

Suporte ao rastreamento de requisitos na ferramenta JGOOSE

Victor Augusto Pozzan, Gustavo Cesar Lopes Geraldino, Victor Francisco Araya Santander,
Ivonei Freitas da Silva

UNIOESTE – Universidade Estadual do Oeste do Paraná, Cascavel/PR – Brasil

victor_pozzan@hotmail.com, {gustavo.geraldino, victor.santander,
ivonei.silva}@unioeste.br

Abstract: Requirements engineering is an important phase of software engineering. The Requirements traceability is an essential activity for requirements management, by enabling a system development team to assess the impact of the changes in the requirements, ensuring the quality of the system. However, nowadays the computational tools that support the software process also need to contain some sort of traceability. This paper will be taken into consideration studies related to the requirements traceability techniques and the requirements traceability means which will be the basis for the implementation of features that allow tracking and managing artifacts generated by the JGOOSE tool. The JGOOSE tool allows, from the i* and BPMN organizational models, to generate UML use cases. Thus, this paper aims to show the implementation that provides support for traceability in the JGOOSE tool, from horizontal and vertical traceability metamodels, generating traceability matrices. This new support will help requirements engineers to assess and make changes appropriately and accurately, maintaining the system's consistency and their generated documentation.

Keywords: Requirements, traceability, JGOOSE.

1. Introdução

A importância do rastreamento de requisitos no desenvolvimento de software tem sido reconhecida tanto em âmbito acadêmico quanto industrial [7]. Na engenharia de requisitos entende-se que a capacidade de rastrear requisitos ao longo de sua vida possibilita gerenciar requisitos de maneira mais efetiva. Além disso, possibilita avaliar o impacto de mudanças no software de forma mais precisa. Evolução em requisitos e outros artefatos do processo de software é uma realidade na qual engenheiros de requisitos e demais profissionais da computação se defrontam no dia a dia de suas funções [9]. Qualquer evolução implica em avaliar o impacto dessa mudança em termos de esforço, tempo e custos associados. Para esse fim, é necessário que seja possível avaliar, no menor tempo possível, quais artefatos serão afetados pela mudança. Neste contexto, vários trabalhos têm sido propostos visando auxiliar profissionais da área na rastreabilidade de requisitos [9] [21] [22] [23]. São propostas técnicas, ferramentas e métodos [19] [23] para permitir que o processo de gestão de mudanças de requisitos seja realizado com o maior suporte possível.

Rastrear requisitos implica em que se estabeleçam meios de manter documentadas as ligações entre requisitos e demais artefatos tanto em relação à origem do requisito quanto em relação aos artefatos gerados posteriormente à documentação de requisitos. Para tanto, prover suporte automatizado para gerar essa documentação de apoio à rastreabilidade é fundamental [18]. Neste sentido, o presente trabalho tem seu foco na proposição de uma solução para o rastreamento de requisitos no âmbito da ferramenta *Java Goal into Object Oriented Standard Extension* (JGOOSE) [15]. Esta ferramenta apoia engenheiros de requisitos no processo de derivação de casos de uso *Unified Modeling Language* (UML) [20] a partir de modelos iStar (i*) [4] e *Business Process Model and Notation* (BPMN) [3]. A mesma contém editores gráficos [2] [5] [10] [12] para manipular os artefatos

envolvidos e tem sido alvo de experimentos no âmbito acadêmico [2] [24]. Contudo, uma das grandes deficiências da ferramenta está em não possuir apoio automatizado para gerar a documentação de rastreamento de requisitos, o que permitiria aos engenheiros de requisitos avaliar de forma mais adequada o impacto de mudanças nos artefatos suportados. Para este fim, foram estudadas propostas que norteiam a criação de meta-modelos de rastreamento, bem como o uso prático destes em processos de software. O trabalho proposto em [8] foi escolhido como base por apresentar um processo detalhado passível de ser aplicado no âmbito da JGOOSE. Todas as etapas previstas nesse trabalho são aplicadas com o intuito de prover suporte automatizado à rastreabilidade de requisitos na referida ferramenta. Os elementos das técnicas i*, BPMN e Casos de uso UML são incluídos nesse suporte bem como os meta-modelos de rastreamento propostos já consideram outros artefatos que a JGOOSE pretende suportar no futuro. A ferramenta com o novo suporte ao rastreamento é aplicada em alguns exemplos mostrando a sua viabilidade.

Entre as vantagens percebidas nesta proposta está a facilidade de visualizar, via matrizes de rastreamento [7] [21], as dependências e associações entre elementos dos artefatos utilizados no processo de derivação de casos de uso. Isto permite que engenheiros de requisitos possam visualizar quais elementos serão afetados em caso de evoluções, auxiliando inclusive na estimativa de esforço e tempo necessários para realizar mudanças. Apesar desta proposta estar limitada a uma ferramenta, neste caso a JGOOSE, é possível verificar que os meta-modelos gerados podem ser úteis em outros trabalhos, já que os mesmos permitem correlacionar e classificar em níveis de informação associados ao processo de software, os elementos internos dos artefatos utilizados. Também a experiência e processo utilizados para dar suporte automatizado à rastreabilidade de requisitos, pode ser base e motivar profissionais que desejam automatizar esta atividade da engenharia de requisitos.

Neste artigo, além da seção inicial, o mesmo está estruturado em: seção 2 apresenta brevemente a fundamentação teórica necessária ao entendimento do trabalho; seção 3 apresenta a proposta de suporte à rastreabilidade na JGOOSE; seção 4 apresenta a implementação da proposta na referida ferramenta; seção 5 descreve o estudo para aferir a viabilidade de uso da ferramenta; na seção 6 são realizadas as considerações finais.

2. Fundamentação Teórica

O rastreamento de requisitos corresponde à habilidade para descrever e seguir o ciclo de vida de um requisito tanto para frente como para trás no processo de software [6]. Em [7] detalha-se os tipos de rastreamento como sendo: *backward-from* - liga requisitos as suas fontes, outros documentos ou pessoas; *forward-from* - liga requisitos aos artefatos de projeto e implementação; *backward-to* - liga componentes de projeto e implementação de volta aos requisitos; *forward-to*: liga outros documentos (os quais precedem o documento de requisitos) para requisitos relevantes.

Neste contexto, destaca-se o trabalho de [8], no qual os quatro conceitos de Rastreamento *backward/forward* são relacionados aos conceitos de Rastreamento Vertical e Rastreamento Horizontal [9]. Tais conceitos representam, respectivamente, a capacidade de tratar os requisitos ou artefatos produzidos ao longo do ciclo de vida do projeto e a capacidade de tratar as diferentes versões de requisitos ou artefatos em uma determinada fase do ciclo de vida. Toranzo [8] também propõe classificar as informações a serem rastreadas; um meta-modelo; um modelo intermediário e um processo para desenvolver um modelo conceitual de rastreamento. Para a classificação das informações são propostos quatro níveis de informação, descritos a seguir. Externo: representa informações relacionadas com o contexto político e econômico, leis e normas no qual a organização está inserida; Organizacional: representa requisitos da organização para definir regras, processos e decidir mudanças; Gerencial: engloba as tarefas subordinadas a uma ou mais pessoas com o objetivo de planejar e controlar as atividades no âmbito do gerenciamento de um projeto de software; Desenvolvimento:

representa os elementos/artefatos do processo de desenvolvimento (documentos de requisitos, diagramas e programas). Estes conceitos guiam o processo de criação das matrizes de rastreamento na ferramenta JGOOSE.

Conforme já mencionado, a ferramenta JGOOSE [5], a qual será alvo da implementação de suporte à rastreabilidade utilizando o processo e elementos propostos em [8], suporta os processos propostos em [10] e [11]. A versão atual da ferramenta [12] permite que modelos organizacionais sejam construídos via framework *i** [4] ou BPMN [3] e, a partir destes, gerar requisitos funcionais de sistemas computacionais na forma de casos de uso UML [17]. Também é possível gerar casos de uso na forma textual seguindo um *template* adaptado de [14]. Quando usado o framework *i**, casos de uso são gerados com base nos atores e nas suas dependências e razões estratégicas. Já a derivação de Casos de Uso a partir de modelos BPMN utiliza os fluxos de conectividades, atividades e demais elementos do BPMN [1]. A ferramenta possui editores próprios para todos os artefatos utilizados. Esta ferramenta está disponível em [15] e um vídeo da mesma pode ser visualizado em [16].

Entre os trabalhos relacionados podemos destacar [29] [30] como estudos preliminares e [17] [18] [19] os quais fazem relatos e análise de ferramentas de gerência e a rastreabilidade de requisitos no âmbito acadêmico e industrial. Estes trabalhos focam em descrever a importância de desenvolver ferramentas de apoio neste contexto, bem como nas funcionalidades essenciais que estas devem conter. Já os trabalhos de [27] [28] possuem maior semelhança com este estudo, entretanto, com distinções em seus objetivos, sendo [27] direcionado às questões interdisciplinares (produtos mecatrônicos) e [28] relacionada ao desenvolvimento ágil o qual propõe a implementação (semiautomática) da rastreabilidade de requisitos em ambiente ágil. Diferentemente, nossa proposta objetiva em descrever a aplicação de um processo proposto em âmbito acadêmico [8] para dar suporte automatizado à rastreabilidade de requisitos em uma ferramenta já existente. Maiores detalhes sobre o framework *i**, BPMN e Casos de Uso podem ser consultados em [4] [3] [20], respectivamente.

3. Proposta de suporte à rastreabilidade

Para que o rastreamento de requisitos seja realizado eficazmente deve-se documentar todos os artefatos produzidos durante o desenvolvimento de um software. Neste sentido, foram observados todos os artefatos (*i**, BPMN, Casos de Uso) produzidos na ferramenta JGOOSE durante um ciclo de desenvolvimento de software. A elaboração de um meta-modelo de rastreamento de requisitos para essa ferramenta segue um conjunto de diretrizes propostas por [8] as quais estão descritas na Tabela 1.

Inicialmente, ressalta-se que os modelos *Strategic Dependency* (SD) (Figura 1) e *Strategic Rationale* (SR) (Figura 2) do framework *i** podem ser construídos na ferramenta JGOOSE em um contexto de modelagem do ambiente organizacional com o objetivo de gerar os casos de uso para o sistema computacional pretendido. Inclusive, um ator que representa esse sistema computacional juntamente com suas intencionalidades (dependências e razões estratégicas) é incluído na modelagem. Assim, a classificação dos elementos dos modelos SD e SR nos segmentos de informação e classes propostos em [8] (seção 2) observa este contexto. Desta forma, aplicando as diretrizes 1 a 8 (Tabela 1) para os elementos do modelo SD é possível classificar os elementos (atores e dependências) observando os envolvidos na relação *dependor*→*dependum*→*dependee*, sendo que as dependências (*dependum*) podem ser do tipo *Softgoal*, *Goal*, *Task* e *Resource* (Figura 1).

Em uma primeira análise, foi possível detectar a relação ator_sistema→*dependum*→*stakeholder*, sendo que o *dependor* é classificado em uma classe que representa o ator_sistema pretendido e a classe *stakeholder* classifica atores que influenciam ou são influenciados direta ou indiretamente pelo sistema computacional. Neste caso, as dependências (*dependum*) pertencerão ao segmento de InformaçãoExterno e classe denominada InformaçãoExterna. Cabe destacar que as classes ator_sistema e *stakeholder* não estão presentes na proposta apresentada em [8] mas foram criadas para classificar

elementos importantes que devem constar nos meta-modelos de rastreamento para a JGOOSE (Figuras 6 e 7).

Ainda no modelo SD, nessa mesma relação ator_sistema→dependum→stakeholder, se a dependência (dependum) for do tipo goal e o Stakeholder (dependee) não possuir dependências como dependem em relação ao ator_sistema, o goal será classificado como Objetivo do Sistema. Também, uma dependência do tipo goal pode ser classificada como Informação Organizacional quando representa uma dependência entre dois atores classificados como stakeholders.

Tabela 1. Diretrizes Propostas por Toranzo [8]

Diretriz	Descrição
Diretriz 1	Identificar classes no segmento de informação externo.
Diretriz 2	Identificar classes no segmento organizacional.
Diretriz 3	Identificar classes no segmento de informação gerencial.
Diretriz 4	Identificar classes no segmento de desenvolvimento: identificar subsistemas.
Diretriz 5	Identificar classes no segmento de desenvolvimento: identificar requisitos.
Diretriz 6	Identificar classes no segmento de desenvolvimento: identificar diagramas.
Diretriz 7	Identificar classes no segmento de desenvolvimento: identificar programas.
Diretriz 8	Identificar classes no segmento de desenvolvimento: identificar teste.
Diretriz 9	Remover as classes de informação irrelevantes.
Diretriz 10	Integrar as classes com o mesmo significado.
Diretriz 11	Integrar novas classes.
Diretriz 12	Organizar as classes.
Diretriz 13	Estabelecer Relacionamentos.
Diretriz 14	Recomendar atributos sobre as classes.
Diretriz 15	Definir uma matriz para cada relacionamento do modelo.
Diretriz 16	Validar o modelo de rastreamento.

Por fim, na relação stakeholder→dependum→ator_sistema todo dependum é classificado na classe Requisitos, já que estes elementos representam requisitos de atores da organização que necessitam ser satisfeitos pelo sistema computacional.

Já utilizando as diretrizes 1 a 8 no modelo SR do framework i*(Figura 2) é necessário observar as dependências do modelo SD que podem ser satisfeitas internamente (razões internas - SRs) pelos atores que atuam como dependee na relação. Inicialmente consideramos as SRs do ator que representa o sistema computacional, expressas através dos elementos goals, tasks, resources e softgoals e relacionamentos entre os mesmos do tipo meio-fim e decomposição de tarefas. Observando estas SRs é possível concluir que os elementos tasks e resources pertencem ao segmento de informação Desenvolvimento, sendo classificados na classe de Requisitos já que apresentam características que proporcionam a visão interna do sistema computacional.

Já os goals são classificados como Objetivos do Sistema pertencendo ao segmento Gerencial, já que o sistema busca satisfazer estes objetivos por meio de suas funcionalidades. Por fim, os softgoals representam requisitos não-funcionais que o sistema deve satisfazer e as tasks são operações que o sistema deve atender, pertencendo desta forma ao segmento de informação Desenvolvimento.

O outro artefato que deve ser classificado seguindo as diretrizes (Tabela 1) é casos de uso. Nos modelos de Casos de Uso existem dois elementos sendo o Ator (Actor) e os Casos de Uso (Use Cases) (Figura 4). Os casos de uso foram associados ao nível de informação Desenvolvimento, pois nesta fase do projeto verifica-se por meio de um diagrama quais são os requisitos funcionais do sistema a serem implementados e, por este motivo, pertencem à classe Requisitos. Actors são stakeholders que possuem objetivos associados aos use cases, sendo assim, mapeados para a classe Stakeholders.

As Figuras 1, 2, 3 e 4 apresentam um resumo dos elementos dos modelos classificados. Já a Figura 5 apresenta a classificação desses elementos em segmento de informação e classe. Em seguida, aplicamos as diretrizes 1 a 8 nos elementos pertencentes à modelagem BPMN (Figura 3).

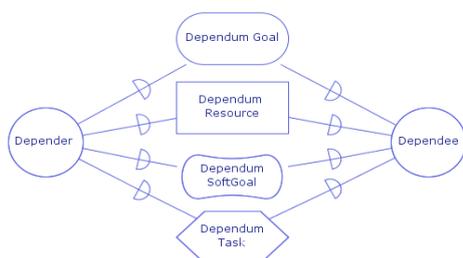


Fig. 1. Modelo SD

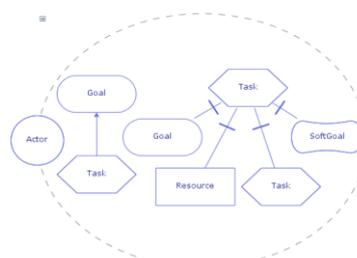


Fig. 2. Modelo SR

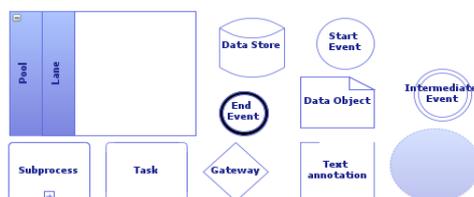


Fig. 3. Elementos básicos do BPMN

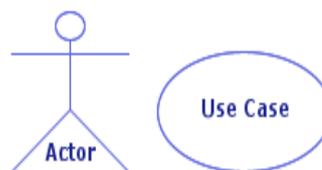


Fig. 4. Elementos de Casos de Uso

Events são utilizados no contexto de questões externas e organizacionais levando à conclusão de que devem pertencer aos segmentos de Informações Organizacional e Externo. Já os *Artifacts* são elementos organizacionais e pertencem à classe InformaçãoOrganizacional quando complementam o modelo com novas informações organizacionais. Contudo, há o caso específico dos *Artifacts Data-Store* e *Data-Object* que pertencem ao segmento Gerencial e classe ObjetivodoSistema quando representam um resultado de uma *task* ou de um *subprocess* que precisa ser obtido no modelo de negócios.

Modelagem i*	Segmento(s) de Informação → Classe
Actor	Sem segmento de Informação → Stakeholder Sem segmento de Informação → AtorSistema
Task	Externo → InformaçãoExterna Desenvolvimento → Requisito
Softgoal	Externo → InformaçãoExterna Desenvolvimento → Requisito
Goal	Externo → InformaçãoExterna Organizacional → InformaçãoOrganizacional Gerencial → ObjetivoSistema
Resource	Externo → InformaçãoExterna Desenvolvimento → Requisito
Modelagem Caso de Uso	Segmento(s) de Informação → Classe
Actor	Sem segmento de Informação → Stakeholder
Use Case	Desenvolvimento → Requisito
Modelagem BPMN	Segmento(s) de Informação → Classe
Events	Externo → InformaçãoExterna Organizacional → InformaçãoOrganizacional
Artifacts	Externo → InformaçãoExterna Organizacional → InformaçãoOrganizacional Gerencial → ObjetivoSistema
Gateways	Organizacional → InformaçãoOrganizacional
Swirlanes	Sem segmento de Informação → Stakeholder
Activities	Desenvolvimento → Requisito

Fig. 5. Classificação dos artefatos suportados pela ferramenta JGOOSE

Além desta funcionalidade, o *Data-Store* pode pertencer ao nível Externo quando servir como base de dados externa ao modelo de negócio, ou seja, vinculado à classe InformaçãoExterna. Os *Gateways* devem pertencer ao segmento de informação Organizacional, já que estes elementos decidem por meio de regras organizacionais qual(is) o(s) fluxo(s) de execução a serem tomados. As *Activities* apresentam características que permitem associá-las ao segmento de informação Desenvolvimento, pois *task* e *subprocess* podem prover uma visão interna de ações dentro do sistema

computacional pertencendo à classe Requisitos (apresentam requisitos do sistema). Isto ocorre quando *Activities* estão dentro de uma *Swinlane* classificada como *stakeholder*.

Por fim, ressalta-se que a utilização das diretrizes 9 a 12 não impactou na remoção de classes de informação irrelevantes e também não foi necessário integrar as classes com mesmo significado em nenhuma das classificações.

3.1 Meta-modelos Propostos

A diretriz 13 (Tabela 1) visa estabelecer relacionamentos entre os elementos do modelo. A utilização do rastreamento vertical e horizontal no contexto da ferramenta JGOOSE possibilita a rastreabilidade de requisitos. A representação de meta-modelos Horizontal e Vertical é proposta neste trabalho. Os tipos de associação entre classes são descritos por [8] e são usadas abreviações para cada tipo de associação, sendo elas: Generalização, Aloca (aloc), Agregação, Satisfação (sat), Recurso (rec), Responsabilidade (resp) e Representação (rep). Para os tipos de associação Generalização e Agregação não existem abreviaturas e possuem sua definição conforme a UML [20].

A rastreabilidade vertical (Figura 7) permite visualizar a vida do requisito ao longo das várias fases do processo de software. Neste sentido, cabe destacar que as classes Subsistema, Diagrama, Programa e ModelodeRaciocínio não existem na ferramenta JGOOSE, porém estes artefatos serão integrados em projetos futuros, o que conduz a já os incluir no meta-modelo de rastreamento. Já a rastreabilidade Horizontal (Figura 6) permite a visualização das relações entre elementos em uma mesma fase do processo de software.

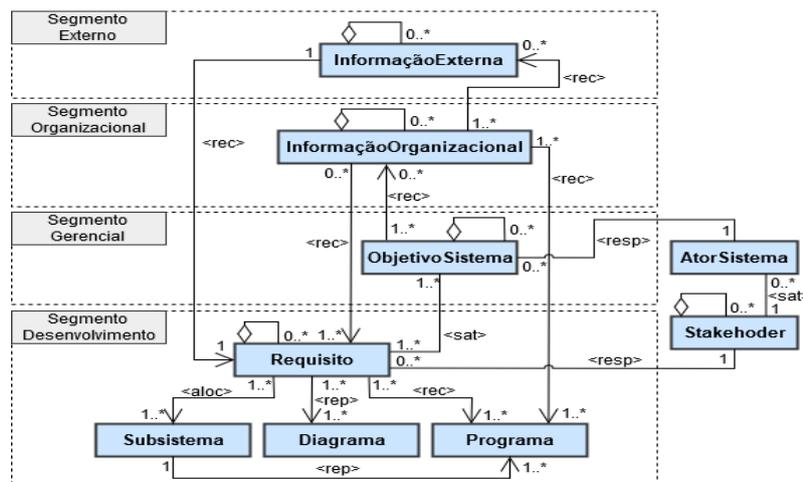


Fig. 6. Meta-Modelo Horizontal

Por exemplo, relacionamentos entre requisitos na engenharia de requisitos ou entre elementos do framework i* na fase de modelagem organizacional. Observando a figura 6 verificamos que a classe **InformaçãoExterna** pertence ao nível de segmento Externo, a classe **InformaçãoOrganizacional** pertence ao nível de segmento Organizacional, a classe **Objetivo Sistema** pertence ao nível de segmento Gerencial e a classe **Requisito**, **Subsistema**, **Diagrama** e **Programa**, pertencem ao nível de segmento Desenvolvimento. Na Figura 7, as classes **InformaçãoExterna** e **InformaçãoOrganizacional** pertencem às fases de *Requisitos Iniciais (Early Requirements)*, *Requisitos finais (Late Requirements)* e *Projeto Arquitetural*. As classes **Requisito** e **Objetivo Sistema** pertencem às fases *Requisitos Finais* e *Projeto Arquitetural*. A classe **ModelodeRaciocínio** pertence à fase *Projeto Arquitetural*. A classe

Diagrama faz parte da fase Projeto Detalhado. Por fim, na fase de Implementação as classes que fazem parte são Programa e Subsistema.

Seguindo a diretriz 14, (especificação de atributos sobre as classes) todas as classes pertencentes aos meta-modelos Horizontal e Vertical terão como atributos um ID para a identificação na matriz de rastreabilidade e um atributo Nome (nome do artefato gerado).

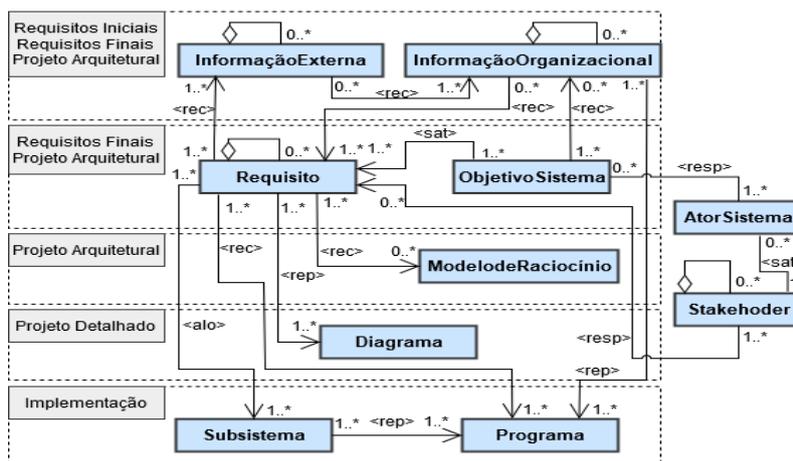


Fig. 7. Meta-Modelo Vertical

A classe Requisito terá um atributo responsável por identificar se este requisito é funcional ou não-funcional. Para a diretriz 15 (elaboração de matrizes de relacionamento entre as classes), a matriz contém: no canto superior esquerdo apresenta-se a abreviação do tipo da associação e tipo de associação (“→” ou “←”) indicando o sentido da relação. A associação entre elementos pode ser representada pelo símbolo “A” (Alta dependência), “M” (média dependência) ou “B” (baixa dependência), nas intersecções entre colunas /VS linha (Figura 9, seção 4). Cada associação do diagrama de classes gera uma possível matriz de rastreamento. Entre as possíveis matrizes de rastreamento a serem geradas podemos destacar: Associação do tipo Agregação entre as classes Requisito x Requisito; Associação do tipo Recurso, entre as classes InformaçãoExterna x Requisito; Satisfação, entre as classes Requisito x ObjetivoSistema; Associação do tipo Recurso, entre as classes Requisito x InformaçãoOrganizacional; Responsabilidade, entre as classes Stakeholder e Requisito. Na seção 5 são apresentados alguns exemplos destas matrizes.

4. Implementação do suporte à rastreabilidade na JGOOSE

O processo de rastreamento de requisitos aplicando as classificações apresentadas na seção anterior pode ser um processo demorado e suscetível a falhas humanas, já que requer uma análise de todos os artefatos presentes à medida que se tornam mais complexos. Com o objetivo de automatizar os processos de rastreabilidade vertical e horizontal foi implementado um rastreamento de requisitos na ferramenta JGOOSE com a finalidade de suportar a rastreabilidade a partir de modelos organizacionais i*, BPMN e Casos de uso.

Na rotina de rastreamento horizontal, o objetivo principal é encontrar os elementos de um modelo BPMN, i* ou Casos de uso previamente selecionado. Na rastreabilidade vertical, antes de realizar o rastreamento é verificado se um dos modelos BPMN ou i* foi utilizado para derivar os casos de uso. Se isto não ocorreu é retornada ao usuário uma mensagem de erro informando a necessidade de derivação dos casos de uso. O usuário pode então usar a opção “Generate use cases” para esse fim

(Figura 8) [12]. Isto ocorre porque o rastreamento vertical prevê a relação de elementos e artefatos entre fases do processo de software. Neste caso, entre a *early* e *late requirements*.

Concluída a identificação do tipo de rastreamento, inicia-se a rotina de identificação dos elementos a serem rastreados e armazenamento dos mesmos em estruturas de dados (armazena informações obtidas: código do elemento e nome). Primeiramente é realizada a análise dos elementos instanciando-os nas classes de rastreabilidade: InformaçãoExterna, InformaçãoOrganizacional, ObjetivoSistema, Requisitos, Stakeholder, AtorSistema, ModeloRaciocinio, Programa, Diagrama ou Subsistema (Figuras 6 e 7) de acordo com a situação analisada (seção 3).

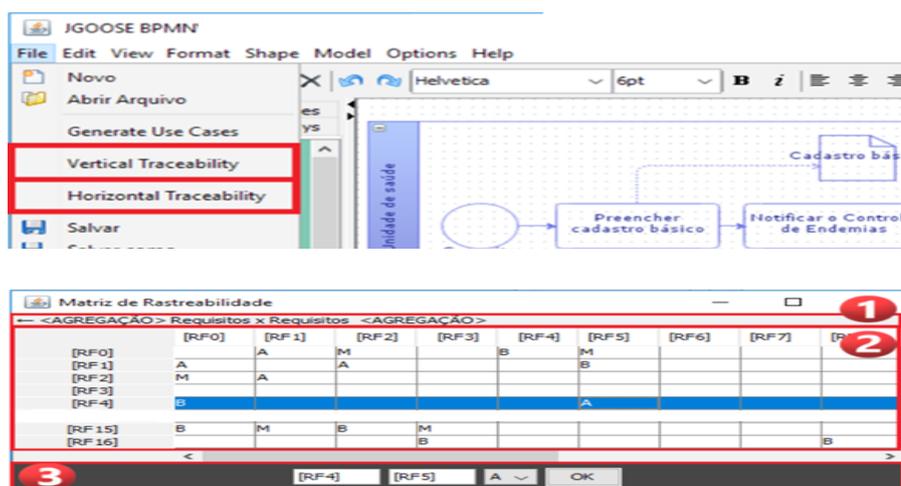


Fig. 9. Matriz de rastreabilidade

Após o rastreamento de todos os elementos pertencentes a um modelo verifica-se a existência de conflitos e dependências entre os elementos, sendo estes conflitos armazenados da seguinte forma: código e seu grau de dependência (alta, média ou baixa).

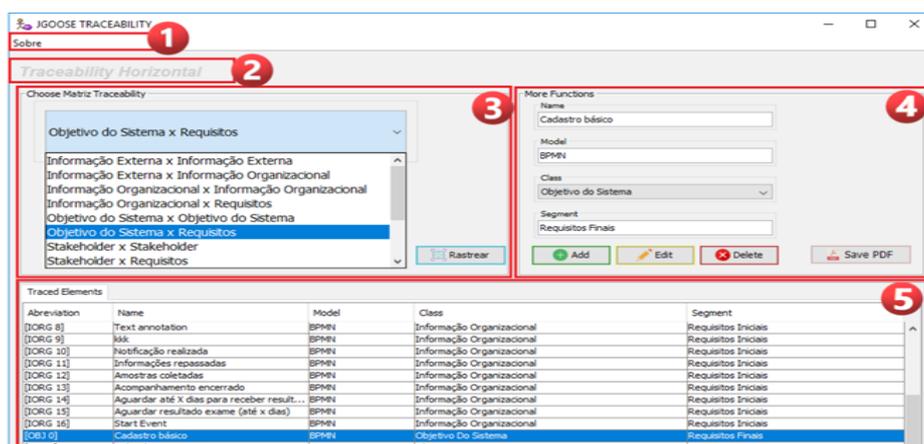


Fig. 10. Interface de rastreabilidade

Para facilitar a visualização das informações rastreadas foi implementada uma interface de visualização de Rastreabilidade (Figura 10), a qual foi dividida em 5 áreas diferentes: (1) obter mais informações sobre a rastreabilidade; (2) tipo da rastreabilidade escolhida; (3) escolha de matriz de rastreabilidade a ser gerada (botão rastrear exibe a matriz escolhida); (4) mais funcionalidades – ao

selecionar o elemento (permite incluir, excluir e modificar elementos rastreados e gerar um documento em formato *Portable Document Format – PDF*); (5) tabela de elementos rastreados.

A matriz de rastreamento gerada (Figura 9) é dividida em três partes: (1) título: permite ao usuário saber o título da matriz assim como o tipo de associação entre as duas classes rastreadas; (2) matriz: permite ao usuário ver a matriz, sendo os elementos rastreados representados por suas abreviações na primeira coluna e na primeira linha, e o grau da dependência destes elementos nas intersecções da matriz; (3) editar: caso o usuário desejar mudar o tipo de dependência entre dois elementos, basta clicar na intersecção desejada, alterar o grau de dependência e clicar em “OK”. Utiliza-se a mesma lógica de apresentação para todas as matrizes geradas na JGOOSE.

5. Estudo de viabilidade

Esta seção apresenta um estudo de viabilidade realizado seguindo a metodologia proposta em [25]. O objetivo foi avaliar se os recursos de rastreabilidade vertical e horizontal implementados na JGOOSE geram resultados satisfatórios (tempo e recursos) e se apresentam resultados positivos em relação à utilidade do rastreamento de requisitos para o usuário.

5.1 Planejamento do Estudo de Viabilidade

O *Goal/Question/Metric* (GQM) [26] é utilizado para definir o objetivo e as questões a serem respondidas no estudo de viabilidade. Para o planejamento são definidos objetivos, sendo: objetivo global e objetivo do estudo. O *objetivo global*: avaliar a implementação do suporte à rastreabilidade horizontal e vertical de requisitos na ferramenta JGOOSE. Já o *objetivo do estudo*: analisar os elementos presentes nos modelos BPMN, i* e Casos de Uso e o seu mapeamento para matrizes de rastreamento, com a finalidade de avaliar a correte e utilidade das matrizes geradas do ponto de vista dos autores do trabalho assumindo o papel de engenheiros de requisitos. O contexto do estudo é uma avaliação acadêmica com exemplos já utilizados pelos autores em trabalhos prévios.

Tabela 2. Questões e métricas adotadas

Questões (Q) Diretriz	Métricas (M)
Q01: A rastreabilidade Horizontal gerada pela ferramenta a partir de modelos BPMN, i* e Casos de Uso está correta?	M01: matrizes de rastreamento de requisitos bem como relações e elementos extraídos dos artefatos encontrados nos modelos BPMN, i* e Casos de Uso.
Q02: A rastreabilidade Vertical gerada pela ferramenta a partir de modelos BPMN para Casos de Uso está correta?	M02: matrizes de rastreamento de requisitos bem como relações e elementos extraídos dos artefatos encontrados nos modelos BPMN e Casos de Uso gerados automaticamente pela ferramenta.
Q03: A rastreabilidade Vertical gerada pela ferramenta a partir de modelos i* para Casos de Uso está correta?	M03: matrizes de rastreamento de requisitos bem como relações e elementos extraídos dos artefatos encontrados nos modelos i* e Casos de Uso gerados automaticamente pela ferramenta.
Q04: As informações obtidas são úteis?	M04: as informações rastreadas tanto horizontalmente quanto verticalmente fazem sentido e podem ser utilizadas por engenheiros de requisitos, gerentes de projetos e analistas.
Q05: O tempo de resposta para a obtenção das informações rastreadas é aceitável?	M05: as informações rastreadas são obtidas em tempo aceitável de acordo com o ponto de vista esperado por engenheiros de requisitos, gerentes de projetos e analistas.

As Questões (Q)/Métricas (M) (Tabela 2) seguem as premissas definidas pelo GQM [26], o qual especifica métricas como um conjunto de dados, o qual é associado a todas *Questions* com a finalidade de respondê-las (quantitativamente) e os dados poderão ser: objetivos ou subjetivos.

O primeiro exemplo usado foi extraído de [2] o qual modela usando o framework i* uma organização que busca com o apoio de um sistema computacional realizar o processo de compra e venda. Os casos de uso para esse sistema são derivados usando a JGOOSE. O segundo exemplo foi extraído

de [10] e corresponde aos modelos BPMN e casos de uso gerados via JGOOSE para um processo de negócio de identificação e acompanhamento de um caso suspeito de uma doença endêmica.

5.2 Resultados do Estudo de Viabilidade

Para o segundo exemplo - BPMN para casos de uso, após a modelagem e derivação dos casos de uso é possível visualizar na Figura 11 uma das matrizes geradas. Esta matriz apresenta a relação entre *stakeholders* classificados dos elementos do BPMN e *stakeholders* classificados dos modelos de Casos de Uso. Por exemplo, [STK 6] “Unidade de Saúde” pertencente ao modelo BPMN e associado à classe *Stakeholder*, possui uma alta dependência com o [STK 0] “Unidade de Saúde” que pertence ao modelo de Use Cases e à classe *Stakeholder*. Isto pode indicar ao engenheiro de requisitos que como esse membro da organização foi mapeado para ator em casos de uso, qualquer mudança no processo organizacional envolvendo esse *stakeholder*, pode refletir em mudanças nos casos de uso associados ao mesmo.

Tabela 3. Alguns dos elementos dos modelos BPMN e Casos de Uso mapeados verticalmente

Abreviation	Name	Model	Class
[STK 0]	Unidade de saúde	BPMN	Stakeholder
[STK 6]	Unidade de saúde	Use Cases	Stakeholder
[RF33]	Comunicar Controle de Endemias	Use Cases	Requisito

← <AGREGAÇÃO> Stakeholder x Stakeholder <AGREGAÇÃO>												
	[STK 0]	[STK 1]	[STK 2]	[STK 3]	[STK 4]	[STK 5]	[STK 6]	[STK 7]	[STK 8]	[STK 9]	[STK 10]	[STK 11]
[STK 0]							A					
[STK 1]								A				
[STK 2]									A			
[STK 3]										A		
[STK 4]											A	
[STK 5]												A
[STK 6]	A											
[STK 7]		A										
[STK 8]			A									
[STK 9]				A								
[STK 10]					A							
[STK 11]						A						

Fig.11. Matriz de rastreabilidade vertical *Stakeholder x Stakeholder*

Para a rastreabilidade vertical do primeiro exemplo - modelo *i** para casos de uso (Tabela 4 e Figura 12) apresentam os elementos rastreados e a matriz de rastreabilidade. Esta última apresenta a responsabilidade de um *stakeholder* em relação a um requisito.

Tabela 4. Amostra dos elementos dos Artefatos *i** e Casos de Uso mapeados verticalmente

Abreviation	Name	Model	Class
[AS 0]	Sistema	<i>i*</i>	Ator Sistema
[STK 0]	Funcionário	<i>i*</i>	Stakeholder
[RF15]	Consultar Produtos	Use Cases	Requisito
[RF17]	Emitir Nota Fiscal	Use Cases	Requisito

← <RESP> Stakeholder x Requisitos <RESP>																		
	[RF0]	[RF1]	[RF2]	[RF3]	[RF4]	[RF5]	[RF6]	[RF7]	[RF8]	[RF9]	[RF10]	[RF11]	[RF12]	[RF13]	[RF14]	[RF15]	[RF16]	[RF17]
[STK 0]																		
[STK 1]	A	A	A	A												A	A	A
[STK 2]																		

Fig.12. Matriz de Responsabilidade entre *Stakeholder x Requisito*

Esta matriz possibilita identificar, por exemplo, que o [STK 0] “Funcionário” (origem no modelo *i** e classificado na classe *Stakeholder*) possui alta dependência (altamente responsável) com o [RF 15] “Consultar Produtos” (modelo *Use Cases* e classe Requisito). Isto implica em afirmar que qualquer mudança envolvendo esses dois elementos terá um forte impacto em ambos. De forma mais geral, esta matriz auxilia engenheiros de requisitos a obter uma visão ampla de quais requisitos estão

associados a quais *stakeholders* e vice versa, podendo ser vital em uma mudança em potencial. Para investigar a rastreabilidade horizontal nos dois exemplos adotados, foi realizada a modelagem usando os editores da JGOOSE e utilizada a opção na interface da ferramenta (Figura 8) para gerar as matrizes horizontais. As Tabelas 5, 6 e 7 apresentam alguns elementos dos artefatos rastreados bem como as Figuras 13, 14 e 15 apresentam exemplos de matrizes horizontais.

Tabela 5. Alguns elementos do artefato BPMN rastreados horizontalmente

Abreviation	Name	Model	Class
[STK 0]	Unidade de saúde	BPMN	Stakeholder
[OBJ 3]	Formulário busca ativa	BPMN	Objetivo Do Sistema
[RF2]	Preencher cadastro básico	BPMN	Requisito

	[OBJ 0]	[OBJ 1]	[OBJ 2]	[OBJ 3]
[IORG 0]			M	
[IORG 1]		B		
[IORG 2]		M		
[IORG 3]				
[IORG 4]		B	B	
[IORG 5]				
[IORG 6]				M
[IORG 7]				
[IORG 8]				B
[IORG 9]				
[IORG 10]			M	
[IORG 11]		M		
[IORG 12]				B

Fig. 13. Recurso entre Objetivo do Sistema x Informação Organizacional

As matrizes de rastreabilidade (Figuras 13, 14 e 15) apresentam a agregação entre a classe “Objetivo de Sistema” e a classe “Informação Organizacional”, entre elementos da classe “Requisitos”, e entre elementos da classe “Stakeholders”, respectivamente.

Por exemplo, na matriz de rastreabilidade da figura 15 é possível identificar que o [STK 3] “*Entomologista*” possui uma alta dependência com o [STK 5] “*Controle de Endemia*”. Isto pode indicar ao engenheiro de requisitos que qualquer mudança no escopo de um stakeholder, por exemplo, restrições ou ampliações com processos ou casos de uso do sistema, implicará na revisão do escopo do stakeholder dependente.

Tabela 6. Alguns artefatos do modelo i* rastreados horizontalmente

Abreviation	Name	Model	Class
[AS 0]	Sistema	i*	Ator Sistema
[STK 0]	Funcionário	i*	Stakeholder
[STK 1]	Cliente	i*	Stakeholder
[IORG 0]	Dados para Pagamento	i*	Informação Organizacional
[IORG 1]	Fazer Compra	i*	Informação Organizacional
[RF0]	Consultar Produtos	i*	Requisito

	[RF0]	[RF 1]	[RF 2]	[RNF 3]	[RF 4]	[RF 5]	[RF 6]	[RF 7]	[RF 8]	[RF 9]	[RF 10]	[RF 11]	[RF 12]	[RF 13]	[RF 14]
[RF0]															
[RF 1]	M														
[RF 2]	M	M													
[RNF 3]	M	M	M												
[RF 4]	B			A											
[RF 5]	B	M		A	M										
[RF 6]	B	M	A	A			M	M							
[RF 7]	B	M	B	A			M	M							
[RF 8]	B	M	B	A			M	M							
[RF 9]	B			A											
[RF 10]	B			A						M					
[RF 11]	M	B		A	A	A	B	B	B						
[RF 12]	M	B		A	B	B	B	M	B	A	A	B			
[RF 13]	M	A	M	A	B	B	A	A	B	B	M	M			
[RF 14]	A	M	B	A	M	M	A	M	M	A	A	A			

Fig. 14. Agregação entre Requisito x Requisito

Considerando todos os artefatos rastreados pela JGOOSE horizontalmente e verticalmente, as questões definidas no planejamento GQM (subseção anterior) foram respondidas conforme segue: todas as métricas (M01, M02, M03, M04 e M05) obtiveram a resposta “sim”. Em relação às métricas M01, M02 e M03 todos os elementos mapeados presentes nas Tabelas 3, 4, 5, 6 e 7 tem origem nos

elementos presentes nos artefatos modelados (BPMN e i*) e derivados (Casos de uso). Isso condiz que nenhum elemento a mais foi criado ou excluído do mapeamento automático gerado pela ferramenta JGOOSE, gerando assim uma consistência sobre estes artefatos mapeados tanto horizontalmente como verticalmente. Já as matrizes apresentadas anteriormente também abordam somente os elementos os quais estão presentes nas tabelas desta seção e as intersecções das matrizes contemplam o grau de dependência mapeado.

Tabela 7. Alguns artefatos do modelo Casos de Uso rastreados horizontalmente

Abreviation	Name	Model	Class
[STK 0]	Unidade de saúde	Use Cases	Stakeholder
[STK 1]	Suspeito / Contato	Use Cases	Stakeholder
[STK 2]	Laboratório	Use Cases	Stakeholder
[STK 3]	Entomologista	Use Cases	Stakeholder
[STK 4]	Secretária	Use Cases	Stakeholder
[STK 5]	Controle de Endemia	Use Cases	Stakeholder

← <AGREGAÇÃO> Stakeholder x Stakeholder <AGREGAÇÃO>						
	[STK 0]	[STK 1]	[STK 2]	[STK 3]	[STK 4]	[STK 5]
[STK 0]						
[STK 1]					M	
[STK 2]				M		A
[STK 3]			M			A
[STK 4]		M				
[STK 5]			A	A		

Fig.15. Agregação entre Stakeholder x Stakeholder

A resposta sobre a métrica M04 considera que as matrizes de rastreamento e tabelas sobre elementos mapeados são úteis e fazem sentido permitindo ajudar os engenheiros de requisitos na tomada de decisões envolvendo mudanças de requisitos, alterações nos modelos BPMN, i* e Casos de Uso, verificação de inconsistências entre artefatos, visualização da complexidade de um requisito através da análise do grau de dependência do mesmo em relação a outros elementos de outros artefatos, dentre outros aspectos úteis advindos da capacidade de poder rastrear as informações.

Referente à métrica M05, detectou-se que o mapeamento automático dos elementos dos artefatos usados para criar as matrizes de rastreabilidade é quase instantâneo demorando menos de 3 segundos para todos os modelos construídos. Isto demonstra um tempo aceitável, visto que a geração manual deste mapeamento demandaria algumas horas.

Assim, é possível constatar que a utilização do suporte à rastreabilidade implementada na ferramenta JGOOSE torna-se viável considerando que os artefatos rastreados em conjunto com as matrizes de rastreabilidade podem contribuir para as etapas posteriores do processo de software e os resultados são obtidos de forma rápida não necessitando utilizar uma ferramenta externa ou até mesmo esforço manual o que traria inúmeras desvantagens.

6. Considerações finais

Neste artigo, apresentou-se a proposta de implementação de rastreabilidade de requisitos alicerçado em [8] gerando dois meta-modelos de rastreabilidade, Vertical e Horizontal. A elaboração e a implementação da rastreabilidade de requisitos desta nova funcionalidade (JGOOSE) auxilia profissionais na área de engenharia de software, viabilizando a análise de possíveis impactos de mudanças dos artefatos. Esta funcionalidade ainda corrobora na descoberta de inconsistência nos artefatos, possibilitando a identificar, de forma rápida, das dependências entre os artefatos mapeados.

O processo de rastreabilidade, utilizando as diretrizes propostas por [8], ajudou na classificação de artefatos presentes na ferramenta JGOOSE, gerando assim os meta-modelos de rastreabilidade

Horizontal e Vertical consistentes e bem definidos (a rastreabilidade de requisitos foi implementada na ferramenta JGOOSE).

As limitações desta pesquisa estão relacionadas a utilização de exemplos de modelos de menor complexidade e pouca diversificação de exemplos de modelos. Outro aspecto limitante é o número reduzido de avaliadores, o qual se restringiu apenas aos autores no processo de avaliação da rastreabilidade de requisitos nas matrizes vertical e horizontal.

Para os trabalhos futuros, há possibilidades de realizar melhorias na interface construída para gerar as matrizes de rastreamento, refinamento do processo de rastreamento e geração de matrizes, elaboração de mecanismos de auxílio na compreensão de cada etapa pelo usuário e definição de novas diretrizes de rastreabilidade. Também consideramos a ampliação e aprimoramento do processo de avaliação bem como realizar todo o processo de validação proposto por [25] utilizando uma equipe com analistas, engenheiros de requisitos e gerentes de projetos, buscando satisfazê-los e cumprir todas as etapas do estudo sendo estes: Estudo de Viabilidade, Estudo Observacional, Estudo de Caso e Estudo de Caso na indústria. Por fim, a realização de experimentos como *survey* e/ou experimento controlado considerando os diferentes benefícios advindos de cada uma destas categorias propostas na Engenharia de Software Experimental.

Referências

1. OUTA, C. T.; SANTANDER, V. F. A. O uso de modelos de processos de negócio e de modelagem organizacional em metodologias ágeis: uma revisão sistemática da literatura. In: Workshop on Requirements Engineering, 2019, Recife-PE, Brasil, agosto 13-16, 2019.
2. PELISER, D.; SANTANDER, V. F. A.; FREITAS, I.; ANDRADE, S. C.; SCHEMBERGER, E. E4J Use Cases: um editor de diagrama de casos de uso integrado à ferramenta JGOOSE. In: 35th International Conference of the Chilean Computer Science Society (SCCC 2016), Valparaíso, Chile. NY 12571 USA: IEEE Catalog Number CFP16139-ART, 2016
3. BPMN. Business Process Model and Notation (BPMN) Version 2.0.2. [S.l.]. 2013.
4. YU, E. S. K. MODELLING STRATEGIC RELATIONSHIPS FOR PROCESS REENGINEERING. Tese (Tese de Doutorado) — University of Toronto, 1995.
5. MERLIM, L. P.; SILVA, A. L. B.; SANTANDER, V. F. A.; SILVA, I. F.; CASTRO, J. F. B. (2015) Integrating the E4J editor to the JGOOSE tool. In: XVIII Workshop de Engenharia de Requisitos (WER), Universidad Ricardo Palma, Lima.
6. GOTEL, O. Contribution Structures for Requirements Engineering. PhD (Tese). Imperial College of Science, Technology and Medicine, Department of Computing, U.K. 1996.
7. KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering: Processes and Techniques. 1st. ed. [S.l.]: Wiley Publishing, 1998. ISBN 0471972088, 9780471972082.
8. TORANZO, M. Uma Proposta para Melhoria do Rastreamento de Requisitos de Software. Tese (Doutorado). Centro de Informática, Universidade Federal de Pernambuco. 2002.
9. RAMESH, B.; EDWARDS, M. (1993). Issues in the Development of a Model of Requirements Traceability. In Proceedings of the 1st International Symposium on Requirements Engineering, IEEE Computer Society Press. San Diego, USA.
10. GIROTTO, A. N.; SANTANDER, V. F. A. (2017). Uma proposta para derivar Casos de Uso a partir de modelos BPMN com suporte computacional. Universidade Estadual do Oeste do Paraná, Cascavel-PR.
11. SANTANDER, V. F. A.; CASTRO, J. F. B. Deriving use cases from organizational modeling. IEEE Joint International Requirements Engineering Conference - RE' 02, Essen Germany, p. 32–39, 2002.
12. GERALDINO, G. C. L.; SANTANDER, V. F. A. (2019) The JGOOSE Tool. 12th International i* Workshop (iStar 2019) in: 38th International Conference on Conceptual Modeling (ER 2019). Salvador, Brasil.

13. VICENTE, A. A.; SANTANDER, V. F. A.; CASTRO, J. B.; FREITAS DA SILVA I.; REYES, M.; FRANCISCO, G. JGOOSE: A Requirements Engineering Tool to Integrate i* Organizational Modeling with Use Cases in UML. *Ingeniare. Revista chilena de ingeniería* 17, 6-20. 2009.
14. COCKBURN, A. *Writing Effective Use Cases*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 2002.
15. JGOOSE. Java Goal into Object Oriented Standard Extension. Laboratório de Engenharia de Software - LES. Disponível em: <www.inf.unioeste.br/~les/index.php/listadownload>. Acessado em: 30.mar.2020.
16. POZZAN, V. A.; SANTANDER, V. F. A. JGOOSE: Uma ferramenta para derivar casos de uso a partir de modelos em i* e BPMN. 2019 (Video). In: *Workshop on Requirements Engineering, 2019, Recife-PE, Brasil, 13-16.ago.2019*
17. YOSHIDOME, E.; SOUZA, M. R. D. A.; LIRA, W. M.; OLIVEIRA, S. R. B.; VASCONCELOS, A. M. L. Uma Apoio Sistematizado à Implementação do Processo de Desenvolvimento de Requisitos do MPS. BR e CMMI a partir do Uso de Ferramentas de Software Livre. In *WER. 2012*.
18. JUNIOR, A. B. C.; BENTES, L. N.; RONNY, M.; OLIVEIRA, S. R. B.; YOSHIDOME, E. Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS. BR. In *WER. 2010*.
19. DE GEA, J. M. C.; NICOLÁS, J.; ALEMÁN, J. L. F.; TOVAL, A.; EBERT, C.; VIZCAÍNO, A. Requirements engineering tools: Capabilities, survey and assessment. *Information and Software Technology*. 2012. 54(10), 1142-1157.
20. BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *Unified Modeling Language User Guide, The (2Nd Edition) (Addison-Wesley Object Technology Series)*. [S.l.]: Addison-Wesley Professional, 2005. ISBN 0321267974.
21. RAMESH, B. Factors Influencing Requirements Traceability Praticce. *Communication of the ACM*, Vol. 41, No.12, pp. 37-44, 1998.
22. GOTEL, O.; FINKELSTEIN, A. Extended Requirements Traceability Results of an Industrial Case Study. In *Proceedings of the 2nd International Conference on Requirements Engineering, IEEE Computer Society Press*, pp. 169-178. Colorado Springs, USA, 1996.
23. RAMESH, B.; JARKE, M. Towards Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering*, Vol. 27, No.1, pp. 58-93, 2001.
24. SANTANDER, V. F. A.; SILVA, I. F. Avaliando a utilização da Ferramenta JGOOSE no Processo de Ensino e Aprendizagem na Engenharia de Requisitos: Um Relato de Experiência. In: *XIX Conferência Internacional sobre Informática na Educação (TISE)*. 2014.
25. SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. In: *Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. New York, NY, USA: ACM, 2001. (ESEC/FSE-9), p. 288-296.
26. BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. *Encyclopedia of software engineering*, p. 528-532, 1994.
27. DOMINIK, W.; UDO, L. Manage Interdisciplinarity Based on Requirements Traceability - A Graph-based Tool Support for Requirements Traceability. *PICMET '17: Technology Management for Interconnected World*, 2017.
28. ELAMIN, R.; OSMAN, R. Towards Requirements Reuse by Implementing Traceability in Agile Development. *41st Annual Computer Software and Applications Conference - IEEE*, 2017.
29. GOTEL, O.; FINKELSTEIN, A. An analysis of the requirements traceability problem. in *Proceedings of the First international conference on requirements engineering*. Colorado Springs, CO. pp. 94-101, apr, 1994.
30. RAMESH, B.; POWERS, T.; STUBBS, C; EDWARDS, M. Implementing Requirements Traceability: A Case Study. *IEEE International Symposium on Requirements Engineering*, 1995.