

Utilizando a Engenharia de Software Experimental para avaliar a implementação das Regras de Garantia de Consistência na ferramenta JGOOSE

Kauan Campos^[0009-0004-9344-4555], Rafael Ferreira Lima^[0009-0007-3812-0103] and Victor Santander^[0000-0002-0584-4777]

Universidade Estadual do Oeste do Paraná, Cascavel CEP 85819-110, Brasil
kauan.campos@unioeste.br; rafael.limal4@unioeste.br;
victor.santander@unioeste.br

Resumo. A Engenharia de Requisitos de Software busca identificar, documentar e gerenciar requisitos ao longo do ciclo de vida de sistemas, sendo a consistência entre diferentes modelos um desafio recorrente. Modelos como iStar e Casos de Uso da UML são amplamente utilizados, mas sua integração exige mecanismos que assegurem coerência entre artefatos ao longo da evolução do software. Este estudo tem como objetivo avaliar a implementação das Regras de Garantia de Consistência na ferramenta JGOOSE, verificando sua capacidade de manter a consistência semiautomática entre casos de uso e modelos iStar. A pesquisa foi conduzida sob princípios da Engenharia de Software Experimental, utilizando a estratégia de quase-experimento. O planejamento envolveu hipóteses formais, definição de variáveis e aplicação da abordagem GQM para estruturar métricas de avaliação. Os resultados indicam que, mesmo havendo oportunidades para ajustes futuros, as Regras de Garantia de Consistência foram aplicadas corretamente em diversos cenários, garantindo a derivação semiautomática consistente entre os modelos e contribuindo para a evolução da ferramenta.

Palavras-Chave: Engenharia de Software Experimental, Derivação Semiautomática, Ferramenta CASE.

1 Introdução

A Engenharia de Requisitos (ER) é uma área essencial da Engenharia de Software, responsável por identificar, analisar, documentar, validar e gerenciar requisitos ao longo do ciclo de vida de um sistema [1]. Um dos maiores desafios relatados na literatura é garantir a consistência entre diferentes modelos de requisitos [1], de modo que as necessidades dos *stakeholders* sejam corretamente refletidas e mantidas durante a evolução do software. Várias abordagens podem ser utilizadas para modelar requisitos. Cabe destacar o framework iStar [20] e Casos de Uso da UML [4]. O primeiro permite representar o ambiente organizacional e intencionalidades associadas a atores do ambiente com um foco típico de abordagens pertencentes à engenharia de requisitos

orientada a objetivos. O segundo permite descrever as funcionalidades esperadas e interações com o sistema. Nesta perspectiva, em [19] foram propostas diretrizes para derivar Casos de Uso a partir de modelos iStar. Estas diretrizes foram atualizadas em [21] considerando as modificações propostas em [20] que resultaram na iStar 2.0. Uma ferramenta computacional denominada JGOOSE tem sido desenvolvida e evoluída para apoiar a execução semiautomática dessas diretrizes. Contudo, um dos desafios detectados na ferramenta está associado à garantia de consistência de ambos os modelos, especialmente na derivação de modelos iStar a partir de casos de uso. Após gerar o iStar a partir de casos de uso, qualquer mudança realizada nos casos de uso deve ser refletida no modelo iStar associado. Para apoiar a manutenção da consistência entre estes modelos, um conjunto de Regras de Garantia de Consistência (RGCs) foi proposto em [2]. Estas RGCs foram incorporadas à JGOOSE conforme pode ser consultado em [7].

Apesar desse avanço, ainda é necessário avaliar a efetividade dessas regras em cenários reais, envolvendo usuários e experimentação controlada. Para isso, a Engenharia de Software Experimental (ESE) oferece métodos sistemáticos para caracterizar, avaliar e validar práticas e técnicas [6], permitindo analisar a correte e completude das derivações realizadas pela ferramenta. Diante desse contexto, este trabalho tem como objetivo validar as RGCs implementadas na JGOOSE, verificando sua capacidade de manter a consistência entre modelos de casos de uso e modelos iStar. Além disso, busca-se compreender as percepções dos usuários sobre a adequação dos modelos gerados, contribuindo para o avanço das práticas de consistência na Engenharia de Requisitos e para a consolidação da ferramenta como apoio ao processo de modelagem.

Para garantir o rigor metodológico, a estrutura deste artigo reflete as etapas sistemáticas do experimento conduzido. O trabalho está organizado da seguinte forma: na seção 2 apresenta-se o referencial teórico com alguns conceitos importantes para o entendimento do trabalho, na seção 3 o Planejamento do experimento, na seção 4 a Execução do experimento, na seção 5 é apresentada a Análise dos Resultados, as limitações encontradas e sugestões de Trabalhos Futuros e, por fim, na seção 6 as Considerações Finais.

2 Referencial Teórico

As regras de garantia de consistência foram definidas em [2] [24], com o objetivo fundamental de mapear mudanças realizadas em diagramas de casos de uso para o modelo iStar associado. Cabe ressaltar que as regras não consideram as descrições textuais dos casos de uso. As 12 regras abrangem todos os elementos dos dois artefatos a fim de traduzir de forma coerente cada caso de uso, suas associações e interdependências para elementos na iStar 2.0. As regras são descritas na Tabela 1.

No cenário da engenharia de software, casos de uso possuem o objetivo de especificar o comportamento do sistema, definindo sequências de ações para cada requisito e incluindo possíveis variações nestas ações, o que produz um resultado

observável para um ator [4], sendo em suma um conjunto de cenários que representam interações entre o ator e o sistema. O conceito é adotado pela linguagem gráfica *Unified Modeling Language* (UML), sendo a linguagem gráfica padrão para projetos de sistemas [14].

Tabela 1. Cada linha representa uma regra de garantia de consistência e para cada uma há um resumo de sua definição

RGC	Resumo
1	Alterações em casos de uso devem ser refletidas no elemento correspondente do modelo iStar
2	Ator primário dos casos de uso torna-se um ator genérico no iStar
3	O limite do sistema nos casos de uso torna-se o ator “sistema” no iStar
4	Todo caso de uso torna-se uma dependência no iStar, satisfeita pelo ator sistema
5	Relação include torna-se AND no iStar (Execução obrigatória)
6	Relação extend torna-se OR no iStar
7	Subcaso de uso torna-se uma tarefa (AND/OR) no iStar, ligada ao caso pai e gera dependência com os atores relacionados
8	Ator secundário torna-se ator no iStar com dependência de outro ator (sempre o dependente)
9	Associação entre ator e caso de uso torna-se dependência no modelo iStar
10	Herança entre atores torna-se relação IS-A no iStar
11	Remover algo nos casos de uso remove o elemento correspondente no iStar
12	Um caso de uso pai, no iStar só pode ser refinado por AND ou OR, nunca ambos ao mesmo tempo

O framework iStar [3] [20] modela contextos organizacionais, avaliando as interdependências entre atores e elementos relacionados às necessidades do sistema e do ambiente. Assim, permite compreender as intenções de cada ator, fornecendo informações valiosas para a equipe de desenvolvimento.

Com relação a JGOOSE [22], este é um projeto de ferramenta CASE, *open source* de acesso livre pela comunidade acadêmica e está disponível em [7]. Mais especificamente, o software atua como um suporte computacional para o mapeamento de modelos de casos de uso a partir de modelos iStar e BPMN [9] e permite gerar algumas matrizes de rastreabilidade [23] entre esses artefatos.

A Fig. 1 apresenta um resumo das telas envolvidas no experimento realizado e analisado. O editor de casos de uso [8] é utilizado pelos participantes no experimento para modificar os casos de uso, o botão de gerar iStar é utilizado para refletir as mudanças realizadas no modelo iStar. Na Fig. 1 é possível observar a execução genérica da funcionalidade.

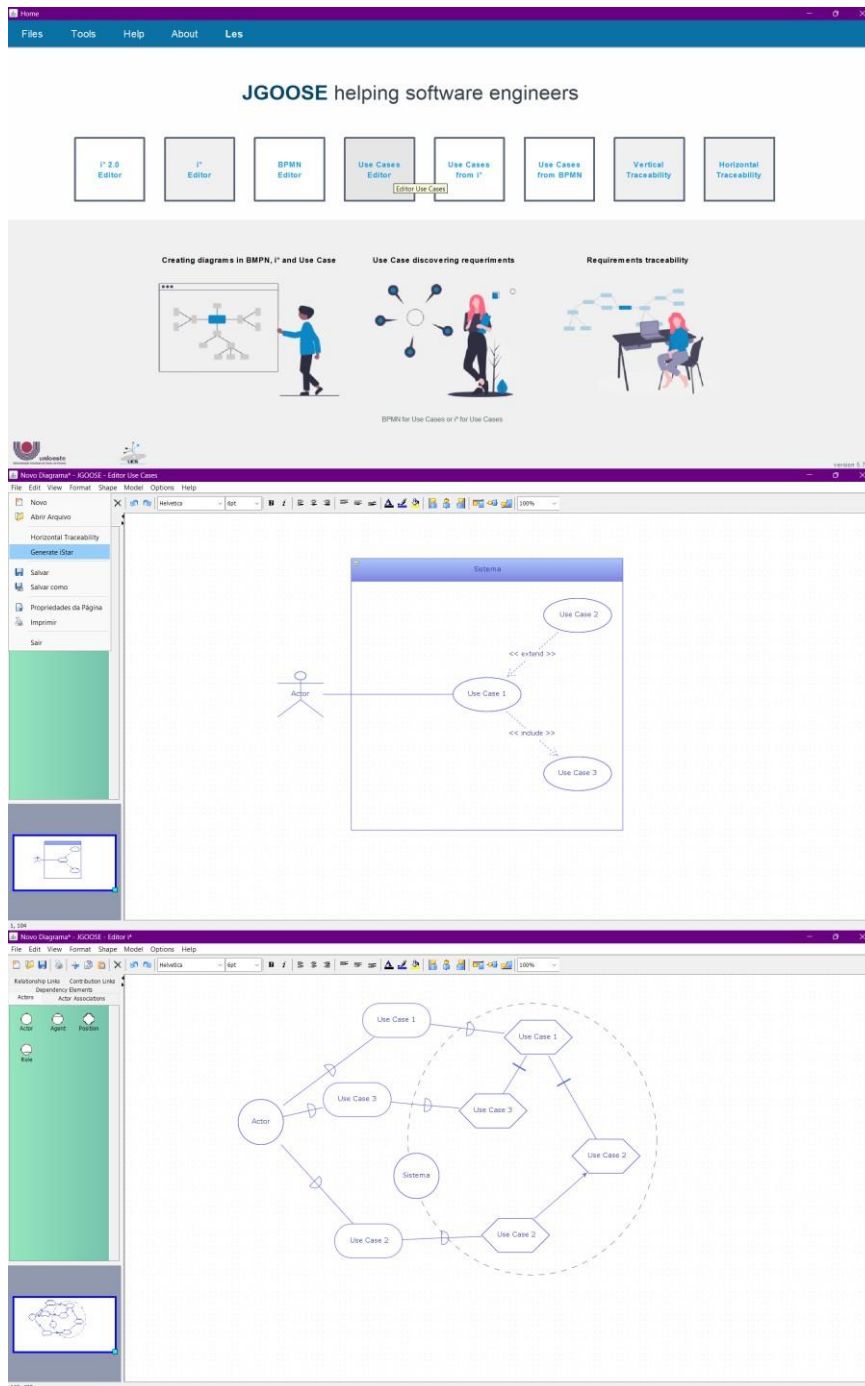


Fig. 1. Fluxo de telas na JGOOSE para derivação semiautomática.

A respeito da ESE, Wohlin [6] propõe que a engenharia de software também é uma ciência de laboratório. Nesse contexto, pesquisadores e profissionais da área visam compreender o objeto de estudo, seu ciclo de vida e a relação entre ambos no contexto do sistema, por outro lado é papel do profissional criar e aprimorar softwares com base nas informações adquiridas e disponíveis. Desta forma, o experimento foi planejado com base em [6] e optou-se por realizar um quase-experimento, dada a natureza do ambiente acadêmico e a disponibilidade dos participantes. Conforme aponta a literatura clássica [11], nessa modalidade os participantes não são atribuídos aleatoriamente aos tratamentos, utilizando-se de grupos de conveniência para a coleta de dados. No cenário desse tipo de experimento, recomenda-se utilizar a técnica GQM [17], a qual visa a partir de um ou mais objetivos definir questões e métricas a fim de executar a análise e interpretação dos dados de forma sistemática.

3 Planejamento do experimento

O experimento foi conduzido por um dos pesquisadores envolvidos no estudo, responsável pelo planejamento, execução das atividades experimentais e análise dos dados coletados, sob supervisão do orientador do projeto. O planejamento experimental seguiu as diretrizes propostas por Wohlin [6], sendo estruturado em etapas, conforme detalhado nos subtópicos a seguir, com o objetivo de sistematizar a condução do experimento e sua aplicação junto aos participantes. Nesse contexto, os participantes foram organizados em 11 grupos, contendo entre 1 e 4 integrantes cada. A formação dos grupos ocorreu por afinidade e disponibilidade, sem randomização formal, caracterizando o estudo como um quase-experimento.

3.1 Definição do Escopo

O resumo a seguir utiliza o formato proposto pela técnica GQM [17].

Resumo do escopo: Analisar o modelo iStar derivado do diagrama de casos de uso, bem como o próprio diagrama de casos de uso e a funcionalidade de derivação semiautomática na ferramenta JGOOSE. Com o propósito de obter dados qualitativos e quantitativos a respeito da corretude na aplicação de cada RGC por meio da ferramenta. Com respeito a derivação do modelo iStar a partir de um diagrama de casos de uso. Do ponto de vista do pesquisador. Na condição de um projeto de pesquisa realizado dentro da universidade à qual os autores pertencem, utilizando-se de sua estrutura disponibilizada, além disso, usando o próprio JGOOSE como ferramenta e os alunos do segundo ano de graduação do curso de Ciência da Computação, matriculados na disciplina de Engenharia de Software como voluntários participantes do projeto.

3.2 Formulação das hipóteses

A base para a análise de um experimento é o teste das hipóteses [6]. Foram formulados dois tipos de hipótese: Hipótese Nula e Hipótese alternativa [6]. Assim, foram definidas

ao todo seis hipóteses que abrangem uma análise completa a respeito de cada tarefa proposta para os participantes do experimento.

Hipótese Alternativa 1: Os elementos dos modelos iStar gerados a partir da funcionalidade fornecida pelo software JGOOSE, foram derivados conforme definido pelas regras de garantia de consistência descritas por Paza [2]. Ou, os modelos iStar, foram gerados parcialmente corretos conforme as RGCs definidas por Paza [2]. Além disso, os participantes consideraram que o modelo gerado reflete idealmente o modelo de casos de uso, ou seja, sem erros.

Hipótese Alternativa 2: Os elementos dos modelos iStar gerados a partir da funcionalidade fornecida pelo software JGOOSE, foram derivados conforme definido pelas regras de garantia de consistência descritas por Paza [2]. Ou, os modelos iStar, foram gerados parcialmente corretos conforme as regras de garantia de consistência definidas por Paza [2]. Além disso, os participantes consideraram o modelo iStar gerado adequado com o diagrama de casos de uso, entretanto o modelo não reflete idealmente o modelo de casos de uso e sugerem alterações.

Hipótese Alternativa 3: Os elementos dos modelos iStar gerados a partir da funcionalidade fornecida pelo software JGOOSE, foram derivados conforme definido pelas regras de garantia de consistência descritas por Paza [2]. Ou, os modelos iStar, foram gerados parcialmente corretos conforme as regras de garantia de consistência definidas por Paza [2], todavia os participantes consideraram que o modelo gerado reflete inadequadamente o modelo de casos de uso.

Hipótese Alternativa 4: Para os modelos iStar produzidos pela ferramenta, nenhuma das regras de garantia de consistência propostas por Paza [2], foi aplicada corretamente, contudo, os participantes consideraram o modelo iStar gerado coerente com o diagrama de casos de uso, ou seja, para os participantes, o modelo gerado reflete idealmente o modelo de casos de uso, ou seja, sem erros.

Hipótese Alternativa 5: Para os modelos iStar produzidos pela ferramenta, nenhuma das regras de garantia de consistência propostas por Paza [2], foi aplicada corretamente, contudo, os participantes consideraram o modelo iStar gerado adequado com o diagrama de casos de uso, ainda assim, para os participantes, o modelo gerado não reflete idealmente o modelo de casos de uso, e sugerem alterações.

Hipótese Nula: Os modelos iStar produzidos pela ferramenta em análise não foram gerados conforme as regras de garantia de consistência propostas por Paza [2], ademais, os participantes mediante questionário consideraram que o modelo gerado reflete inadequadamente o modelo de casos de uso.

3.3 Seleção das variáveis

Um quase-experimento é, em resumo, uma investigação de hipóteses, na qual variáveis independentes são manipuladas a fim de que uma relação de causa e efeito seja vista em variáveis dependentes [11]. O estudo é sobre o efeito da mudança de uma ou mais variáveis independentes, estas também podem se chamar fatores, sendo o tratamento um valor particular de um fator [6].

As variáveis dependentes são os objetos centrais do estudo, pois o objetivo foi compreender os efeitos gerados pelas mudanças atribuídas a cada fator, também podem

se chamar variáveis responsivas [6]. A partir dos conceitos apresentados, houve uma reflexão acerca dos objetos participantes do experimento, e, portanto, as seguintes variáveis foram definidas:

Variáveis independentes: Implementação das RGCs, a ferramenta JGOOSE e em particular a funcionalidade que permite a derivação de modelos iStar a partir de diagramas de casos de uso, os diagramas de casos de uso e modificações realizadas, experiência dos participantes da experimentação.

Variável dependente: se todas as RGCs implementadas na ferramenta JGOOSE estão conforme proposto por Paza [2]. Também os elementos do modelo iStar resultantes da derivação a partir dos casos de uso.

3.4 Design do Experimento

Cada grupo recebeu o mesmo questionário inicial, contendo questões e informações que auxiliam e norteiam a diagramação de Casos de uso, dessa maneira, cada grupo construiu o seu diagrama de casos de uso baseando-se na experiência prévia, que possibilita diversos entendimentos e diagramas para um mesmo problema, caso este seja suficientemente complexo.

Foram ao todo 3 tarefas, a tarefa 1 designava, igualmente para cada grupo, que construíssem o diagrama de casos de uso e enviassem o diagrama e o modelo iStar derivado pela ferramenta. A tarefa 2 avaliava a percepção dos grupos acerca das RGCs, isto é, cada equipe recebeu a definição escrita das regras acompanhada de exemplos de uso, devendo julgá-las de acordo com o seu próprio entendimento. A tarefa 3 demandava que os grupos, propositalmente, alterassem o nome de elementos e excluíssem outros, essa abordagem garante que todas as RGCs implementadas na ferramenta de derivação sejam executadas durante o experimento. O experimentador ficou disponível para esclarecer dúvidas dos alunos durante a realização do experimento.

Como parte que compõe o design do experimento segundo Wohlin [6], tem-se algumas características definidas como segue:

Aleatoriedade: Os participantes foram divididos em grupos (1 a 4 pessoas), a seleção dos destes foi conforme disponibilidade e consenso entre eles.

Bloqueio: A partir do momento em que os participantes são agrupados, seus níveis de conhecimento se dissolvem com os outros membros do grupo, o que inibe qualquer efeito negativo que surge por meio da experiência de cada voluntário nos resultados que serão avaliados.

Balanceamento: Como o experimento depende da disponibilidade e consenso de participantes, sua distribuição e balanceamento podem não ficar homogêneos, contudo, o objetivo é chegar o mais próximo possível por meio do agrupamento.

O tempo reservado foi de cem minutos. Local: laboratórios de Ciência da Computação da universidade. Vale destacar que os participantes possuíam experiência prévia na modelagem de Casos de uso e iStar. Essa experiência foi possível por meio da disciplina de Engenharia de Software, na qual o professor apresentou previamente aos alunos, não somente conceitos e uso do diagrama de Casos de uso e modelos iStar, como também utilizou a JGOOSE como ferramenta para trabalhos acadêmicos.

3.5 Avaliação sobre a validade do experimento

A análise das ameaças à validade é tão fundamental quanto a execução do próprio experimento, visto que a utilidade dos dados obtidos depende diretamente da confiabilidade e da integridade do processo de coleta. Quatro tipos de ameaças à validade do experimento foram consideradas [6][13], são elas:

A validade de conclusão, diz respeito à relação entre os tratamentos aplicados às variáveis independentes e os efeitos observados nas variáveis dependentes; A validade interna, busca assegurar que a relação entre o tratamento e as variáveis dependentes seja causal e não fruto do acaso; A validade de construção, verifica a aderência entre a teoria e a observação prática; A validade externa, é voltada à capacidade de generalização dos achados para outros contextos.

Dessa forma, cada validade foi contemplada a partir das soluções específicas apresentadas a seguir. Em relação à validade interna, esta será atendida tanto a partir do uso da abordagem GQM [17] quanto a partir do número de sujeitos escolhidos, além do questionário de abertura que auxiliará os voluntários na produção de diagramas de casos de uso diversos.

Correspondente à validade externa, esta será atendida pela experiência prévia dos voluntários. Nesta linha, a homogeneidade dos grupos foi um fator de atenção, pois um ambiente homogêneo é menos realista, todavia com base nos artefatos analisados, houve diferentes percepções quanto a como diagramar e modelar os softwares.

Em relação à validade de conclusão esta foi reforçada pela complexidade do cenário proposto no questionário de abertura, garantindo que os diagramas resultantes demandassem a ativação de todas as 12 RGCs. A análise baseou-se em métricas binárias objetivas (utilização e corretude da regra), reduzindo a subjetividade na interpretação dos dados quantitativos e qualitativos.

Em último lugar, a validade de construção demanda união entre teoria e prática observada. Identificou-se um risco caso as RGCs fossem utilizadas e aplicadas corretamente, mas o modelo iStar derivado não refletisse o diagrama de casos de uso segundo a opinião dos voluntários, sob esta ótica, perguntas relacionadas a percepção sobre a coerência das regras e do modelo foram aplicadas.

3.6 Uso da abordagem GQM como auxiliar no processo de medição e análise dos resultados que serão obtidos

A técnica consiste em definir previamente objetivos para o experimento e, a partir disso, gerar questões utilizadas posteriormente para definir se as métricas, também definidas, foram ou não alcançadas [17]. A análise dos resultados do experimento foi estruturada a partir de dois objetivos, são estes: **Objetivo 1:** Avaliar a completude e corretude de cada modelo iStar, com base na teoria estipulada pelas regras de garantia de consistência, classificando o emprego correto ou não de cada uma. **Objetivo 2:** Avaliar a percepção dos sujeitos acerca da coerência de cada regra, isto é, uma regra que define uma transformação que faz sentido, e do modelo em geral, baseando-se no

conhecimento prévio de cada grupo quanto à modelagem de sistemas na área de engenharia de software. Para o objetivo 1 definiu-se a seguinte questão e métrica:

Questão 1: Para os momentos em que a RGC em análise foi ou deveria ser instanciada, ela foi aplicada de forma absolutamente correta? Métrica 1: Total de execuções corretas desta RGC perante a demanda do modelo.

Para o objetivo 2 definiu-se as seguintes questões e métricas:

Questão 1: Qual porcentagem melhor traduz a percepção dos grupos acerca da coerência da RGC em análise? Métrica 1: Porcentagem de grupos que avaliaram esta RGC como sendo coerente, ou não.

Questão 2: Qual porcentagem melhor traduz a percepção dos grupos acerca da coerência do modelo? Métrica 13: Distribuição percentual das escolhas por grupo.

4 Execução do experimento

Os resultados foram obtidos a partir do experimento realizado no dia 07/08/2025, das 9 horas e 50 minutos até às 12 horas do mesmo dia, seguindo o planejamento do experimento. Ao todo, foram 11 equipes de 1 a 4 participantes. Os grupos realizaram as seguintes tarefas: Tarefa 1: Construir o diagrama de casos de uso e enviar com o modelo iStar derivado pela ferramenta. Tarefa 2: Avaliar a percepção de cada grupo sobre as RGCs, garantindo a validade do experimento. Tarefa 3: Alterar nomes e excluir elementos dos diagramas para testar todas as RGCs e responder às questões acerca das RGCs testadas e sobre o iStar, derivado pela ferramenta, na totalidade.

Na tarefa 1, dos onze grupos, dez enviaram corretamente via e-mail o diagrama e o modelo para análise. Com relação à tarefa dois, houve um equívoco por parte de um grupo, este enviou mais de uma resposta por meio dos vários integrantes, felizmente, outros grupos seguiram o padrão recomendado, mantendo assim a validade das respostas.

5 Análise dos resultados

A partir da coleta dos dados, a validação das hipóteses do experimento foi realizada por meio do método GQM [17]. Para cada objetivo estratégico previsto pela técnica, foram definidas questões e métricas seguindo o paradigma, conforme detalhado no planejamento. A seguir é realizada uma análise para cada questão relacionada ao objetivo 1 (um), a partir das métricas definidas.

A Questão 1 é definida como: “Para os momentos em que a RGC em análise foi ou deveria ser instanciada, ela foi aplicada de forma absolutamente correta?”. Pode-se afirmar, individualmente para cada RGC segundo a métrica: as RGCs 1, 2, 3, 6, 8, 9, 10, 11 e 12 foram aplicadas corretamente e, perante a demanda do modelo, todas as suas execuções estiveram corretas. Em contrapartida, as métricas associadas às RGCs 4, 5 e 7 revelaram comportamentos divergentes; ou seja, ao aplicar-se a mesma questão com foco de análise sobre as RGCs 4, 5 e 7, segundo a métrica definida, pode-se

afirmar: as regras em análise, em parte das vezes em que foram ativadas, não obtiveram execuções corretas e não supriram a demanda do modelo. Com base na análise dos modelos iStar, a tabela a seguir compara a aplicação prática das regras com suas definições teóricas, segundo a avaliação do experimentador.

Tabela 2. Cada linha representa uma equipe, cada coluna representa uma RGC. C representa correto, PC parcialmente correto e I incorreto.

RGC	1	2	3	4	5	6	7	8	9	10	11	12
Equipe A	C	C	C	C	C	C	C	C	C	C	C	C
Equipe B	C	C	C	C	C	C	C	C	C	C	C	C
Equipe C	C	C	C	C	C	C	C	C	C	C	C	C
Equipe D	C	C	C	C	C	C	C	C	C	C	C	C
Equipe E	C	C	C	PC	PC	C	PC	C	C	C	C	C
Equipe F	C	C	C	PC	C	C	C	C	C	C	C	C
Equipe G	C	C	C	C	C	C	C	C	C	C	C	C
Equipe H	C	C	C	C	C	C	C	C	C	C	C	C
Equipe I	C	C	C	PC	C	C	C	C	C	C	C	C
Equipe J	C	C	C	C	C	C	C	C	C	C	C	C
Equipe K	C	C	C	C	C	C	C	C	C	C	C	C

A tarefa 2 estava relacionada a percepção dos grupos com a coerência da regra aplicada. Com base na técnica GQM, podemos valorar as métricas definidas a fim de atender as questões correspondentes ao objetivo 2 (dois).

A Questão 1 do Objetivo 2 é definida como: “Qual porcentagem melhor traduz a percepção dos grupos acerca da coerência da RGC em análise?”. Pode-se afirmar, individualmente para cada RGC segundo a métrica: as RGCs 1, 2, 3, 4, 9 e 10 obtiveram 100% de aprovação, ou seja, todos os grupos avaliaram estas regras como sendo coerentes. Em contrapartida, as métricas associadas às RGCs 5, 6, 7, 8, 11 e 12 revelaram percepções divergentes; ou seja, ao aplicar-se a mesma questão sobre estas regras, segundo a métrica definida, constatou-se que a percepção de coerência não foi unânime: as RGCs 5, 6, 8 e 12 apresentaram 92,3% de aceitação, enquanto as RGCs 7 e 11 obtiveram, respectivamente, 84,6% e 75% de avaliações como coerentes pelos grupos.

A Fig. 2 ilustra as métricas (azul: positivo, vermelho: negativo), englobando as RGCs 1 e 11 e a percepção geral da tarefa 3. Apesar de envios duplicados e incompletos por um dos grupos, os dados mantêm sua validade.

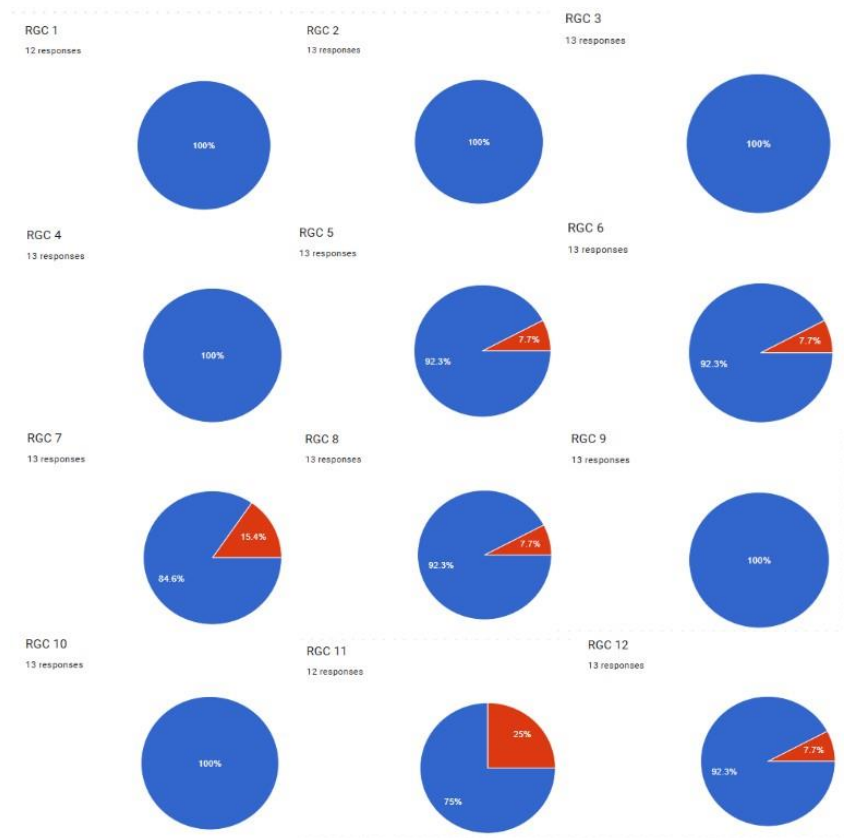


Fig. 2. Porcentagem dos grupos que concordam ou discordam sobre cada RGC.

Por fim, a tarefa 3 foi uma questão sobre a percepção dos participantes acerca da assertividade geral dos modelos iStar, assim sendo, correspondente ao objetivo 2.

Nesse contexto, a Questão 2 do Objetivo 2 define-se como: “Qual porcentagem melhor traduz a percepção dos grupos acerca da coerência do modelo?”. Pode-se afirmar, a partir da métrica de distribuição percentual das escolhas por grupo, que a percepção geral sobre a coerência do modelo derivado é majoritariamente positiva. Segundo os dados, 75% dos grupos avaliaram o modelo como “Bem assertivo”, encontrando poucos pontos de divergência em relação ao algoritmo de derivação; somado a isso, 8,3% dos participantes consideraram o modelo “Sem erros”, visualizando-o como ideal. Em contrapartida, as métricas revelaram que 16,7% dos grupos perceberam o modelo como “Pouco assertivo”, indicando que fariam várias mudanças. Não houve registros de grupos que avaliaram o modelo como “Nada assertivo”.

A base para análise da questão e valor da métrica pode ser vista na Fig. 3.



Fig. 3. Questão sobre a assertividade geral dos modelos gerado.

O quase-experimento permitiu constatar que as RGCs 4, 5 e 7 possuem problemas na implementação. Ademais, apesar da baixa porcentagem de erros em comparação com o número total de regras aplicadas, essa ferramenta demanda tolerância zero a erros, visto que pode desencadear problemas técnicos, financeiros e de confiança para a equipe e empresa de desenvolvimento.

Além disso, a tarefa 2 possuía um objetivo qualitativo com foco em entender a percepção sobre a coerência da transformação acerca das regras por parte de cada grupo. Nesse aspecto, algumas RGCs obtiveram respostas negativas, como, por exemplo, as regras 5, 6, 7, 8, 11 e 12, o que aponta falta de entendimento sobre as mesmas. Isto indica possíveis causas: falta de conhecimento aprofundado sobre os artefatos pelos participantes considerando que eram acadêmicos em formação, falta de clareza das RGCs, falta de tempo para uma análise mais detalhada da RGC entre outros aspectos. Estas causas, apesar de terem sido mitigadas no início do experimento, precisam ser mais bem avaliadas em futuros experimentos. Contudo, uma reflexão sobre a clareza e coerência das RGCs está sendo realizada pelos pesquisadores envolvidos.

Sobre a questão de percepção geral, ou seja, em relação a uma visão macro dos artefatos gerados, as respostas, majoritariamente, apontam um acerto na implementação e na funcionalidade da ferramenta.

Como trabalhos futuros, recomenda-se a correção dos erros dentro da implementação, além disso, uma revisão da definição de cada RGC pode ser proposta a fim de incentivar mais discussões sobre o assunto e possíveis melhorias e acertos relativamente a cada regra.

6 Considerações finais

O experimento realizado demonstrou que a implementação das Regras de Garantia de Consistência na JGOOSE é parcialmente correta e eficaz, validando a proposta inicial do projeto. As métricas utilizadas mostraram que a maior parte das RGCs foram aplicadas sem erros, especialmente aquelas que tratam da correspondência direta entre elementos centrais dos diagramas de casos de uso e dos modelos iStar.

Além disso, a percepção dos participantes reforçou a coerência dos modelos gerados, indicando aceitação prática da abordagem. Contudo, algumas regras específicas, como a RGC 7, RGC 5 e RGC 4, ainda apresentaram falhas pontuais de aplicação ou de interpretação, revelando os pontos de melhoria citados anteriormente.

Esses resultados confirmam que a hipótese nula pode ser rejeitada, enquanto as hipóteses alternativas 1 e 2 encontram respaldo nos dados, indicando que a JGOOSE gera modelos iStar corretos ou parcialmente corretos, mas ainda sujeitos a ajustes finos em situações mais complexas. Assim, a pesquisa atingiu seu objetivo de validar experimentalmente as RGCs implementadas, contribuindo para a evolução da ferramenta JGOOSE e para o uso da ESE como meio de avaliação.

Referências

1. Kotonya, G., Sommerville, I.: Requirements Engineering: Processes and Techniques. 1st edn. John Wiley & Sons, Chichester (1998).
2. Paza, D. N., Anjos, M. A., Gilnek, V. L. C., Santander, V. F. A.: "Maintaining Consistency in the Coevolution of iStar Models and Use Cases in Requirements Engineering," *2024 Latin American Computer Conference (CLEI)*, Buenos Aires, Argentina, (2024), pp. 1-10, doi: 10.1109/CLEI64178.2024.10700503.
3. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.): Social Modeling for Requirements Engineering. MIT Press, Cambridge (2011).
4. Booch, G., Jacobson, I., Rumbaugh, J.: The Unified Modeling Language User Guide, Addison-Wesley, (1999).
5. Thedoldi, D., Santander, V., Gilnek, V., Morgenroth, A.: Mantendo a consistência entre diagramas de Casos de Uso e modelos iStar com apoio de suporte computacional. In: Anais do XXI Congresso Latino-Americano de Software Livre e Tecnologias Abertas, pp. 188–196. SBC, Porto Alegre (2024).
6. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012).
7. Laboratório de Engenharia de Software - Unioeste, <https://www.unioeste.br/portal/les>, último acesso 2026/04/02.
8. Peliser, D., Santander, V. F. A., Silva, I. F., 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, vol. 12571, (2016).
9. Pessini, T.: BP2UC: Técnica para derivar Casos de Uso a partir de modelos BPMN. Trabalho de Conclusão de Curso. Universidade Estadual do Oeste do Paraná, Cascavel (2014).
10. Cockburn, A.: Writing Effective Use Cases. 1st edn. Addison-Wesley Professional, Boston (2000).
11. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Shull, F., Singer, J., Sjöberg, D.I.K. (eds.) Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, London (2008).
12. Silva, R.G.: Comparação do processo de derivação de Casos de Uso a partir de modelos BPMN e i* utilizando a ferramenta de apoio JGOOSE. Tese de Bacharelado, Universidade Estadual do Oeste do Paraná, Cascavel (2019).
13. Cook, T.D., Campbell, D.T.: Quasi-Experimentation: Design & Analysis Issues for Field Settings. Houghton Mifflin, Boston (1979).

14. Object Management Group (OMG): Unified Modeling Language (UML), <http://www.uml.org>, último acesso 2026/03/19.
15. Maldonado, J.C., Carver, J., Shull, F., Fabbri, S., Dutra, E., Spínola, R.O., Mendonça, M., Basili, V.R.: Perspective-Based Reading: A Replicated Experiment Focused on Individual Reviewer Effectiveness. *Empirical Software Engineering* 11, 119–142 (2006).
16. Hadar, I., Reinhartz-Berger, I., Kuflik, T., Perini, A., Ricca, F., Susi, A.: Comparing the comprehensibility of requirements models expressed in Use Case and Tropos: Results from a family of experiments. *Information and Software Technology* 55(10), 1823–1843 (2013).
17. Basili, V.R., Caldiera, G., Rombach, H.D.: Goal Question Metric Paradigm. In: Marciniak, J.J. (ed.) *Encyclopedia of Software Engineering*, vol. 1, pp. 528–532. John Wiley & Sons, New York (1994).
18. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: *Proceedings of the Conference on the Future of Software Engineering*, pp. 35–46. ACM, Limerick (2000).
19. Santander, V.F.A., Castro, J.F.B.: Deriving use cases from organizational modeling. In: *Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE 2002)*, Essen, Germany, pp. 32–39 (2002). <https://doi.org/10.1109/ICRE.2002.1048503>.
20. Dalpiaz, F., Franch, X., Horkoff, J.: *istar 2.0 language guide*. arXiv preprint arXiv:1605.07767 (2016).
21. Casarotto, B.L., Geraldino, G.C.L., Santander, V.F.A., Silva, I.F., Cespedes, M.A.T.: Using of ix(iStar) 2.0 for Improving the Use Cases Derivation. *IEEE Latin America Transactions* 20(2), 198–207 (2022).
22. Pozzan, V.A., Santander, V.F.A.: *JGOOSE: Uma ferramenta para derivar casos de uso a partir de modelos em i* e BPMN*. In: *Proceedings of the 22nd Workshop on Requirements Engineering (WER2019)*, Brazil, 2019. 10.29327/1298731.22-18.
23. Pozzan, V.A., Geraldino, G.C.L., Santander, V.F.A., Silva, I.F.D.: *Suporte ao rastreamento de requisitos na ferramenta JGOOSE*. In: *Proceedings of the 23rd Workshop on Requirements Engineering (WER2020)*, Brazil, 2020. 10.29327/1298730.23-7.
24. Paza, D.D.N., Santander, V.: *Mantendo a consistência na coevolução de modelos iStar, BPMN e Casos de Uso na Engenharia de Requisitos*. In: *Proceedings of the 24th Workshop on Requirements Engineering (WER2021)*, Brazil, 2021. 10.29327/1298728.24-25.