

UM EDITOR PARA A LINGUAGEM DE ESPECIFICAÇÃO DE REQUISITOS RELAX

Rafael Torres¹

Luiz Paulo Franz²

Joao Pablo Silva Da Silva³

Resumo:

Sistemas autoadaptativos apresentam características únicas, que os diferenciam de outros tipos de sistemas. Como a principal delas, podemos destacar as situações de incerteza que tais sistemas devem enfrentar. Por conta disso, os processos usuais de engenharia de requisitos não são satisfatórios nesse contexto. Uma das propostas para auxiliar na engenharia de requisitos para sistemas autoadaptativos é a linguagem RELAX, que tem como objetivo evidenciar as incertezas inerentes à esse tipo de sistema. Neste artigo, apresentamos o desenvolvimento de uma ferramenta que dá suporte à escrita e edição de requisitos RELAX, explorando sua sintaxe e semântica e facilitando o uso da linguagem. Como resultado, obtemos um editor funcional baseado na plataforma eclipse.

Palavras-chave: Engenharia de Requisitos, Sistemas autoadaptativos

Modalidade de Participação: Iniciação Científica

UM EDITOR PARA A LINGUAGEM DE ESPECIFICAÇÃO DE REQUISITOS RELAX

¹ Aluno de graduação. torresrafa22@gmail.com. Autor principal

² Outro. luizpaulofranz@gmail.com. Co-autor

³ Docente. joaosilva@unipampa.edu.br. Orientador



UM EDITOR PARA A LINGUAGEM DE ESPECIFICAÇÃO DE REQUISITOS RELAX

1. INTRODUÇÃO

Sistemas autoadaptativos, de acordo com Macías-Escrivá (2013), podem ser definidos como softwares capazes avaliar e mudar seu comportamento em tempo de execução com base nos estímulos que recebem do ambiente e seu estado atual. Whittle (2010) comenta que esses sistemas atuam em condições e contextos diversos, o que dificulta a previsão dos estados pelos quais esse software vai passar ao longo de seu ciclo de vida.

Engenharia de requisitos é um conjunto de processos que tem como objetivo elicitar e avaliar os requisitos que motivam um projeto. De acordo com Sommerville (2007) através da engenharia de requisitos, é possível manter um artefato de documentação de requisitos, que deve ser seguido pela equipe de desenvolvimento durante a implantação do sistema.

Considerando as situações de imprevisibilidade que os sistemas autoadaptativos enfrentam, fica claro que as abordagens convencionais de engenharia de requisitos não são adequadas para lidar com tais sistemas. Tendo isso em vista, Whittle (2010) propôs a linguagem RELAX, com o objetivo de elicitar os requisitos de um sistema autoadaptativos evidenciando suas características de incerteza.

Neste trabalho, temos como objetivo apresentar uma ferramenta para especificação de requisitos que dê suporte a linguagem RELAX, explorando seus aspectos sintáticos e semânticos, tornando seu uso mais transparente.

2. Desenvolvimento

Apresentada por Whittle (2010), a linguagem RELAX é uma linguagem natural estruturada que tem como objetivo evidenciar características de incerteza, que por sua vez, são inerentes à sistemas autoadaptativos.

A RELAX utiliza operadores para organizar o requisito de forma mais formal, porém ainda com trechos em linguagem natural. A classificamos, portanto, como linguagem semi-formal. Os operadores RELAX são apresentados na Tabela Tabela 1.

Tabela 1: O vocabulário da Linguagem RELAX.

Operadores	Descrição
Operadores Modais	
<i>SHALL</i>	Um requisito deve ser fornecido/mantido.
<i>MAY... OR MAY</i>	Um requisito especifica uma ou mais alternativas.
Operadores Temporais	
<i>EVENTUALLY</i>	Um requisito deve ser fornecido/mantido eventualmente.
<i>UNTIL</i>	Um requisito deve ser fornecido/mantido até uma posição futura.
<i>[BEFORE — AFTER]</i>	Um requisito deve ser fornecido/mantido antes ou depois de um evento em particular.

<i>IN</i>	Um requisito deve ser fornecido/mantido em/durante um particular intervalo de tempo.
<i>AS [EARLY — LATE]</i> <i>AS POSSIBLE</i>	Um requisito especifica algo que deveria ser fornecido/mantido tão cedo quanto possível ou deveria ser postergado por tanto tempo quanto possível.
<i>AS CLOSE AS POSSIBLE</i> [frequência]	Um requisito especifica algo que acontece repetidamente, mas a frequência pode ser relaxada tão próxima quanto possível de uma frequência.
Operadores Ordinais	
<i>AS CLOSE AS POSSIBLE</i> [quantidade]	Um requisito especifica uma quantidade contável, mas o número exato pode ser relaxado tão próxima quanto possível de uma quantidade.
<i>AS [MANY — FEW]</i> <i>AS POSSIBLE</i>	Um requisito especifica uma quantidade contável, mas o número exato pode ser relaxado o maior número possível ou o menor número possível.
Fatores de incerteza	
<i>ENV</i>	Define um conjunto de propriedades que definem o ambiente do sistema.
<i>MON</i>	Define um conjunto de propriedades a serem monitoradas pelo sistema.
<i>REL</i>	Define a relação entre as propriedades de ENV e MON.
<i>DEP</i>	Identifica as dependências entre os requisitos relaxados e invariantes.

Para realizarmos o desenvolvimento da ferramenta, optamos pela utilização do Xtext. Xtext é um framework pertencente à plataforma Eclipse Foundation (2017) para o desenvolvimento de *Domain Specific Language* (DSL), bem como linguagens de propósito geral. Além da facilidade que o framework oferece para desenvolver linguagens, ele ainda oferece recursos avançados de forma simples, sendo alguns deles: *Syntax Highlight*, evidenciando palavras reservadas, comentários e erros de sintaxe; *Content assist*, opção de *auto-complete* e *Hyperlinking*, possibilitando saltar para a declaração de algum bloco ou variável.

Para o desenvolvimento da ferramenta, estendemos a gramática original da Linguagem RELAX e adicionamos alguns operadores: o operador REQ para demarcar o início de um novo requisito; os operadores ENV, MON, REL e DEP que, por mais que tenham sido apresentadas no artigo de Whittle (2010), não haviam sido formalmente definidos.

Como terminal da Linguagem RELAX utilizado para o desenvolvimento da ferramenta, temos o conjunto T. Nele, temos os operadores da linguagem definidos, o terminal ID, que representa os caracteres alfanuméricos para os trechos em linguagem natural, e o símbolo ϕ representando uma expressão RELAX.

$T = \{REQ, ID, \phi, true, false, SHALL, MAY..., OR MAY..., EVENTUALLY, UNTIL, BEFORE, AFTER, IN, AS CLOSE AS POSSIBLE TO, AS CLOSE AS POSSIBLE TO, AS \{EARLY; LATE; MANY; FEW\} AS POSSIBLE, ENV, MONS, RELS, DEP\}$

Como variáveis da gramática, $V = \{\varphi, \varphi1, \varphi2, e, f, q, t, p\}$, sendo que φ representa uma expressão RELAX (ou seja, tratando a recursividade da linguagem), e representa um evento, f representa frequência, q representa quantidade, t representa tempo e p representa uma frase em linguagem natural.

Para a regra maior da linguagem, definimos que ela será composta pelo operador REQ, um identificador único para o requisito, uma sentença RELAX, suas propriedades, e encerrará com ponto e vírgula. Temos, portanto, a seguinte regra de construção:

$P = \{S \rightarrow REQ \ p: \varphi; Prop \mid true \mid false,$
 $Prop \rightarrow ENV \mid MON \mid ENV \mid ENV \ MON \ REL \mid ENV \ MON \ REL \ DEP \mid DEP \mid ENV \ DEP$
 $\mid MON \ DEP \mid \varepsilon,$
 $ENV \rightarrow ENV \{ (p : p;)^+ \},$
 $MON \rightarrow MON \{ (p : p;)^+ \},$
 $DEP \rightarrow DEP: p; \mid (p,)^+ p; ,$
 $REL \rightarrow REL \{ (p : p;)^+ \};$

A seguir, apresentamos um exemplo de requisito escrito utilizando a Linguagem RELAX e depois a expressão que é gerada a partir dele.

REQ prenat01 : The smartphone SHALL send a help message \to
 another authorized user when the pregnant woman is \in a
 hospital ;
 ENV { currentPosition : Current position of the pregnant woman ; }
 MON {gps: The pregnant woman 's smartphone GPS ;}
 REL {gps - currentPosition ;}

Expressão gerada:

ReqId := prenat01
 exp := SHALL (p1);
 p1 := The smartphone send a help message \to another authorized
 user when the pregnant woman is \in a hospital
 ENV := currentPosition : Current position of the pregnant woman
 MON := gps: The pregnant woman 's smartphone GPS
 REL := mon=gps -> env= currentPosition

Para testarmos a ferramenta, foram desenvolvidos testes baseados em contextos distintos e cobrindo várias possibilidades oferecidas pela Linguagem RELAX, bem como todos os seus operadores:

- SmartCampus
- SmartCar
- SmartHome
- SmartTractor

3. RESULTADOS e DISCUSSÃO

Nos testes, obtivemos uma baixa taxa de erro, sendo que todos eles estão relacionados ao operador MAY... OR MAY.... O resultado dos testes é apresentado na Tabela 2.

Tabela 2: Resultado dos testes.

Cenário de teste	Testes	Erros
SmartCampus	12	1
SmartCar	12	1
SmartHome	6	0
SmartTractor	11	0
Total	41	2

A ferramenta obtida é apresentada na Figura 1.

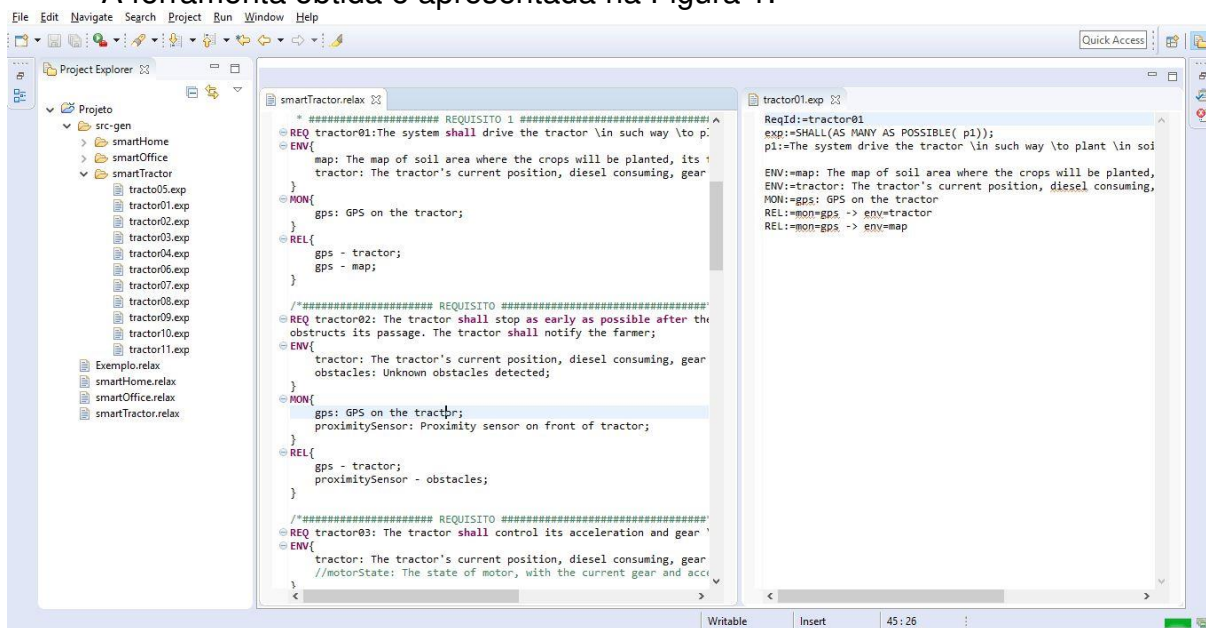


Figura 1. Screenshot da tela do RelaxEditor.

4. CONSIDERAÇÕES FINAIS

O principal objetivo deste trabalho é apresentar o processo de criação de uma ferramenta para elicitac o de requisitos com suporte   linguagem RELAX proposta por Whittle (2010). Baseado no que foi apresentado, podemos concluir que cumprimos o objetivo de produzir tal ferramenta. Com ela podemos escrever senten as complexas utilizando os operadores RELAX e extrair as express es formais delas. Entretanto, a ferramenta ainda apresenta limita  es relacionadas ao operador MAY... OR MAY.... Como trabalhos futuros, propomos a melhoria da ferramenta em rela  o ao operador que apresenta problema e ampliar ainda mais os cen rios de teste para aumentar a confiabilidade na ferramenta.

5. REFER NCIAS

FOUNDATION, E. Xtext - reference documentation; 2017 [acesso em 02 out 2017]. Dispon vel em: <https://eclipse.org/Xtext/documentation/index.html>

MAC IAS-ESCRIV ; FRANK D, HABER, RODOLFO, DEL TORO, RAUL, AND HERNANDEZ, VICENTE. Self-adaptive systems: A survey of current approaches, research challenges and applications. Expert Systems with Applications, Londres, v. 40, n. 18, p. 7267-7279, 2013.

SOMMERVILLE, I. Engenharia de Software. 8.ed. S o Paulo: Pearson Addison-Wesley, 2007.

WHITTLE, J.; SAWYER, P., BENCOMO, N., CHENG, B. H., AND BRUEL, J.-M.
RELAX: a language to address uncertainty in self-adaptive systems requirement,
Londres, v. 15, n. 2, p. 177-196, mar. 2010.