

Editor iStar 2.0 na ferramenta JGOOSE

Eduardo Pimentel dos Santos

Centro de Ciências Exatas e Tecnológicas - CCET
Universidade Estadual do Oeste do Paraná - UNIOESTE
Cascavel, Brazil
eduardo.santos47@unioeste.br

Rafael Ferreira Lima

Centro de Ciências Exatas e Tecnológicas - CCET
Universidade Estadual do Oeste do Paraná - UNIOESTE
Cascavel, Brazil
rafael.lima14@unioeste.br

Victor Francisco Araya Santander

Centro de Ciências Exatas e Tecnológicas - CCET
Universidade Estadual do Oeste do Paraná - UNIOESTE
Cascavel, Brazil
victor.santander@unioeste.br

Edineia dos Santos Brizola Brum

Centro de Ciências Exatas e Tecnológicas - CCET
Universidade Estadual do Oeste do Paraná - UNIOESTE
Cascavel, Brazil
edineia.brum@unioeste.br

Alexssandro Morgenroth

Centro de Ciências Exatas e Tecnológicas - CCET
Universidade Estadual do Oeste do Paraná - UNIOESTE
Cascavel, Brazil
alexssandro.morgenroth@unioeste.br

Abstract - Requirements elicitation is a crucial phase in the development of any software system. It consists of collecting, analyzing, documenting, validating and managing the organizational, functional and non-functional requirements of the software to be developed. In this context, some models are used to capture the understanding of the business environment and others to capture the functional requirements of the system in detail. For the first model, frameworks and techniques such as iStar and BPMN are usually used. UML (Unified Modeling Language) use cases are among the most widely used for capturing functional requirements. A proposal was presented for the extraction of UML use cases from iStar models and for computational support using a tool called JGOOSE. Recently, however, a new version of iStar has been introduced, iStar 2.0, which incorporates and modifies several graphical elements of the framework. This article presents the results of a project whose main goal was to further develop the iStar editor of the JGOOSE tool to support these new features of iStar 2.0. A video of the tool can be found at [15].

Keywords—JGOOSE; iStar 2.0; Computational tool; Requirements engineering.

Resumo - Elicitação de requisitos é uma fase crucial no desenvolvimento de qualquer sistema de software. Consiste em coletar, analisar, documentar, validar e gerenciar os requisitos organizacionais, funcionais e não funcionais do software a ser

desenvolvido. Nesse contexto, alguns modelos são utilizados para capturar o entendimento do ambiente de negócios e outros para capturar os requisitos funcionais do sistema em detalhe. Para o primeiro modelo, frameworks e técnicas como iStar e BPMN são normalmente empregados. Os casos de uso em UML (Unified Modeling Language) estão entre os mais amplamente utilizados para a captura de requisitos funcionais. Foi apresentada uma proposta para a extração de casos de uso UML a partir de modelos iStar e para o suporte computacional utilizando uma ferramenta chamada JGOOSE. Recentemente, porém, uma nova versão do iStar foi introduzida, o iStar 2.0, que incorpora e modifica diversos elementos gráficos do framework. Este artigo apresenta os resultados de um projeto cujo principal objetivo foi aprimorar o editor iStar da ferramenta JGOOSE para dar suporte a esses novos recursos do iStar 2.0. Um vídeo da ferramenta pode ser encontrado em [15].

Palavras-chave—JGOOSE; iStar 2.0; Ferramenta Computacional; Engenharia de requisitos.

I. INTRODUÇÃO

A obtenção de artefatos a partir de outros previamente desenvolvidos é uma prática bastante utilizada na Engenharia de Software para reaproveitar informações entre modelos diferentes. Neste sentido, em [1], uma proposta inicial de derivação de casos de uso UML a partir de modelos iStar foi apresentada.

Esta proposta foi gradativamente sendo suportada via ferramenta computacional denominada JGOOSE (Java Goal into Object Oriented Standard Extension). Contudo, em [2], uma nova versão do framework iStar foi proposta, denominada de iStar 2.0 e tem como objetivo principal facilitar o uso na indústria. Entretanto, a versão inicial do editor E4J iStar proposta em [3] e, posteriormente, melhorada em [4], não considerava a versão iStar 2.0 e suas modificações.

Assim, este trabalho apresenta a implementação de um protótipo do E4J iStar 2.0 como parte da ferramenta JGOOSE, incorporando a este editor as mudanças gráficas e de nomenclatura, como também a inclusão de novos elementos propostos na iStar 2.0. Cabe ressaltar que este novo editor permite que usuários possam utilizar a versão mais recente do framework iStar, se beneficiando, em especial, das vantagens propostas nesta nova versão para uso na indústria. Além disto, esse novo editor viabiliza implementar a última versão das diretrizes, as quais permitem apoiar a obtenção de casos de uso UML a partir do iStar 2.0 proposta em [1]. Desta forma, usuários da ferramenta, além de poder usar a versão iStar 2.0, a qual é mais próxima da realidade da indústria, também poderão gerar casos de uso com melhorias significativas nas suas descrições textuais. Essas melhorias são possíveis a partir do mapeamento de elementos presentes apenas na versão iStar 2.0, agora suportada pela JGOOSE.

Este artigo, além desta seção, apresenta na seção II trabalhos envolvendo projetos de ferramentas similares, na seção III alguns conceitos importantes sobre a ferramenta JGOOSE, como também uma breve descrição da arquitetura da mesma. Na seção IV apresenta-se o novo editor iStar 2.0 e na seção V, a validação do editor através de prova de conceito e uso do método TAM (*Technology Acceptance Model*). Por fim, na seção VI são realizadas as considerações finais do trabalho.

II. PROJETOS SIMILARES

Ao pesquisar por projetos similares, destacou-se a ferramenta piStar Tool, proposta em [10]. Essa ferramenta oferece suporte ao iStar 2.0, sendo de código aberto, de uso via navegador (online) e com compatibilidade para diversos alfabetos, além de possuir uma interface intuitiva e uma experiência de usuário agradável.

No entanto, a piStar Tool limita-se ao modelo iStar 2.0, o que significa que, para a geração de outros tipos de diagramas como BPMN ou Diagramas de Caso de Uso, é necessário recorrer a uma outra ferramenta adicional. Essa é uma diferença significativa em relação ao JGOOSE, que, além do suporte a vários diagramas, também permite a geração automática de outros diagramas, como diagramas de casos de uso a partir de diagramas iStar já existentes.

Também é possível observar em [12], algumas ferramentas que permitem construir modelos iStar. Contudo, estas ferramentas também não permitem gerar modelos UML a partir dos modelos iStar elaborados, conforme proposto na JGOOSE.

III. JGOOSE

Para realizar as mudanças propostas, considerando a nova versão iStar 2.0, foi realizado inicialmente um estudo da última versão da ferramenta JGOOSE [7] e seus componentes, em especial, os elementos utilizados no editor E4J (*Editor for JGOOSE*) iStar. O JGOOSE é um projeto do Laboratório de Engenharia de Software (LES), no qual os pesquisadores colaboram para explorar e impulsionar tópicos variados na Engenharia de Software [8]. Cabe ressaltar que esta ferramenta tem como objetivo demonstrar e mapear a derivação de casos de uso de forma visual a partir de modelos organizacionais elaborados via iStar [6] ou BPMN (*Business Process Modeling Notation*) [19]. Ela automatiza a obtenção de casos de uso com base em diretrizes apresentadas em [5] e [18]

Após o entendimento da estrutura da ferramenta, foi realizado o estudo da versão iStar 2.0, apresentada em [2] e as diferenças em relação à versão iStar 1.0 [6]. O entendimento das principais diferenças entre uma versão e outra será apresentado em detalhes na Seção IV.

A ferramenta JGOOSE é *open source*, foi desenvolvida utilizando a linguagem de programação Java e seu uso pode ser acadêmico ou comercial. Essa versão elaborada em 2021 pode ser encontrada em [9]. No que tange à arquitetura da JGOOSE, a mesma foi construída utilizando o padrão arquitetural MVC (*Model View Controller*) sendo particionada em quatro grandes pacotes, cuja estrutura interna segue o padrão mencionado.

O pacote denominado “iStar” está relacionado com a importação de modelos iStar elaborados em outra ferramenta denominada OME [11], oriunda ainda da primeira versão da ferramenta, antes de possuir editor próprio para iStar; o pacote denominado “log” foi criado para gravar o log do sistema; o pacote denominado “E4J” concentra todas as classes relacionadas aos editores para iStar e BPMN; e o pacote JGOOSE o qual utiliza os outros pacotes para apoiar a implementação das rotinas de mapeamento/derivação de BPMN — > Casos de Uso e iStar — > Casos de uso. A Figura 1 apresenta estes pacotes na plataforma *Github* [23].

Cabe ressaltar que, para a criação do editor iStar 2.0, optamos por replicar o editor iStar já existente, sendo possível reaproveitar todas as classes que já haviam sido implementadas no projeto JGOOSE, sem necessidade de alteração e fornecendo uma base sólida para começar a implementação. Deste modo, foram aplicadas diversas alterações nos arquivos reutilizados do editor iStar 1.0.

documentation	add images documentation
e4j	finish implement mapping and fixing bugs
istarm1-jgoose	mapping uc to istar
jgoose	(add) examples
jgraphx-master/jgraphx-master	update java and jgraphx
log-jgoose	finish implement mapping and fixing bugs
target	finish implement mapping and fixing bugs
.gitignore	update java and jgraphx
CONTRIBUTING.md	Update CONTRIBUTING.md
README.md	Update README.md

Fig. 1. Estrutura da JGOOSE no Github.

Em particular, as alterações de nomenclatura, como por exemplo de “*softgoal*” para “*quality*”, foram relativamente simples, mas outras mudanças, como a inclusão de novos elementos, tais como os relacionamentos “*needed-by*” e “*qualification*”, demandaram um pouco mais de esforço.

O repositório que armazena o código e documentação desta nova versão pode ser acessado em [13]. É relevante observar que no momento, a ferramenta se encontra sob desenvolvimento do suporte para que usuários da ferramenta possam fazer a derivação em ambos sentidos iStar 2.0 < – > Casos de Uso. Isto tornará possível que, principalmente, profissionais envolvidos na documentação de requisitos possam evoluir modelos de casos de uso e refletir essas alterações em modelos iStar 2.0, mantendo assim ambos os modelos consistentes.

Neste sentido, entendemos que os potenciais usuários da ferramenta são todos aqueles envolvidos na elaboração da documentação de requisitos de sistemas (ou leitura desse documento), entre os quais destacam-se: engenheiros de requisitos, analistas de negócio, arquitetos, projetistas, programadores, testadores, entre outros, além de usuários e clientes do *software*, com conceitos básicos sobre os artefatos envolvidos.

Para executar a ferramenta é necessário possuir a ferramenta JDK (Java SE Development Kit) [14] e também a ferramenta JGOOSE que pode ser obtida pelo *github* [13]. Após fazer o *download* (utilizar o ícone “Code” e em seguida a opção “Download ZIP”), basta descompactar os arquivos, abrir a pasta JGOOSE, e executar o arquivo “*jgoose.bat*”.

No que tange às tecnologias, a JGOOSE foi construída utilizando a *Integrated Development Environment* (IDE) *Netbeans*, que comporta a linguagem de programação Java já mencionada, o pacote *Jgraph* para a construção dos elementos gráficos, entre outras. Na Figura 3 é possível observar a tela inicial da ferramenta antes de incluir o editor iStar 2.0.

IV. EDITOR ISTAR 2.0

Para a atualização para a versão do iStar 2.0 foi tomada a decisão de manter o editor que atualmente utiliza o iStar 1.0 pois muitas funções legadas seriam extremamente prejudicadas ao realizar essa atualização. Além disso, também observou-se que seria benéfico permitir o uso de editores diferentes para cada versão do iStar, possibilitando que o usuário possa escolher qual versão do editor utilizar. Outra decisão de projeto tomada, foi iniciar removendo elementos e ligações desnecessárias, renomear os elementos e relações que tiveram seu nome alterado e, por fim, adicionar as novas funcionalidades. A Figura 4 apresenta a tela inicial do editor, sendo possível notar a mudança na primeira funcionalidade à esquerda, onde está localizado o ícone iStar 2.0 Editor.

A seguir são descritas as alterações realizadas nas novas telas de criação do editor iStar 2.0. As alterações podem ser visualizadas na Figura 5. Na área dedicada a criar atores, os mesmos podem ser representados de três formas diferentes: “Actor”, “Agent” e “Role”. O elemento “Position”, presente na versão anterior, foi removido nessa nova tela.

Na área dedicada a criar as ligações entre os atores, neste novo editor permite-se apenas o uso de duas ligações: “ISA” que foi mantida, e “Participates-in” que foi criada com base na ligação “ISA”, sendo removidas as ligações “Is-part-of”, “Plays”, “Covers”, “Occupies”, e “Instance-of” da versão anterior. Na área dedicada a criar os elementos de dependência, foi realizada a alteração da nomenclatura do elemento intencional “*softgoal*” para “*quality*”.

Na área dedicada a criar as ligações de contribuição temos apenas “Make”, “Hurt”, “Help” e “Break” neste novo editor. Estas ligações são consideradas essenciais para representar os efeitos (contribuição) de elementos intencionais nas “*quality's*”, sendo que as ligações da versão anterior “Some +”, “Some -”, “And”, “Or” e “Unknown” foram removidas.

Na área dedicada à criação das ligações entre elementos intencionais temos: “Dependency”, a qual não sofreu alteração e os elementos “Or” e “And” que são respectivamente, a renomeação dos elementos “Means-End” e “Task-Decomposition”.

Por fim, foram criados os elementos de ligação entre intencionalidades “*Needed-By*” e “*Qualification*”. Estes elementos são novos na versão iStar 2.0 e sua representação gráfica no editor não foi algo trivial, exigindo conhecimentos aprofundados na linguagem e componentes utilizados no projeto e implementação.

Na Figura 2 apresenta-se, de forma sintetizada, estas diferenças quanto aos elementos modificados ou incluídos da iStar 1.0 para a versão iStar 2.0. Um exemplo utilizando os elementos do iStar 2.0 pode ser visualizado na Figura 6.

No que tange ao uso do editor iStar 2.0, podemos observar a figura 7, a qual resume as ações necessárias para inserir e relacionar elementos no editor. Por exemplo, para realizar a inclusão de um novo elemento, basta arrastar o elemento do painel à esquerda na Figura 6, para a janela de edição ao centro. Mais detalhes sobre o uso do editor também podem ser observados no vídeo apresentado em [15].

V. VALIDAÇÃO DA FERRAMENTA

Para validação da evolução da ferramenta proposta neste artigo, aplicou-se um questionário a especialistas baseado no Modelo de Aceitação de Tecnologia (TAM, do inglês *Technology Acceptance Model*) [17] bem como foi realizada uma prova de conceito (PoC, do inglês *Proof of Concept*) [22] abordando um exemplo de organização na área de comércio de pneus. A prova de conceito passa pela construção de um modelo iStar 1.0 e, em seguida, o modelo adaptado e melhorado usando o iStar 2.0.

Removidos	Renomeados	Adicionados
 Position	 Softgoal	 Participat...
 Is-part-of	 Task-De...	 Need...
 Covers	 Means-end	 Qualificat...
 Some +	 AND	
 Some -	 OR	
 Unknown		
 Or		
 And		

Fig. 2. Diferenças entre as versões iStar 1.0 e 2.0.

A. Modelo de aceitação de tecnologia (TAM)

O Modelo de aceitação de tecnologia é um modelo teórico que busca explicar os processos de adoção e uso de tecnologias da informação. Desenvolvido por [17], o TAM fundamenta-se em dois constructos psicológicos centrais: a Utilidade Percebida (*Perceived Usefulness* - PU), que representa a crença do usuário sobre como o sistema pode melhorar seu desempenho, e a Facilidade de Uso Percebida (*Perceived Ease of Use* - PEOU), referente à percepção do esforço necessário para utilizar a tecnologia.

O instrumento de pesquisa consistiu em um questionário composto por dez questões, sendo nove objetivas e uma

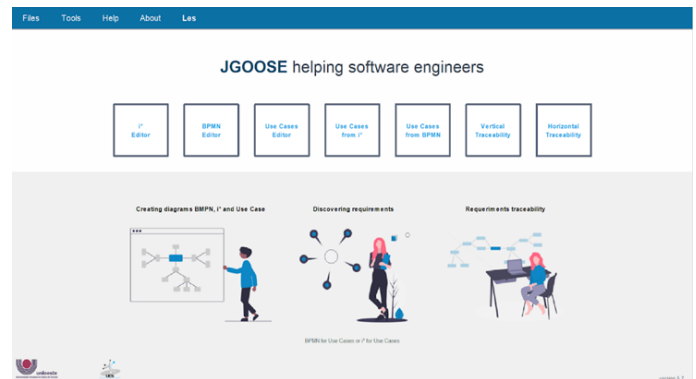


Fig. 3. Tela inicial da antiga versão da ferramenta JGOOSE.



Fig. 4. Nova tela inicial da ferramenta JGOOSE.

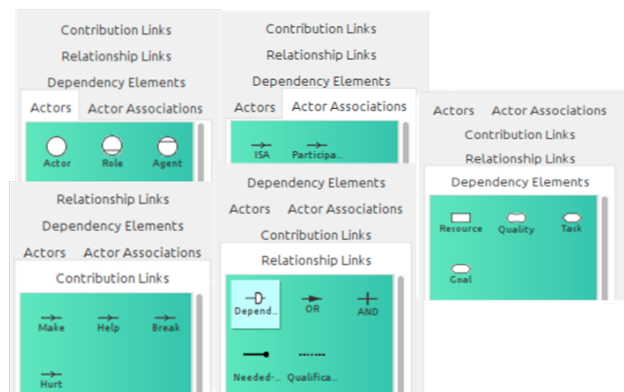


Fig. 5. Grade de recortes das novas telas dedicadas à criação de modelos iStar 2.0.

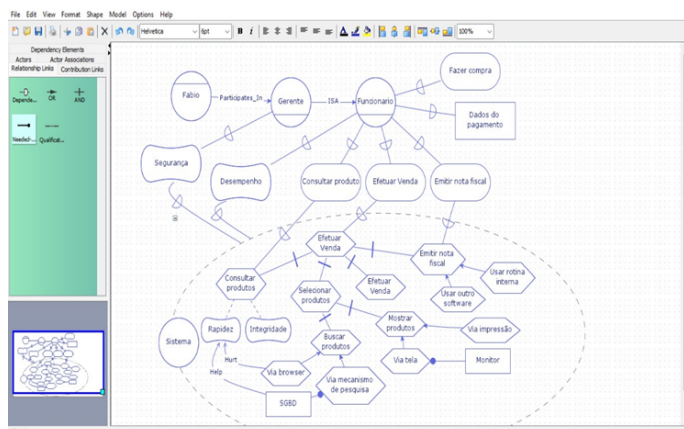


Fig. 6. Tela do editor de modelos iStar 2.0 com exemplo.

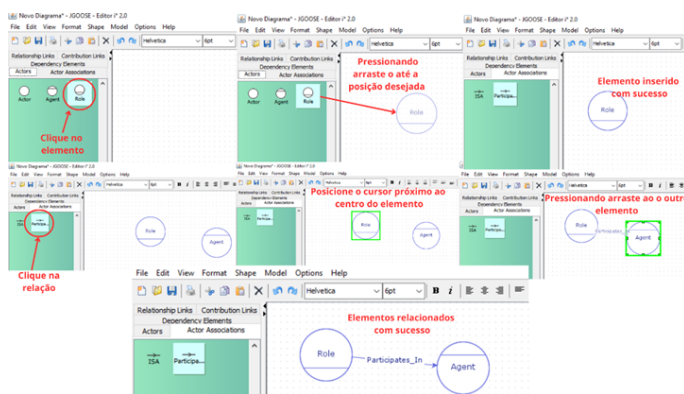


Fig. 7. Curto tutorial feito de recortes de tela.

subjetiva, estruturadas da seguinte forma: oito das questões objetivas utilizaram uma escala *Likert* de cinco pontos, com as seguintes opções: “Discordo totalmente”, “Discordo parcialmente”, “Neutro”, “Concordo parcialmente” e “Concordo totalmente”; a questão objetiva restante empregou uma escala numérica de 0 a 5, em que 0 representava a ideia de “Muito difícil” e 5 indicava “Muito fácil”; e uma questão final aberta, ficando a critério livre do participante a resposta proposta.

Este modelo foi aplicado visando receber um parecer inicial de especialistas sobre a utilidade e facilidade de uso da ferramenta. Os resultados obtidos são baseados nas respostas de seis especialistas qualificados, com três diferentes perfis de atuação: (1) Desenvolvedores, (2) Docentes e (3) Analistas de Qualidade. A Figura 8 ilustra a distribuição percentual desses perfis entre os participantes da avaliação. Também cabe destacar que todos os profissionais consultados apresentam mais de um ano de experiência na sua atividade bem como no uso da técnica iStar.

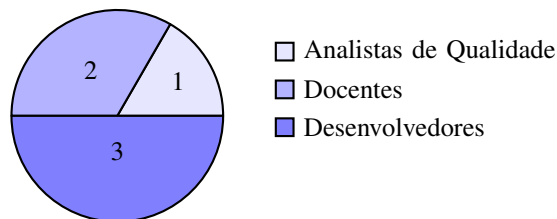


Fig. 8. Perfil dos Especialistas.

Para facilitar o entendimento das informações coletadas, após cada questão apresenta-se um gráfico demonstrativo com os resultados juntamente com um breve resumo dos resultados. O questionário, também disponível em [20], possui as seguintes questões para avaliação:

Questão 1 - A nova interface do editor iStar 2.0 na JGOOSE proporciona uma experiência de uso mais intuitiva em comparação com ferramentas similares, como o piStar?

Conforme demonstrado na Figura 9, os resultados indicam uma predominância de respostas positivas, sendo possível inferir que a maioria dos usuários concorda que a ferramenta possui uma usabilidade mais intuitiva em relação às alternativas atualmente disponíveis, contudo podemos notar que ainda existe preferência por outras ferramentas e espaço para melhorias conforme indicado pela resposta discordante.

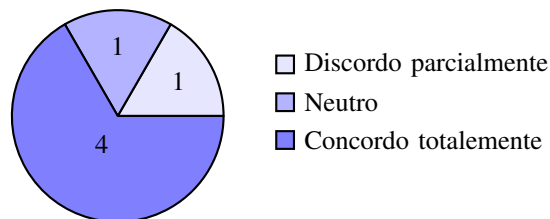


Fig. 9. Usabilidade mais intuitiva em comparação a outras ferramentas.

Questão 2 - Os elementos da versão iStar 2.0 foram implementados de maneira clara e consistente no editor?

Os seis especialistas avaliaram esta questão como “Concordo totalmente”, este resultado demonstra consenso unânime entre os especialistas quanto à implementação clara e consistente dos elementos do iStar 2.0 no editor, isto indica precisão semântica na implementação e consistência visual, sem inconsistências significativas ao realizar a tradução dos conceitos teóricos para uma ferramenta prática.

Questão 3 - O editor facilita a visualização das intenções e dependências entre os atores nos modelos organizacionais?

Conforme apresentado na Figura 10, todas as respostas distribuíram-se igualmente entre as opções “Concordo parcial-

mente” e “Concordo totalmente”. Esses resultados demonstram uma avaliação unanimemente positiva, indicando que os usuários consideram a ferramenta eficaz na representação visual das relações entre atores organizacionais.

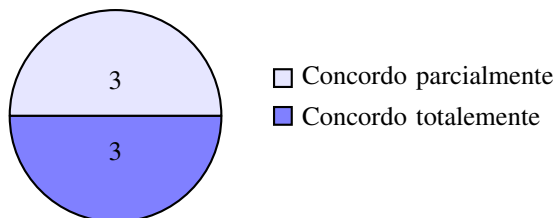


Fig. 10. Visualização das intenções e dependências entre os atores.

Questão 4 - A utilização deste editor contribui significativamente para a compreensão das relações entre modelagem organizacional e requisitos funcionais?

Conforme apresentado na Figura 11, os resultados indicam uma percepção majoritariamente positiva (83% de concordância combinada) sobre a capacidade do editor em articular as relações entre modelagem organizacional e requisitos funcionais, com apenas uma resposta neutra que não reflete discordância.

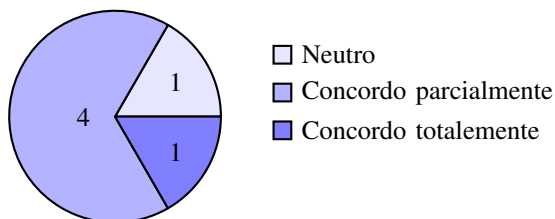


Fig. 11. Contribuição para a compreensão das relações entre modelagem organizacional e requisitos funcionais.

Questão 5 - A possibilidade de manter editores separados para iStar 1.0 e 2.0 contribui para a flexibilidade no processo de modelagem?

Conforme apresentado na Figura 12, os resultados indicam uma percepção significativamente positiva sobre a contribuição da arquitetura dual para a flexibilidade de modelagem, com apenas uma resposta neutra que não indica discordância explícita.

Questão 6 - A importação e adaptação de modelos da versão 1.0 para 2.0 no editor iStar foram práticas e eficientes?

Os resultados demonstram aceitação unânime entre os especialistas quanto à eficiência do processo de importação e adaptação de modelos elaborados no editor iStar 1.0 para o editor iStar 2.0, demonstrando como o processo é simples e eficiente. Contudo, ainda há espaço para melhorar, adicionando

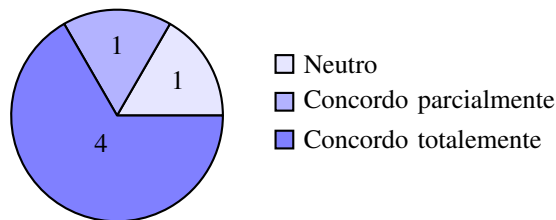


Fig. 12. Editores separados para iStar 1.0 e 2.0 geram maior flexibilidade no processo de modelagem.

uma portabilidade automática de elementos que não dependa da alteração manual.

Questão 7 - O editor iStar 2.0 atende às necessidades específicas de projetos reais de engenharia de requisitos?

Conforme apresentado na Figura 13, os resultados indicam a predominância de avaliações positivas, isto sugere que o editor atende efetivamente às demandas de projetos reais, com espaço para melhorias conforme indicado pela resposta discordante.

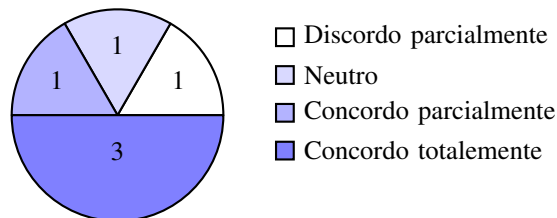


Fig. 13. A ferramenta atende às necessidades específicas de projetos reais.

Questão 8 - A curva de aprendizagem para o uso do editor iStar 2.0 é baixa (é fácil aprender a utilizar o editor)?

Esta questão empregou uma escala numérica de 0 a 5, sendo que o valor 0 representava a ideia de “Muito difícil” e 5 indicava “Muito fácil”. Conforme apresentado na Figura 14, a maioria dos especialistas classificou o aprendizado como “fácil” ferramenta apresenta uma curva de aprendizagem considerada baixa, indicando facilidade de uso para novos usuários. Contudo, a resposta neutra indica que existe uma área a ser refinada e potencializada, onde podem ser desenvolvidos mais tutoriais em vídeo ou diagramas de fluxo.

Questão 9 - Os recursos visuais e nomenclaturas adotadas no editor iStar 2.0 refletem adequadamente as mudanças do framework iStar 2.0?

Os resultados demonstram que os recursos visuais e as nomenclaturas implementadas no editor iStar 2.0 refletem de maneira adequada as mudanças propostas pelo *framework* iStar 2.0. Essa afirmação é respaldada nas respostas dos especialistas consultados, onde contamos com um voto em “Concordo parcialmente” e cinco votos em “Concordo totalmente”. A

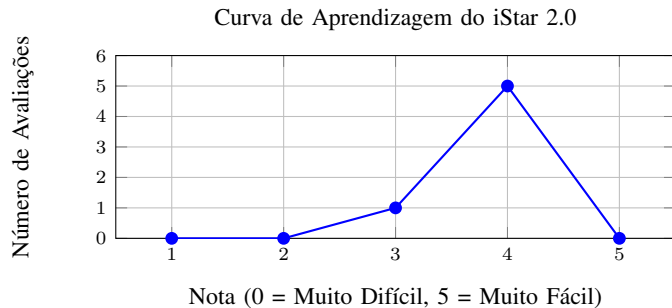


Fig. 14. Percepção da curva de aprendizagem do iStar 2.0.

predominância de concordância total (83,3%) indica que a implementação atual atende satisfatoriamente aos requisitos do *framework*. A única concordância parcial não representa discordância, mas sim a identificação de oportunidades para refinamentos a serem adicionados.

Questão 10 - Quais funcionalidades ou melhorias você sugeriria para futuras versões do editor iStar 2.0 na JGOOSE?

Esta questão era aberta sendo que os especialistas poderiam dar uma resposta mais completa sobre sua experiência ao testar a ferramenta. Ao analisar os *feedbacks* de especialistas sobre o editor iStar 2.0 na JGOOSE, identificamos sugestões valiosas para aprimorar a ferramenta em aspectos técnicos, funcionais e estratégicos. Esta questão revelou desafios de usabilidade, estabilidade e divulgação, além de oportunidades para transformar a ferramenta em uma solução mais robusta e acessível. Dadas as principais demandas, organizamos eixos prioritários que equilibram correções imediatas, inovações de médio prazo e estratégias de longo prazo para ampliar o impacto da plataforma na comunidade de engenharia de requisitos. As propostas refletem tanto necessidades técnicas como a correção de bugs e modernização da interface quanto iniciativas para fortalecer a adoção da ferramenta, garantindo que ela atenda efetivamente às demandas de projetos reais. É importante fazer um adendo de que muitos desses problemas já existiam na ferramenta anterior, e o editor iStar 2.0, em sua atual versão, herdou essas falhas históricas como as inconsistências no canvas e os comportamentos inesperados na interface. Esse contexto ressalta a necessidade de não apenas corrigir deficiências pontuais, mas a necessidade de um estudo que foque na manutenção e atualização de aspectos estruturais da plataforma para evitar a perpetuação desses gargalos.

O editor iStar 2.0 na JGOOSE provou ser uma ferramenta robusta e eficaz para modelagem de requisitos, suprimindo as atualizações necessárias para se adequar ao *framework* iStar 2.0. Os resultados das avaliações revelam consenso entre os especialistas sobre sua implementação consistente dos conceitos

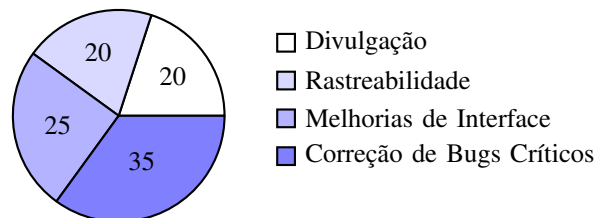


Fig. 15. Atualizações necessárias.

teóricos, com destaque para a clareza dos elementos visuais e nomenclaturas. Além disto, foi possível observar algumas oportunidades de melhoria, principalmente relacionadas à estabilidade da interface e sugestões para modernização visual. A necessidade de maior divulgação e canais dedicados para *feedback* também foi destacada como importante para ampliar a adoção da ferramenta. Essas sugestões devem ser incorporadas em evoluções do editor iStar 2.0.

B. Prova de conceito (PoC)

Este modelo foi aplicado para avaliar a viabilidade e funcionalidades da solução desenvolvida utilizando um dos cenários (problemas exemplos) proposto em [16] na execução de um quase-experimento com discentes da disciplina de Engenharia de Requisitos do curso de graduação de Ciência da Computação da Universidade Estadual do Oeste do Paraná (Unioeste).

Este exemplo considera o desenvolvimento de um *software desktop* para a automação de uma loja de pneus (CONEU). Um questionário de abertura de chamado proposto em [21] foi utilizado como base para obter as informações básicas sobre esse projeto. Em um processo de engenharia de requisitos [5], após essa coleta inicial de requisitos, recomenda-se a construção de modelos organizacionais usando técnicas como BPMN ou iStar. Como prova de conceito, os autores do presente artigo modelaram tanto o diagrama na versão iStar 1.0 quanto na versão iStar 2.0 (usando o novo editor) para esse problema.

Para compreender a modelagem realizada e apresentada nas Figuras 16 e 17, segue um resumo sobre o referido exemplo de organização incluindo a necessidade de desenvolvimento de um sistema computacional de apoio.

Introdução: O projeto tem por finalidade o desenvolvimento de um *software* para automação de uma loja de comércio de pneus, sendo responsável pelo gerenciamento do estoque de produtos assim como parte do processo de venda.

O sistema *desktop* a ser desenvolvido deverá ser capaz de realizar o controle de estoque dos produtos da empresa, sendo necessário possibilitar a sua manipulação pelo gerente de estoque e vendedor assim como a atualização automática

do estoque ao ser realizada uma compra por um cliente cadastrado. A compra de produtos dos fornecedores por parte da empresa cliente faz parte do sistema a ser desenvolvido. Deve ser responsável também por parte do processo de vendas da empresa, não sendo incluso nisto o processo de pagamento (cartão, dinheiro, etc) que será realizado pelo cliente durante seu cadastro.

Funcionamento Atual: Atualmente a empresa cliente conta com uma planilha eletrônica para auxiliá-lo no controle de estoque. O vendedor é encarregado de registrar toda operação do tipo "venda". Enquanto que o responsável por registrar as operações do tipo "compra" é o gerente de estoque. Ambas as tarefas podem ser canceladas durante o processo. As vendas são cadastradas manualmente pelo vendedor em um caderno de anotações, assim como as compras realizadas pelo gerente de estoque. Em uma venda, o cliente solicita o produto ao vendedor e o mesmo verifica se este consta em estoque, se houver, o vendedor anota a venda detalhando a quantidade, produto, cliente e o valor unitário do produto em uma planilha, se o cliente não está inserido no sistema o vendedor pode realizar o cadastro desse cliente. Em uma compra, o gerente de estoque verifica a existência do produto em estoque e solicita o produto ao fornecedor, o gerente de estoque anota a compra registrando a quantidade, produto, fornecedor e o preço unitário do produto em uma planilha, se o fornecedor ou produto não está inserido no sistema o gerente de estoque pode realizar o cadastro desse fornecedor ou produto. O cadastro de fornecedores no presente momento é feito apenas com uma agenda de anotações, assim como o cadastro de cliente. A empresa cliente deseja um *software* capaz de controlar o estoque da empresa e gerenciar as vendas realizadas pela mesma, bem como as compras feitas com o fornecedor e vendas feitas para clientes.

Dependências do Processo: Depende do sistema da Receita Federal para legalização das vendas através de Nota Fiscal e, de acordo com os desejos da empresa cliente, os dados do sistema (base de dados) serão armazenados em um banco de dados integrado ao sistema.

Após este breve resumo da organização e problema, é possível descrever os diagramas iStar gerados. A Figura 16 apresenta a organização e solução apontadas usando a iStar 1.0 e a Figura 17, a modelagem organizacional usando a iStar 2.0, na qual os elementos que representam as diferenças mais relevantes, que foram adicionadas ou alteradas, se encontram representados com a coloração azulada. Estes modelos foram construídos pelos autores deste artigo em um processo de *brainstorming*, visando provar que todos os conceitos associados à ambas versões do iStar são contemplados pelos editores, em especial, do editor iStar 2.0 e as mudanças em relação à iStar 1.0. Ambos modelos gerados visam descrever o problema

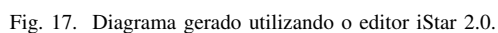
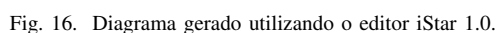
anteriormente descrito da maneira mais completa e objetiva possível, e apesar de semelhantes, possuem diferenças notáveis, entre as quais é possível destacar:

O ator utilizado para representar a função de "Funcionário" é diferente nos dois diagramas, enquanto o iStar 1.0 utiliza o ator "Position", o iStar 2.0 utiliza o ator "Role", consequentemente a ligação de "Covers" entre os atores "Funcionário" para "Gerente de Estoque" e "Funcionário" para "Vendedor" foi alterada para "ISA" de "Gerente de Estoque" para "Funcionário" e "Vendedor" para "Funcionário", sendo que esta alteração traz uma melhor compreensão das ligações entre os atores, sendo mais intuitiva a afirmação de que o vendedor é um funcionário.

Cabe ressaltar a ausência destacável da ligação "Needed-By" na versão iStar 1.0, que faz com que o recurso "SGBD", presente na versão iStar 2.0, não possa ser utilizado, sendo que isto gera uma perda de representatividade significativa para as tarefas de busca e cadastro em comparação à versão gerada no editor iStar 2.0. Além disto, o recurso "SGBD" também faz o uso da ligação "Qualification", que é utilizada para qualificar elementos do tipo objetivo, tarefa ou recurso, vinculando-os com o elemento intencional "Quality". Esta ligação incrementa muito a compreensão do recurso, deixando muito mais visível como consistência, velocidade e segurança são desejáveis e estão vinculados fortemente a este recurso. Infelizmente, estas ligações não podem ser realizadas com a mesma qualidade e clareza no editor iStar 1.0.

Ainda sobre a ligação "Needed-By", podemos observar a diferença na tarefa de exibir os produtos via tela, sendo que com essa ligação é possível adicionar para a tarefa, outros recursos como *tablet* e televisão e estas afetam diretamente elementos como orçamento e praticidade, necessitando de ligações de contribuição para exemplificar o seu impacto. Na tarefa de buscar produtos via mecanismo de busca, foi necessário realizar a substituição da ligação "Some+" para a ligação "help" para o diagrama da versão iStar 2.0, esta ligação foi utilizada para demonstrar a influencia positiva desta tarefa, e a ligação *help* consegue manter o mesmo propósito.

No processo de construção de ambos os modelos apresentados na Figura 16 e 17 é importante salientar que os recursos disponíveis nos editores iStar na JGOOSE, em particular, o editor iStar 2.0, permitiram construir esses modelos conforme elementos e restrições apresentados em [2]. Conclui-se que a prova de conceito permitiu aferir que o editor iStar cumpre com seu objetivo. Não menos importante e coerente com o apresentado no item A desta seção, algumas recomendações de melhorias foram realizadas pelos especialistas em engenharia de requisitos e uso da técnica iStar. Essas sugestões devem ser incorporadas em evoluções do editor iStar 2.0.



VI. CONCLUSÕES

A finalidade principal do presente artigo consiste na apresentação da evolução do editor iStar da ferramenta JGOOSE. Como opção de projeto foi decidido manter operacional a versão iStar 1.0 juntamente com a nova versão do editor iStar 2.0. Isto permite que usuários da ferramenta possam utilizar ambas versões do editor de acordo com sua preferência ou necessidade. Neste contexto, cabe destacar que o editor iStar 2.0 permite importar modelos elaborados no editor iStar 1.0. Contudo, nessa importação, os elementos da iStar 1.0 são visualizados apenas na forma textual, sem a notação gráfica associada e, fica a cargo do usuário, remover ou substituir esses elementos.

Entre as dificuldades percebidas na realização do projeto cabe destacar a falta de uma documentação atualizada para melhor entendimento lógico do sistema implementado, sendo de grande auxílio a realização de reuniões com antigos desenvolvedores da ferramenta visando buscar maior entendimento de todos os componentes implementados anteriormente.

É importante ressaltar que tanto o questionário aplicado e respondido por especialistas quanto a prova de conceito realizada, permitiram observar que o editor iStar 2.0 tem uma boa percepção de utilidade e facilidade de uso bem como considera todos os elementos e restrições expressas em [2].

Entre as limitações atuais do editor iStar 2.0 e ferramenta JGOOSE, destaca-se a necessidade do usuário realizar o *download* bem como instalação da ferramenta, assim como do *Java JDK* [14] (necessário para o correto funcionamento da ferramenta). Isto implica em esforço e conhecimento extra do usuário para poder usar a JGOOSE, o que poderia ser evitado se a mesma estivesse disponível para uso como uma aplicação WEB. Também, apesar de funcionar em sistemas operacionais Linux, alguns relatos de usuários apontam problemas com elementos gráficos não reconhecidos nesses ambientes operacionais. Os aspectos mencionados motivam trabalhos futuros, em especial, a disponibilização de uma versão da JGOOSE na WEB. Outro trabalho futuro envolve a implementação das novas diretrizes de mapeamento de modelos iStar 2.0 para Casos de Uso apresentadas em [1]. Isto não era possível anteriormente, uma vez que o editor iStar 1.0 não contemplava uma série de elementos utilizados na evolução do mapeamento, tais como os relacionamentos “*needed by*” e “*qualification*”. Pretende-se também conduzir novos experimentos em ambiente industrial, seguindo os princípios da Engenharia de Software Experimental [24].

REFERÊNCIAS

- [1] B. L. Casarotto, G. C. L. Geraldino, V. F. A. Santander, I. F. da Silva, and M. A. T. Cespedes, “Using of i*(iStar) 2.0 for Improving the Use Cases Derivation”, *IEEE Latin America Transactions*, vol. 20, no. 2, pp. 198–207, 2022.
- [2] F. Dalpiaz, X. Franch, and J. Horkoff, “iStar 2.0 Language Guide”, 2016.
- [3] L. P. Merlim, A. L. B. Silva, V. F. A. Santander, I. F. Silva, and J. F. B. Castro, “Integrating the E4J editor to the JGOOSE tool”, in *Proc. XVII Workshop de Engenharia de Requisitos (WER)*, Universidad Ricardo Palma, Lima, 2015.
- [4] V. A. Pozzan and V. F. A. Santander, “Evolução de uma ferramenta computacional que suporta o processo de derivação de casos de uso a partir de modelos BPMN e i*”, in *Anais do 7º EAICTI – Encontro Anual de Iniciação Científica e Inovação da Unioeste*, 2021.
- [5] V. F. A. Santander and J. F. B. Castro, “Deriving Use Cases from Organizational Modeling”, in *Proc. IEEE Joint Int. Requirements Engineering Conf. (RE)*, 2002.
- [6] E. Yu, “Modelling Strategic Relationships for Process Reengineering”, Ph.D. dissertation, Dept. of Computer Science, Univ. of Toronto, 1995.
- [7] JGOOSE-2025, [Online]. Available: <https://www.unioeste.br/portal/arq/files/LES/jgoose/JGOOSE-2025-iStar2.0.zip>. [Accessed: Apr. 4, 2025].
- [8] Laboratório de Engenharia de Software - LES, [Online]. Available: <https://www.unioeste.br/portal/les/>. [Accessed: Jun. 8, 2025].
- [9] JGOOSE-2021, [Online]. Available: <https://www.unioeste.br/portal/arq/files/LES/jgoose/JGOOSE-2021.zip>. [Accessed: Apr. 4, 2025].
- [10] J. Pimentel and J. Castro, “piStar Tool – A Pluggable Online Tool for Goal Modeling”, in *Proc. 2018 IEEE 26th Int. Requirements Engineering Conf. (RE)*, pp. 498–499, 2018.
- [11] OME Tool. [Online]. Available: <https://www.cs.toronto.edu/km/ome/>. [Accessed: Apr. 4, 2025].
- [12] iStar Wiki.org. [Online]. Available: <http://istarwiki.org/tiki-index.php?page=i%2A+Tools>. [Accessed: Apr. 4, 2025].
- [13] “Código e documentação E4J iStar 2.0.” [Online]. Available: <https://github.com/EduPim/JgoosePrivTests>. [Accessed: Apr. 4, 2025].
- [14] Oracle, “JDK - Java SE Development Kit.” [Online]. Available: <https://www.oracle.com/java/technologies/downloads/#jdk24-windows>. [Accessed: Apr. 4, 2025].
- [15] “Vídeo da ferramenta JGOOSE editor iStar 2.0”, YouTube. [Online]. Available: <https://youtu.be/bQ2MklQfIIE>. [Accessed: Jun. 7, 2025].
- [16] R. G. Silva, Comparação do processo de derivação de Casos de Uso a partir de modelos BPMN e i* utilizando a ferramenta de apoio JGOOSE, Monografia, Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste do Paraná – Campus de Cascavel, 2019.
- [17] F. D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, v. 13, p. 319–, 09 1989.
- [18] A. N. Giroto, V. F. A. Santander, I. F. Silva, and M. A. Toranzo, “Uma proposta para derivar Casos de Uso a partir de modelos BPMN com suporte computacional,” in 36th International Conference of the Chilean Computer Science Society (SCCC 2017), Arica, Chile, 16-20 Oct. 2017.
- [19] Object Management Group, Business Process Model and Notation (BPMN), Version 2.0 [Online]. Available: <https://www.omg.org/spec/BPMN/2.0/>. [Accessed: Jun. 8, 2025].
- [20] GOOGLE FORMS. Questionário de Pesquisa. [Online]. Available: <https://docs.google.com/spreadsheets/d/1tBqfDadydH27O614br9KeGD0ilwS3dVogNfg6qXp0dc/edit?usp=sharing>. [Accessed: Jun. 8, 2025].
- [21] C. F. de Souza and V. F. A. Santander, “Uma Proposta de Elicitação e Análise de Requisitos no Contexto de Médias e Pequenas Empresas de Desenvolvimento de Software,” in *CIBSE*, 2011, pp. 285–296.
- [22] A. J. R. Neto, *Um Modelo de Contexto das Práticas e Redes de Conhecimento nas Provas de Conceito*. Tese de Doutorado em Ciência da Informação, Faculdade de Letras, Universidade de Coimbra, Coimbra, Portugal, Set. 2023. Disponível em: <https://estudogeral.uc.pt/handle/10316/114191>.
- [23] Plataforma GitHub, GitHub.[Online]. Available: <https://github.com>. [Accessed: Jun. 8, 2025].
- [24] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén, Experimentation in software engineering (Vol. 236). 2012. Berlin: Springer.