

Uma Extensão do STREAM para Escolha de Padrões Arquiteturais baseada em Requisitos Não-Funcionais

Fábio Silva^{1,2}, Marcia Lucena¹, Leonardo Lucena², Roniceli Moura¹

¹Universidade Federal do Rio Grande do Norte (UFRN), Natal, RN, Brasil
marciaj@dimap.ufrn.br, roniceli.moura@yahoo.com.br

²Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN), Natal, RN, Brasil

{fabio.silva, leonardo.lucena}@ifrn.edu.br

Resumo. Cada vez mais os requisitos não funcionais são considerados primordiais aos sistemas computacionais. Satisfazer esses requisitos exige uma atenção especial com a arquitetura já que uma arquitetura inadequada introduz uma complexidade maior, além da complexidade intrínseca do sistema. Estudos mostram que apesar das atividades de engenharia de requisitos e de arquitetura de software atuarem em diferentes aspectos do desenvolvimento, é preciso executá-las de forma iterativa e entrelaçada para produzir sistemas computacionais satisfatórios. O processo STREAM apresenta uma abordagem sistemática para diminuir a lacuna entre requisitos e o desenvolvimento da arquitetura, enfatizando os requisitos funcionais, enquanto que os requisitos não-funcionais são usados de forma *ad hoc*. No entanto, alguns estudos mostram que requisitos não-funcionais impactam o sistema como um todo. Este artigo apresenta uma proposta para auxiliar o processo STREAM na realização da escolha de padrões arquiteturais a partir de requisitos não-funcionais, com propósito de guiar o refinamento da solução arquitetural.

Palavras Chave: *Requisitos Não-Funcionais, Arquitetura de Software, Padrão Arquitetural.*

1 *Introdução*

O relacionamento entre requisitos e arquitetura é considerado a parte central de um Processo de Desenvolvimento de Software (PDS) [5]. Enquanto a atividade de Engenharia de Requisitos (ER) é responsável por tratar os Requisitos Funcionais (RFs) e Não-Funcionais (RNFs) de um software, a Arquitetura de Software (AS) está relacionada com os componentes do software e a comunicação entre eles. No processo de construção de um sistema de software, a AS é considerada uma questão crítica, já que a qualidade da arquitetura está relacionada com a satisfação ou não dos requisitos chave requeridos pelo software [4]. O processo STREAM (Strategy for Transition between REquirements models and Architectural Models) [8] surgiu com o intuito de contribuir para a diminuição da lacuna existente entre a AS e a ER. Esta estratégia,

baseada em MDD (Model-Driven Development) [11], tem como objetivo principal a geração de modelos arquiteturais a partir de modelos de requisitos i^* . O STREAM é constituído por: (i) um conjunto de regras de transformação para preparar os modelos de requisitos i^* , chamadas de regras horizontais; (ii) um conjunto de regras de transformação para gerar modelos arquiteturais, chamadas de regras verticais; e (iii) um processo sistemático para auxiliar os desenvolvedores. O processo sistemático para auxiliar os desenvolvedores é dividido em 4 atividades: (i) refatorar modelos de requisitos; (ii) gerar modelos arquiteturais; (iii) selecionar uma Solução Arquitetural; e (iv) refinar uma Solução Arquitetural. As regras horizontais são utilizadas na Atividade (i) e as verticais na Atividade (ii). Para executar a Atividade (iii) os RNFs são usados para a seleção de Solução Arquitetural, quando ocorrer mais de uma. E, de acordo com os RNFs utilizados, alguns Padrões Arquiteturais (PAs) podem ser aplicados na atividade (iv). Os PAs a serem utilizados na Atividade (iv) são selecionados a partir de RNFs. O problema é que o STREAM apresenta uma limitação na escolha desses PAs. Não existe uma sistematização nessa escolha. Ela é realizada de forma ad hoc.

Em um processo de construção de AS são levados em consideração tanto os requisitos funcionais quanto os não-funcionais [2]. No entanto, os RNFs normalmente exercem influência sobre todo o sistema [6]. Eles são fundamentais em dois momentos do projeto de AS [1],[6]: (i) seleção de uma AS dentre várias; e (ii) escolha de um ou mais PAs que serão usados no refinamento da AS.

No processo de projeto e construção de um sistema complexo de software, a AS se apresenta como uma questão crítica, já que a qualidade da arquitetura está relacionada com a satisfação ou não dos requisitos chaves requeridos pelo software [4]. Segundo [12], decisões tomadas durante a fase de AS são particularmente significativas uma vez que tem implicações em todo o sistema, especialmente sobre os RNFs. Se de um lado temos os PAs e do outro os RNFs a serem atendidos por uma AS, como relacioná-los de forma a guiar a uma AS de qualidade de acordo com os RNFs? Uma vez que os RNFs foram elicitados e representados no modelo de requisitos, eles precisam ser analisados com o objetivo de se chegar a conclusão de: (i) quais impactam na AS; e (ii) dentre eles, quais são os prioritários.

Neste contexto, este trabalho visa levar em consideração RNFs, que sejam arquiteturalmente significantes, para guiar o refinamento de uma AS. Para isso um processo sistemático foi proposto tendo como entrada modelos de requisitos com RNFs elicitados, e como saída PAs selecionados para o refinamento da AS. Este processo será incorporado ao STREAM com o objetivo de contribuir para solucionar a deficiência existente nesta abordagem.

O artigo está organizado da seguinte forma: na seção 2 serão apresentados os objetivos da pesquisa. Na seção 3 são apresentadas as contribuições científicas realizadas. Na seção 4 são apresentadas as conclusões encontradas. Na seção 5 são apresentados os trabalhos futuros e em andamento.

2 *Objetivos da Pesquisa*

A pesquisa deseja responder a seguinte pergunta: quais PAs são os mais adequados para refinar uma determinada AS afim de que ela atenda a RNFs específicos? A AS é a Solução Arquitetural gerada e escolhida na aplicação do processo STREAM. O seu refinamento é a atividade realizada com o objetivo de fazer com que ela atenda a RNFs específicos. Neste contexto, a pesquisa assumiu como objetivo geral a proposição de um processo, o STREAM-AP, que sistematize a escolha de PAs, a partir de RNFs, para que possam ser usadas no refinamento de uma AS.

No processo de sistematização serão utilizadas contribuições científicas disponíveis nas áreas de RNFs e de PAs. Assim, a ideia principal do trabalho foi reunir em uma única abordagem o necessário para a realização do processo de escolha de PAs e inseri-lo no contexto do STREAM.

Os objetivos específicos deste trabalho são: (i) pesquisar bases de dados de RNFs; (ii) pesquisar bases de dados de PAs; (iii) disponibilizar e utilizar essas bases de dados de forma integrada; (iv) definir as atividades e sub-atividades do processo de sistematização proposto; (iv) aplicar o processo de sistematização proposto em um exemplo.

3 *Contribuições Científicas*

O trabalho proposto surgiu com o objetivo de suprir a deficiência apresentada pela abordagem STREAM no que diz respeito a escolha de PAs a partir de RNFs. Neste sentido, bases de dados de RNFs e de PAs são reunidas e disponibilizadas em um único ambiente. Tanto os RNFs quanto os PAs serão relacionados com tipos de sistemas. E, para contribuir na direção do objetivo principal da nossa proposta, as bases de dados serão relacionadas entre si, de forma a demonstrar o impacto que os PAs exercem sobre os RNFs, ou seja, se um PA impacta positivamente, de forma neutra ou negativamente em um determinado RNF. Essas bases de dados são resultados dos seguintes trabalhos científicos: (i) RNFs e seus atributos [9]; (ii) RNFs por tipos de sistemas [9]; (iii) conflitos entre os RNFs [10]; (iv) PAs por tipos de sistemas [7]; (v) impacto de PAs em RNFs [6]; (vi) pares de PAs de sistemas que utilizam até dois PAs [7]. Elas serão consultadas e/ou atualizadas no decorrer da aplicação da abordagem proposta. A partir dos dados consultados, decisões poderão ser tomadas no que diz respeito a utilização ou não de determinados PAs com o objetivo de se alcançar RNFs específicos.

A partir da incorporação do STREAM-AP no processo STREAM pretende-se tornar a escolha de PAs a partir de RNFs menos empírica e mais científica. A Figura 1 apresenta uma visão geral da abordagem STREAM com o STREAM-AP incorporado a ela. O significado da sigla STREAM-Ap é: STREAM (*Strategy for Transition between REquirements models and Architectural Models*) - AP (*Architectural Pattern*).

A Figura 2 apresenta as principais atividades e artefatos utilizados e gerados pelo STREAM-AP.

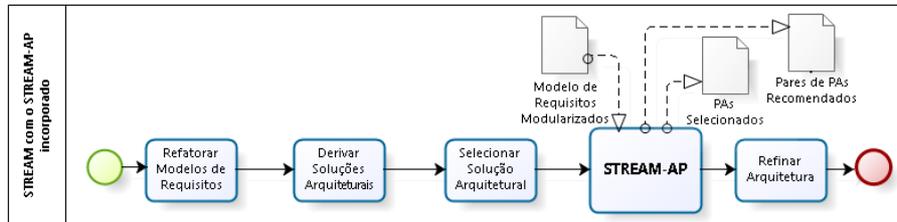


Fig. 1. Visão geral do STREAM com o STREAM-AP incorporado

É importante acrescentarmos que o processo proposto não se restringe e não é dependente da abordagem STREAM. Para que o STREAM-AP seja utilizado é necessário apenas que se tenha um modelo de requisitos no qual esteja disponível os RNFs a serem atendidos.

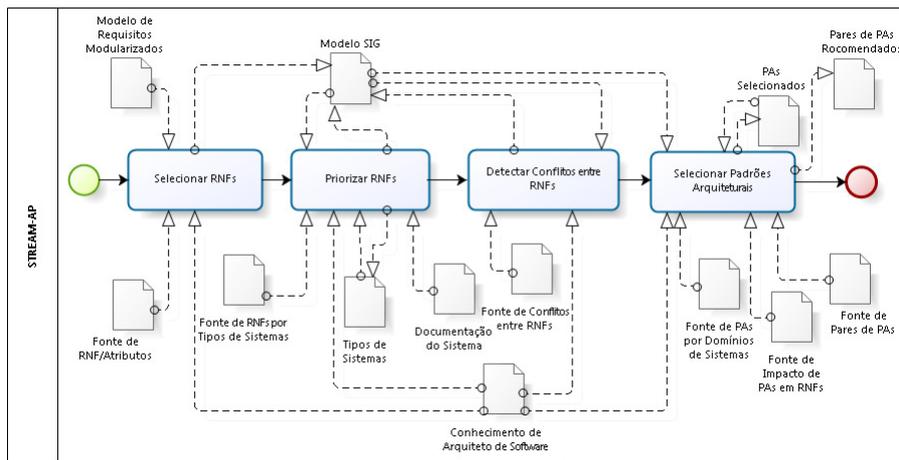


Fig. 2. Visão geral do STREAM-AP

4 Conclusões

Neste trabalho nós propomos o STREAM-AP. Trata-se de um processo que objetiva sistematizar a escolha de PAs a partir de RNFs. Esses PAs serão utilizados no refinamento da AS de forma que ela atenda a RNFs específicos.

Algumas abordagens foram propostas com o objetivo de reduzir a lacuna existente entre a ER e a AS, por exemplo [3],[8]. Entretanto, nas nossas pesquisas, percebemos uma deficiência no aprofundamento da escolha de PAs a partir de RNFs. Em específico, a abordagem STREAM [8] realiza a seleção de RNFs e PAs de forma ad hoc.

Com a utilização do STREAM-AP pretende-se aumentar a qualidade da AS e auxiliar o arquiteto de software na difícil tarefa de decisão arquitetural. A ajuda oferecida

a ele/ela é de fundamental importância uma vez que tira dele/dela a responsabilidade única da tomada de decisão baseada apenas em seus conhecimentos, situação em que em muitos momentos ele/ela não tem conhecimento suficiente para sozinho realizar a tomada de decisão.

5 *Trabalhos Futuros e Em Andamento*

O processo de escolha de PAs a partir de RNFs, proposto pelo STREAM-AP, precisa ser aprimorado no que diz respeito a dois aspectos: (i) semi-automatização do processo; e (ii) bases de dados de apoio ao processo.

No que diz respeito a semi-automatização do processo, ela se faz necessária pelo fato da manipulação das bases de dados de forma manual ter se mostrado trabalhosa, uma vez que o volume de dados a ser analisado é considerável e existem vários cruzamentos de dados que necessitam ser realizados. Tal procedimento contribui para a diminuição da carga de trabalho e também da responsabilidade única imposta sobre o arquiteto de software.

A escolha pela semi-automatização ao invés da automatização total foi considerada levando-se em consideração a natureza do processo em questão. Neste tipo de processo é de fundamental importância a intervenção manual do arquiteto de software com o objetivo de ajustes nas decisões realizadas.

A semi-automatização deverá ser realizada através da construção de um software que a partir dos parâmetros de entrada e das informações armazenadas em sua base de dados, gere para o arquiteto de software as informações que ele necessita para a tomada de decisão. No que diz respeito as bases de dados, o arquiteto de software terá a possibilidade de usar os dados armazenados e/ou armazenar novos dados.

O software será constituído de dois módulos: (i) gerenciamento da base de dados; e (ii) execução do processo de seleção dos PAs. Quanto a tecnologia utilizada no desenvolvimento do software, será utilizada a tecnologia Java, mais especificamente do framework para desenvolvimento web JSF. Já no que diz respeito a tecnologia de banco de dados, será usado o SGBD MySQL e o framework de persistência de dados Hibernate. A escolha dessas tecnologias são justificadas pela sua aceitação no meio acadêmico e industrial, e por serem adequadas ao tipo de software que se deseja construir. Além disso, todas as tecnologias são *Open Source* e são utilizadas no ambiente de ensino e pesquisa da UFRN. O processo de construção desse software está em fase de análise e em breve será iniciado o processo de desenvolvimento da ferramenta.

Como segundo aspecto temos a questão das bases de dados que auxiliam todo o processo. Elas precisam ser mais enriquecidas no que diz respeito a quantidade e aos tipos de dados disponibilizados: (i) disponibilizar maior quantidade de RNFs e seus respectivos atributos; (ii) disponibilizar a relação entre os atributos de RNFs e os tipos de sistemas, e não apenas por RNFs; (iii) disponibilizar a relação entre os atributos de RNFs e os PAs, e não apenas por RNFs. O objetivo do enriquecimento das bases de dados é proporcionar uma análise/escolha mais refinada dos PAs. Para que isso ocorra, é de fundamental importância que outras fontes de informações, da academia e/ou da indústria, relacionadas a RNFs, conflitos entre RNFs, PAs, impacto de PAs em

RNFs e a relação entre eles, sejam consultadas de forma a enriquecer e dar um maior embasamento a este trabalho de pesquisa.

Referências

1. Castro, J., Lucena, M., Silva, C. T., Alencar, F. M., Santos, E., & Pimentel, J.: Changing attitudes towards the generation of architectural models. *Journal of Systems and Software*, 85 (3), 463-479 (2012)
2. Chung, L., Gross, D., & Yu, E. S.: Architectural Design to Meet Stakeholder Requirements. *WICSA*, (pp. 545-564) (1999)
3. Chung, L., Supakkul, S., Subramanian, N., Garrido, J. L., Noguera, M., Hurtado, M. V., et al.: Goal-Oriented Software Architecting. In: *Relating Software Requirements and Architectures* (pp. 91-109) (2011)
4. Garland, D.: Software architecture: a roadmap. *Proceedings of the Conference on The Future of Software Engineering* (pp. 91-101). New York, NY, USA: ACM (2000)
5. Hall, J. G., Grundy, J., Mistrik, I., Lago, P., & Avgeriou, P.: Introduction: Relating Requirements and Architectures. In: *Relating Software Requirements and Architectures* (pp. 1-9) (2011)
6. Harrison, N. B., Avgeriou, P.: Leveraging Architecture Patterns to Satisfy Quality Attributes. *ECSA*, (pp. 263-270) (2007)
7. Harrison, Neil B.; Avgeriou, Paris. Analysis of architecture pattern usage in legacy system architecture documentation. In: *Software Architecture, 2008. WICSA 2008. Seventh Working IEEE/IFIP Conference on. IEEE, 2008. p. 147-156. (2008)*
8. Lucena, M., Castro, J., Silva, C. T., Alencar, F. M., & Santos, E.: Stream: a strategy for transition between requirements models and architectural models. *SAC*, (pp. 699-704) (2011)
9. Mairiza, D., Zowghi, D., & Nurmaliani, N.: An investigation into the notion of non-functional requirements. *SAC*, (pp. 311-317) (2010)
10. Mairiza, D., Zowghi, D., & Nurmaliani, N: Towards a Catalogue of Conflicts Among Non-functional Requirements. *ENASE 2010: 20-29 (2010)*
11. Kleppe, A., Warmer, J., & Bast, W. *MDA Explained: The Model Driven Architecture*. Prentice Hall. (2003).
Harrison, N. B., Avgeriou, P., & Zdun, U. Using Patterns to Capture Architectural Decisions. *IEEE Software*, 24 (4), 38-45. (2007)