iStar4Safety: Uma Extensão de iStar para Modelagem de Requisitos de Segurança em Sistemas Críticos

Moniky Ribeiro[1], Jaelson Castro[1], Jéssyka Vilela[1], João Pimentel[2]

[1] Centro de Informática, Universidade Federal de Pernambuco, Brasil
{smsr,jbc, jffv}@cin.ufpe.br
[2] Universidade Federal Rural de Pernambuco, Brasil
joao.hcpimentel@ufrpe.br

Resumo. Contexto: Sistemas Críticos de Segurança são caracterizados por serem sistemas que caso falhem ou não se comportem como esperado, podem levar à danos ou até perdas de vidas, destruição de propriedade, perdas de missões e/ou dano ambiental. A literatura reporta uma maior probabilidade de erros relacionados à segurança (do inglês safety) estarem associados às fases iniciais do processo de desenvolvimento (elicitação e especificação de requisitos) do que às fases finais (tais como a codificação). **Objetivo:** Este trabalho propõe uma extensão da linguagem iStar 2.0, a denominada iStar4safety, para possibilitar a modelagem de requisitos de segurança. **Método:** A definição dos novos construtores para modelar características relevantes a segurança foi realizada analisando os conceitos essenciais definidos pela literatura especializada. Além disso, propõe-se o metamodelo da linguagem juntamente com as regras de validação. **Resultados:** A linguagem iStar4Safety tem como foco a modelagem dos atributos de segurança no estágio inicial do desenvolvimento de sistemas, adotando uma abordagem orientada a objetivos. Um Sistema Crítico, que visa controlar o cruzamento de ferrovias, foi modelado usando a extensão proposta. **Conclusões:** Os resultados de uma avaliação preliminar indicam que iStar4Safety é conservativa, isto é, preserva os construtores da linguagem iStar 2.0, sendo adequada para descrever os requisitos iniciais de Segurança de Sistemas Críticos.

Palavras-chave: Engenharia de Requisitos · Sistemas Críticos · Safety · iStar · Extensão.

1 Introdução

Um Sistema Crítico de Segurança, do inglês *Safety-Critical System (SCS)*, envolve tanto componentes de hardware quanto de software, que caso falhem ou se comportem de maneira inesperada, podem levar a acidentes que podem resultar em danos a pessoas ou propriedades, grandes prejuízos financeiros, danos ao ambiente ou até mesmo perda de vidas [11, 12]. Portanto, para que sejam evitados comportamentos inaceitáveis ou indesejados nesse tipo de sistema, as entidades

regulatórias exigem um maior nível de cuidado e rigor em seu desenvolvimento quando comparados aos sistemas de informação tradicionais. Observa-se que os componentes de software dos *SCSs* executam funções cada vez mais críticas em áreas tais como aeronáutica, automotiva, médica, robótica, geração de energia, entre outras.

Esses sistemas devem ser desenvolvidos visando garantir a mitigação de perigos (do inglês *hazards*), com o intuito de prevenir acidentes. Em um trabalho prévio [20], detectamos a falta de construtores nas linguagens GORE (do inglês *Goal-Oriented Requirements Engineering*) para a modelagem da Análise Preliminar de Segurança (do inglês *Preliminary Safety Analysis*). Além disso, identificamos KAOS [1] e iStar [21] como sendo as linguagens GORE mais promissoras para modelar conceitos de segurança. Devido a sua natureza social e simplicidade dos seus construtores visuais, decidimos estender a linguagem iStar para torná-la adequada para modelar SCS. De fato, várias extensões tem sido propostas para iStar visando possibilitar seu uso em outros domínios [13]. Em trabalhos prévios [7, 8] propusemos o processo PRISE (PRocess to conduct iStar Extensions) que sistematiza a criação de extensões de qualidade para a linguagem iStar.

Portanto, o objetivo desse trabalho é apresentar a extensão **iStar4Safety** definida segundo os preceitos do processo PRISE. Esta extensão permitirá a modelagem de requisitos iniciais de segurança que atendam à Análise Preliminar de Segurança. Desta forma, os requisitos de segurança serão definidos o mais cedo possível no processo de desenvolvimento do Sistema Crítico. O restante deste artigo está estruturado da seguinte forma: na Seção 2 são apresentados os trabalhos relacionados; na Seção 3 é descrita a extensão **iStar4Safety** e seu processo de criação; na Seção 4 é apresentada uma ilustração do uso da linguagem proposta e, na Seção 5 são apresentadas as discussões e propostas para trabalhos futuros.

2 Trabalhos Relacionados

A linguagem KAOS, do inglês *Knowledge Acquisition in autOmated Specification* ou *Keep All Objects Satisfied* (descrita em [10, 1]) é uma linguagem de modelagem orientada à objetivos com um conjunto de técnicas de análise formal bem estabe-lecidas. Um importante conceito na linguagem KAOS é a ideia de obstáculo que permite a representação de situações onde o objetivo, expectativa ou requisito pode ser obstruído por algum fato. Contudo, não existe na literatura extensões de KAOS para a área de Sistemas Críticos.

O trabalho de Mouratidis e Giorgini [14], Secure Tropos, propõe uma extensão da metodologia Tropos [2] que adiciona o conceito de restrição de segurança (do inglês *Security*), além de estender os conceitos de dependência, objetivo, tarefa e recurso da linguagem nativa para o tipo *Secure*. As ameaças aos objetivos de segurança são circunstâncias que podem causar perdas, representadas por construtores ameaças (do inglês *threats*). Contudo, Secure Tropos não contempla conceitos para especificação completa dos atributos de segurança (*Safety*) desejados de Sistemas Críticos de Segurança.

RiskML [18] é uma estrutura de modelagem que usa modelagem conceitual para avaliar riscos na adoção de componentes de software de código aberto. Os conceitos utilizados no framework não atendem aos conceitos necessários para a especificação da segurança (*safety*) [20] na modelagem da Análise Preliminar de Segurança, como a abordagem iStar4Safety pretende fazer.

A próxima seção descreve a extensão iStar4Safety que visa permitir a modelagem de requisitos iniciais de segurança.

3 iStar4Safety

A extensão **iStar4Safety** tem o objetivo de modelar requisitos iniciais de segurança o mais cedo possível durante o desenvolvimento de Sistemas Críticos de Segurança. Para atender a este objetivo e criar uma extensão de forma sistemática e de qualidade foi usado o processo PRISE [7] para criação de extensões de iStar. O processo PRISE define seis subprocessos: (1) Análise da necessidade de proposta de nova extensão; (2) Conceitualização da extensão de iStar; (3) Desenvolvimento da extensão; (4) Validação e avaliação da extensão; (5) Verificação de novos construtores a serem inseridos; e (6) Publicação da extensão. É importante ressaltar que a realização do quinto subprocesso não foi necessária, pois não foi identificada a necessidade de novos construtores após a avaliação e validações da extensão, preconizadas pelo PRISE, serem realizadas. Apresenta-se, portanto, o processo de criação da linguagem, por meio dos subprocessos e suas respectivas atividades descritas nas seções a seguir.

3.1 Análise da Necessidade de Proposta de Nova Extensão

Como primeira atividade em busca do desenvolvimento da nova extensão, a área de segurança foi estudada. Os conceitos candidatos a integrar a extensão foram definidos tendo como base os quinze conceitos indicados como necessários para modelagem de uma Análise Preliminar de Segurança em Vilela et al. [20], bem como reuniões com especialistas em iStar e proponentes de extensões para iStar.

Nesse sentido, concluiu-se que um subconjunto de nove conceitos atenderia à modelagem de requisitos iniciais de segurança em iStar, sem ferir as características candidatas propostas em [20]e estando também de acordo com as diretrizes do processo PRISE. O objetivo dessa extensão é permitir uma modelagem consistente da Análise Preliminar de Segurança que contribua para o entendimento dos requisitos iniciais de segurança de um Sistema Crítico. Os conceitos selecionados após análise foram os seguintes:

1. Modelagem de acidentes (Conceito chave).
2. Modelagem de perigos (Conceito chave).
3. Modelagem de causas de perigos (Conceito chave).
4. Modelagem de condições ambientais (Conceito chave).
5. Modelagem de requisitos funcionais de segurança (Conceito chave).
6. Representação dos relacionamentos entre os construtores relacionados à perigo.

7. Modelagem e raciocínio de estratégias de segurança: devido à necessidade de propor estratégias para a resolução de situações de perigo prezando por um sistema seguro.
8. Habilidade de modelar recursos de segurança.
9. Modelagem do nível de impacto do acidente: pois ele deve ser considerado em uma Análise Preliminar de Segurança.

Conforme detalhado a seguir, os demais 6 (seis) conceitos apontados por Vilela et.al. [20] foram modelados por meio de outros conceitos já existentes ou não foram cobertos pela extensão. Os conceitos modelados através de outros conceitos foram:

- Representação de obstáculos: Um obstáculo, no domínio de segurança, à um objetivo de segurança é representado pelo conceito de perigo [19].
- Representação de como um evento afeta a segurança do sistema: por meio dos links de contribuição (*make, help, hurt, break*) é possível definir o nível de contribuição de dado evento para segurança como sugerido em [20].

Já os conceitos não modelados por essa extensão e as respectivas justificativas foram:

- Representação de restrições: Apesar da possibilidade apontada por [20] de desenhar a restrição através de links de contribuição (*hurt, break, help, make*), Leveson [11] cita que quando se trata de sistemas críticos é importante separar restrições de segurança de restrições não relacionadas à essa propriedade. Porém, os atuais links existentes em iStar permitem somente a associação entre elementos de qualidade e outros elementos, não possibilitando modelar restrições que podem ser representadas por outros construtores além de qualidades.
- Representação de pre/pós-condições: A linguagem base iStar 2.0 não permite a inserção de descrição textual que represente a modelagem de pré e pós-condições. Portanto, conservamos esta característica na extensão **iStar4Safety**.
- Representar o nível de criticidade de um elemento crítico ou suas contribuições à situações de falha: O nível de criticidade em **iStar4Safety** é relacionado ao acidente que pode acontecer e não a cada elemento isoladamente. O motivo é que o objetivo da extensão é modelar requisitos iniciais de segurança, não abrangendo, portanto, o detalhamento de cada elemento. Além disso, as contribuições de um elemento à situações de falhas podem ser indicados pelos links de contribuição.
- Apoio de descrição textual de requisitos de segurança: Considerando que a extensão trata da modelagem de requisitos iniciais de segurança, e que a linguagem nativa iStar 2.0 não permite a inserção de descrição textual, consideramos desnecessário para essa fase inserir descrições textuais dos elementos.

Um exemplo de um Sistema Crítico de Segurança (Sistema de Bomba de Infusão de Insulina) foi modelado [16], demonstrando que a linguagem nativa iStar 2.0 não estava apta a modelar os requisitos de segurança iniciais de tais sistemas. Após uma busca pelo catálogo **CATIE** [8][3], repositório de extensões de iStar 2.0, observou-se que nenhuma extensão já havia sido criada com esse objetivo. Portanto, foi definida a necessidade de desenvolvimento de uma extensão da linguagem para modelagem de requisitos iniciais de segurança.

3.2 Conceitualização da extensão de iStar

Neste subprocesso, foi realizada uma busca por construtores relacionados à todos os conceitos escolhidos (e listados à seguir) no catálogo CATIE que fossem de outras extensões e pudessem ser reusados. Não foram encontrados construtores que atendessem às necessidades de modelagem da extensão.

Seguindo o processo, os conceitos a serem modelados foram definidos e descritos:

1. **Acidente**: É um evento não planejado e nem desejado, mas não necessariamente não esperado, que resulta em um nível específico de perda [11, 5].
2. **Perigo**: Trata-se de um estado ou condições de um dado sistema que somado à outras condições do ambiente em torno dele levará inevitavelmente à um acidente [11]. O acidente é consequência de um perigo.
3. **Causa de Perigo**: É representada por uma condição que sozinha ou associada à outras, é/são suficiente(s) para o perigo relacionado à ela(s) ocorrer. As causas do perigo podem ser controladas ou até eliminadas em alguns casos [5, 20, 11].
4. **Condição Ambiental**: Trata-se de um conjunto de componentes e suas propriedades, incluindo elementos físicos, culturais, demográficos, econômicos, políticos, regulatórios ou tecnológicos que, apesar de não serem parte do sistema, podem afetar seu comportamento.
5. **Requisitos Funcionais de Segurança**: São os requisitos funcionais usados para mitigar ou prevenir os efeitos de falhas identificadas na análise de segurança.
6. **Estratégias de Segurança**: São ações que visam mitigar as consequências de um possível acidente. O objetivo dessas ações é eliminar ou reduzir o risco associado a uma situação perigosa. Cada mitigação tem um custo para sua realização que, na maioria das vezes, envolve o consumo de algum recurso [20, 4].
7. **Recursos**: Na linguagem iStar 2.0, recursos são apresentados como entidades informacionais ou físicas que são requeridas pelo ator à fim de realizar uma tarefa [6]. No contexto de Sistemas Críticos de Segurança, recursos são os ativos necessários para o correto funcionamento de requisitos críticos. Portanto, são especializações de recursos, indicando assim a sua criticidade em relação à outros recursos [20].

[3] O catálogo CATIE encontra-se na página https://istarextensions.cin.ufpe.br/catalogue/.

8. **Nível de Impacto do Acidente**: Define quão crítico é o acidente em relação à segurança do sistema. Esse nível pode ter cinco valores: (1) Catastrófico; (2) Muito Severo; (3) Considerável; (4) Pequeno; e, (5) Sem Efeito.
9. **Relacionamento entre construtores relacionados à perigos**: É necessário definir como os elementos se relacionam com os perigos, seja como sendo suas causas, consequências ou mitigando-os.

3.3 Desenvolvimento da extensão iStar4Safety

Após a conceitualização de iStar4Safety ter sido realizada, o processo PRISE propõe o desenvolvimento da extensão. O desenvolvimento de **iStar4Safety** consistiu na criação do metamodelo da linguagem, definição das regras de validação e definição da sintaxe concreta. Foi realizada também uma análise da qualidade da extensão. Por fim, uma ferramenta de modelagem foi adaptada para proporcionar um apoio semi-automatizado para a criação de modelos de **iStar4Safety**. Apresenta-se, portanto, cada marco desse subprocesso.

Metamodelo A primeira atividade realizada foi a criação do metamodelo de **iStar4Safety**. Conforme apresentado na Figura 1, o metamodelo foi criado conservando o metamodelo original de iStar 2.0 [6] (elementos em amarelo) e adicionando os novos elementos de iStar4Safety (elementos em roxo). O metamodelo foi representado por meio de um diagrama de classes de UML (Unified Modeling Language) seguindo o padrão MOF 2.4.1 da OMG [15].

Pode-se visualizar, quanto aos elementos inseridos em **iStar4Safety**, que a classe **Recurso de Segurança** é uma especialização de um recurso. Os elementos **Tarefas de segurança** serão, por sua vez, especializações da classe Tarefa, de iStar 2.0. Esses dois elementos, associados ou não, podem mitigar ou solucionar os *perigos* ao qual estão associados, sendo portanto, estratégias de segurança.

As classes **Objetivo de Segurança** e **Perigo** são especializações da classe Objetivo, de iStar 2.0. Os elementos da classe **Perigo** podem **obstruir** os **Objetivos de Segurança**. Os perigos são conceituados como uma especialização de objetivos porque podem ser considerados anti-objetivos [19] - isto é, objetivos que deseja-se que não aconteçam. Elementos da classe **Perigo** relacionam-se entre si por uma relação de causa, representada pelos links E/OU de iStar, já que perigos podem ser causas de outros perigos, sendo tais causas indicadas por perigos filhos que refinam um dado perigo pai. Os elementos da classe **Objetivo de Segurança** possuem a propriedade de nível do acidente que pode assumir um de cinco valores de acordo com o nível de acidente que acontecerá se o objetivo de segurança for obstruído.

Metamodelos podem ter restrições quanto à representação de todas as especificidades da linguagem, sendo necessárias geralmente descrições adicionais em linguagem natural ou através de alguma linguagem formal. Como iStar4Safety é uma extensão conservativa, as restrições de iStar 2.0 definidas em [6] foram mantidas e as restrições relacionadas à iStar4Safety foram adicionadas.

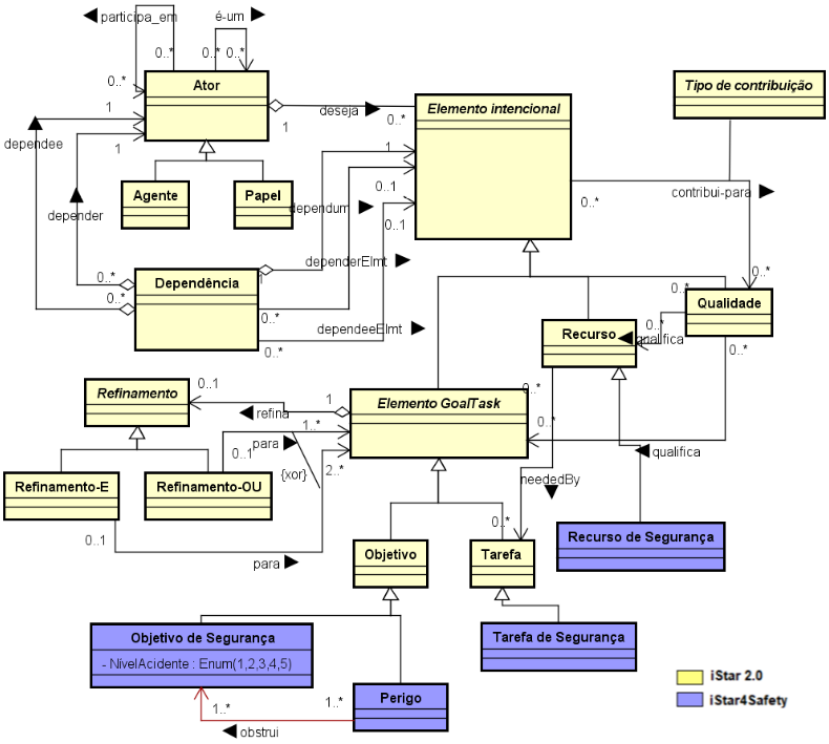

Fig. 1. Metamodelo de iStar4Safety.

Regras de validação Foram definidas as seguintes regras de validação para a modelagem com **iStar4Safety**:

- Os construtores da extensão iStar4Safety não podem ser elementos ***dependum***.
- Um **objetivo de segurança** só pode ser refinado ou por **objetivos de segurança** ou por **perigos**, não podendo ser pelos dois elementos ao mesmo tempo.
- Somente **perigos**-raiz podem estar relacionados à **objetivos de segurança**.
- Somente **perigos**-folha podem se associar à estratégias de segurança: **recurso de segurança** e **tarefa de segurança**.
- Todo **perigo**-folha deve ter pelo menos uma estratégia de segurança associada a ele.
- **Recursos de segurança** associam-se pelo relacionamento ***neededBy*** à **perigos**, **tarefas de segurança** ou ainda à **tarefas**.
- Toda estratégia de segurança deve estar relacionada ao **ator** que a realiza, exceto aquelas realizáveis pelo próprio **ator** que possui o elemento em sua fronteira.

Sintaxe Concreta A definição da sintaxe concreta de iStar4Safety foi escolhida visando a simplicidade de modelagem, ou seja, construir extensões com a menor quantidade possível de novos construtores e representações para não dificultar o uso da linguagem e propor representações gráficas simples que possibilitem a modelagem sem ferramenta, caso seja necessário [9]. Ao mesmo tempo, proporcionar a modelagem dos conceitos que são necessários durante a Análise Preliminar de Segurança. Portanto, optou-se por utilizar o mecanismo de extensão de peso-leve, usado também em extensões da linguagem UML. O estereótipo textual, que trata-se do nome do construtor entre os símbolos "<< >>", é a opção peso-leve mais utilizada para representar nós especializados [3].

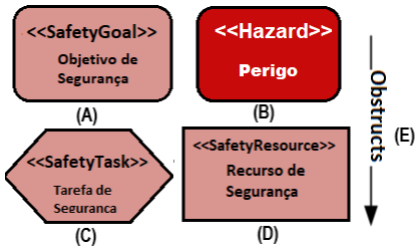

Fig. 2. Construtores gráficos adicionados pela extensão iStar4Safety.

Para a criação dos construtores gráficos, foram utilizadas as mesmas representações dos nós pais, associadas à estereótipos com o nome do construtor especializado. Outra diferença entre os construtores da linguagem nativa e da extensão **iStar4Safety** está na seleção das cores: os elementos de segurança são representados pela cor rosa e o elemento **Perigo** pela cor vermelha. Contudo, estas cores poderiam ser customizadas pelo usuário da linguagem.

Portanto, como resultado de todo o processo de definição de construtores gráficos, representações foram escolhidas e sua representação gráfica pode ser vista na Figura 2. O elemento (A) da figura representa o construtor **Objetivo de Segurança (Safety Goal)**. O elemento (B) por sua vez indica o construtor gráfico **Perigo (Hazard)**. O construtor **Tarefa de segurança** (**Safety Task**) é representado pelo elemento (C). Um **Recurso de Segurança** (**Safety Resource**) é representado pelo elemento (D) da figura. Por fim, o elemento (E) é a representação gráfica do link **obstrui** (**Obstructs**).

Análise da Qualidade de iStar4Safety Após o desenvolvimento da extensão, é necessário verificar se a mesma é conservativa e atende à requisitos de completude, consistência e é livre de conflitos com outras extensões de iStar. Após a realização da análise, conclui-se que a linguagem **iStar4Safety** é capaz de modelar os conceitos de segurança identificados como necessários no subprocesso de análise da necessidade da extensão descritos na subseção 3.1, além de possuir todos esses conceitos descritos tanto no metamodelo desenvolvido, como na sintaxe concreta. **iStar4Safety** não alterou quaisquer aspectos nativos da

linguagem iStar 2.0. Além disso, foi verificado que **iStar4Safety** não possui conflitos entre construtores gráficos e conceitos em relação à outras extensões já definidas para iStar, tendo como base para tal afirmação o catálogo CATIE [8] de extensões de iStar [16].

Uma ferramenta de modelagem para iStar4Safety Para facilitar a adoção da nova linguagem de modelagem iStar4Safety, foi disponibilizada a ferramenta **piStar-4Safety**[4], que é uma extensão da ferramenta piStar, criada para gerar modelos iStar 2.0.

Diretrizes para modelagem com iStar4Safety Para a criação do modelo de Raciocínio Estratégico (SR) com o uso de construtores de **iStar4Safety**, foi definido um conjunto de diretrizes que podem ser executadass durante a criação do modelo. Tais diretrizes encontram-se disponíveis em [16].

3.4 Ilustração da extensão iStar4Safety

O uso da nova extensão é ilustrado por meio da modelagem de um Sistema de Cruzamentos de Ferrovias [16] seguindo as diretrizes criadas e indicadas na subseção 3.3. O sistema modelado, adaptado do trabalho de [17], realiza o controle do cruzamento de uma linha férrea com a rodovia, sendo ponto de passagem do tráfego padrão de ciclistas, automóveis além de pessoas. A área de intersecção entre as duas vias é chamada zona de perigo, devido a criticidade envolvida na passagem de usuários e trens.

Apresenta-se na Figura 3, o Modelo de Requisitos Iniciais deste Sistema Crítico. Para melhor detalhamento da aplicabilidade da extensão, o ator Sistema Controlador da Via foi detalhado através da expansão via Modelo de Raciocínio Estratégico.

Ator Sistema Controlador do Cruzamento O ator Sistema Controlador do Cruzamento, representado pela Figura 4, possui dois objetivos de segurança. O *objetivo de segurança* "*Que a passagem simultânea de usuário da via e trem seja impedida*", pode ser obstruído pelos *perigos* "*Que a aproximação do trem não seja sinalizada pelo sistema*" e "*Que a zona de perigo esteja aberta para passagem de usuário e trem*". As causas para a não-sinalização de aproximação do trem serão a luz amarela ou vermelha de sinalização não estarem ativas ou as barreiras não estarem abaixadas. Para mitigar o *perigo* da luz amarela não estar ativa, deve-se realizar a *tarefa de segurança* "*ativar luz vermelha*" e que o ator *Centro de Operações* repare a falha. Para mitigar o *perigo* da luz vermelha não estar ativa, deve-se "*Cancelar fechamento da via*", além de ter a falha reparada pelo *Centro de Operações*.

Quanto ao *perigo* das barreiras não serem abaixadas, a fim de mitigá-lo, o sistema deve declarar a falha e realizar a tarefa de segurança "*Solicitar reinicialização do sistema de controle de cruzamento*" para o ator *Centro de Operações*. O *perigo* das barreiras abertas pode causar também o *perigo* "*Que a zona de*

[4] piStar-4Safety Tool encontra-se em: http://www.cin.ufpe.br/∼jhcp/pistar/4safety/.

perigo esteja aberta para passagem de usuário e trem". Já o *objetivo de segurança* "*Que as passagens na via não sejam bloqueadas indefinidamente*" pode ser violado pelo sistema não ser informado sobre o término da passagem do trem. O que pode causar esse *perigo*, é o fato do sensor do trem falhar. Nesse caso, para sanar esse *perigo*, o *Sistema de Controle de Cruzamento* deve bloquear permissões de passagem e o *Centro de Operações* deve tratar o bloqueio.

4 Discussões e Trabalhos Futuros

O objetivo principal desse trabalho é apresentar a criação de uma extensão para a linguagem de modelagem iStar 2.0 para modelar requisitos de segurança iniciais de Sistemas Críticos.

iStar4Safety foi desenvolvida seguindo o processo PRISE [7] que permite a criação de uma extensão de iStar que atenda às melhores práticas propostas por especialistas [9]. Já para a definição dos conceitos necessários para modelagem de requisitos iniciais, usou-se como base o trabalho de Vilela et.al [20] que identificou quinze conceitos necessários para a Análise Inicial de Requisitos de Segurança de Sistemas Críticos. Desses conceitos candidatos, nove foram escolhidos para integrar a nova extensão.

Para modelar o conceito de Perigo, que impede a satisfação de objetivos de segurança, foi usado o conceito de obstáculo [19]. O relacionamento entre os conceitos necessários à Análise Preliminar de Segurança para a modelagem de requisitos iniciais de segurança e iStar4Safety é sumarizada na Tabela 1.

Para melhor definir os elementos da nova linguagem, o metamodelo de iStar4Safety foi desenvolvido, além das regras de validação da extensão. Os construtores gráficos, ou seja, sua sintaxe concreta, foram também definidos.

Como preconizado pelo PRISE, foi realizada uma verificação sobre completude, consistência e falta de conflitos entre a extensão **iStar4Safety** e a linguagem iStar 2.0 e demais extensões. O resultado gerado mostrou que iStar4Safety atendeu satisfatoriamente estas exigências.

Como passo adicional, foi proposta uma ferramenta para apoiar a modelagem com a nova linguagem. Além disso, um conjunto de diretrizes foi proposta para guiar a modelagem de requisitos de segurança iniciais utilizando a extensão **iStar4Safety**.

A ilustração do uso extensão foi realizada por meio da modelagem de um Sistema de Cruzamento de Ferrovias. Uma avaliação de iStar4Safety foi realizada por meio da modelagem de um cenário além da coleta das opiniões dos participantes através da aplicação de um survey em uma turma de graduação (detalhes disponíveis em [16]). O objetivo do survey foi avaliar a extensão considerando a modelagem de uma versão simplificada de uma Bomba de Infusão de Insulina bem como as respostas do survey aplicado após a modelagem. Um método empírico foi aplicado com o objetivo de avaliar a extensão iStar4Safety [16].

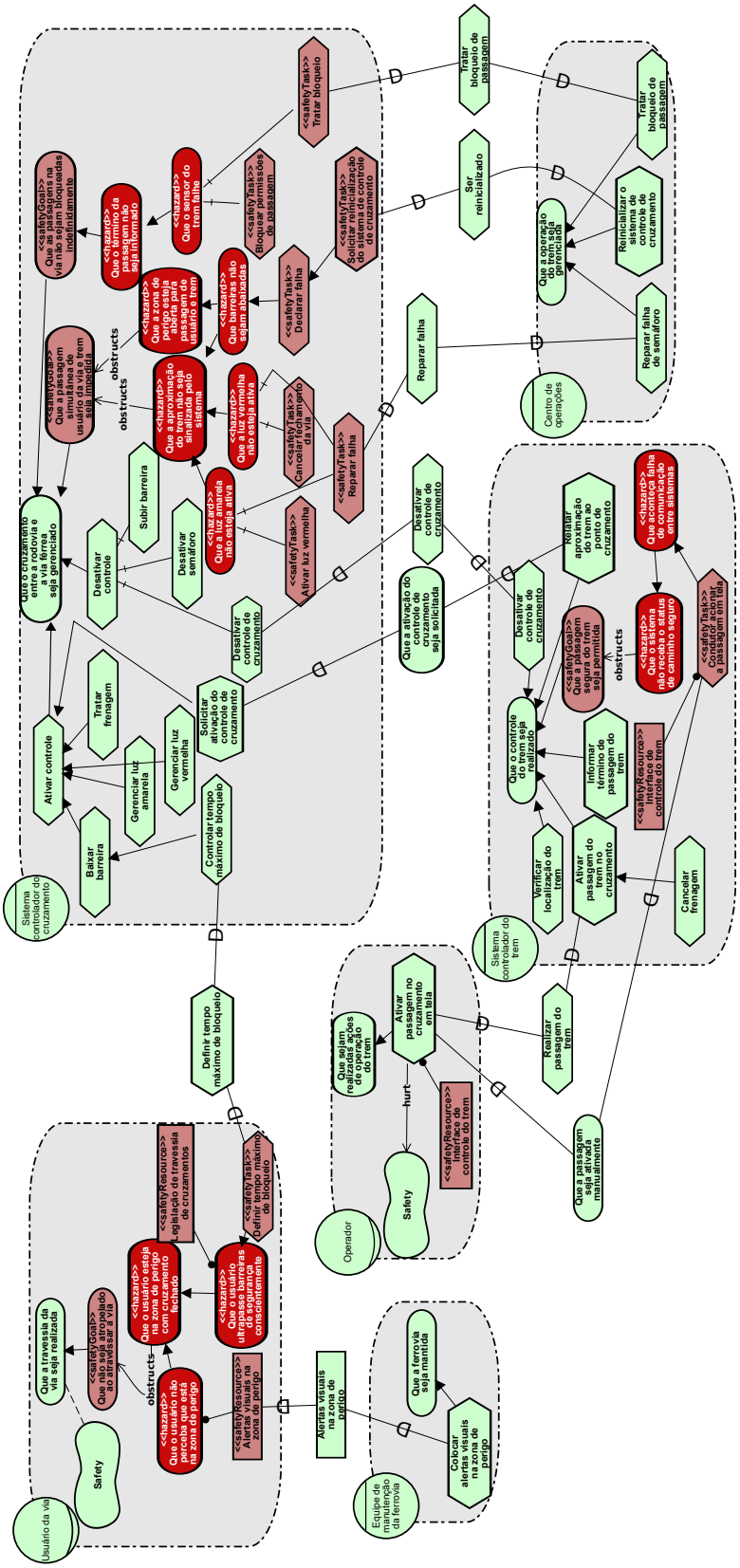

Fig. 3. Modelo SR do Sistema de Controle de Travessia da Ferrovia modelado com iStar4Safety.

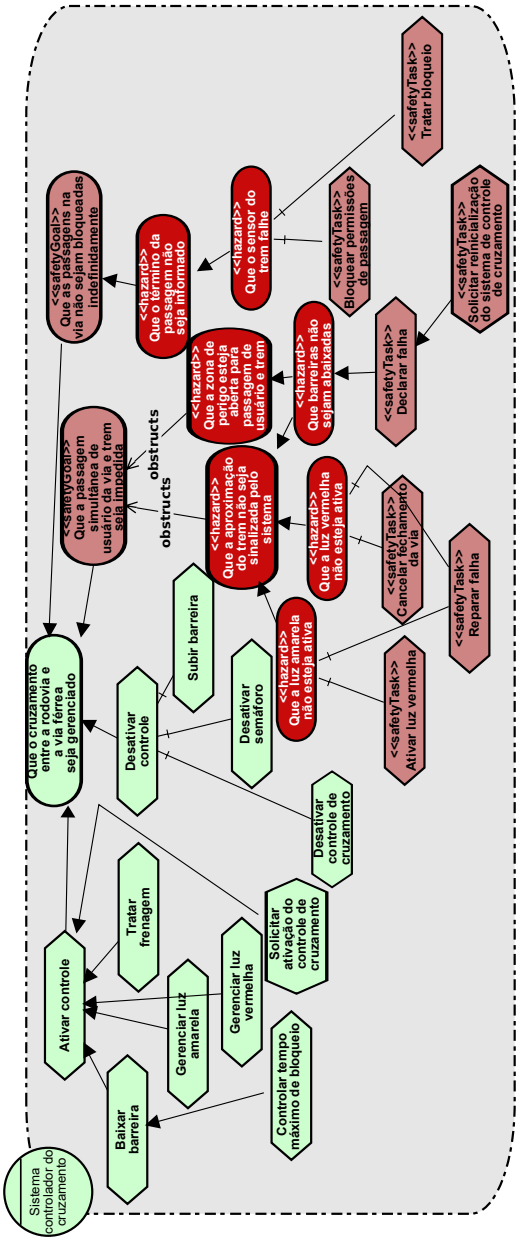

Fig. 4. Ator Sistema Controlador do Cruzamento - Parte do modelo SR do Sistema de Controle de Travessia da Ferrovia modelado com iStar4Safety.

Nesta avaliação, os participantes, dezesseis (16) alunos de graduação da disciplina de Especificação de Requisitos da UFPE, foram convidados a desenvolver

a modelagem de uma Bomba de Infusão de Insulina, conforme documento de requisitos fornecido.

Os modelos criados foram então avaliados quanto a corretude sintática, complexidade estrutural e similaridade comportamental. Além disso, um questionário foi aplicado para avaliar a opinião dos participantes quanto à usabilidade, efetividade e eficiência da extensão. Por fim, a extensão foi submetida ao catálago CATIE e aprovada para integrá-lo[5].

Table 1. Relacionamento entre conceitos da Análise Preliminar de Segurança e iStar4Safety.

Conceito da Análise Preliminar de Segurança	iStar4Safety
Acidente	O conceito de um acidente está implícito em iStar4Safety, já que um acidente é uma consequência da obstrução de um objetivo de segurança por um perigo.
Perigo	Elemento perigo
Causa de perigo	Um elemento perigo, refinando outro elemento perigo
Condição ambiental	Elemento perigo
Requisito funcional de segurança	Elemento tarefa de segurança
Estratégia de Segurança	Árvore de tarefas de segurança e recursos de segurança
Recurso	Elemento recurso de segurança
Nível de impacto do acidente	Propriedade accidentImpactLevel no elemento objetivo de segurança
Relacionamento entre construtores relacionados à perigos	Link obstrui e refinamentos e/ou

Como trabalhos futuros sugere-se:

- Analisar como modelar Requisitos Finais de Sistemas Críticos de Segurança, buscando apoiar a modelagem de outros níveis de Análises de Segurança;
- Definir uma forma adequada de representar o conceito de restrições, separando as restrições relacionadas à segurança de outras restrições.
- Implementar as regras de validação na **piStar-4Safety** Tool;
- Avaliar a extensão através de experimentos controlados, comparando a extensão criada a outras formas de modelar Requisitos Iniciais de Sistemas Críticos de Segurança;
- Avaliar a sintaxe concreta desenvolvida, realizando estudos empíricos;
- Reavaliar a forma de representação do nível de impacto do acidente.

5 Agradecimentos

Os autores agradecem ao apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) pelo apoio financeiro.

[5] O link para acessar iStar4Safety no repositório de extensões de iStar é: https://istarextensions.cin.ufpe.br/catalogue/publication/view/112.

Referências bibliográficas

1. Goal-directed requirements acquisition. Science of Computer Programming **20**(1), 3 – 50 (1993)
2. Towards requirements-driven information systems engineering: the tropos project. Information Systems **27**(6), 365 – 389 (2002)
3. A systematic literature review of istar extensions. Journal of Systems and Software **137**, 1 – 33 (2018)
4. Asnar, Y., Giorgini, P., Mylopoulos, J.: Goal-driven risk assessment in requirements engineering. Requir. Eng. **16**(2), 101–116 (Jun 2011)
5. Berry, D.M.: The safety requirements engineering dilemma. In: Proceedings of the 9th International Workshop on Software Specification and Design. pp. 147–. IWSSD '98 (1998)
6. Dalpiaz, F., Franch, X., Horkoff, J.: istar 2.0 language guide. CoRR **abs/1605.07767** (2016), http://arxiv.org/abs/1605.07767
7. Gonçalves, Enyo, A.J.C.J.: Prise: A process to conduct istar extensions. (under review) (2019)
8. Gonçalves, E., Heineck, T., Araújo, J., Castro, J.: A catalogue of istar extensions. WER18, Rio de Janeiro (sep 2018)
9. Gonçalves, E., de Oliveira, M.A., Monteiro, I., Castro, J., Araújo, J.: Understanding what is important in istar extension proposals: the viewpoint of researchers. Requirements Engineering (Jul 2018)
10. Lapouchnian, A.: Goal-oriented requirements engineering: An overview of the current research (01 2005)
11. Leveson, N.G.: Safeware: System Safety and Computers. ACM, New York, NY, USA (1995)
12. Leveson, N.G.: Engineering a Safer World: Systems Thinking Applied to Safety. Mit Press, Massachusetts, London, England (2011)
13. Morandini, M., Penserini, L., Perini, A., Marchetto, A.: Engineering requirements for adaptive systems. Requir. Eng. **22**(1), 77–103 (Mar 2017)
14. Mouratidis, H., Giorgini, P.: Secure tropos: A security-oriented extension of the tropos methodology. Int. J. Soft. Eng. Knowl. Eng. (02), 285–309 (apr)
15. OMG: OMG Meta Object Facility (MOF) Core Specification, Version 2.4.1 (Jun 2013), http://www.omg.org/spec/MOF/2.4.1
16. Ribeiro, M.: Desenvolvimento de uma extensão da linguagem de modelagem iStar para Sistemas Críticos de Segurança - iStar4Safety. Master's thesis, Universidade Federal de Recife (feb 2019)
17. Schnieder, E.: Specification Methodology, Case Studies, and Experiments – An Introduction to the Subject Area of Traffic Control Systems, pp. 89–95. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
18. Siena, A., Morandini, M., Susi, A.: Modelling risks in open source software component selection. In: Yu, E., Dobbie, G., Jarke, M., Purao, S. (eds.) Conceptual Modeling. pp. 335–348. Springer International Publishing, Cham (2014)
19. Van Lamsweerde, A., Letier, E.: Handling obstacles in goal-oriented requirements engineering. IEEE Trans. Softw. Eng. **26**(10), 978–1005 (Oct 2000)
20. Vilela, J., Castro, J., Martins, L.E.G., Gorschek, T., Silva, C.: Specifying safety requirements with gore languages. In: Proceedings of the 31st Brazilian Symposium on Software Engineering. pp. 154–163. SBES'17 (2017)
21. Yu, E.S.K.: Modelling Strategic Relationships for Process Reengineering. Ph.D. thesis, Toronto, Ont., Canada, Canada (1995), aAINN02887