Confiança está relacionado na questão de acreditar ou se ter confiança em alguém ou alguma coisa, além de ser um fator psicológico que é reforçado a partir de questões de informação. O expectador necessita de informação ao longo de algum processo para que possa se sentir seguro de que o ocorrido é confiável. Daí nossa percepção de que a Transparência de software é capaz de influenciar em questões de Confiança e vice-versa.

Por outro lado, há características de Proveniência que estão intimamente ligadas às questões de Transparência e Confiança. A análise de taxonomias de Proveniência sugere que pontos como “Qualidade do Dado” (*data quality*), que permeia questões de “rastros ou linhagem do dado” que permitem estimar sua qualidade e “confiança” (*reliability*). Além de outras características de “trilhas de auditoria” (*audit trail*), atribuição ou “responsabilidade” pelo dado (*liability*) e informação que estão intimamente ligados aos conceitos de Transparência de Processo e Informação, presentes em [1].

Proveniência, nos casos citados, tem sido tratada na literatura científica como um processo para manter rastros de dados (origem, transição e destino) com o objetivo de permitir posteriormente a verificação de como o dado foi produzido, bem como a qualidade do processo que o produziu.

Portanto, umas das principais preocupações com a pesquisa é estabelecer políticas de avaliação da Transparência baseadas em critérios técnicos, a partir da evidência de sua relação com outras características capazes de potencializar o nível de confiança do usuário no processo de monitoração. Como a arquitetura do software também é um elemento passível de Transparência, tais políticas necessitam estabelecer ou sugerir formas de auditoria dos próprios agentes de monitoração, garantindo a transparência também para as ações dos mesmos.

Outro ponto importante na pesquisa é a condição de estabelecer os domínios do conceito de rastreabilidade, utilizado nesse artigo como Rastreabilidade, termo já classificado e definido na Engenharia de Software (ES) e relacionado à Engenharia de Requisitos (ER), e sua fronteira ou relação com as questões de Proveniência, termo amplamente usado na área de Banco de Dados.

Um ponto chave para o estudo destas abordagens está na construção de modelos baseados no *NFR framework*, proposto por [13]. Uma primeira impressão é que Rastreabilidade, tal como propõe a literatura da ES, trata-se da rastreabilidade de artefatos em um processo de software; do outro lado, a Proveniência trata de questões de persistência de rastros de execução e de sua recuperação.

No caso da Proveniência, há uma condição particular de rastros em diferentes camadas: na coleta de rastros na composição/concepção de determinado *workflow*, definida como prospectiva; e na coleta de rastros na execução de um *workflow*,

definida como retrospectiva. Apesar de algumas semelhanças há fortes indicações de que as duas abordagens trabalham em níveis de abstração diferentes.

A intenção é procurar entender como processos de Rastreabilidade podem dar suporte sistematizado na construção de processos de avaliação da Transparência e como que estruturas de Proveniência podem dar suporte às questões de estruturação, semântica, captura, persistência, e recuperação de rastros de um software avaliado.

3 Contribuições Científicas

Uma das principais expectativas com relação à contribuição da pesquisa é a definição de uma estratégia de monitoração com o objetivo de entender se atributos de Transparência estão sendo observados em um software em execução.

Com isso, pretende-se definir modelos explícitos de políticas de avaliação de Transparência de software baseados em *framework* i*. Tal contribuição pretende satisfazer às necessidades de suporte metodológico, sistematizado e baseado em critérios técnicos para futuras implementações de sistemas que irão atuar sob aspectos dos requisitos não funcional de Transparência. Além disso, nesse primeiro momento tratar a construção e evolução de um SMA que automatize o processo de monitoração e validação.

Uma das primeiras contribuições elaboradas pela pesquisa foi detalhada em [2], que propôs a primeira versão de um SMA de coleta de dados e comparação com catálogo de Transparência [1]. Sua sistematização é realizada a partir da construção de 4 agentes: MONITOR, CANONIZADOR, CONSOLIDADOR e ANALISADOR. Em suma, o agente MONITOR trata de monitorar traços de execução de determinado software e gera um novo traço para dentro do SMA. A partir disso, o agente CANONIZADOR marca os traços de execução com delimitadores condizentes com os atributos do catálogo de Transparência [1]. O CONSOLIDADOR limpa os traços não marcados e gera um novo traço de execução para que o ANALISADOR possa comparar com o catálogo de Transparência. Ou seja, a partir dos traços de execução e padrões estabelecidos no catálogo é verificado se o software avaliado está em conformidade com atributos de Transparência.

Os modelos representados na ocasião foram baseados no *framework* i* com o objetivo de representar e facilitar a análise de execução dos agentes com o uso da representação da intencionalidade distribuída [2], comum aos SMA. Feitas as correções, evolução do SMA e melhorias no modelo intencional, o trabalho foi submetido e aceito [14] pela Revista de Informática Teórica e Aplicada.

4 Conclusões

Pelo andamento das pesquisas bibliográficas, escrita de artigos preliminares, de uma primeira versão do SMA e das discussões em grupo, entendemos que o caminho que menos onera a compreensão de propostas de monitoração de requisitos não funcionais perpassa por criação de modelos. O uso de modelos baseados no *framework* i*

permite-nos estabelecer tanto objetivos, ou metas flexíveis para a estratégia de monitoração, como também permite com que *rationales*, ou operacionalizações, da monitoração sejam evidenciadas e melhor compreendidas.

Por outro lado, envolver na pesquisa conceitos de Transparência, Confiança, Rastreabilidade e Proveniência requer uma representação de suas características e também de suas relações. Entendemos que o uso do *NFR framework* [13] possa ser uma ferramenta adequada para tais representações e os catálogos de cada uma dessas abordagens serão construídos para que fiquem evidenciadas características individuais, comuns e suas relações.

5 Trabalhos Futuros

Os integrantes da pesquisa, juntamente com o Grupo ER - PUC-Rio discutem atualmente a respeito de conceitos e inter-relacionamentos entre Rastreabilidade e Proveniência para que modelos descritos em *NFR framework* sejam estabelecidos e as fronteiras e interseções entre as duas abordagens sejam delineadas.

Alguns trabalhos futuros começam a surgir em um *brainstorm* a partir da análise das possibilidades com a evolução da pesquisa. Uma primeira visão de trabalho que pode ser promissor é o desenvolvimento de estratégias para sistemas de recomendação. A partir do momento em que a execução da monitoração é realizada há o registro da monitoração, como também, o registro de conformidades e não conformidades do *software* monitorado em relação aos padrões de qualidade de Transparência. Modelos intencionais para sistemas de recomendação poderão ser criados e ser úteis para definir sistemas que possam, a partir da base de conhecimento criada e das não conformidades, sugerir alternativas para softwares monitorados e que não estejam de acordo com os padrões pré-estabelecidos de Transparência.

Outra possibilidade de trabalho futuro é avaliar se a proposta da pesquisa é aderente a outros requisitos não funcionais. Entendemos que a pesquisa indica que outros requisitos não funcionais poderão ser operacionalizados e validados a partir da proposta da política que está sendo criada. Apesar disso, uma questão que já nos inquieta é se uma estratégia de monitoração tem que ser obrigatoriamente diferente para requisitos não funcionais distintos.

Como também tratamos na pesquisa sobre questões de persistência de traços de execução e de monitoração seria importante que meta-modelos do registro das informações fossem sugeridos de forma padronizada com o objetivo de reuso.

Referências bibliográficas

1. Grupo Transparência de Software, PUC-Rio. http://www.er.les.inf.puc-rio.br/~wiki/index.php/Transpar%C3%Aancia_de_Software.
2. Leal, A.L. de C., Sousa, H.P., Leite, J.C.S.P.; Lucena, C.J.P. de.: Modelagem intencional de políticas e implementação de agentes de monitoração de transparência em sistemas de software. ftp://ftp.inf.puc-rio.br/pub/docs/.../11_20_castro.pdf (2011).
3. Leite, J.C.S.P., Capelli, C.: Exploring i* Characteristics that Support Software Transparency. In.: Proc. of the 3rd International i* Workshop pp. 51-54. Recife-PE (2008)

4. Oliveira, A.P.A., Leite, J.C.S.P., Cysneiros, L.M., Cappelli, C.: Eliciting Multi-Agent Systems Intentionality: from Language Extended Lexicon to i* Models. In: Jornadas Chilenas de Computação 2007 - UNAP, 2007, Iquique.: Proceedings of the XXVI International Conference of the Chilean Computer Science Society. Iquique: Los Alamitos: IEEE Computer Society Press, v. 16. p. 40-49 (2007)
5. Cysneiros, L.M., Werneck, V.M.B.: An Initial Analysis on How Software Transparency and Trust Influence each other. In: 12th Workshop on Requirements Engineering, Proceedings of, v. 1. pp. 27-32. Valparaiso. Proceedings of 12th Workshop on Requirements Engineering, Universidad Tecnica Federico Santa Maria (2009)
6. Leite, J. C. S. P.. Sistemas de Software Transparentes. Palestra Convidada. In.: Simpósio Brasileiro de Engenharia de Software. <http://www-di.inf.puc-rio.br/~julio/S1ct-pub/transpsbes.pdf>. Florianópolis, SC (2006)
7. Leal, A.L.C., Sousa, H.P., Leite, J.C.S.P., Braga, J.L.: Transparência aplicada a modelo de negócios. In: XIV WER - Workshop em Engenharia de Requisitos, v. XIV. pp. 321-332. Proceedings XIV CibSE, Rio de Janeiro (2011)
8. Lei de Acesso. Lei nº 12.527, de 18 de novembro de 2011, <http://www.acessoinformacao.gov.br/acessoinformacaogov/acesso-informacao-brasil/legislacao-integra-completa.asp>
9. Clifford L. A.: When documents deceive, Trust and provenance as new factors for information retrieval in a tangled web. In.: Journal of the American Society for Information Science and Technology, pp. 12-17 (2001)
10. Dai, C., Lin, D., Bertino, E., Kantarcioglu, M.: Trust evaluation of data provenance. Technical Report CERIAS TR, Purdue University (2008)
11. Dividino, R., Schenk, S., Sizov, S., Staab, S.: Provenance, trust, explanations—and all that other meta knowledge. Künstliche Intelligenz (2009)
13. Chung, L., Nixon, B., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers, Massachusetts (2000)
14. Leal, A. L. C., Sousa, H. P., Leite, J. C. S. P., Lucena, C. J. P.. Aplicação de Modelos Intencionais em Sistemas Multi-agentes para Estabelecer Políticas de Monitoração de Transparência de Software. Revista de Informática Teórica e Aplicada: RITA, v. 20, p. 111-138, 2013.