

UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA
2010.2



Extensão de um Editor BPMN para anotação de
Requisitos Não-Funcionais

TRABALHO DE GRADUAÇÃO EM ENGENHARIA DE
SOFTWARE E SISTEMAS DISTRIBUÍDOS

Aluno: Aluisio Rodrigo Fonseca de Santana (arfs@cin.ufpe.br)

Orientador: Nelson Souto Rosa (nsr@cin.ufpe.br)

Recife, 31 de agosto de 2010.



Universidade Federal de Pernambuco

Centro de Informática

Graduação em Engenharia da Computação

Aluisio Rodrigo Fonseca de Santana

Extensão de um Editor BPMN para anotação de Requisitos
Não-Funcionais

*Este trabalho foi apresentado ao
Centro de Informática da
Universidade Federal de
Pernambuco como requisito parcial
para a obtenção do Grau de Bacharel
em Ciências da Computação.*

ORIENTADOR: NELSON SOUTO ROSA

Orientador

Nelson Souto Rosa

Aluno

Aluisio Rodrigo Fonseca de Santana

Recife, 14 de Dezembro de 2010

“A educação é aquilo que permanece quando alguém esquece tudo o que
aprendeu no colégio.”
(Albert Einstein.)

Agradecimentos

Primeiramente agradeço a Deus pelas oportunidades que me foram dadas na vida. Aos meus pais, Edinize Santana e Clóvis Santana, que acreditaram em mim dando-me forças para prestar o vestibular por duas vezes, pois fui jubilado da Universidade (vestibular de 1999), por estar em Brasília, construindo um outro pedaço de minha vida: o Corpo de Bombeiros. Do mesmo modo, agradeço a minha recém criada família: eu, Marlane Santana (nega véia) e Maria Júlia (minha pequena M.J.), hoje com cinco meses.

Agradeço também aos meus amigos do CIn (D'oleron, Bruno Silva, Cristian Diego e outros), que contribuíram com minha aprendizagem nesta longa jornada que foi minha graduação. Em especial, agradeço aos meus amigos Xico Bento (André Souza), que mesmo turbinado com atividades soube dedicar seu precioso tempo para esclarecimentos quanto a questões de implementação, principalmente *bugs*, e Rita de Cássia, outra pessoa cheia de atividades, mas que sempre esteve sempre pronta para esclarecimentos quando solicitada, além de ter contribuído sobremaneira para com a conclusão do presente Trabalho de Graduação.

Ao Orientador, Prof. NELSON SOUTO ROSA, pelo incentivo, compromisso e presteza no auxílio às atividades.

Por último, agradeço aos professores e demais funcionários do Centro de Informática e todos aqueles que contribuíram direta ou indiretamente na conclusão de minha graduação.

Resumo

Com a adoção da abordagem orientada a serviços nos processos de automação de negócio, há uma crescente necessidade de se considerar requisitos não funcionais no nível de processo de definição de negócio. Apesar do reconhecimento de sua importância, a incorporação de requisitos não funcionais, no nível de definição do processo de negócio para sua subsequente automação, ainda é considerada um desafio, pois, dentre outros fatores, as linguagens de modelagem de processos de negócio, tal como o BPMN, não oferecem artefatos para expressar esses requisitos. Portanto, o presente trabalho apresenta uma extensão do BPMN Editor, para incorporação de requisitos não funcionais no processo de modelagem de negócio permitindo que anotações sejam feitas aos elementos do BPMN Editor existentes.

Abstract

With the adoption of service-oriented approach in business process automation, there is a growing need to consider non-functional requirements at the business level process. Despite the recognized importance, the incorporation of non-functional requirements in the process level for its subsequent automation is still considered a challenge because, among other factors, the modeling languages of business processes, such as BPMN, does not provide artifacts to express these requirements. Therefore, this paper presents an extension of BPMN Editor for incorporation of non-functional requirements in business process modeling by allowing annotations to be made to existing elements of BPMN Editor

Sumário

1. Introdução.....	1
1.1. Contexto/Motivação.....	1
1.2. Objetivos.....	3
1.3. Estrutura.....	3
2. Conceitos Básicos	4
2.1 . BPMN/Editor.....	4
2.2 Requisitos Não Funcionais/ RNFs	6
2.3 Abstrações Não Funcionais.....	7
3. Proposta.....	12
3.1 Visão Geral	12
3.2 Extensões.....	12
3.3 Implementação.....	13
4. Exemplo.....	18
5. Conclusão e Trabalhos Futuros	24
Referências Bibliográficas.....	25

1. Introdução

1.1. Contexto/Motivação

Engenharia de Requisitos tem seu destaque no processo de produção de software, pois o sucesso de um sistema, depende de um bom levantamento de requisitos. Na prática, problemas podem ser facilmente identificados nesta fase [1]. Tais problemas ocorrem devido a fatores como falta de compreensão do negócio pelo analista do sistema, mau entendimento da finalidade do sistema, falhas na comunicação entre analistas de negócio e analistas de sistemas. Desta forma, dificuldades podem comprometer o funcionamento do produto que se pretende fornecer.

Com o intuito de refletir o ambiente de negócio nas primeiras fases do desenvolvimento de um software, a comunicação de engenharia de requisitos tem reconhecido a importância do uso de preocupações relativas ao negócio para direcionar a elicitação de requisitos[1]. Modelos de negócio descrevem a estrutura e o comportamento de uma organização, descrevem a lógica de uma organização e como ela cria, entrega ou agrega valor econômico, social, ou outras formas de valor ao seu produto.

Diante desse cenário, surge a especificação BPMN (Business Process Modeling Notation), visando aproximar analistas de negócios, analistas de sistemas, desenvolvedores e usuários finais. A notação BPMN tem sido amplamente adotada entre analistas de domínio e arquitetos de sistemas como uma linguagem para definição de modelos de processo de negócio [2].

BPMN tem sido usado em diversos tipos de processo, tais como processos de negócio privado, os quais representam os processos internos de uma organização (recursos humanos, compras, estoque, etc), processos abstratos, que representam interação entre processos privados e outro participante (uma consulta médica), processos colaborativos, que descrevem entre duas ou mais entidades de negócio (agência de viagens).

Tradicionalmente, a modelagem do processo de negócio focaliza na captura de tarefas funcionais que produzem um produto ou serviço específico na forma de processos e sub-processos. Apesar de existir uma cobertura abrangente das características funcionais de negócio, tais como especificação de linguagens e métodos formais, os requisitos não funcionais (deste ponto em diante referidos

apenas por RNFs), em particular os relacionados a tarefas de negócios, não são identificados[3][4].

Consequentemente, informações importantes do negócio podem não ser consideradas no momento da modelagem, ou até mesmo podem ser detectadas em outras fases, como por exemplo, na implementação, fazendo com que o custo de correção seja elevado, pois segundo Cysneiros [5] erros por não tratar os RNFs adequadamente são os mais caros de se corrigir. Portanto, RNFs desempenham um papel importante no processo de desenvolvimento de software, sendo necessária sua representação na fase de modelagem do processo de negócio.

Em termos computacionais, a definição do processo de negócio tem sido realizada por meio da composição de serviço. Uma vez que se espera que o processo de negócio seja executado por meio de uma composição de serviços (orquestração em WS-BPEL[6], por exemplo), fica evidente a necessidade de identificação e representação de requisitos não funcionais no processo de modelagem. Provedores de serviços precisam garantir que os RNFs sejam satisfeitos, quando postos à disposição, uma vez que acordos de nível de serviço (*Service Level Agreement* - SLA) são feitos com os consumidores de serviços [7].

Apesar do reconhecimento de sua importância, a incorporação de requisitos não funcionais no nível de definição do processo de negócio para sua subsequente automação, ainda é considerada um desafio pelas seguintes razões [8] [9]:

- Em geral, as linguagens de modelagem de processos de negócio, tal como o BPMN, não oferecem artefatos para expressar esses requisitos;
- Ausência de uma taxonomia padrão que descreva requisitos não funcionais no contexto de processos de negócio;
- Ausência de mecanismos de verificação e mediação de conflitos entre os requisitos que permitam satisfazer as restrições e preferências elicitadas para o processo;
- Não é clara a definição de requisitos não funcionais na composição de serviços; e,
- Complexidade em traduzir esses requisitos para serem

realizados nos atuais ambientes de execução.

Baseado nesse contexto, o problema estudado é como representar requisitos não funcionais em processos de negócio modelados em BPMN.

1.2. Objetivos

A notação BPMN possui diversas vantagens, dentre as quais podemos destacar o uso de construções familiares (os elementos gráficos são de fácil entendimento, intuitivos) e um conjunto de regras (baseadas na técnica de fluxogramas) comuns em processos de negócio [10]. Entretanto, segundo Recker (2006) [11] BPMN tem algumas limitações relacionadas à construção de modelos, sugerindo que os usuários finais não estão aptos a descrever todos os fenômenos encontrados em situações reais. Pavlovski [12] observa que BPMN possui uma deficiência em se tratando de descrição de requisitos não funcionais.

Há várias ferramentas que dão suporte ao BPMN, como podem ser verificadas mais adiante, entretando, em geral, elas não fornecem suporte para representar requisitos não funcionais (RNFs). Sendo assim, o objetivo do presente trabalho é estender uma ferramenta de modelagem do BPMN, a saber, o BPMN Editor[13], com intuito de possibilitar a incorporação de requisitos não funcionais.

1.3. Estrutura

Além deste capítulo de introdução, o presente trabalho é estruturado como segue:

No Capítulo 2 são introduzidos os conceitos básicos associados ao presente trabalho. Os conceitos relacionados ao padrão BPMN, por meio de uma descrição básica de seus elementos, e contextualizando o leitor ao padrão BPMN. Em seguida, há ainda uma breve descrição de requisitos não funcionais. Por fim, são apresentados os conceitos de abstrações não funcionais;

No Capítulo 3 é apresentada a proposta do presente trabalho. Será apresentada uma descrição de como foi implementada a ferramenta, produto deste trabalho, além de suas extensões.

No Capítulo 4 apresentamos um exemplo clássico, mostrando as anotações de RNFs, realizados pela própria ferramenta.

Por fim, o Capítulo 5 apresenta a conclusão e os próximos passos do presente trabalho.

2. Conceitos Básicos

2.1 . BPMN/Editor

BPMN (*Business Process Modeling Notation*) [14], atualmente mantido pela OMG, é um padrão de modelagem de processo de negócio, que provê uma notação gráfica para a modelagem de processos em um BPD (*Business Process Diagram*), o qual consiste basicamente em uma rede de objetos gráficos, tais como *Pools*, *Activities*, *Gateways* e *Sequence/Message Flows* que definem seu negócio. A Figura 1 apresenta um conjunto básico de elementos BPMN.

O elemento BPMN *Pool* representa um participante no processo de negócio. Um participante pode ser uma entidade de negócio específica, (e.g. uma empresa), ou pode ser uma função de negócio mais genérica (e.g. um comprador, vendedor, ou fabricante). O *Pool* pode conter *Lanes*, que são elementos que agrupam atividades internas de um participante. Uma *Activity* é uma tarefa desenvolvida dentro do processo de negócio, que pode ser classificada como um *Process*, *Sub-Process* ou *Task*. O elemento *Process* é uma atividade desempenhada dentro de (ou entre) organizações ou empresas [14]. No BPMN, um *Process* é descrito como um grafo de Flow Objects, os quais são um conjunto de outras atividades. O elemento *Sub Process* é uma atividade composta que, na modelagem do processo de negócio, indica o fluxo de outras atividades, como por exemplo, elementos do tipo *Task*.

Task representa uma atividade atômica no diagrama. Esse elemento é usado quando uma tarefa não pode ser decomposta em outras. *Sequence Flow* indica a ordem de execução de *Tasks* e *Sub-Process* no *Pool*. *Message Flow* representa o fluxo de mensagens entre os participantes.

Pools podem ainda conter *Gateways* e *Events*. O primeiro controla o paralelismo de *Sequence Flows*, o segundo dispara uma subsequente *Sequence Flow*.

Além dos elementos básicos, BPMN também define *Artifacts* (artefatos), os quais permitem que informações adicionais sejam inseridas no processo de negócio, não afetando *Sequence/Message Flows*.

São exemplos de *Artifacts*: *Data Object*, que representam dados de

entrada e saídas produzidas pelas atividades, *Groups*, que permitem agrupar várias atividades facilitando assim a análise e/ou melhorar a documentação e *Text Annotations* que incorporam informações adicionais ao diagrama (BPD).

O principal objetivo do BPMN foi prover uma notação que fosse fácil de entender por todos os envolvidos no processo de negócio, desde analistas de negócios, os quais criam o primeiro esboço dos processos, desenvolvedores, responsáveis pela implementação, até as pessoas de negócio que irão administrar e monitorar aqueles processos [14].

BPMN visa à criação de um mecanismo padronizado para reduzir a distância entre o projeto e a implementação de processos de negócio [15], visto que ela permite ser mapeada para diversos formatos de execução, tal como o WS-BPEL [6].

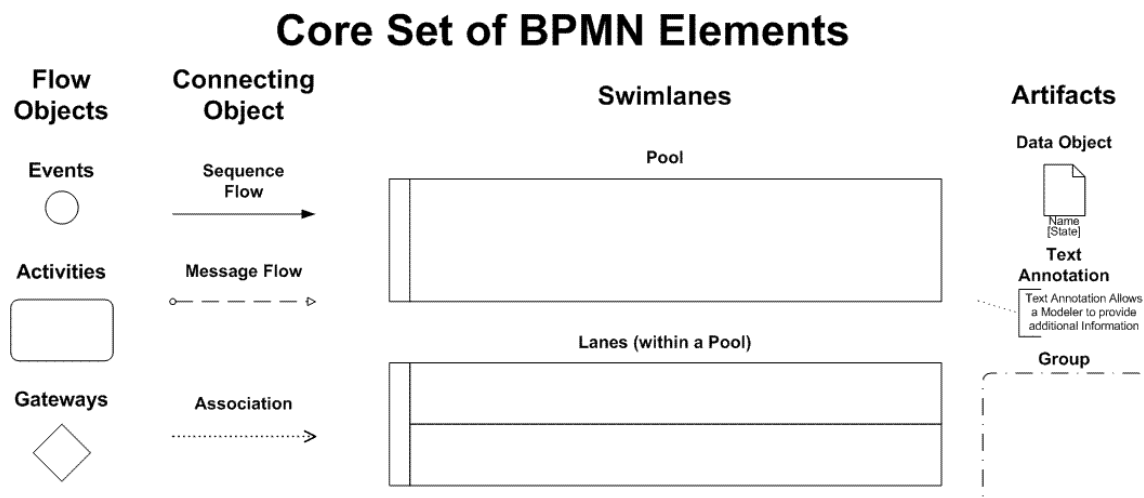


Figura 1 - Conjunto básico de Elementos do BPMN

BPMN tem sido amplamente adotada por analistas de negócio e arquitetos de sistemas como linguagem de definição de modelo de processo de negócio, facilitando o entendimento nos campos de negócio e tecnológico [2].

Atualmente, o BPMN é suportado por mais de 60 ferramentas, dentre as quais podemos citar: Oryx [16], BizAgi [17], ActiveVOS [18], Visual Architect™ [19], WBI Modeler™ [20], BPMN Modeler [6]. Este último é parte do *SOA Tools Platform Project* da plataforma Eclipse.

O BPMN Modeler, um *plugin* desenvolvido para plataforma Eclipse, é

um editor de Diagrama de Processo de Negócio (*Business Process Diagram* - BPD) para analistas de negócio, desenvolvedores e usuários em geral, possibilitando que diferentes *stakeholders* estejam envolvidos em diferentes estágios de desenvolvimento. O editor BPMN Modeler utiliza dois *frameworks*, o EMF [22] (*Eclipse Modeling Framework*), usado para representar o modelo, e o GEF [21], usado para proporcionar uma notação gráfica. O editor é flexível, o que possibilita a inclusão de novos elementos gráficos.

2.2 Requisitos Não Funcionais/ RNFs

Diferentemente de requisitos funcionais, que descrevem o que um software deve fazer, RNFs especificam restrições globais sobre como o software opera ou como a funcionalidade é apresentada [23]. RNFs não expressam nenhuma função a ser desempenhada pelo software [24].

Enquanto requisitos funcionais possuem efeitos localizados, RNFs geralmente estão dispersos em partes distintas do software [23]. Outra diferença importante entre requisitos funcionais e RNFs é que, o primeiro tipo é incorporado ao sistema gradualmente, de forma que no final do processo de elaboração do software, todos os requisitos devem estar satisfeitos, enquanto o segundo não é diretamente implementado.

RNFs são tão importantes quanto os requisitos funcionais. Não são apenas propriedades desejadas do sistema, mas também são aspectos sem os quais a aplicação pode ter seu funcionamento comprometido [23].

Segundo [25], não há uma definição formal ou lista completa de RNFs, nem se é possível apresentar um esquema de classificação universal, único, que acomode todas as necessidades de aplicações de diferentes domínios [20]. Assim, para cada aplicação e domínio diferentes são feitas classificações.

A satisfação de RNFs não é trivial, pois podem surgir conflitos, tanto com outros RNFs como também com requisitos funcionais. São exemplo de requisitos não funcionais: acessibilidade, segurança, confidencialidade, desempenho, portabilidade, consistência, manutenabilidade, robustez, preço e disponibilidade.

Segundo [23], apesar de sua importância, RNFs não são considerados no domínio de sistemas dinâmicos, em geral, devido a diversos fatores, tais como:

1. Descrição: RNFs são descritos informalmente;
2. Abstração: RNFs não são implementados diretamente;
3. Integração: RNFs não são integrados à parte funcional;
4. Suporte Técnico: RNFs são raramente suportados por linguagens, metodologias ou ferramentas;
5. Natureza distinta: RNFs são mais complexos de se tratar;
6. Validação: é difícil validar os RNFs no produto final;
7. Correlação: frequentemente RNFs são conflitantes uns com os outros (*e.g.* Segurança *versus* Desempenho);
8. Implementação: RNFs são comumente incorporados ao sistema pela infraestrutura na qual o software está sendo desenvolvido, ao invés de serem implementados por programadores;
9. Competência Funcional: a separação de requerimentos funcionais e não funcionais não é facilmente definida; e
10. Execução: ausência de notação para descrição de RNFs de um modo preciso dificulta sua verificação em mudanças em tempo de execução.

Da mesma forma, apesar de se estar inserido no contexto dinâmico, os fatores acima elencados são facilmente percebidos no contexto de RNFs em geral.

Apesar das dificuldades, a necessidade de tratar explicitamente os RNFs é evidente, pois (i) há uma crescente demanda por aplicações tolerante a falhas, onde RNFs cumprem um papel importante e é imprescindível que tais requisitos sejam satisfeitos, (ii) como um tipo de requisito, é natural que sua integração ao processo de desenvolvimento de software, (iii) interações entre requisitos funcionais e RNFs são tão coesas, na maioria dos casos, que RNFs podem não ser satisfeitos apenas como uma consequência de decisões de projeto tomadas para satisfazer requisitos funcionais e (iv) um tratamento explícito de RNFs nos possibilita prever a qualidade do produto final de uma forma mais confiável e razoável.[26]

2.3 Abstrações Não Funcionais

Nesta seção, apresentaremos um conjunto de abstrações para raciocinar e representar as propriedades não funcionais, a saber: NF-Attributes, NF-Statements, NF-Actions, NF-Effects, NF-Properties e NF-Groups

A especificação BPMN permite que novos tipos de informação sejam incorporados ao diagrama de processo de negócio [14], e conforme dito anteriormente, a representação de RNFs poderia ser feita através de Artefatos, especificamente, por meio de *Text Annotation*. Entretanto, para se ter melhor expressão e compreensão, e com intuito de modelar nosso módulo não funcional (NF-Module), descrevemos a seguir as abstrações não funcionais definidas para compor o diagrama de processo de negócio não funcional (Non Functional Process Diagram).

O **Non Functional Attribute** (NF-Attribute) representa características não funcionais. Um NF-Attribute pode ser simples ou composto. Um NF-Attribute simples não pode ser decomposto (subrequisito não funcional primitivo), enquanto que o composto pode ser dividido em outros NF-Attributes (sub-requisitos não funcionais simples ou compostos). Para um NF-Attribute composto, seus sub-atributos participam de três formas diferentes de composição: *All*, todos sub-requisitos são necessários para compor a definição do NF-Attribute; *OneMore*, um ou mais sub-requisitos são necessários na definição do NF-Attribute; e *OneAny*, exatamente um sub-requisito é necessário para definir o NF-Attribute. Para um NF-Attribute simples o valor da composição é *None*. O NF-Attribute *Performance* (Desempenho) pode ser decomposto em *Reaction Time*, *Send Time*, *Execution Time*, *Response Time*, *Throughput* e *Scalability*.

Non Functional Action (NF-Action) representa um aspecto de software ou uma característica de hardware que realiza (operacionaliza) ou afeta (positiva ou negativamente) um ou mais NF-Attribute. Aspectos de software significam decisões de projeto, algoritmos, estrutura de dados, e assim por diante, enquanto que aspectos de hardware se referem a recursos computacionais disponíveis para execução do sistema. O impacto de uma NF-Action sobre um NF-Attribute é representado por um **Non Functional Effect** (NF-Effect).

Cada efeito (NF-Effect) possui as seguintes propriedades: Tipo de ação (implementa, afeta, ou monitora), que representa o efeito de uma NF-Action relacionado a um dado NF-Attribute; *Effect* que define o impacto (positivo ou negativo) da NF-Action sobre um dado NF-Attribute e *Degree*, que define o grau de efeito (-3,-2,-1, None, +1,+2,+3) da NF-Action sobre o requisito não funcional.

Non Functional Statement (NF-Statement) representa restrições definidas num NF-Attribute que ajudam a tomar decisões na sua implementação. Ela representa as restrições sobre as quais as atividades de um processo de negócio devem respeitar como parte de seu requisito. NF-Statement possui as seguintes propriedades: *Constraint* representa o nível de satisfação desejado para o RNF. O nível define que ações podem ser reunidas para satisfazer um requisito; *Priority* representa a prioridade da restrição relacionada aos demais. Dado um conjunto de NF-Actions que afetam ou implementam um NF-Attribute, a satisfação de um NF-Statement consiste em implementar as NF-Actions de acordo com as seguintes restrições: *Strong*, *Medium*, *Weak*, *Present*. A representação gráfica da restrição é feita numa *Task*, conforme tabela 1 adiante.

Non Functional Property (NF-Property) representa o conjunto de propriedades específicas (propriedades e valor) de uma NF-Action de acordo com o NF-Statement.

Non Functional Group (NF-Group) representa um grupo de NF-Attributes, já com as restrições definidas pelo NF-Statement, organizados na forma de um predicado. NF-Group possui a propriedade tipo de operador que define o operador lógico AND, OR ou XOR.

Na Figura 2, temos um esboço macro do metamodelo do Diagrama do Processo de Negócio (BPD), tal como comumente é implementado pelas ferramentas de modelagem em BPMN existentes. Nele foi incorporada uma extensão, representada pelo conjunto de elementos em cinza, que possibilitará a representação de RNFs (NF-Module).

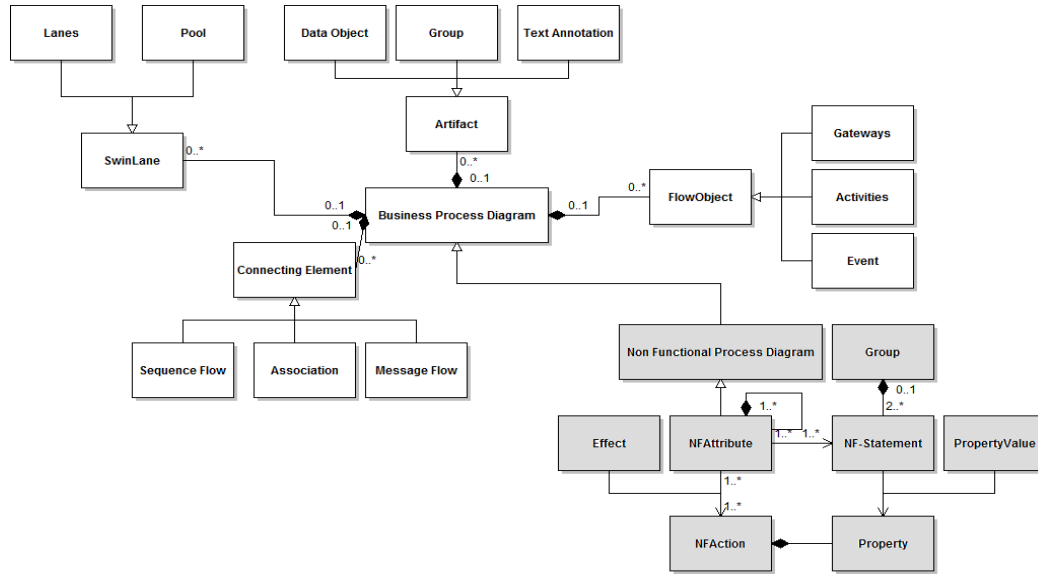


Figura 2 - Metamodelo do Diagrama do Processo de Negócio e NF-Module

Segundo [9], podemos representar as abstrações referentes aos RNFs num diagrama de classes, o qual norteará a implementação do presente trabalho. O diagrama de classe pode ser visto na Figura 3 abaixo.

Non-Functional Abstractions

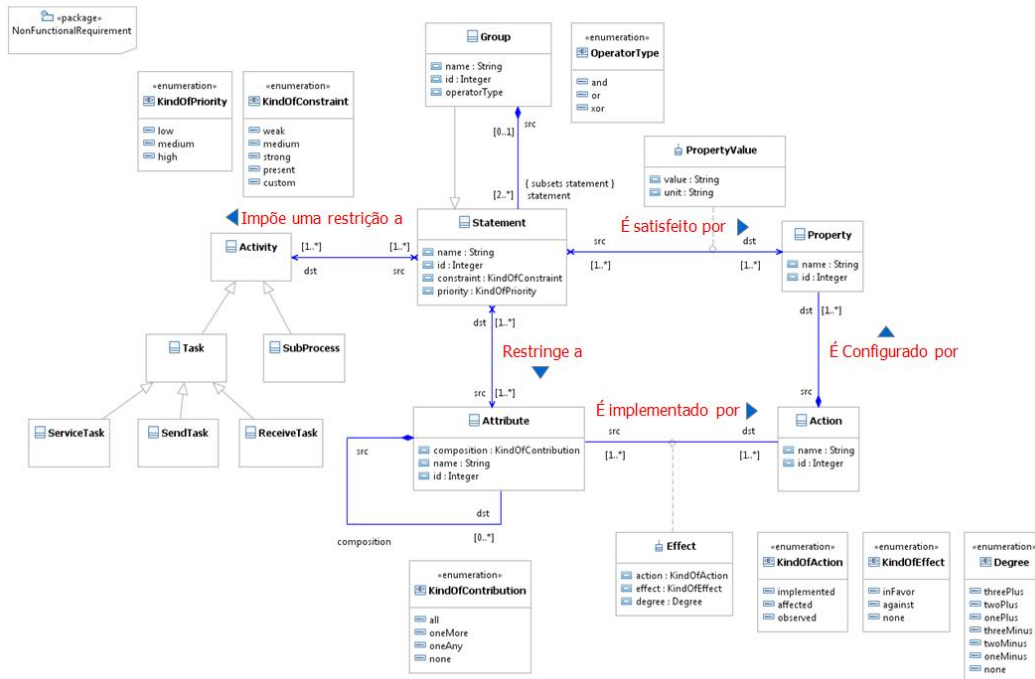


Figura 3 - Diagrama de Classes das Abstrações Não Funcionais[9]

Uma especificação mais detalhada e formal das Abstrações aqui apresentadas pode ser encontrada em [26]

3. Proposta

Nesta seção, apresentaremos a proposta que realiza parte do conjunto de abstrações não funcionais citadas na seção anterior, que possibilitou a realização da extensão do editor BPMN, o NF-Module

3.1 Visão Geral

Com o intuito de capturar os RNFs dentro do processo de modelagem de negócio, faz-se necessário dispor de uma notação que dê suporte gráfico para representação semântica de requisitos não funcionais. Assim, a ferramenta tornará possível a incorporação de requisitos não funcionais na fase de modelagem, permitindo que as abstrações referentes aos RNFs sejam representadas graficamente. Além disso, permite que anotações sejam feitas automaticamente no respectivo arquivo XML gerado.

Conforme descrito na seção 2.1, BPMN não provê mecanismos para representação de requisitos não funcionais. No entanto, dentre os símbolos usados na construção do BPD, Artefatos (*Artifacts*) podem ser usados para representar tais requisitos. Artefatos foram definidos com o fim de permitir a extensão da notação básica de modelagem BPMN, possibilitando que as extensões representem situações específicas [15].

Apesar de Artefatos poderem ser usados para expressar RNFs, conforme dito anteriormente, particularmente por meio de *Text Annotations*, a identificação explícita de RNFs por meio de uma notação facilita a modelagem e ajuda a obter uma melhor interpretação por especialistas de um determinado requisito não funcional [27].

O NF-Module estabelece um conjunto de anotações para representar RNFs em processos definidos em BPMN, o que permite a participação de diferentes *stakeholders*, tais como analista de domínio, analista de sistemas além de provedores de serviços, facilitando ainda a comunicação e entedimento entre eles. Vale salientar que não pretendemos inserir abstrações não funcionais ao padrão BPMN, de modo que esse padrão permanece inalterado.

3.2 Extensões.

No contexto das abstrações não funcionais, que são tratadas na seção

2.3, para a realização do módulo estendido do BPMN Editor (o NF-Module) foram implementados, do ponto de vista do modelo, os seguintes conceitos: NF-Attribute, NF-Action, NF-Statement, NF-Effect e NF-Property.

A interface gráfica, detalhada mais adiante na seção 3.3, foi realizada a partir da extensão do pacote *org.eclipse.iu.views*, na forma de um plugin, incorporado ao arquivo XML (*MANIFEST.MF file*) do BPMN Editor.

O módulo estendido da ferramenta é implementado como um plugin para o Eclipse, a partir da extensão do BPMN Editor. Os conceitos de NF-Attribute e NFAction foram estendidos de um elemento existente, o Artefato (*Artifact*), para que ambos pudessem ser representados graficamente. Apesar de estenderem Artefato, cada conceito possui seus atributos, o que faz com que cada um seja um elemento novo representado no Editor estendido.

Do ponto de vista da modelagem de processo de negócio, através do uso da ferramenta produto do presente trabalho, apenas os elementos NF-Attribute, NF-Action e NF-Statement foram implementados.

3.3 Implementação

Nesta seção serão apresentados aspectos gerais referentes ao desenvolvimento da ferramenta estendida do BPMN Editor, o módulo de requisito não funcional (NF-Module).

A linguagem de programação usada para construir a extensão foi Java, pois o BPMN Editor, base no qual o módulo (NF-Module) foi estendido, utiliza tal linguagem. Isso não trouxe custo adicional, pois além de ser uma linguagem poderosa, orientada a objetos, possibilitou o reuso de código durante a implementação. A versão do eclipse utilizada foi o Eclipse for RCP/Plugin Developer do Eclipse Galileo.

Para criação de um ambiente para modelagem gráfica foram utilizados um conjunto de plugins do Eclipse, sendo o principal deles o GEF (*Graphical Editing Framework*), que representa o editor gráfico em si.

NF-Module dá suporte para modelagem de RNFs e os vincula a elementos BPMN existentes. O módulo estendido do BPMN é responsável pelas

anotações referentes aos RNFs, gerando os elementos graficamente, fornecendo ainda um arquivo XML que reflete, de maneira estruturada, as anotações feitas no diagrama. As anotações de RNFs são feitas através da associação dos elementos que representam os RNFs aos elementos BPMN do Editor, especificamente elementos do tipo *Task*.

Como dito anteriormente, a representação gráfica dos elementos NF-Attribute e NF-Action foi possível extendendo-se um elemento BPMN conhecido: o Artefato. A Figura 3 mostra a hierarquia do NF-Attribute e NF-Action.

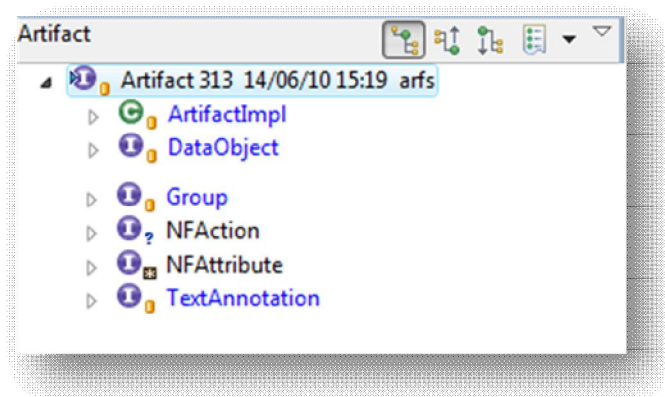


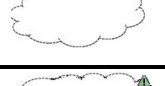
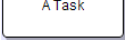
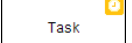
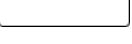
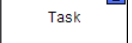
Figura 4 – NF-Action e NF-Attribute extendem Artefato

O NF-Statement representa restrições definidas para um NF-Attribute e que devem ser atendidas pelas atividades funcionais (*Activities* BPMN) do processo de negócio, e desta forma, sua representação gráfica é realizada por meio de anotações nos elementos existentes do BPMN, especificamente no elemento *Task*, e nos elementos de RNFs. Como exemplo, consideremos uma *Task* associada a um NF-Attribute de *Performance*. Quando definimos, através da interface de *NR-Requirements*, a restrição *Strong*, significa que estamos impondo uma restrição àquela *Task*, de modo que aquela *Task* deve satisfazer o NF-Attribute com a restrição *Strong*. Quanto ao NF-Attribute, a restrição imposta a ele é a definição de ser High, Medium, ou Low.

A Tabela 1 mostra a representação gráfica das abstrações definidas na Seção 2.3 que foram implementadas do presente trabalho. Os NF-Statements representados na Tabela 1, por motivos ilustrativos, são válidos para o NF-Attribute

de Desempenho (*Performance*).

Tabela 1 – representação gráfica das principais abstrações

Non Functional Attribute		
Non Functional Attribute	Composition = All	
	Composition = One More	
	Composition = One Any	
Non Functional Statement	Priority = Low	
	Priority = Medium	
	Priority = High	
Non Functional Statement	Constraint = Weak	
	Constraint = Medium	
	Constraint = Strong	
	Constraint = Present	
	Constraint = Custom	
Non Functional Action		

As demais abstrações (NF-Effect e NF-Property) são anotações feitas aos elementos gráficos, e não possuem um elemento gráfico que as representem. As anotações são realizadas a partir da *View NF-Requirements*, interface por meio da qual serão feitas as anotações de RNFs.

A Figura 4 mostra uma visão geral da ferramenta e seus principais componentes, os quais serão detalhados a seguir.

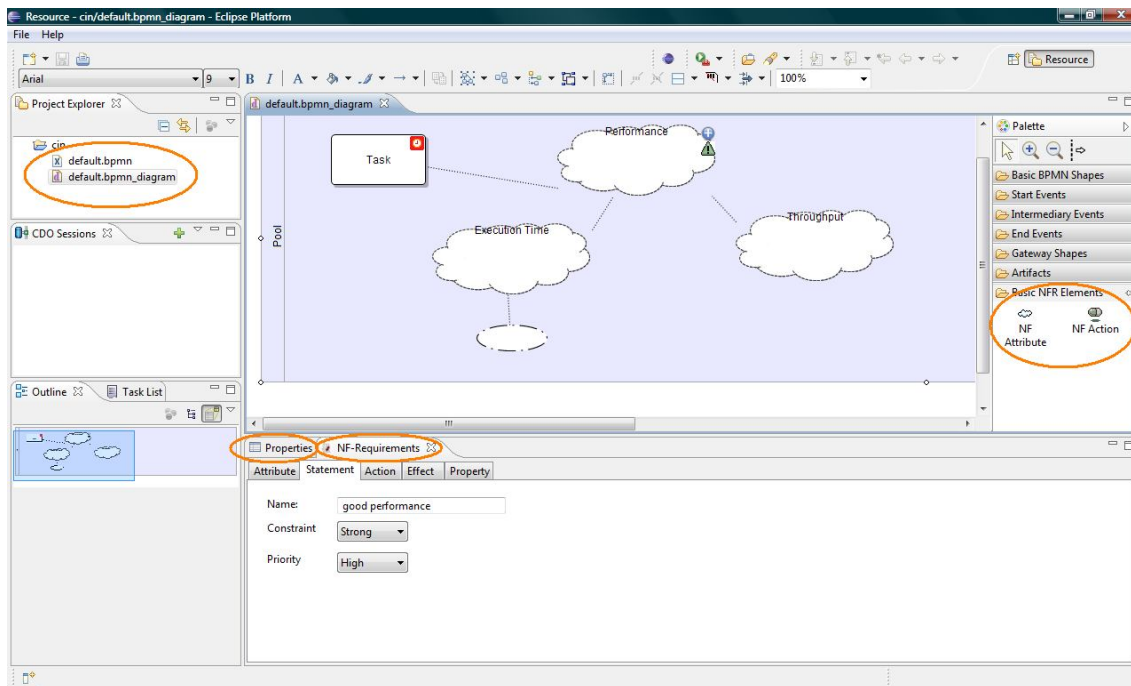


Figura 5 - Visão Geral BPMN Editor com plugin NF-Modulo

Na Figura 4, podemos destacar as principais partes do Ambiente Gráfico, são elas:

O *palette*, à direita, inclui elementos BPMN além dos elementos básicos para fornecer tratamento aos RNFs, estes incorporados ao BPMN Editor através do NF-Module.

NF-Attributes são representados com elementos gráficos na forma de nuvem. NF-Actions são representados por cápsulas. As restrições impostas por NF-Statements às *Tasks* BPMN são representadas por elementos gráficos que indicam os RNFs, por exemplo, o relógio indica o requisito de Desempenho, e uma cor associada para indicar o grau de restrição, por exemplo, vermelho indica uma forte restrição (*Strong*). Enquanto que outros elementos que compõem a modelagem de RNFs (NF-Effect e NF-Property) não constituem elementos gráficos em si, tendo sua representação por meio de anotações, as quais podem também ser visualizadas nos outros elementos não funcionais que compõem o diagrama.

A parte central representa a área de modelagem, a qual contém a definição do processo de negócio usada na modelagem gráfica.

No canto superior esquerdo, temos dois arquivos gerados

automaticamente, um com extensão “.bpmn” e outro com a extensão “.bpmn_diagram”, ambos podem ser visualizados com um XML editor no Eclipse. No primeiro são inseridas as informações que correspondem ao gráfico em si, enquanto no segundo, constam as informações referentes aos relacionamentos entre os elementos gráficos, além de suas anotações. As anotações referentes aos elementos, as quais representam os RNFs, são também armazenadas no arquivo XML com extensão “.bpmn”.

Na parte inferior da Figura 4, temos uma *View* para os RNFs. Ela representa a interface para interação do usuário, onde poderão ser realizadas as anotações referentes às abstrações não funcionais. A view é composta por cinco abas, cada uma permitindo a definição e/ou alteração das abstrações.

Ainda na parte inferior da Figura 4, ao lado esquerdo da View de RNFs, podemos visualizar outra *View*, destinada à exibição e alteração de propriedades existentes em elementos gráficos. Todas as propriedades referentes a esses elementos, *Contribution*, na aba de NF-Attribute, por exemplo, podem ser alteradas por essa *View*.

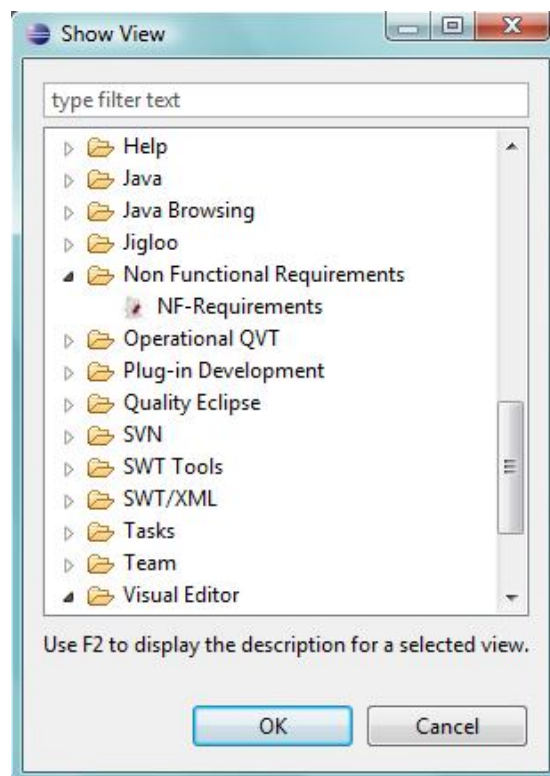


Figura 6 - NF-Requirements View

A View de RNFs pode ser visualizada através da barra de menu do Eclipse Window -> Show View -> Other-> Non Functional Requirements, conforme a Figura 6.

4. Exemplo

Nesta seção, apresentaremos um exemplo utilizando a ferramenta implementada como parte deste trabalho de graduação. Para demonstrar um exemplo, consideramos uma Agência de Viagens Virtual (VTA), que disponibiliza seus serviços através de seu Portal na Internet para assuntos de viagem.

Clientes interagem com a VTA para uso de serviços, desde pagamentos a assuntos não computacionais (por exemplo, recebimento do contrato). O portal VTA permite que seus clientes realizem operações tais como: comprar uma passagem aérea (nacional e internacional), alugar de carro, reservar um hotel, comprar um pacote turístico, etc.

Neste exemplo, vamos considerar o Caso de Uso Comprar uma passagem aérea nacional [28].

Para simplificar, vamos considerar neste exemplo que a modelagem da compra de passagem aérea nacional já está pronta (Figura 7), apenas faltando a modelagem de requisitos não funcionais, onde vamos descrever aqui os passos para incorporação de anotações de RNFs.

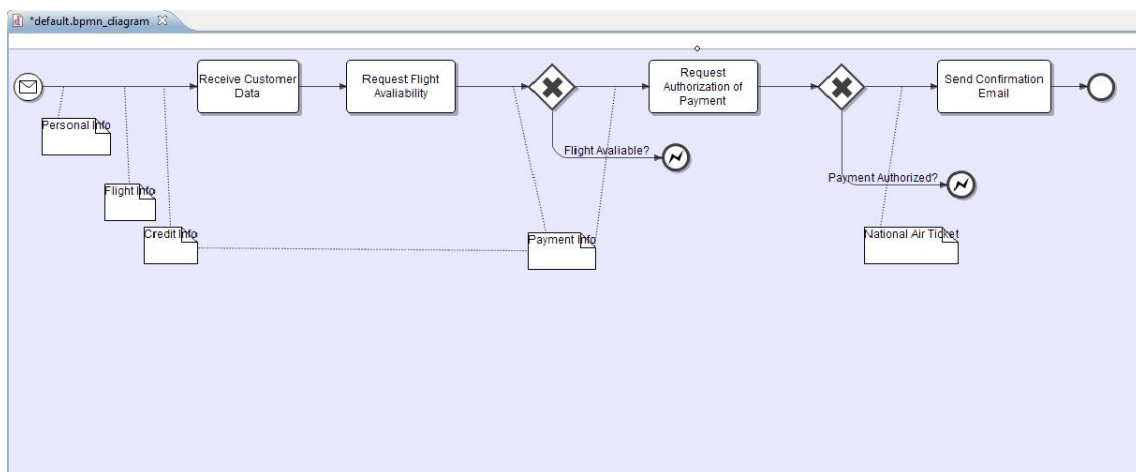


Figura 7 - Compra de Passagem Nacional Aérea sem anotações de RNFs

Para anotar requisitos não funcionais de segurança na *Task Receive*

Customer Data, vamos considerar que VTA deve manter os dados do cliente dentro da empresa, e deve garantir que apenas os funcionários do departamento comercial tenham acesso aos dados (e.g. o projetista do sistema não tem acesso a esses dados).

Fazendo uso da *pallette* mostrada na Figura 5, clicamos na nuvem (que representa o RNF) e, em seguida, clicamos no diagrama. Com a View *NR-Requirements* aberta, podemos escolher quais RNF definiremos, neste caso vamos escolher *Security*, como mostra a Figura 8. Na *View*, no lado direito são mostrados o RNF selecionado, enquanto no lado esquerdo são mostrados os outros possíveis RNF.

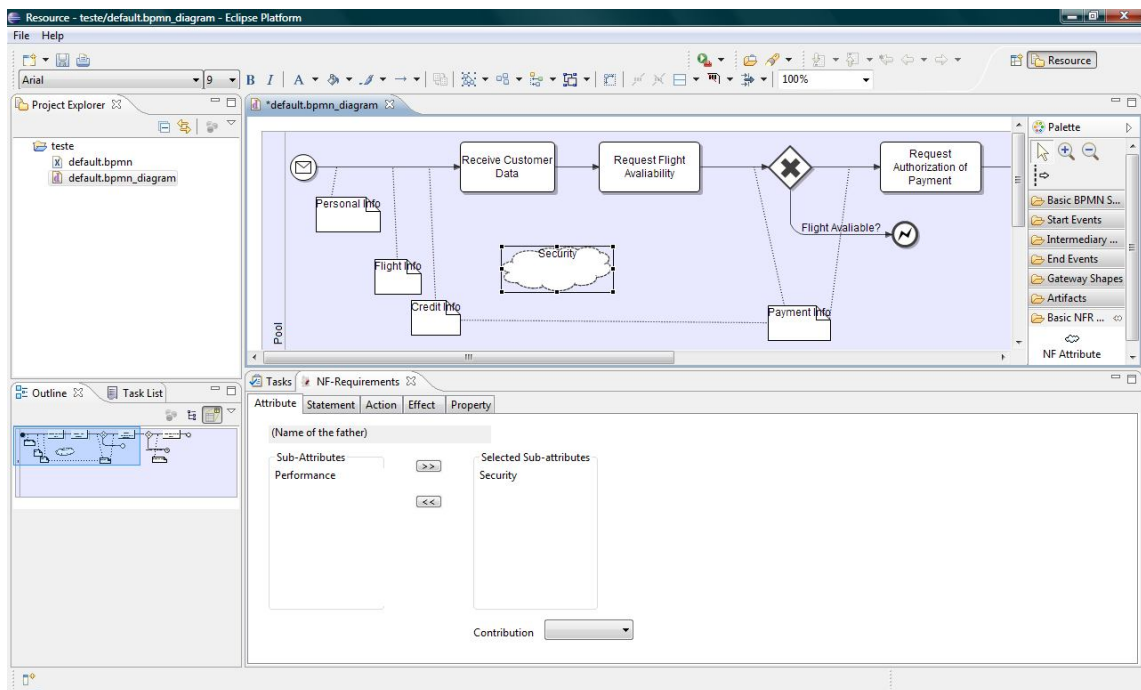


Figura 8 - Definindo *Security*

Em seguida, associamos o RNF *Security* a *Receive Customer Data*, onde este, logo após a associação, ficará anotado com tal requisito, podendo ser visualizado por meio de um cadeado. Do mesmo modo, definimos quais os sub-requisitos de *Security* devem compor a anotação, e logo após associamos os sub-requisitos ao requisito “pai” (neste caso, *Security*). Na aba de *Attribute*, na *View* de RNFs, definimos a contribuição dos sub-requisitos em relação ao NF-Attribute (no caso, *Security*). Assim vamos definir *All*, pois todos os sub-requisitos serão necessários para que a *Security* seja garantida.

De forma semelhante a definição do NF-Attribute, utilizando a *palette* à direita, escolhemos o NF-Action. Em seguida, associamos a ação (NF-Action) ao sub-requisito (NF-Attribute), conforme mostra a Figura 9.

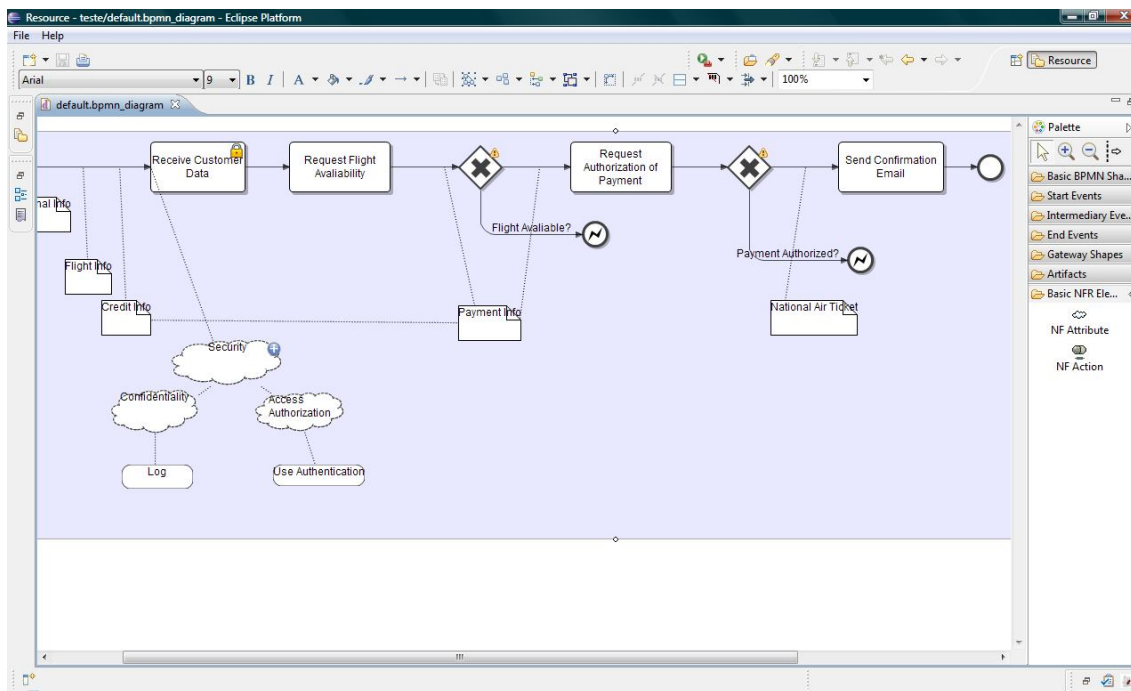


Figura 9 – Receive Customer Data com anotação de Security

Da mesma forma, vamos anotar o RNF de Segurança na *Task Request Authorization of Payment*. Para tal, vamos considerar que :

1. A agência de viagem VTA, precisa criptografar os dados do cartão de crédito;
2. VTA deve garantir que apenas os funcionários do departamento comercial tem acesso aos dados do cliente; e
3. VTA e seus parceiros digitais (por exemplo, empresas de cartão de crédito) precisam validar assinaturas digitais.

Após seguirmos o mesmo passo a passo descrito anteriormente, podemos ver a *Task Request Authorization of Payment* anotada com requisito de Segurança, conforme Figura 10.

Por fim, vamos anotar RNF de Performance, ao Request Authorization of Payment. Para tal, vamos considerar que:

1. Empresas de cartão de crédito devem responder, no máximo, em 5 segundos.
2. Empresas de cartão de crédito devem garantir diariamente uma frequência de 30

solicitações, no período entre meio dia às duas da manhã.

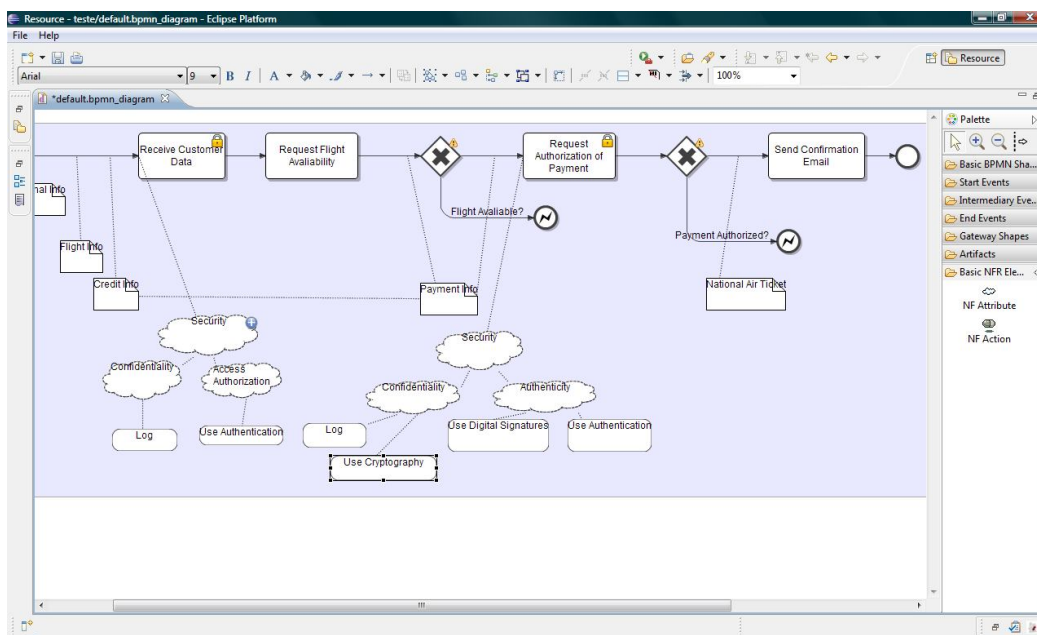


Figura 10 – Request Authorization of Payment com anotação de Security

De modo semelhante, anotamos o RNF de Performance ao Request Authorization of Payment, como pode ser visto na figura 11.

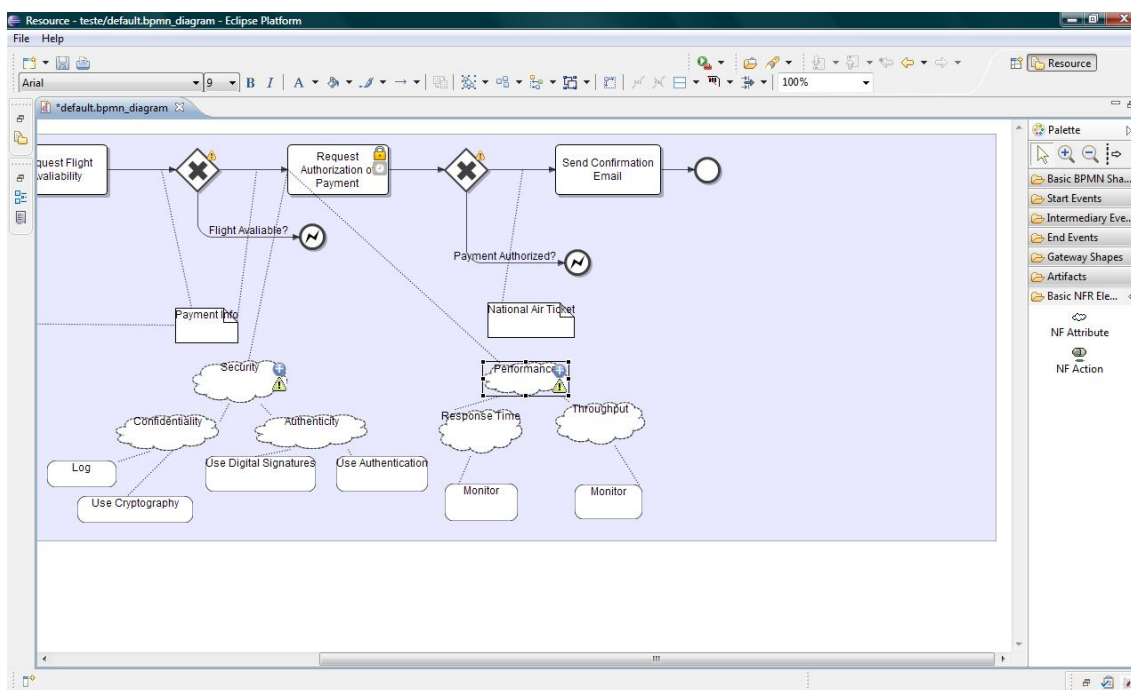


Figura 11 – Request Authorization of Payment com anotação de Performance

Na figura acima, além do cadeado, que representa segurança, temos

um relógio que representa a anotação de Performance. A restrição (Weak, Medium, Strong, Present ou Custom) pode ser configurada através da aba Statement.

Sendo assim, vamos configurar o NF-Statement para a anotação de *Performance*. Como dito anteriormente, na aba de *Statement* vamos definir um nome para o *Statement* (*Good Performance*) e escolher um valor para a restrição (*Constraint*) e Prioridade (*Priority*). A Figura 12 mostra o diagrama após essas configurações.

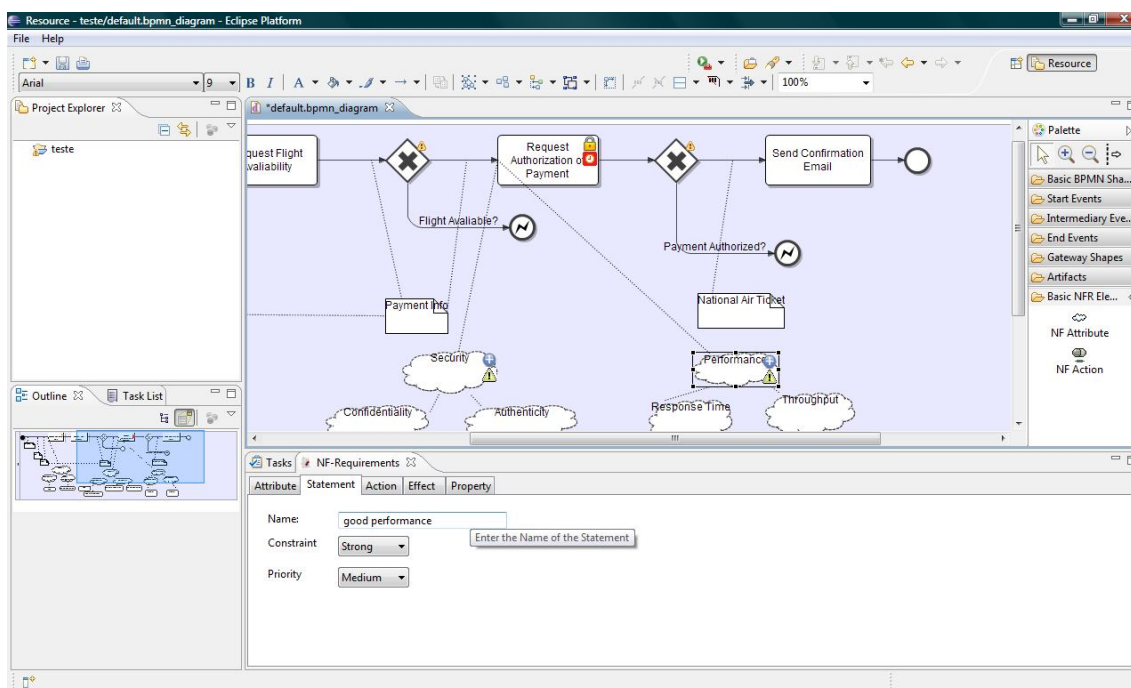


Figura 12 –Performance com anotações de NF-Statement

Por fim, a figura 13 apresenta o diagrama após feitas as configurações aqui descritas.

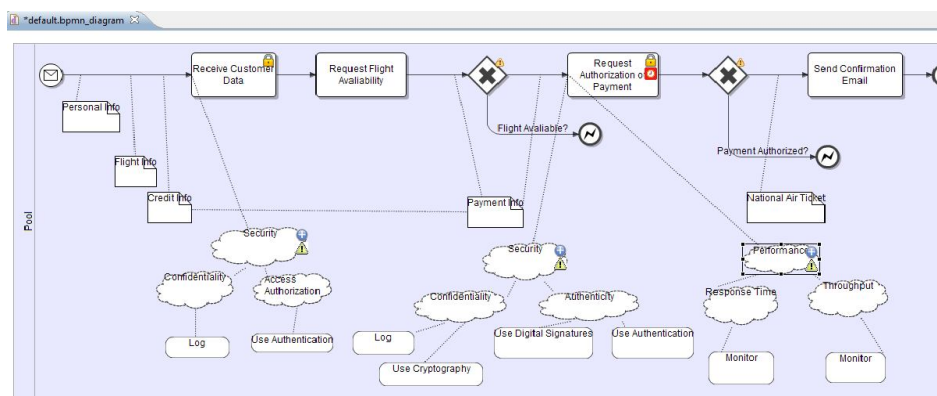


Figura 13 - Compra de passagem aérea com RNF de Segurança e Performance

Conforme dito anteriormente, além de gerar a visualização gráfica dos elementos, a ferramenta gera também o arquivo XML dos elementos correspondentes. Mostramos abaixo o arquivo XML para o NF-Attribute Performance.

```
<artifacts xmi:type="bpmn:NFAAttribute" xmi:id="rxMPvQbuEeCTUojaB34aMv" id="rxMPvAbuEeCTUojaB34aMv" name="Performance">
  <eAnnotations xmi:type="ecore:EAnnotation" xmi:id="LA-skAbvEeCTUojaB34aMv" source="composition">
    <details xmi:type="ecore:EStringToStringMapEntry" xmi:id="LA-skQbvEeCTUojaB34aMv" key="type" value="All"/>
  </eAnnotations>
  <eAnnotations xmi:type="ecore:EAnnotation" xmi:id="-8GwsAb0EeCTUojaB34aMv" source="statement">
    <details xmi:type="ecore:EStringToStringMapEntry" xmi:id="-8GwsQb0EeCTUojaB34aMv" key="name" value="good performance"/>
    <details xmi:type="ecore:EStringToStringMapEntry" xmi:id="-8Gwsqb0EeCTUojaB34aMv" key="priority" value="Medium"/>
  </eAnnotations>
  <associations xmi:type="bpmn:Association" xmi:id="9-Xx4AbuEeCTUojaB34aMv"/>
</artifacts>
<artifacts xmi:type="bpmn:NFAAttribute" xmi:id="tOabsQbuEeCTUojaB34aMv" id="tOabsAbuEeCTUojaB34aMv" name="Response Time"/>
<artifacts xmi:type="bpmn:NFAAttribute" xmi:id="vikPoQbuEeCTUojaB34aMv" id="vikPoAbuEeCTUojaB34aMv" name="Throughput"/>
<artifacts xmi:type="bpmn:NFAAction" xmi:id="CEfSsQbvEeCTUojaB34aMv" id="CEfSsAbvEeCTUojaB34aMv" name="Monitor"/>
<artifacts xmi:type="bpmn:NFAAction" xmi:id="EdS9IQbvEeCTUojaB34aMv" id="EdS9IAbvEeCTUojaB34aMv" name="Monitor"/>
```

Figura 14 – Arquivo XML mostrando anotações referentes ao NF-Attribute de Performance

5. Conclusão e Trabalhos Futuros

Neste presente trabalho foi implementada uma extensão do Editor BPMN: o NF-Module. Tal extensão possibilitou a incorporação de requisitos não funcionais ao processo de modelagem de negócio, fazendo com que tais requisitos sejam considerados no momento anterior a implementação, reduzindo assim custos relativos à correção. Com as anotações de RNFs geradas pela ferramenta, podem ser feitas traduções para linguagens de execução de serviço tal como WS-BPEL.

Entretanto, a ferramenta representa um protótipo, sendo necessário o término das seguintes pendências:

- Permitir a anotação das abstrações NF-Effect, NF-Propety e NF-Group, por meio de sua interface aos elementos gráficos;
- Permitir a interação completa entre a interface e o editor; e
- Definir um novo elemento gráfico de associação entre NF-Action e sub-requisitos (tal elemento representa uma seta “saindo” de NF-Action e “chegando” no sub-requisito).

Referências Bibliográficas

- Xavier, Lais. (2009), 'Integração de Requisitos Não-Funcionais a Processos de Negócio. Integrando BPMN e RNF', PhD master, Centre of Informatics, Federal University of Pernambuco.
- [1] J. Recker, M. Indulska, M. Rosemann, and P. Green. Do process modelling techniques get better? A comparative ontological analysis of BPMN. In D. Bunker B. Campbell, J. Underwood, editor, Proceedings of the 16th Australasian Conference on Information Systems. Australasian Chapter of the Association for Information Systems, Sydney, Australia, 2005.
- [2] Christopher J. Pavlovski and Joe Zou. Non-functional requirements in business process modeling. In APCCM '08: Proceedings of the fifth on Asia-Pacific conference on conceptual modelling, pp. 103–112, 2008.
- [3] Franch, Xavier. The Convenience for a Notation to Express Non-Functional Characteristics of Software Components. In Foundations of Component-based Systems Workshop (FoCBS), pp. 101-109, 1997.
- [4] Cysneiros, L. M. (2001). Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual. Ph.D. thesis, PUC/RJ.
- [5] OASIS. Business Process Execution Language (WSBPEL). Version 2.0. May 2007. Disponível em: <http://docs.oasisopen.org/wsbpel/2.0>. Acesso em: Jun, 2010.
- [6] Souza, A., Silva, B., Lins, F., Damasceno, J., Rosa, N., Maciel, P., Medeiros, R., Stephenson, B., Motahari-Nezhad, H. R., Li, J., & Northfleet, C.
- [7] "Incorporating Security Requirements into Service Composition: From Modelling to Execution". In ICSOC/ServiceWave 2009: The 7th International Joint Conference on Service Oriented Computing, pp. 24-27, November 2009.
- [8] Michael Menzel, Ivonne Thomas, Christoph Meinel, "Security Requirements Specification in Service-Oriented Business Process Management," ares, pp.41-48, 2009 International Conference on Availability, Reliability and Security, June 2009.
- [9] Lima, Rita de Cássia, Rosa, N., Technical Report. "Incorporating Non-Functional Requirements into Business Processes". 2010.
- [10] Dubray, J.-J. (2004). Business process modeling notation. ebPML.org.
- [11] Recker, J., Indulska, M., Rosemann, M., and Green, P. (2006). How good is bpmn really? insights from theory and practice. In Proceedings 14th European Conference on Information Systems, pages 1582–1593, Goeteborg, Sweden.
- [12] Pavlovski, C. J. and Zou, J. (2008). Non-functional requirements in business process modeling. In APCCM '08: Proceedings of the fifth on Asia-Pacific conference on Conceptual modelling, pages 103–112, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- [13] Eclipse Foundation (2008), The BPMN Modeler. Disponível em: www.eclipse.org/bpmn. Acesso em Mai, 2010.
- [14] OMG. Business Process Modeling Notation (BPMN) Version 1.2. OMG Final Adopted Specification. OMG, March 2009. Disponível em: <http://www.bpmn.org/>. Acesso em: Abril, 2010.
- [15] White, S. A. (2004), 'Introduction to BPMN', Technical report, IBM Corporation.
- [16] The Oryx Project Disponível em <http://www.bpmn-editor.org/> Acesso em: Ago, 2010.
- [17] BizAgi Process Modeler Disponível em <http://www.bizagi.com/> Acesso em: Ago, 2010.
- [18] ActiveVOS Dinponível em <http://www.activevos.com/products-activevos.php> Acesso em: Ago, 2010.
- [19] Visual Architect Disponível em <http://www.visual-paradigm.com/product/bpva/> Acesso em: Ago, 2010

- [20] IBM: WBI Modeler™Disponível em< <http://www.ibm.com/us/en/sandbox/ver2/>> Acesso em:Ago, 2010.
- [21] Graphical Editing Framework Disponível em< <http://www.eclipse.org/gef/>> Acessado em:Jun, 2010.
- [22] Eclipse Modeling Framework Project (EMF) Disponível em < <http://www.eclipse.org/modeling/emf/>>Acessado em: Jun, 2010.
- [23] Rosa, N. S. (2001), 'NFi: An Architecture-based Approach for Treating Non-Functional Properties of Dynamic Distributed Systems', PhD thesis, Centre of Informatics, Federal University of Pernambuco.
- [24] Cysneiros, L. M. and do Prado Leite, J. C. S. (1998). Utilizando requisitos não funcionais para análise de modelos orientados a dados. Workshop de Engenharia de Requisitos-XII SBES.
- [25] Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (2000). Non-Functional Requirements in Software Engineering. Springer.
- [26] Rosa, N. S.; Cunha, P.; Justo, G.(2004) An approach for reasoning and refining non-functional Requirements. Journal of the Brazilian Computer Society version ISSN 0104-6500
- [27] Rodriguez, A.; Fernández-Medina, E. & Piattini, M. (2007). A BPMN Extension for the Modeling of Security Requirements in Business Processes. In IEICE - Trans. Inf. Syst. E90-D(4), 745--752.
- [28] Rosa, N Virtual Travel Agency – Case Study. (2009)