



Universidade de Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Micro-ondas e Óptica



DOCUMENTAÇÃO E TESTE APLICADOS A UM SOFTWARE DE SIMULAÇÃO ELETROMAGNÉTICA

Autor (a): Daniela dos Santos Santana

Orientador: Prof. Dr. Hugo Enrique Hernández Figueroa

Co-orientador: Dr. Carlos Henrique da Silva Santos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: **Telecomunicações e Telemática.**

Banca Examinadora

Dr. Carlos Henrique da Silva Santos (presidente) — DMO/FEEC/UNICAMP

Prof. Dr. Antônio Manoel Ferreira Frasson — UFES

Prof. Dr. Leonardo Lorenzo Bravo Roger — FT/UNICAMP

Campinas – SP
30 de Janeiro de 2012

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Sa59d Santana, Daniela dos Santos
Documentação e teste aplicados a um software de
simulação eletromagnética / Daniela dos Santos Santana.
--Campinas, SP: [s.n.], 2012.

Orientador: Hugo Enrique Hernández Figueroa.
Co-orientador: Carlos Henrique da Silva Santos.
Dissertação de Mestrado - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Campos eletromagnéticos - Modelos matemáticos.
2. Sistema homem máquina. 3. Heurística. 4. Semiótica
computacional. I. Figueroa, Hugo Enrique Hernández.
II. Santos, Carlos Henrique da Silva. III. Universidade
Estadual de Campinas. Faculdade de Engenharia Elétrica
e de Computação. IV. Título.

Título em Inglês: Documentation and testing applied a software of
electromagnetic simulation

Palavras-chave em Inglês: Electromagnetic fields - Mathematical models, Man
machine system, Heuristic , Computational semiotic

Área de concentração: Telemática e Telecomunicações

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Antônio Manoel Ferreira Frasson, Leonardo Lorenzo Bravo
Roger

Data da defesa: 30-01-2012

Programa de Pós Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Daniela dos Santos Santana

Data da Defesa: 30 de janeiro de 2012

Título da Tese: "Documentação e teste aplicados a um software de simulação eletromagnética"

Dr. Carlos Henrique da Silva Santos (Presidente):

Prof. Dr. Antônio Manoel Ferreira Frasson:

Prof. Dr. Leonardo Lorenzo Bravo Roger:



Dedico este trabalho,
Ao meu pai José Dias
de Santana (in memoriam),
pelo amor e incentivo a
mim sempre dedicado.

Agradecimentos

Agradeço primeiramente a Deus por todas as oportunidades na minha vida e por estar sempre ao meu lado.

Agradeço a toda minha família, em especial meu querido e amado pai (José Dias de Santana), que veio a falecer no decorrer deste trabalho, a minha mãe (Edileuza Severina dos Santos) e aos meus irmãos (Wanderley dos Santos Santana e Tamirys dos Santos Santana) por todo carinho, amor, compreensão e orações.

Agradeço aos meus tios Serafim e tia Terezinha por toda força e carinho.

Agradeço ao meu orientador Prof. Hugo Enrique Hernández Figueroa pelo apoio, orientação, compreensão, criticidade, paciência, objetividade e pelas conversas objetivas e sinceras que sem dúvidas acrescentaram muito na minha vida, tanto acadêmica quanto profissional.

Agradeço ao meu co-orientador Carlos Henrique da Silva Santos pela orientação, objetividade, amizade, paciência, compreensão e oportunidades oferecidas.

Agradeço aos meus colegas do laboratório DMO, Zady Salazar, Iury Zotelle, Ruth Rubio, Roger Garay, Leonardo Ambrosio e Adriano Ferreira, pela ajuda no decorrer do trabalho.

Agradeço as melhores amigas que poderia ter no mundo que considero como irmãs Patrícia Leal Souza e Manoela Thiago dos Santos, as quais sempre estão presentes na minha vida nos momentos de alegrias e nos momentos de tristeza e dor, e espero ter sempre a amizade delas por toda minha vida.

Agradeço também aos amigos Willian Lira, Jamile Cristina, Márcio, Renata Marques, Adriano dos Santos, Jenifer, Gerson, Monica, Thalita, Robson, Emi, Antonio Carlos Zambon, Regina Lucia de Oliveira Moraes, Roque e Cleyton pelo apoio e amizade.

Agradeço a UNICAMP em especial a FEEC (Faculdade de Engenharia Elétrica e Computação) pela utilização do laboratório e toda estrutura fornecida.

Por fim, agradeço à Coordenação de Aperfeiçoamento de Pessoa de Nível Superior (CAPES) pelo apoio financeiro.

Resumo

Este trabalho apresenta a integração de algumas técnicas computacionais para validar e melhorar recursos de interface para o provimento de simulador eletromagnético chamado de SEM-BR (Simulador Eletromagnético Brasileiro). Para isso, foi integrada a avaliação heurística com as técnicas de usabilidade no SEM-BR, com foco na detecção de erros e avaliação da interface do software. Foram utilizadas as dez heurísticas de Nielsen, juntamente com os graus de severidade, semiótica computacional, percurso cognitivo e os testes com *checklist* e ferramenta UPPAAL.

O estabelecimento desse procedimento sequenciado de aplicação das referidas técnicas computacionais apresentou bons resultados, pois foi possível detectar e corrigir um grande número de erros de funcionalidades da interface, com variados níveis de severidade do problema, e também foram alterados alguns recursos visuais da interface para melhorar sua qualidade de visualização e padronização. Contribuindo com a melhoria das interações homem-máquina.

Além disso, também foi estabelecido um procedimento documental para a continuidade desse simulador eletromagnético. Isso é importante devido a complexidade apresentada por esse aplicativo e quantidade de recursos disponíveis, pois possibilita que futuras intervenções sejam melhor planejadas e mais rapidamente executadas. Outro nível de documentação também foi elaborado, focando-se nos usuários que agora possuem um manual devidamente escrito e com exemplos de uso.

Palavras-chave: *Simulador Eletromagnético, Interface Humano-Computador, avaliação heurística, semiótica computacional e percurso cognitivo.*

Abstract

This work presents the integration of some computational techniques to validate and to improve resources of interface for the provisions of electromagnetic simulator SEM-BR call (Brazilian Electromagnetic Simulator). For this, it was integrated the heuristical evaluation with the techniques of usability in the SEM-BR, with focus in the detection of errors and evaluation of the interface of software. The ten heuristical of Nielsen had been used, together with the degrees of severity, computational semiotics, cognitive route and the tests with checklist and tool UPPAAL.

The establishment of this sequenced procedure on the application of the related computational techniques presented good results, therefore it was possible to detect and to correct a great number of errors in the functionalities of the interface, with various levels of problems severities, some visual resources of the interface had been also modified to improve its quality of visualization and standardization. Contributing with the improvement of the man-machine interactions.

Moreover, also a documentary procedure for the continuity of this electromagnetic simulator was established. This was important because the complexity presented by this application and amount of available resources, therefore it makes possible that future interventions better are planned and more quickly executed. Another level of documentation also was elaborated, focusing in the users who now possess a manual duly written and with examples of use.

Keywords: *Electromagnetic Simulator, Human-Machine Interface, heuristic evaluation, computational semiotic and cognitive route.*

Sumário

Agradecimentos.....	v
Resumo.....	vii
Abstract.....	viii
Lista de Figuras.....	xi
Lista de Tabelas.....	xiii
Lista de Abreviações.....	xiv
Capítulo 1.....	16
Introdução.....	16
Motivação.....	16
Objetivos.....	17
Estrutura do Trabalho.....	18
Capítulo 2.....	19
Simulador Eletromagnético - SEM-BR.....	19
Capítulo 3.....	27
SysML e a modelagem de sistemas.....	27
3.1. Motivação do capítulo.....	27
3.2 Conceitos de SysML.....	29
3.2.1 Diagramas SysML.....	30
3.3 Diferenças entre SysML e a UML.....	32
3.4. Uso da SysML no Simulador Eletromagnético SEM-BR.....	33
3.4.1 Diagrama de Blocos.....	34
3.4.2 Diagrama de Pacotes.....	35
3.4.3 Diagrama de requisitos.....	36
3.4.4 Diagrama de Estados.....	38
3.4.5 Diagrama de Atividades.....	39
3.4.6 Diagrama de caso de uso.....	40
3.4.7 Diagrama de Seqüência.....	41
3.5 Ferramentas utilizadas.....	42
3.5.1 Topcased Modeling.....	42

3.5.2 Artisan Studio	43
3.6. Conclusão do capítulo	44
Avaliação heurística, semiótica computacional e percurso cognitivo utilizados na validação da interface do simulador eletromagnético SEM-BR.....	46
4.1 Avaliação Heurística	48
4.2 Semiótica computacional	51
4.3 Percurso cognitivo	53
4.4. Realização das avaliações na interface	54
4.4.1 Resultados da Avaliação heurística	55
4.4.2 Resultados da Semiótica Computacional	61
4.4.3 Resultados do Percurso Cognitivo	63
4.5.1 Análise dos resultados obtidos na avaliação heurística e semiótica computacional realizadas no simulador eletromagnético SEM-BR	68
4.6. Comparação das técnicas de avaliação de interface.....	69
4.7 Conclusões do capítulo.....	71
Capítulo 5.....	72
Testes utilizando o UPPAAL.....	72
5.1 Validação e verificação	72
5.2 Checklist de teste do Simulador Eletromagnético	74
5.3. Ferramenta UPPAAL.....	77
5.4 Verificação e validação do Simulador Eletromagnético SEM-BR.....	77
5.5 Conclusões do capítulo.....	80
Capítulo 6.....	81
Conclusões.....	81
Referências Bibliográficas	83
Anexo	87
Documentação e Avaliação de interface	87

Lista de Figuras

FIGURA 1 - FIGURA GEOMÉTRICA INSERIDA NO SIMULADOR SEM-BR	20
FIGURA 2 - IMAGEM DAS ETAPAS DE PROCESSAMENTO DO SIMULADOR SEM-BR	21
FIGURA 3 - IMAGEM DAS ETAPAS DE PROCESSAMENTO DO SIMULADOR SEM-BR	21
FIGURA 4 - IMAGEM GERADA PELO SIMULADOR ELETROMAGNÉTICO SEM-BR	22
FIGURA 5 - ESQUEMÁTICO DOS RECURSOS DO SIMULADOR SEM-BR E SUAS TECNOLOGIAS (FREITAS, 2010).	24
FIGURA 6 - ARQUITETURA DO SIMULADOR SEM-BR	24
FIGURA 7 - DIAGRAMA SYSML MODIFICADO (OMG, 2011).	31
FIGURA 8 - COMPARAÇÃO DE SYSML COM UML 2.0 (BALMELLI, 2006)	33
FIGURA 9 - DIAGRAMA DE BLOCOS	35
FIGURA 10 - DIAGRAMA DE PACOTES	36
FIGURA 11 - DIAGRAMA DE REQUISITOS	37
FIGURA 12 - DIAGRAMA DE ESTADOS	38
FIGURA 13 - DIAGRAMA DE ATIVIDADES	39
FIGURA 14 - DIAGRAMA DE CASO DE USO	41
FIGURA 15 - DIAGRAMA DE SEQUÊNCIA	42
FIGURA 16 - TOPCASED - DIAGRAMADOR SYSML	43
FIGURA 17 - ARTISAN STUDIO - DIAGRAMADOR SYSML	44
FIGURA 18: REPRESENTAÇÃO TRIÁDICA DE PIERCE (PEIRCE, 2005)	51
FIGURA 19: PRIMEIRA AVALIAÇÃO HEURÍSTICA	57
FIGURA 20: VISÃO DO SOFTWARE ANTES DAS MODIFICAÇÕES (A) INTERFACE CADASTRAR NOVO MATERIAL E A (B) INTERFACE INSERIR FONTE ELETROMAGNÉTICA SIMULAÇÃO.	58
FIGURA 21: SOFTWARE APÓS AS MODIFICAÇÕES (A) INTERFACE CADASTRAR NOVO MATERIAL. (B) INTERFACE INSERIR FONTE ELETROMAGNÉTICA SIMULAÇÃO	59

FIGURA 22: SEGUNDA AVALIAÇÃO HEURÍSTICA	59
FIGURA 23: TELAS DE AJUDA DO SISTEMA	60
FIGURA 24 - INTERFACE DO PAINEL GERENCIADOR	61
FIGURA 25 - INTERFACE DO PAINEL GERENCIADOR MODIFICADA	61
FIGURA 26: APLICAÇÃO DA TIPOLOGIA POR ANDERSEN NA TELA PRINCIPAL DO SOFTWARE SEM-BR	62
FIGURA 27 - TELA DE INÍCIO ANTES DAS MODIFICAÇÕES	62
FIGURA 28 - APLICAÇÃO DA TIPOLOGIA POR ANDERSEN NA TELA CADASTRAR NOVO MATERIAL	63
FIGURA 29 - MÉDIA DAS AVALIAÇÕES HEURÍSTICAS	68
FIGURA 30 - TESTE UTILIZANDO O UPPAAL	79
FIGURA 31 - RESULTADOS DOS TESTES COM O UPPAAL	80

Lista de Tabelas

TABELA 1 - EQUAÇÕES DE MAXWELL'S EM FORMA INTEGRAL E DIFERENCIAL (LIMA, 2012)	25
TABELA 2 - REQUISITOS DO SIMULADOR ELETROMAGNÉTICO	36
TABELA 3 - GRAU DE SEVERIDADE DOS PROBLEMAS DE USABILIDADE	50
TABELA 4 - RELATÓRIO DOS PARTICIPANTES DA PRIMEIRA AVALIAÇÃO	55
TABELA 5 - RELATÓRIO DOS PARTICIPANTES DA SEGUNDA AVALIAÇÃO	56
TABELA 6 - ANÁLISE PERCURSO COGNITIVO DA TAREFA 1	64
TABELA 7 - ANÁLISE PERCURSO COGNITIVO DA TAREFA 2	65
TABELA 8 - ANÁLISE PERCURSO COGNITIVO DA TAREFA 3.	66
TABELA 9: ANÁLISE PERCURSO COGNITIVO DA TAREFA 4	67
TABELA 10 - COMPARAÇÃO DAS AVALIAÇÕES DE INTERFACE	70
TABELA 11 - CHECKLIST DE TESTE DO SIMULADOR ELETROMAGNÉTICO.	74

Lista de Abreviações

ABNT – Associação Brasileira de Normas Técnicas
CTL – Computational Tree Logical
DMO – Departamento de Microondas e Óptica
DXF - AutoDesk Drawing Interchange Format
FDTD – Finite Difference Time Domain
FEEC – Faculdade Engenharia Elétrica e Computação
FEM – Método dos Elementos Finitos
HTML – Hyper Text Markup Language
IHC – Interface Homem Computador
IME – Instituto de Matemática e Estatística
IMOC - International Microwave and Optoelectronics Conference
INCOSE – Uniform Interface to Computing Resources
ISO – International Organization for Standardization
IEEE – Instituto de Engenheiros Eletricistas e Eletrônicos
NIED – Núcleo de Informática Aplicada à Educação
OMG – Object Management Group ou Grupo de Gerenciamento de objetos
SAR – Specific Absortion Rate
SysML – Systems Modeling Language
STL - Standard Tessellation Language
UML - Unified Modelling Language
USP- Universidade de São Paulo
V & V – Verificação e validação
VTK – The Visualization Toolkit
XML - *eXtensible Markup Language*

Capítulo 1

Introdução

MOTIVAÇÃO

Atualmente, tem crescido a preocupação com a documentação, avaliação da interface e testes nos sistemas, tanto no ambiente acadêmico quanto nas empresas. O que antes era visto como algo que demandava muito tempo e “desnecessário” aos olhos dos desenvolvedores, nos dias atuais é visto como imprescindível, tendo em vista que muitas empresas desenvolvem softwares, sendo um diferencial oferecer um produto de qualidade (Bruneli, 2006).

Tendo como base o cenário descrito acima, devemos levar em consideração que quanto mais complexo for o software, mais necessário ele deverá ter uma interface inteligível e de simples uso, facilitando a interação entre os usuários e o computador. O usuário deve conseguir navegar no software sem dificuldades, sem necessitar consultar o manual diversas vezes e sem ter muitas dúvidas de como avançar em cada uma das tarefas, a fim de alcançar o resultado esperado, senão o usuário fica perdido, sem saber o que deve fazer no término de cada tarefa.

Para conseguir navegar sem dificuldades em todas as telas do software, alguns elementos na interface desempenham papéis importantes como os ícones e as expressões que devem ser construídos de forma clara e objetiva para ajudar os usuários.

Em termos de qualidade de software também é necessário que haja uma boa documentação, não só para ajudar os usuários em possíveis dúvidas que venham a ter, como também ajudar os desenvolvedores na manutenção do software. Um projeto sendo bem

documentado otimiza tempo dos desenvolvedores.

Finalmente, em conjunto com a documentação e as melhorias na interface são realizados os testes para fornecer maior qualidade ao software. Os testes são realizados com intuito de comprovar que o simulador tem o menor número de erros possíveis, fornecendo, portanto, confiabilidade ao software e aos seus usuários.

O software a ser analisado neste trabalho é o simulador eletromagnético SEM-BR (Simulador Eletromagnético Brasileiro), desenvolvido no Departamento de Microonda e Óptica (DMO) na Faculdade de Engenharia Elétrica e Computação (FEEC) – UNICAMP, o qual tem como objetivo estudar o comportamento dos campos eletromagnéticos.

Portanto, este trabalho é motivado em buscar formas de avaliar a interface, com intuito de proporcionar aos usuários uma interface de fácil uso e interativa, uma documentação que auxilie tanto os usuários quanto os desenvolvedores e finalmente os testes para encontrar possíveis defeitos agregando, portanto, qualidade ao software, tendo como preocupação otimizar tempo e os gastos. Sendo aplicado em um software com elevada complexidade e quantidade de recursos e que está principalmente focado em análises de dispositivos de telecomunicações.

OBJETIVOS

O objetivo deste trabalho é de apresentar, através das técnicas de documentação, avaliação de interface e testes, que é possível detectar erros e melhorar a qualidade de um simulador ou produto final sem demandar grandes esforços em questões de tempo e gastos. Visando contribuir para a melhoria final do simulador eletromagnético SEM-BR, para que futuramente ele possa ser disponibilizado para uso.

ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte maneira:

No capítulo 2 são apresentados conceitos sobre o simulador eletromagnético SEM-BR e seu funcionamento.

No capítulo 3 são apresentados conceitos do SysML e a modelagem de sistemas, foram criados os diagramas de requisitos, caso de uso, atividade, sequência, pacotes e estados utilizando as ferramentas Artisan Studio e Topcased, tendo uma breve descrição das vantagens e desvantagens do uso da SysML. Havendo também uma comparação entre o uso da SysML em relação a UML para esta aplicação específica.

O capítulo 4 é dedicado a explicar como foi realizada a avaliação heurística no simulador, de acordo com as dez heurísticas de Nielsen, a semiótica computacional e o percurso cognitivo, sendo depois comparadas as três técnicas.

No capítulo 5 são discutidos os testes no simulador utilizando a ferramenta UPPAAL para testes automatizados e o checklist para os testes manuais.

E, finalmente no capítulo 6, as conclusões a respeito do trabalho e trabalhos futuros.

Capítulo 2

Simulador Eletromagnético - SEM-BR

2.1 Motivação do capítulo

O foco deste trabalho está na avaliação do simulador eletromagnético SEM-BR, define-se simulador como sendo um objeto ou representação parcial de uma tarefa do mundo real passível de ser replicada (Filho e Scarpelini, 2007).

A maior parte desses simuladores numéricos possui elevada complexidade e muitas exigências de recursos computacionais. Na área do eletromagnetismo computacional são realizados inúmeros esforços para reduzir o tempo de processamento e o desenvolvimento de novos recursos voltados à facilitação do provimento de diferentes dispositivos (Santos, Ambrosio e Hernández-Figueroa, 2005), (Santos, 2010), (Santos, Gonçalves e Hernández-Figueroa, 2010), (Gonçalves, Santos, Bordonalli e Hernández-Figueroa, 2010).

Para isso, o simulador SEM-BR tem como finalidade ser uma ferramenta computacional facilitadora para a modelagem e simulação de diferentes dispositivos e efeitos eletromagnéticos, que na versão atual foca-se na faixa de microondas. Outra importante contribuição desse simulador é o seu uso em análises de impacto provocados pelas radiações não-ionizante nos tecidos humanos, simulando as taxas de absorção específica (*Specific Absorption Rate – SAR*), sendo que este simulador também atende as exigências do padrão IEEE 1528 para análises de SAR na cabeça humana (IEEE, 2011). Podendo também viabilizar a mediação, pois em vários casos se torna inviável por causa da complexidade.

O simulador eletromagnético aqui analisado é dividido em três partes, pré-processamento, processamento e pós-processamento.

Pré-processamento: Entrada de dados no simulador. Essa entrada de dados pode ser por meio de importação de arquivos STL, ou inserindo figuras geométricas tais como (esfera, cilindro, cone e tijolo), definição de materiais e suas propriedades eletromagnéticas, domínio computacional como suas dimensões, tipo de plano (cartesiano, esférico ou cilíndrico) e fontes eletromagnéticas, sendo mostrada na figura 1 a inserção de uma geométrica no simulador SEM-BR, a esfera em azul é pode ser visualizada na posição de cima, frontal, lateral e perspectiva não editável (Figuroa, 2007).

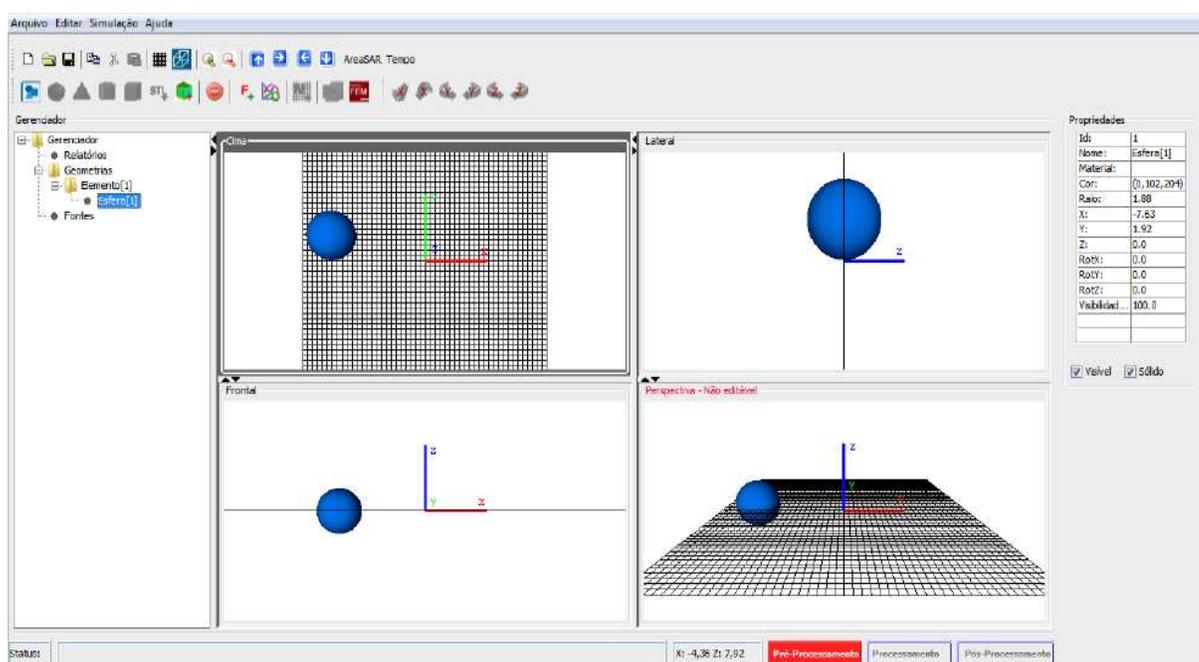


Figura 1 - Figura geométrica inserida no simulador SEM-BR

Processamento: Resultados são gerados de acordo com os cálculos relacionados com os campos eletromagnéticos. Sendo utilizado no SEM-BR dois métodos, o Método das Diferenças Finitas no Domínio do Tempo (FDTD) e o Método dos Elementos Finitos (FEM), tendo esses métodos diferentes comportamentos de acordo com a aplicação e a complexidade do problema em questão. A execução do método numérico (neste caso um FDTD 3D desenvolvido pelo grupo de pesquisa) utilizando os atributos de entrada especificados no pré-processamento.

Pós-processamento: Visualização de forma gráfica dos resultados obtidos durante o processamento. São utilizadas quatro formas de visualização de relatórios, as quais são: relatório filme 2D, filme 3D, propagação eletromagnética no tempo e SAR – *Specific Absortion Rate* – Região do Domínio Computacional. Na figura 2 são apresentados os relatórios aos usuários provenientes do processamento realizado pelo simulador SEM-BR.

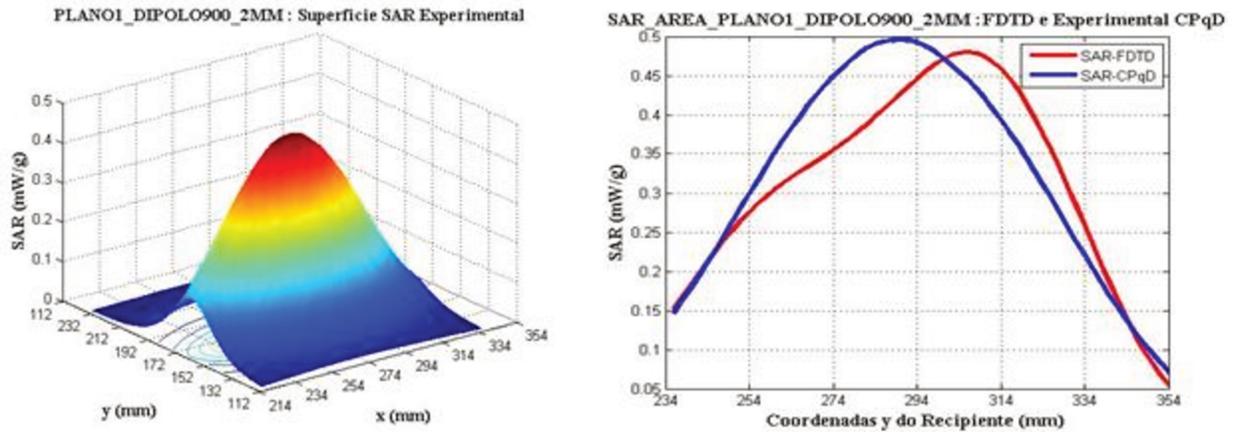


Figura 2 - Imagem das etapas de processamento do simulador SEM-BR

Na figura 3 é mostrada a representação das três etapas de funcionamento do simulador SEM-BR, tendo primeiro a etapa de pré-processamento, segundo o processamento e por último o pós-processamento.

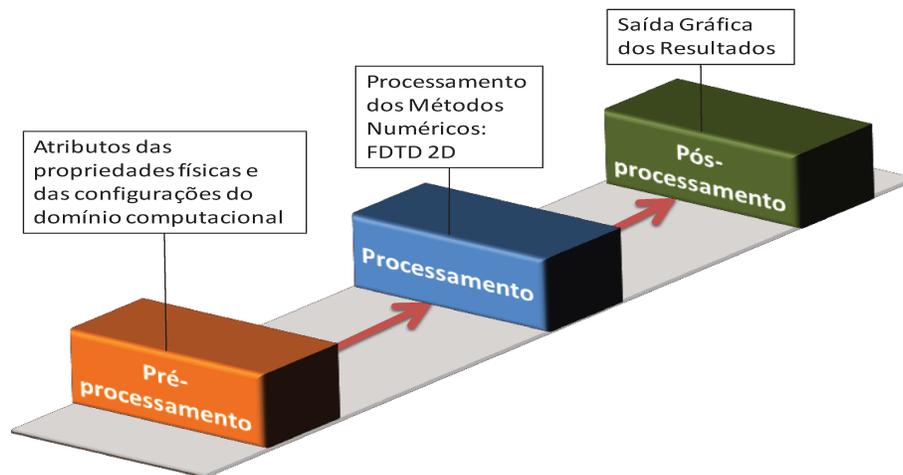


Figura 3 - Imagem das etapas de processamento do simulador SEM-BR

No simulador apresentado neste trabalho estão integrados os métodos dos elementos (FEM) bidimensionais e o método das diferenças finitas no domínio do tempo (FDTD) tridimensional, que neste simulador é encontrado com maior quantidade de recursos integrados e fáceis de ser utilizado. Por isso, a interface com os usuários foca-se no FDTD 3D.

O FDTD 3D é um método explícito, ou seja, não há a necessidade de inversões matriciais. Este método também permite discretizar o espaço (Δx , Δy , Δz) e o tempo Δt para solução das equações de Maxwell (Dias, 2003). Contudo, tem a desvantagem de não haver o aproveitamento da esparsidade em domínios computacionais, implicando em grande consumo de memória para armazenar todos os dados (matrizes).

Todos os dados realizados nestes equacionamentos são complexos e por esse motivo à interface é um importante recurso que deve ser bem dimensionado, pois abstrai os recursos numéricos através de representações gráficas, onde os usuários inserem seus atributos de entrada e analisam os resultados processados em gráficos gerados dentro e fora do simulador, portanto os usuários não necessitam saber o que tem dentro do simulador, ou seja, como funciona cada um dos processos numéricos.

A Figura 5 mostra um esquemático dos recursos já inseridos neste simulador eletromagnético com as respectivas tecnologias utilizadas, sendo importante notar que este simulador é composto por pacotes desenvolvidos em diferentes linguagens de programação tais como (Java, Fortran, OpenGL e C++), mas que se comunicam através de um gerenciador de serviços, onde temos cada componente com linguagens de programação diferentes como: gerador de malha que é desenvolvido em Java, o método FDTD 3D é desenvolvido em C++, o método FEM 2D utiliza Fortran, o gerenciador utiliza Java e a interface usa java, OpenGL e VTK.

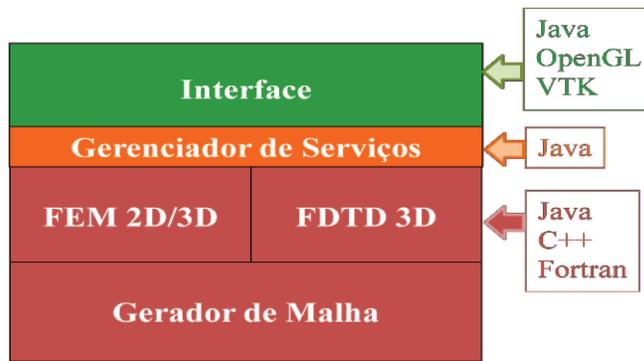


Figura 5 - Esquemático dos recursos do simulador SEM-BR e suas tecnologias (Freitas, 2010).

Na figura 6 é mostrada a arquitetura do simulador eletromagnético SEM-BR, sendo apresentados os pacotes que compõem o simulador os quais são:

- Mediador: faz a comunicação entre os módulos.
- Gerador de malha: Gerar a malha tridimensional com método numérico FDTD.
- Gerenciador e modelador básico: Responsável por checar o status do sistema e fazer solicitação de serviços pelo mediador.
- Pós-processamento: Este módulo é responsável pela visualização dos dados gerados no processamento.
- Importadores: responsável por fazer a importação dos modelos geométricos com formato de arquivo do tipo STL.
- Processamento: Realiza a execução do método numérico FDTD, de acordo com dados inseridos pelos usuários no módulo de Pré-Processamento.
- Utils: Classes que podem ser utilizadas por outros módulos.

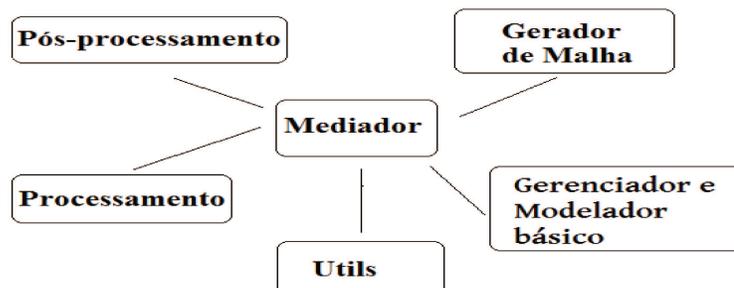


Figura 6 - Arquitetura do simulador SEM-BR

Todos os métodos numéricos empregados nesse simulador são baseados em equações de Maxwell (Santos, 2010), estando na forma integral (FEM) ou diferencial (FDTD), como mostrado na Tabela 1, sendo as equações organizadas de (1) a (4), descritas abaixo:

- 1: Lei de Faraday
- 2: Lei de Ampere
- 3: Lei de Gauss – Fluxo Elétrico
- 4: Lei de Gauss – Fluxo Magnético

De acordo com a Tabela 1, pode ser visualizada a complexidade das equações que estão por trás das interfaces.

Tabela 1 - Equações de Maxwell's em forma integral e diferencial (Lima, 2012)

Forma Integral	Forma Diferencial
$\oint_{sl} \vec{D} \cdot d\vec{S} = \int_v \rho dv \quad (C)$	$\nabla \cdot \vec{D} = \rho (C/m^3) \quad (1)$
$\oint_l \vec{E} \cdot d\vec{L} = 0$	$\nabla \times \vec{E} = 0 \quad (2)$
$\oint_s \vec{H} \cdot d\vec{L} = \int_s \vec{J} \cdot d\vec{S} \quad (A)$	$\nabla \times \vec{H} = \vec{J} (A/m^2) \quad (3)$
$\int_s \vec{B} \cdot d\vec{S} = 0$	$\nabla \cdot \vec{B} = 0 \quad (4)$

A complexidade das equações numéricas descritas anteriormente demonstra como um profissional especializado que pretende modelar e analisar suas propriedades eletromagnéticas tem que desenvolver uma solução numérica para estes problemas. Por estas razões, a interface do usuário de software deve ser desenvolvida para encapsular a maioria destes métodos numéricos. Ele fornece diretamente as interações mais eficientes entre homem-máquina no contexto de modelagem. Estas interfaces computacionais também contribuem para fornecer abordagens mais fáceis centradas na produtividade (tempo de modelo e processo de um método eletromagnético), flexibilidade para modelar um grande número de aplicações diferentes.

Levando em consideração todo o contexto descrito, o simulador SEM-BR foi escolhido para ser analisado por se tratar de um simulador com complexas equações numéricas, possuindo uma grande complexidade de cálculos, necessitando, portanto, ter uma interface que seja fácil de ser operada pelos usuários. O simulador SEM-BR também foi escolhido pelo fato de está em processo de desenvolvimento, podendo ser verificado e analisado seus processos de execução.

A integração entre os métodos numéricos, as necessidades de modelagem de usuários e a interface de computador deve ser previamente informadas e especificadas. Neste contexto, este trabalho aborda nas próximas seções conceitos de engenharia de software, interface humano-computador e testes focado na interação homem-máquina.

Capítulo 3

SYSML E A MODELAGEM DE SISTEMAS

3.1. MOTIVAÇÃO DO CAPÍTULO

A documentação de software pode ser considerada uma tarefa que para alguns desenvolvedores demanda muito tempo, contudo é fundamental para reduzir horas valiosas na correção de problemas e facilitar a comunicação entre os membros da equipe de desenvolvimento, possibilitando que possíveis manutenções no software sejam feitas de forma mais eficiente e rápida.

Além de elaborar uma boa documentação inicial, é importante também que essa documentação seja acompanhada e alterada, de acordo com as demandas oriundas de futuras manutenções no software.

A documentação de software pode ser dividida em duas partes sendo elas:

- Documentação direcionada ao desenvolvedor, composta por fluxogramas, comentários no código e diagramas.
- Documentação direcionada ao usuário final, composta por manuais e apostilas (Michelazzo, 2011).

O foco da documentação neste trabalho está no desenvolvedor, por serem construídos diagramas para auxiliar-lo tanto no desenvolvimento do simulador quanto em possíveis manutenções. Quando forem necessárias manutenções no software, tendo a documentação, este processo será mais fácil de ser executado.

Tendo pronta a documentação do software, algumas informações podem ser reutilizadas em outros projetos e também visa melhorar a comunicação entre os desenvolvedores e os outros membros da equipe do projeto.

Para que seja desenvolvida a documentação há no mercado diversas linguagens de modelagem como a UML, SysML, aSideML, cada uma com suas características, sendo estas descritas abaixo:

A UML é uma linguagem de diagramas que busca especificar, visualizar e documentar modelos de software (Gama, 2003).

A SysML é uma linguagem que auxilia na visualização dos objetos, proporcionando maior compreensão entre os desenvolvedores e outros membros da equipe como testadores, tem como base a UML, porém como diferencial possuir o diagrama de requisitos (SysML, 2011).

A aSideML é uma linguagem que proporcionar ao projetista construir modelos que tenham como principais objetivos os conceitos, mecanismos e propriedades de sistemas orientados a aspectos (Chavez, 2004).

Neste trabalho é utilizada a linguagem SysML por ser uma linguagem de modelagem baseada na UML que possui a mais o diagrama de requisitos. Este diagrama é importante por mostrar a representação gráfica dos requisitos (Soares e Vrancken, 2009). Tendo como vantagem a utilização da SysML o fato dessa possuir o diagrama de blocos, o qual permite a visualização geral do sistema com sua interação entre os elementos, podendo conter não só a relação entre os elementos de software quanto hardware, na UML há apenas o diagrama de pacotes, contendo a relação entre componentes de software.

As duas ferramentas utilizadas para fazer a modelagem do simulador SEM-BR e que possuem suporte para SysML são a Topcased e a Artisan Studio, ambas as ferramentas são *free* e serão utilizadas para criar os diagramas, apresentados a seguir (Rosa, 2006):

- Caso de uso: O diagrama de caso descreve uma sequência de ações de um ou mais atores em um sistema, tendo como funcionalidade descrever o sistema em termos de utilização/objetos pelos atores.
- Estado: O diagrama de estado é usado para modelar o comportamento interno de um objeto ou de um sistema, mostrando os possíveis estados dos objetos. Como, por exemplo, uma lâmpada, seu estado pode ser acesa ou apagada, tendo basicamente como foco representa o ciclo de vida de um bloco.
- Seqüência: O diagrama de sequência é usado para mostra uma interação dos objetos de acordo com a visão temporal, sendo utilizado para permitir a representação do comportamento baseado em mensagens.
- Atividades: usado para modelar o tempo de vida de um objeto ou sistema, exibindo o fluxo de controle de entrada e saída, sendo incluídas as sequências e condições para coordenar as atividades.
- Pacote: O diagrama de pacotes permite juntar diferentes elementos de um sistema em grupos, facilitando assim a visualização do sistema de forma global.
- Requisitos: O diagrama de requisitos tem como meta determinar as atividades ou condições que o sistema deve satisfazer, descrevendo os requisitos contidos numa especificação.
- Blocos: O diagrama de blocos fornece uma visão geral de todos os componentes do sistema, mostrando as conexões entre os elementos do sistema, podendo conter a parte de software e de hardware, sendo utilizado para captura propriedades e sendo reutilizado em vários contextos.

3.2 CONCEITOS DE SYSML

SysML (*Systems Modeling Language*) é uma linguagem de modelagem gráfica que representa um subconjunto da UML 2.0, desenvolvida pela OMG e INCOSE (Hansen, 2011).

Tendo como objetivo o uso geral para especificar, analisar, projetar e verificar sistemas complexos, podendo ser incluídos, hardware, software, informações pessoais, procedimentos e facilidades (IBM, 2011).

A SysML possui nove dos treze diagramas da UML, e um diferencial entre essas duas técnicas é o fato da SysML possuir o diagrama de requisitos e está organizada com intuito de atender as entidades a seguir:

- Requisitos: condição ou capacidade de satisfazer o sistema.
- Blocos: possuem a função de modelar os elementos do sistema, como hardware e software.
- Atividades: descrevem uma sequência de ações com entradas e saídas.
- Blocos restritos: Os blocos restritos possuem duas características: um grupo de parâmetros e uma expressão que limita os parâmetros.

3.2.1 DIAGRAMAS SysML

A SysML é uma extensão da UML, possui nove dos treze diagramas da UML, sendo acrescentados os diagramas de requisitos e o diagrama paramétrico. Na figura 7 é apresentado o diagrama de organização geral da SysML com algumas modificações, visando mostra os diagramas que serão abordados na documentação do simulador eletromagnético SEM-BR, o quadrado destacado em azul, com borda acentuada, representa o diagrama de requisitos, o qual foi acrescentado na SysML.

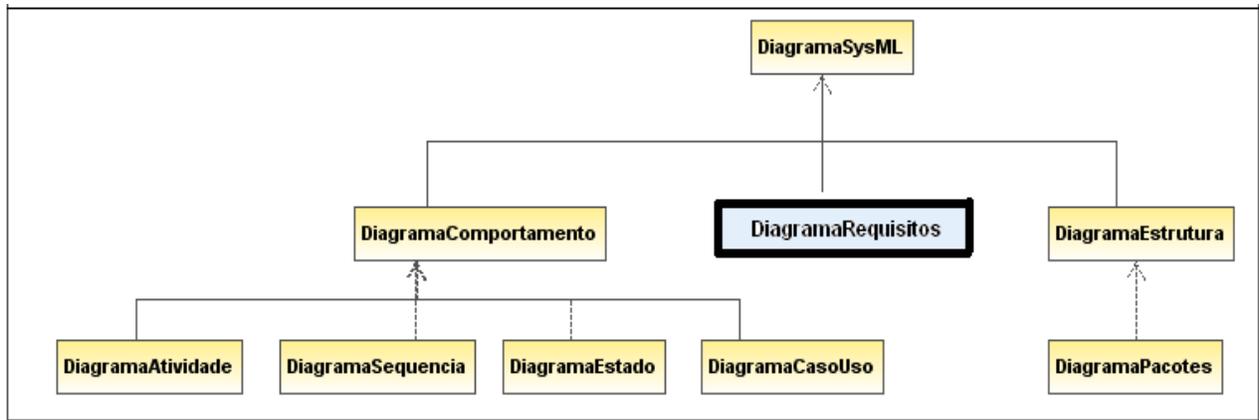


Figura 7 - Diagrama SysML modificado (OMG, 2011).

A SysML contém nove diagramas, apresentados a seguir:

- **Diagrama de requisitos:** É um diagrama novo pertencente a SysML e que não consta na UML 2.0, cuja função é descrever o desempenho dos requisitos de interface, mostrando seus relacionamentos e as condições ou capacidades necessárias para satisfazer o sistema.
- **Diagrama de Atividades:** Mostra o comportamento ou funcionalidade do sistema, como fluxos de controle entre as atividades que realizam as diferentes partes dos sistemas, com suas entradas e saídas, descrevendo, portanto, todo o fluxo que o usuário irá percorrer no software.
- **Diagrama de seqüência:** Mostra as mensagens trocadas entre os objetos, de acordo com a ordem, seguindo uma lógica temporal.
- **Diagramas de Estado:** São utilizados para modelar o comportamento dos objetos utilizando máquina de estado.
- **Diagrama de casos de uso:** Descreve a utilização do sistema por outros sistemas, dispositivos ou usuários, sendo modelada uma parte do sistema, uma funcionalidade.

- **Diagrama de definição de blocos:** é utilizado para especificar as hierarquias e interligação dentro de um projeto do sistema.
- **Diagrama de bloco interno:** é responsável por descrever as estruturas internas de um bloco, com suas propriedades e conectores.
- **Diagrama de pacote:** mostra os pacotes do sistema divididos em agrupamentos descrevendo as dependências entre eles, tendo assim uma visão geral de todos os componentes do sistema.

Todos os diagramas descritos anteriormente são classificados em três grandes grupos, os quais são descritos a seguir:

- **Diagramas estruturais:** Mostram os diagramas que fornecem uma visão geral do sistema, sendo compreendido pelos seguintes diagramas: Diagrama de pacotes, Diagrama de Parâmetros, Diagrama de Blocos e Diagrama de Blocos Internos.
- **Diagramas comportamentais:** Esses diagramas são responsáveis por mostrarem o controle de fluxo das informações sendo compreendidos pelos diagramas: Diagrama de Casos de uso, Diagrama de Seqüência, Diagrama de Atividades e Diagrama de Máquina de Estados.
- **Diagramas transversais:** Mostra o diagrama responsável por conter as condições que satisfaçam o sistema sendo compreendido pelo seguinte diagrama: Diagrama de Requisitos.

3.3 DIFERENÇAS ENTRE SYSML E A UML

A SysML se diferencia da UML pelo fato de ter menos diagramas que a UML. Enquanto a SysML tem nove diagramas a UML possui treze diagramas, não é obrigatório que todos os diagramas sejam utilizados, no entanto menos recursos, facilita na hora que o usuário precisa desenvolver.

Outra diferença entre essas duas técnicas utilizadas para documentação de processos de software é o fato da SysML possuir os diagramas de requisitos e paramétrico, os quais não há na UML (Oliveira e Silva, 2009).

A Figura 8 mostra os diagramas que fazem parte apenas da modelagem SysML os quais são: requisitos, bloco (definição e interno) e paramétrico. E os diagramas que estão presentes tanto na SysML 1.0 quanto na UML 2.0 que são: Caso de uso, máquina de estados, atividades e sequência.

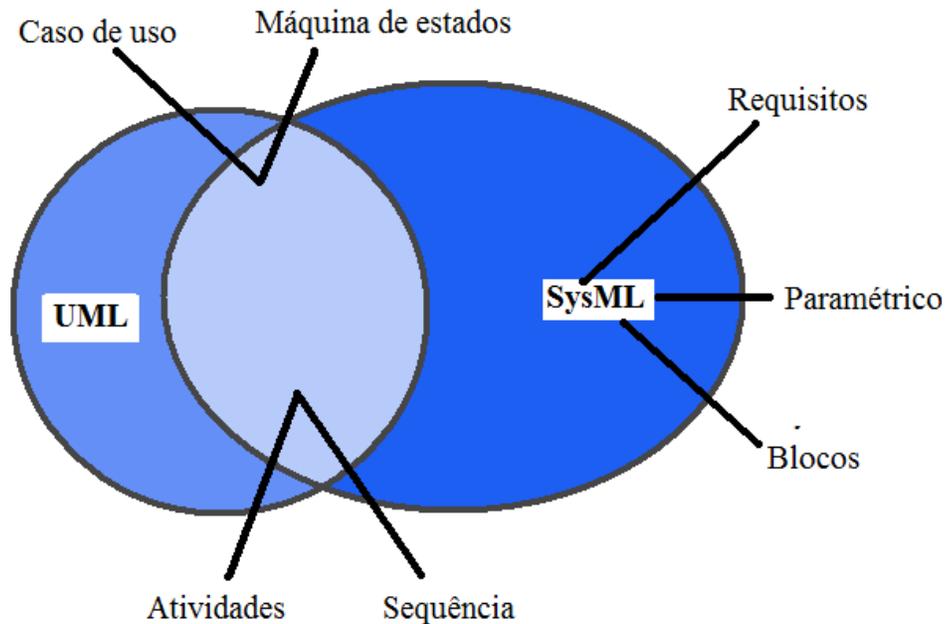


Figura 8 - Comparação de SysML com UML 2.0 (Balmelli, 2006)

3.4. USO DA SYSML NO SIMULADOR ELETROMAGNÉTICO SEM-BR

O estudo de caso aplicado no simulador eletromagnético SEM-BR será descrito nesta seção, na qual serão mostrados alguns dos diagramas SysML utilizados na modelagem.

3.4.1 Diagrama de Blocos

A Figura 9 mostra o diagrama de blocos da arquitetura do simulador eletromagnético SEM-BR. São apresentados os blocos definidos para o SEM-BR e os relacionamentos entre eles, compostos pelos seguintes blocos ou módulos (Freitas, 2010):

- Mediador: Realiza a comunicação entre todos os módulos, sendo utilizado para delegar comandos para fazer o aplicativo funcionar, de acordo com os eventos transmitidos pelo usuário.
- Gerador de malha: módulo cuja funcionalidade é de gerar malha tridimensional com o método numérico FDTD utilizando o algoritmo de cruzamento de raios, sendo feito um mapeamento dos objetos vetoriais que compõem a simulação.
- Gerenciador e Modelador básico: Responsável por checar o status do sistema e fazer solicitação de serviços para o mediador para que sejam realizadas as execuções das operações de simulação.
- Utils: São classes chamadas de utilitárias que podem ser usadas por outros módulos do simulador SEM-BR, como por exemplo, as classes Geometria e Material (que define o tipo de material eletromagnético de cada geometria).
- Processamento: Realiza a execução do método numérico FDTD, de acordo com dados inseridos pelos usuários no módulo de Pré-Processamento, para que seja analisada a propagação dos campos eletromagnéticos.
- Pós-Processamento: Este módulo é responsável pela visualização dos dados gerados no processamento, sendo visualizados em duas ou três dimensões pelos usuários, podendo ser feita análise dos dados.
- Importadores: responsável por fazer a importação dos modelos geométricos com formato de arquivo do tipo STL (*Standard Tessellation Language*) e DXF (*AutoDesk Drawing Interchange Format*).

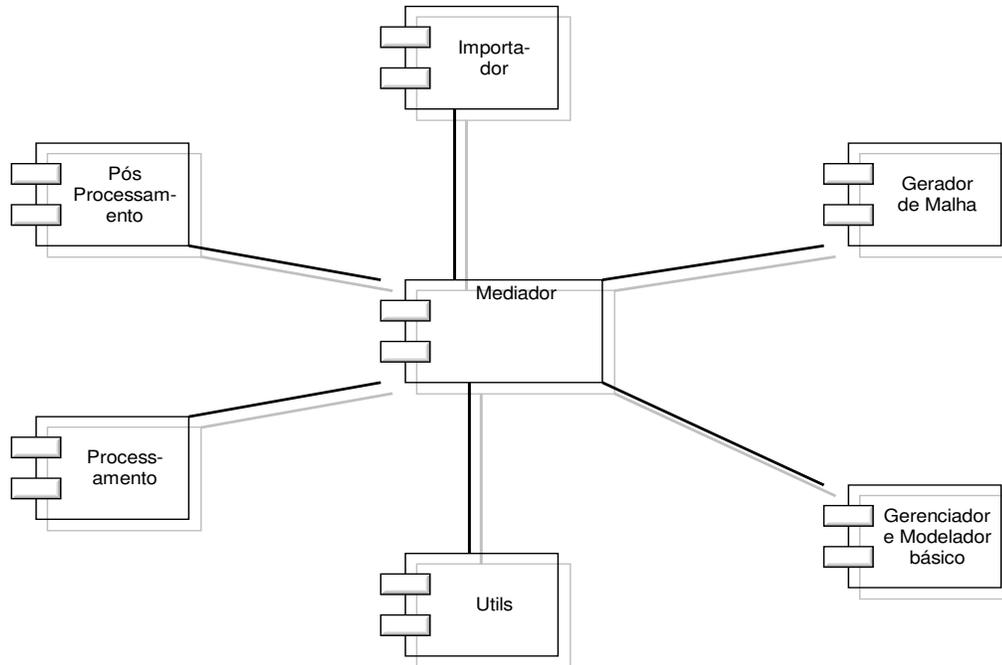


Figura 9 - Diagrama de blocos

3.4.2 DIAGRAMA DE PACOTES

O diagrama de pacotes é responsável por organizar elementos, como classes e diagramas em grupos.

Na Figura 10 é apresentado o diagrama de pacotes, o qual busca mostrar uma visão global do sistema, trazendo os três grandes pacotes que compõem o simulador SEM-BR, sendo descritos a seguir:

- O módulo ou pacote pré-processamento: responsável pelos dados de entrada que são fornecidos pelo usuário. Através da inserção de geometrias (esfera, cilindro, cone e tijolo) ou através da importação de arquivos STL.
- Módulo ou pacote de processamento de dados: processo no qual as classes são processadas e a malha é gerada, sendo logo em seguida, realizada a simulação com método FDTD.

- Módulo ou pacote pós-processamento: processo no qual os resultados do processamento da malha são apresentados na forma de relatório.

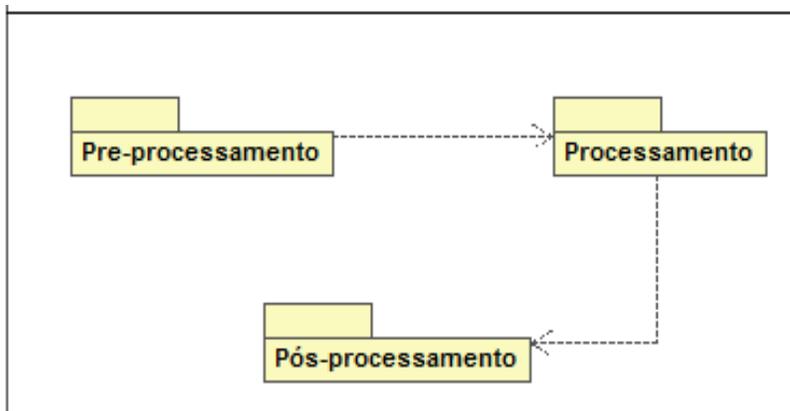


Figura 10 - Diagrama de pacotes

3.4.3 DIAGRAMA DE REQUISITOS

Segundo Macaulay (Macaulay, 1996) o requisito é “*simplesmente algo de que o cliente necessita*”. Os requisitos são responsáveis por fazer a descrição e especificação dos sistemas. Na Tabela 2, são apresentados alguns requisitos do simulador eletromagnético SEM-BR.

Tabela 2 - Requisitos do simulador eletromagnético

Item	Descrição do requisito
01	Requisito principal do simulador
02	O usuário deve fornecer fonte eletromagnética e geometria ou arquivo STL (fase pré-processamento)
03	Exibir dados fornecidos pelo usuário
04	Software deve gerar malha e realizar simulação (fase de processamento)
05	Transmitir informação
06	Software deve emitir relatórios (fase de pós-processamento)

De acordo com a Figura 11 são seis os principais requisitos do simulador eletromagnético SEM-BR apresentados abaixo: O requisito principal do simulador: O principal requisito do sistema é gerar uma malha usando o método FDTD, de acordo com as informações fornecidas pelo usuário, sendo o resultado da simulação apresentada em forma de relatório.

Entrar com dados: Os dados de entrada para gerar a simulação com método FDTD são passados pelos usuários.

Exibir dados fornecidos pelo usuário: Os dados passados pelo usuário podem ser visualizados e confirmados por uma “janela”, podendo ser modificados.

Simular com FDTD: Confirmados os dados passados pelo usuário é realizada a simulação com o método FDTD.

Transmitir informações: As informações geradas são transmitidas para um relatório.

Interpretar relatório: As informações passadas para o relatório são interpretadas pelos usuários.

O software é responsável por pegar os dados fornecidos pelo usuário, gerar a malha e realizar a simular, sendo depois essa malha visualizada na forma de relatórios.

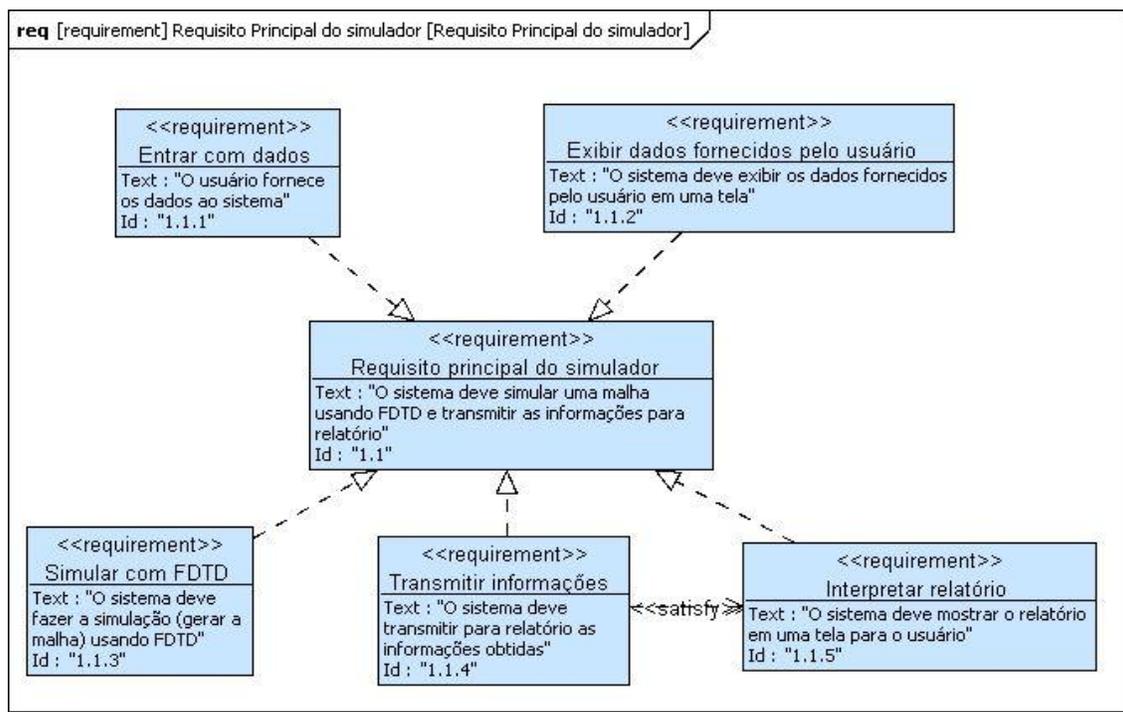


Figura 11 - Diagrama de requisitos

3.4.4 DIAGRAMA DE ESTADOS

O sistema deve estabelecer princípios que proporcionem seu funcionamento. Com base nisso apresenta-se na Figura 12 o diagrama de estados.

Diagrama de estado serve para mostra os eventos os quais o objeto passa ou percorre, sendo representados os estados e as transições entre esses estados, juntamente com as entradas e saídas para esses estados.

No simulador SEM-BR o diagrama de estados segue os seguintes eventos: o usuário fornecer os dados de entrada que são: fonte eletromagnética e geometria (esfera, cilindro, cone e tijolo) e/ou arquivos STL.

Os dados podem ser alterados, sendo configurado o domínio computacional e o material da geometria. Logo após, a entrada dos dados, inicia-se o estado de processamento com a geração da malha. Os resultados obtidos da malha gerada podem ser visualizados através de relatórios no chamado pós-processamento.

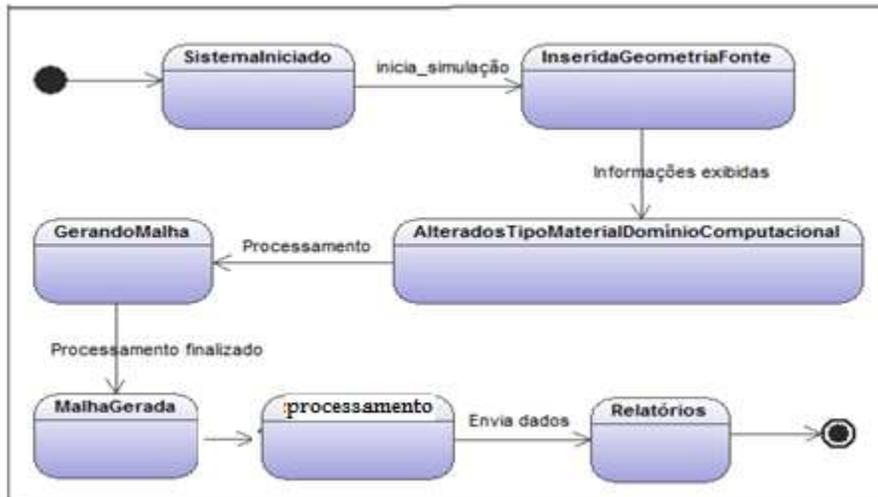


Figura 12 - Diagrama de estados

3.4.5 DIAGRAMA DE ATIVIDADES

Diagrama de atividades é responsável por mostra o fluxo de atividades em um processo ou sistema, com suas entradas e saídas, e a dependência entre as atividades.

Na Figura 13 é ilustrado o diagrama de atividade representando o fluxo dos dados e controle existente entre as atividades realizadas pelo sistema.

A primeira atividade que o usuário deve fazer é fornecer os dados ao sistema, ou importando os dados, ou inserindo uma geometria. Logo após deve ser configurado o domínio computacional, para em seguida ser inserida as propriedades do material, fonte eletromagnética e relatório, podendo então ser gerada a malha e finalmente o usuário pode visualizar os relatórios de pós-processamento.

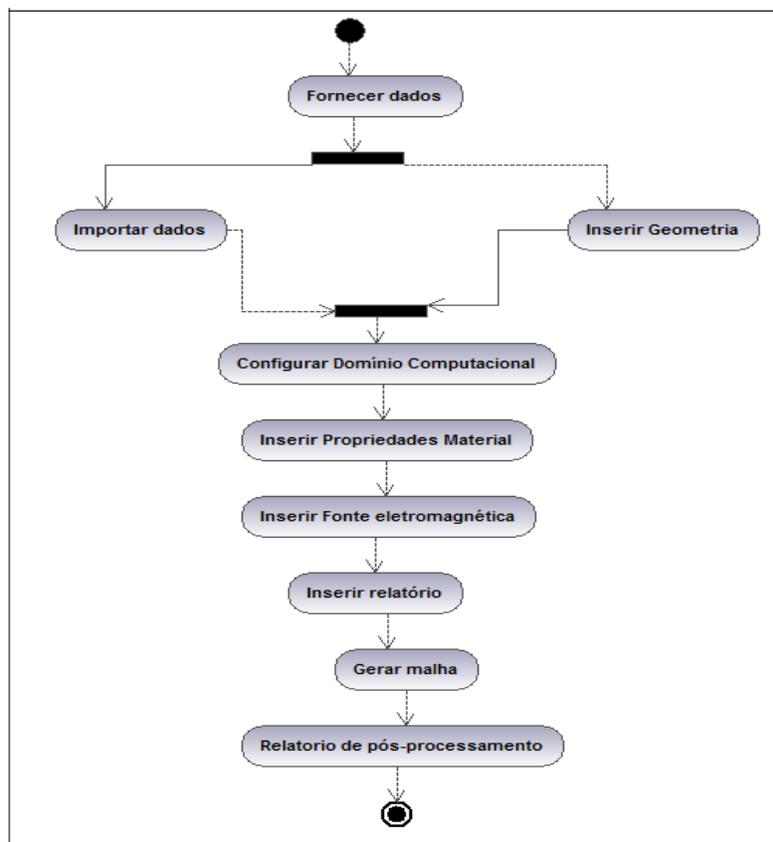


Figura 13 - Diagrama de atividades

Os passos descritos a seguir descrevem como devem ser realizadas as operações até a visualização dos resultados desde a entrada dos dados:

- Fornecer dados: O usuário deve fornecer alguns dados, como fonte eletromagnética e geometria.
- Inserir geometria: O usuário pode inserir: tijolo, cilindro, esfera ou cone.
- Importar dados: podem ser importados arquivos STL.
- Configurar Domínio computacional: O domínio computacional pode ser configurado de acordo com a geometria inserida.
- Inserir fonte eletromagnética: Pode ser inserida uma fonte eletromagnética com campo elétrico ou campo.
- Inserir relatório: podem ser inseridos relatórios:
- Gerar malha: após a inserção de todos os dados a malha pode ser gerada, sendo esse processo parte do processamento.
- Relatório de pós-processamento: Depois da malha gerada os resultados podem ser verificados nos relatórios.

3.4.6 DIAGRAMA DE CASO DE USO

A Figura 14 apresenta o diagrama de caso de uso representado pelas operações desenvolvidas pelo usuário, onde aparece o ator (usuário) e são listados seus relacionamentos que são as informações ou dados de entrada fornecidos ao sistema.

Sendo o diagrama de caso responsável que fazer a interação do usuário podendo ser humano ou máquina com o sistema (Neto, 2011).

O caso de uso da Figura 14 representa as funcionalidades que o usuário pode inserir no simulador que são: inserir fonte eletromagnética e geometria e/ou arquivo STL.

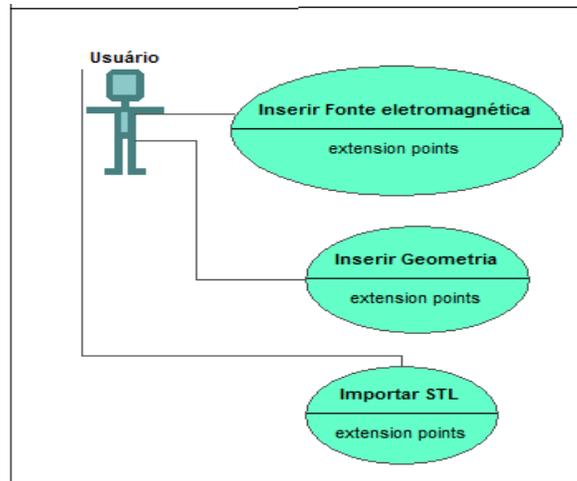


Figura 14 - Diagrama de caso de uso

3.4.7 DIAGRAMA DE SEQÜÊNCIA

O diagrama de seqüência descreve a seqüência dos fatos ocorridos. Mostrando como ocorre a troca de informações, com a preocupação da ordem que ocorrem as trocas de mensagens entre os objetos relacionados no processo (Guedes, 2004). Dando prioridade para a ordem pela qual acontecem os eventos (Correia, 2006).

Na Figura 15 nota-se que o usuário inicia a operação entrando com os dados na interface do sistema, logo depois esses dados são processados pelo S.O., podendo ser realizada simulação, usando o método FDTD ou FEM. Os resultados são obtidos em forma de relatório, sendo visualizados pelo usuário.

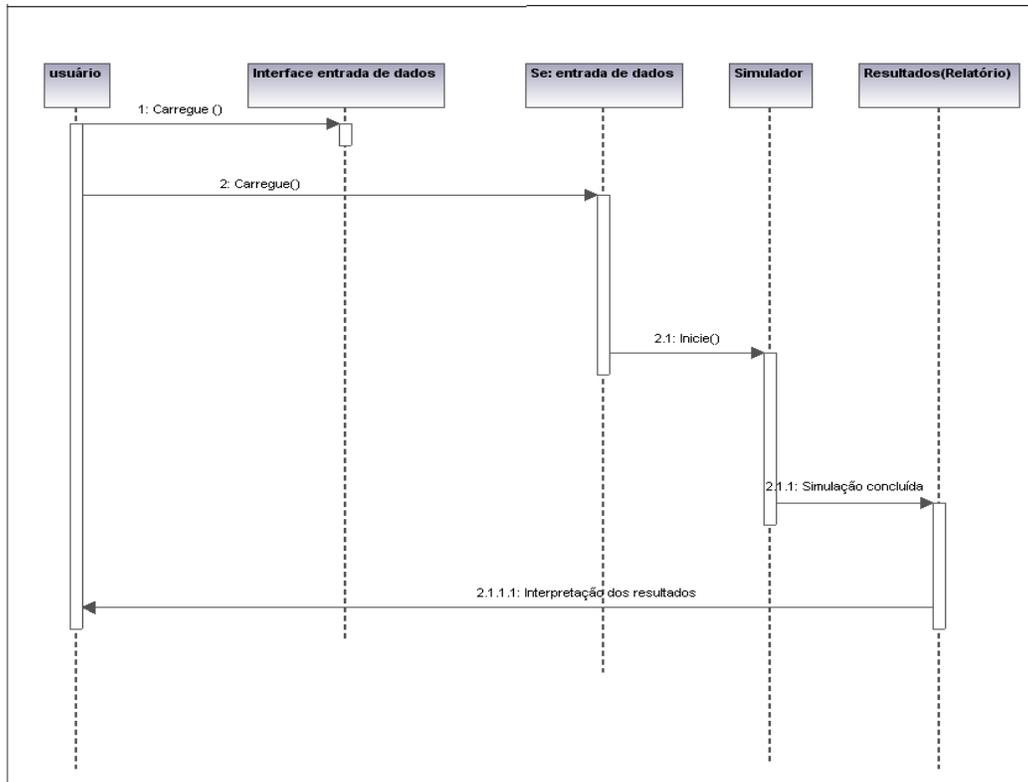


Figura 15 - Diagrama de sequência

3.5 FERRAMENTAS UTILIZADAS

A seguir estão listadas as ferramentas utilizadas para realizar a documentação do simulador eletromagnético SEM-BR, utilizando a linguagem SysML.

3.5.1 TOPCASED MODELING

A Topcased possui suporte a SysML permitindo a criação dos diagramas, possibilitando a documentação do sistema. É uma ferramenta open source, seu download pode ser feito diretamente no site da Topcased (Topcased, 2011).

Topcased pode ser usada para desenvolver documentação em SysML e UML, além de importar arquivos XML (Faudou e Marconato, 2011).

A Figura 16 apresenta a tela principal do software Topcased Modeling, na qual é mostrada a modelagem do diagrama de requisitos.

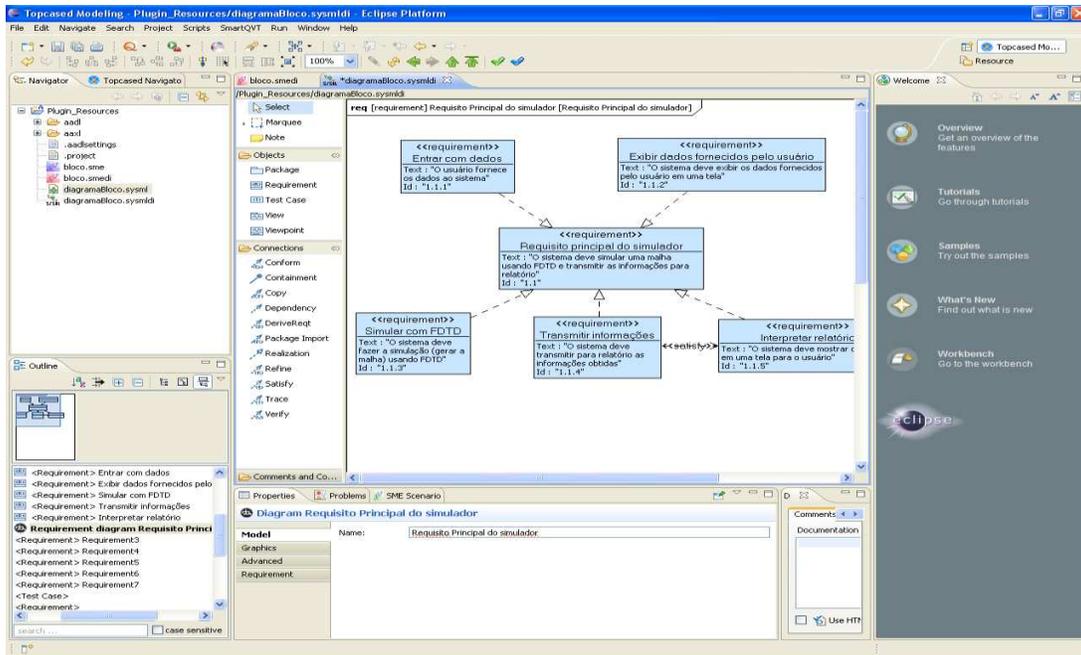


Figura 16 - Topcased - Diagramador SysML

3.5.2 ARTISAN STUDIO

A Artisan Studio é uma ferramenta que dar suporte para SysML, possibilitando criar diagramas, ajudando na documentação (Norfolk, 2009).

Artisan Studio possui suporte para OMG UPDM, SysML e UML, fazendo a verificação e validação de sistemas complexos (Artisan Studio, 2011).

Na Figura 17 é mostrada a tela do software Artisan Studio, na qual é apresentada a modelagem do diagrama de blocos.

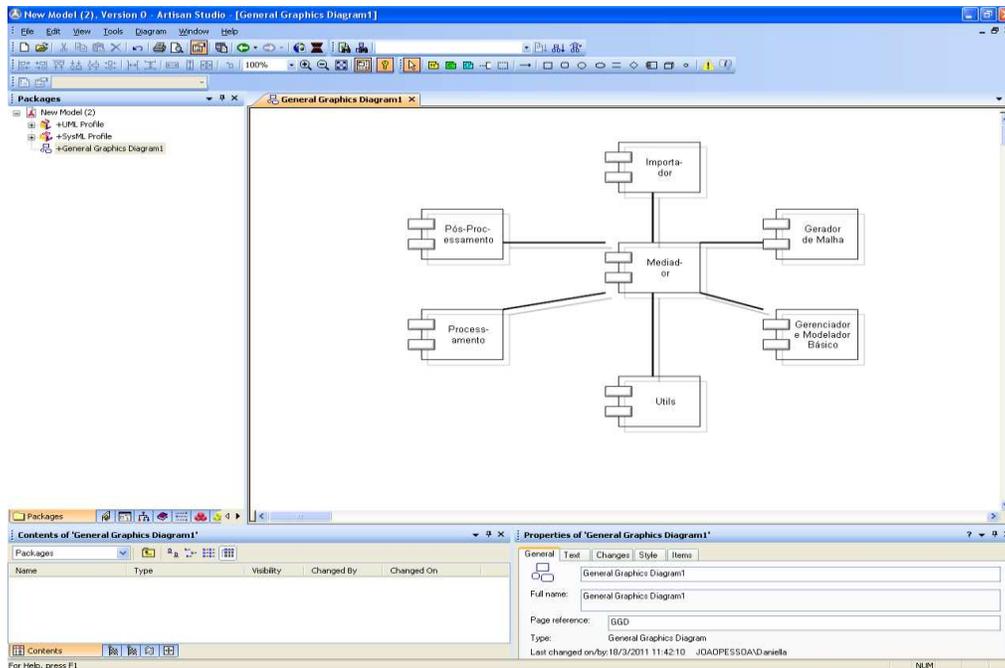


Figura 17 - Artisan Studio - Diagramador SysML

3.6. CONCLUSÃO DO CAPÍTULO

Os softwares de eletromagnetismo precisam de bons princípios de engenharia de software por trabalharem com questões de alta complexidade. Logo é importante que esses softwares possuam uma boa documentação.

Com a documentação a manutenção do software torna-se mais fácil. Informações importantes podem ser reutilizadas em outros projetos e também facilita a comunicação entre os membros da equipe de desenvolvimento no decorrer do projeto, portanto, gerar a documentação do software é muito importante, e atualmente é visto como algo necessário no processo de desenvolvimento de sistemas.

Neste trabalho foi utilizada a técnica SysML para realizar a documentação do simulador eletromagnético SEM-BR, com a ajuda das ferramentas Topcased e Artisan Studio. Sendo construídos os diagramas que ajudam a compreender o funcionamento do simulador. A SysML

foi utilizada por ser uma técnica que contém o diagrama de requisitos e blocos que facilitam a interpretação das funcionalidades do software.

Capítulo 4

AVALIAÇÃO HEURÍSTICA, SEMIÓTICA COMPUTACIONAL E PERCURSO COGNITIVO UTILIZADOS NA VALIDAÇÃO DA INTERFACE DO SIMULADOR ELETROMAGNÉTICO SEM-BR

A interface é um importante componente utilizado em análises de qualidade de software. Segundo (Rocha e Baranauskas, 2003), o conceito de interface foi evoluindo com o passar dos anos, podendo ser definida como sendo uma forma de interação entre diferentes partes como, por exemplo, software e hardware.

Um conceito associado à interface é a usabilidade, que segundo a ISO 9241 pode ser definida como: “medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” (ABNT, 2002).

Já segundo Nielsen (1993), usabilidade está relacionada com facilidade de aprendizado, a lembrança de como uma tarefa deve ser realizada após algum tempo, rapidez no desenvolvimento de tarefas, baixa taxa de erros e a satisfação subjetiva dos usuários.

Cada um desses itens é descrito em (Nielsen, 1993), sendo:

- Facilidade de aprendizado: facilidade de uso do software, necessitando de poucas consultas ao manual, sendo desenvolvido de forma que o usuário consiga rapidamente começar a interagir com o software;

- Lembranças para realizar uma tarefa após algum tempo: facilidade de lembrar passos semelhantes realizados em tarefas anteriores, quando o usuário voltar a usar depois de um tempo saberá como usar sem ter que reaprender como utiliza o sistema;
- Rapidez no desenvolvimento de tarefas: está relacionada à agilidade de solução de problemas com o software. Nos simuladores está relacionado à modelagem e visualização dos resultados;
- Baixa taxa de erros: reduzido número de problemas durante as tarefas, podendo o erro se considerado como uma ação a qual não leva o usuário a concluir a tarefa esperada;
- Satisfação subjetiva do usuário: o usuário está satisfeito com o simulador, quanto aos recursos providos e a baixa taxa de erros durante as tarefas.

Outros aspectos a serem considerados associados à interface são a comunicabilidade e aplicabilidade.

A comunicabilidade esta relacionada com a forma como os usuários entendem o design assim como foram feitos pelos projetistas. Nos sistemas que possuam uma boa comunicabilidade os usuários são capacitados a responder as seguintes perguntas:

- Para que o sistema serve?
- Qual é a vantagem de utilizá-lo?
- Como funciona?

Em relação à aplicabilidade determina a qualidade de uso, a utilidade relacionada com variedades de situações e problemas. Podendo ser determinados os seguintes conceitos:

- O quanto o sistema é útil para o contexto em que foi projetado
- Em que outros contextos o sistema pode ser útil.

A seguir são listados alguns problemas relacionados a baixa qualidade das interfaces:

- Requerem treinamento excessivo

- Desmotivam a exploração
- Confundem os usuários
- Induzem os usuários ao erro
- Geram insatisfação
- Diminuem a produtividade
- Não trazem o retorno de investimento previsto

A avaliação da interface com intuito de proporcionar maior usabilidade, comunicabilidade e aplicabilidade, foi realizada na interface do Simulador eletromagnético SEM-BR, através das técnicas de avaliação heurística, semiótica computacional e percurso cognitivo, sendo descritas abaixo.

4.1 AVALIAÇÃO HEURÍSTICA

Avaliação heurística é um dos métodos de inspeção de usabilidade mais popular. A inspeção é feita sistematicamente na interface de acordo com os dez princípios de Nielsen, que são chamados de heurísticas porque são mais regras valiosas do que diretrizes de usabilidade específicas.

A avaliação heurística possui um baixo custo para ser aplicada, pois é barata, rápida e fácil de ser utilizada.

O objetivo do método é reduzir o custo e simplificar a utilização de técnicas de usabilidade em projetos de software (Nielsen e Mack, 1994), sendo as dez heurísticas de Nielsen apresentadas a seguir:

- Heurística 1: Compatibilidade do sistema com o mundo real: O sistema precisa falar a mesma linguagem do usuário, sem jargões de “computeiros”, utilizando, portanto palavras, frases e conceitos familiares ao usuário. Tendo também a preocupação de seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica.

- Heurística 2: Estética e design minimalista: Diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.
- Heurística 3: Consistência e padrões: Usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções plataforma computacional.
- Heurística 4: Prevenção de erros: Melhor que uma boa mensagem de erro é um design cuidadoso o qual previne o erro antes dele acontecer.
- Heurística 5: Ajuda e documentação: Embora seja melhor um sistema que possa ser usado sem documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário e não muito extensas.
- Heurística 6: Visibilidade do status do sistema: O sistema necessita manter os usuários informados sobre o que está acontecendo, fornecendo um retorno adequado dentro de um tempo razoável.
- Heurística 7: Reconhecimento ao invés de relembração: Tornar objetos, ações e opções visíveis. O usuário não deve ter que lembrar informações de uma parte para outra parte do diálogo. Instruções para o uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.
- Heurística 8: Flexibilidade e eficiência de uso: Usuários novatos se tornam peritos com o uso. Prover aceleradores de forma a aumentar a velocidade da interação. Permitir a usuários experientes “cortar caminho” em ações freqüentes.
- Heurística 9: Liberdade e controle do usuário: Os usuários muitas vezes escolhem por engano funções do sistema e precisam ter claras saídas de emergência para sair do estado indesejado sem ter que percorrer um extenso diálogo.
- Heurística 10: Ajudar os usuários a reconhecer, diagnosticar e corrigir erros: Mensagens de erros devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e construtivamente sugerindo uma solução.

A cada heurística pode-se associar valores que identificam seu grau de severidade e o impacto na interface (Nielsen, 1997). Na Tabela 3 é apresentada a classificação para os problemas de

acordo com o grau de severidade, definida em uma escala de 0 a 4. Sendo o grau de severidade com escala o mais baixo não necessitando ser corrigido de imediato ao passo que o grau de severidade com escala igual a 4 deve ser corrigido imediatamente, pois é um erro que impacta no funcionamento do software.

Tabela 3 - Grau de severidade dos problemas de usabilidade

Grau de severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface.
1	Cosmético	Não é essencial para o funcionamento da interface, portanto não há necessidade imediata de solução.
2	Simple	Este problema deve ser corrigido, porém com baixa prioridade.
3	Grave	Problema deve ser corrigido com alta prioridade. Afeta a funcionalidade da interface.
4	Catastrófico	Um problema catastrófico deve ser corrigido o mais rápido possível antes da liberação do produto, pois afetará na funcionalidade da interface.

Nielsen recomenda que haja de 3 a 5 avaliadores que façam a avaliação heurística, o que garantirá uma cobertura de erros de 85%, por isto a técnica foi aplicada com dois grupos. Um grupo com 4 usuários finais e um outro grupo com 4 especialistas em interface.

O primeiro grupo com 4 usuários, os quais eram usuários finais levaram em média uma hora e meia para percorrer o simulador e fazer a correlação entre as heurísticas de Nielsen e os graus de severidade. No anexo A mostra como foi realizada a correlação dez heurísticas com os graus de severidade pelo usuário.

O segundo grupo com 4 usuários especialistas levaram em média uma hora para percorrer o simulador e fazer a relação entre as dez heurísticas de Nielsen e os cinco graus de severidade.

O intuito de utilizar usuários finais ao longo da avaliação é de garantir que seja levado em consideração o ponto de vista do usuário.

4.2 SEMIÓTICA COMPUTACIONAL

Semiótica é a ciência que estuda como se relacionam os signos. Os signos são usados para expressar uma realidade física ou psicológica.

De acordo com Peirce o conceito de signo está relacionado a três elementos, sendo eles (Peirce, 2005):

- Objeto (o objeto em si): representa o objeto que está sendo analisado;
- Interpretante (interpretação do objeto): representa a interpretação do objeto que esta sendo analisado;
- *Representâmem* (representação do objeto): é a representação do objeto.

A Figura 18 mostra a representação triádica de Pierce, com os três elementos (interpretante, objeto e *representâmem*).



Figura 18: Representação triádica de Pierce (Peirce, 2005)

A semiótica tem como papel primordial ajudar na comunicação, apurando ou transmitindo o sentido de um produto ao consumidor, de forma que uma informação seja passada ao usuário chamando sua atenção de forma clara e objetiva.

Na abordagem da engenharia semiótica, o designer envia uma mensagem ao usuário, contendo funcionalidade (as funções que os usuários podem fazer) e interatividade (como os usuários podem interagir com o sistema).

A idéia ou aplicação da semiótica com uso do computador na transmissão da informação através dos signos nos leva a pesquisa da semiótica computacional, na qual o projetista desenvolve um papel parecido com um escritor, de descrever um cenário para o seu leitor.

A Interface Humano Computador tem a preocupação de transmitir um número de alternativas racionais para cada uma das mensagens, buscando representar dentro de uma interface, um ícone ou mensagem que façam sentido para o usuário.

Andersen classifica os diferentes tipos de signos com base em computador em seis, os quais são (Baranauskas, Rossler e Oliveira, 1999):

- Signo interativo: são os signos usados pelos usuários para interagir diretamente com o software, contato direto do usuário com o software. Exemplo: os botões presentes nas interfaces, que o usuário aciona para interagir com o software, como os botões de confirmar e cancelar.
- Signos atores: são os signos que podem influenciar outros signos, contudo não são influenciados diretamente pelo usuário. Exemplo: barras de progressão, presentes no software.
- Signos controladores: são signos que podem alterar a propriedade de outros signos, contudo as suas próprias propriedades não são alteradas. Exemplo: as bordas das janelas do software.
- Signos objetos: são signos que possuem aspecto de transição e que podem ser realocados. Exemplo: Texto em um editor de texto.

- Signos layout: são signos que servem como enfeite cuja finalidade é de proporcionar uma melhor visão do software. Exemplo: Figura de fundo na tela ou logotipo no caso do simulador SEM-BR.
- Signos fantasma: são signos que possuem impacto no comportamento de outros signos.

4.3 Percurso cognitivo

É uma técnica de inspeção de usabilidade que tenta tornar efetiva a prática da teoria cognitiva na avaliação de interfaces homem-computador. Sendo feita a avaliação pelos membros da equipe de desenvolvimento da interface, portanto uma técnica realizada com os avaliadores e não com os usuários.

O percurso cognitivo pode ser realizado individualmente ou por um grupo de avaliadores, e visa avaliar o quanto a interface é fácil de ser explorada e o quanto o usuário aprende.

Nessa técnica são analisadas as tarefas de forma a cobrir as metas dos usuários na aplicação. Podendo ser aplicada desde início do desenvolvimento do projeto (Faust, 1995). Avaliando o design em relação a facilidade de aprendizagem, mas especificamente com relação a exploração.

O avaliador segue as ações que provavelmente os usuários iriam percorrer, com a intenção de encontrar possíveis defeitos e /ou verificar se o design da interface esta bom, sendo checados ou verificados os menus, se as palavras ou expressões utilizadas não possuem jargões e os *labels* dos botões.

Para dar início ao percurso cognitivo o avaliador deve fazer a preparação da avaliação, com as tarefas que serão inspecionadas e a sequência ou passos das ações para cada uma das tarefas da interface objeto da análise.

4.4. REALIZAÇÃO DAS AVALIAÇÕES NA INTERFACE

Neste tópico podem ser vistos os resultados das avaliações realizadas na interface do Simulador Eletromagnético SEM-BR. Este tem como finalidade realizar a simulação eletromagnética, onde para ser gerada uma malha é necessário que algumas etapas sejam seguidas como descrito abaixo:

- Inserir ao menos uma geometria e definir suas propriedades eletromagnéticas;
- Mudar o tipo do material da geometria escolhida;
- Inserir fonte eletromagnética;
- Inserir algum relatório de interesse para a visualização no pós-processamento;
- Gerar a malha do domínio computacional.

Portanto foi solicitado ao usuário percorrer o simulador para ser realizada a seguinte sequência:

- Criar um arquivo novo
- Clicar em Nova geometria
- Desenhar uma esfera na área de desenho
- Mudar o tipo de material da esfera, por meio do menu editar, clicando com botão direito no elemento esfera.
- Alterar o tipo de material do domínio computacional, que deve ser diferente do material anteriormente selecionado, pois caso o tipo de material do domínio computacional seja igual ao tipo de material da geometria, não será possível visualizar a malha gerada.
- Cadastrar uma fonte eletromagnética
- Inserir uma fonte eletromagnética e
- Gerar uma malha e realizar a simulação.

Durante a avaliação os usuários ficaram livres para acessar o simulador e trabalhar individualmente, sendo observados pelo coordenador da avaliação.

O simulador eletromagnético SEM-BR tem como público alvo, usuários experientes que possuem conhecimentos em informática e eletromagnetismo.

4.4.1 Resultados da Avaliação heurística

Foram realizadas duas avaliações heurísticas. A primeira com um grupo de quatro usuários finais ou potenciais utilizadores do software e a segunda avaliação foi realizada com quatro avaliadores especialistas em interação homem-computador.

O primeiro grupo de avaliadores possui formação em automação, física, engenharia elétrica e pós-doc em engenharia elétrica, todos com conhecimentos de informática e eletromagnetismo, com faixa etária entre 25 a 45 anos. Cada participante realizou a pesquisa individualmente.

Na Tabela 4 pode ser visto um resumo da formação e idade de cada um dos participantes da primeira avaliação heurística.

Tabela 4 - Relatório dos participantes da primeira avaliação

Relatório de Participantes			
Coordenador da avaliação: Daniela Santana			Data: 20/04/2011
Número	Formação	Idade	Data
A1	Automação	45	12/04/2011
A2	Engenharia elétrica	26	13/04/2011
A3	Física	25	14/04/2011
A4	Pós-doc Engenharia Elétrica	31	19/04/2011

O segundo grupo de avaliadores possui formação em engenharia elétrica e tecnologia em informática, todos com conhecimentos de informática e eletromagnetismo, com faixa etária entre 24 a 46 anos. Cada participante realizou a pesquisa individualmente.

Na Tabela 5 pode ser visto um resumo da formação e idade de cada um dos participantes da segunda avaliação.

Tabela 5 - Relatório dos participantes da segunda avaliação

Relatório de Participantes			
Coordenador da avaliação: Daniela Santana			Data: 30/04/2011
Número	Formação	Idade	Data
A1	Engenharia Elétrica	46	22/04/2011
A2	Engenharia elétrica	24	23/04/2011
A3	Tecnologia em informática	25	24/04/2011
A4	Tecnologia em informática	26	29/04/2011

Foram selecionados apenas quatro usuários em cada uma das avaliações, pois segundo Nielsen para realizar uma boa avaliação é necessário apenas de três a cinco avaliadores. Um número acima disso aumenta o custo do projeto, e com três a cinco avaliadores podem ser encontrados até 85% dos problemas (Salgado, Bim e Souza, 2006).

O grupo da primeira avaliação foi treinado para realizar a avaliação, visto que são usuários finais e não especialistas em interface. No treinamento foi explicado o que eram cada uma das dez heurísticas, o grau de severidade e a relação de cada uma das heurísticas com grau de severidade.

Com os resultados da primeira avaliação heurística foi realizada uma média das respostas dos 4 avaliadores, como mostrado na Figura 19. De acordo com a média as heurísticas que obtiveram grau de severidade maior que dois foram: heurística 1 (compatibilidade do sistema com o mundo real), heurística 2 (estética e design minimalista) e heurística 3 (consistência e padrões). Foi realizada esta média para verificar qual heurística deveria ser corrigida, visto que não haveria tempo para corrigir todas as heurísticas. Sendo assim as heurísticas que possuem na média um grau de severidade maior que 2 deveriam ser corrigidas.

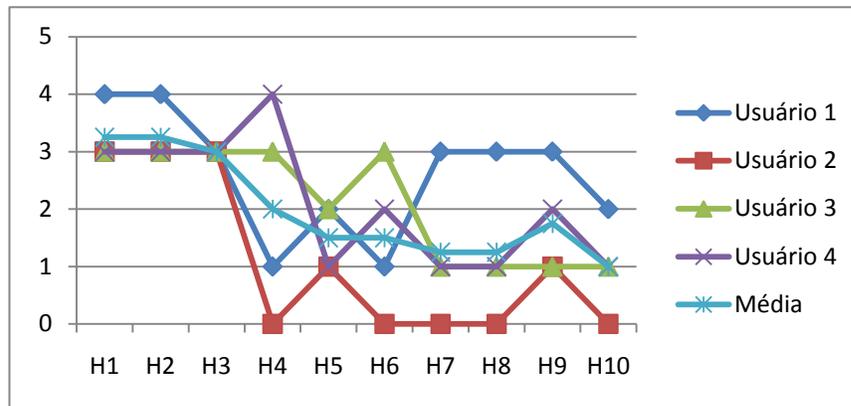
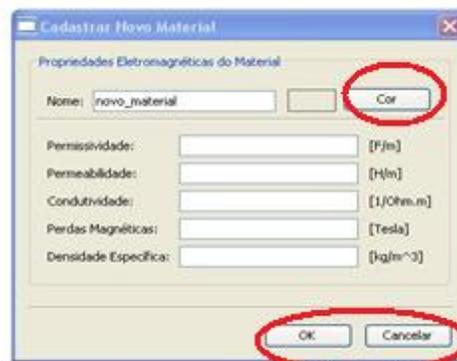


Figura 19: Primeira avaliação heurística

Foram realizadas modificações no simulador com intuito de melhorar sua interface solucionando os problemas relacionados as heurísticas que apresentaram problemas, sendo as modificações descritas a seguir.

Na Figura 20 são apresentadas duas telas do software, a tela de Cadastrar Novo Material e Inserir Fonte Eletromagnética Simulação, de como estavam antes da padronização. Em uma das telas os botões aparecem no centro, na outra no canto direito e ícones apenas em algumas telas como, por exemplo, na tela Inserir Fonte Eletromagnética Simulação.

Outro aspecto importante a destacar na tela Inserir Fonte Eletromagnética Simulação é o fato dos ícones não corresponderem ao nome das funcionalidades, Cancelar e Fechar a janela tem o mesmo ícone.



(a)



(b)

Figura 20: Visão do software antes das modificações (a) Interface Cadastrar Novo Material e a (b) Interface Inserir Fonte Eletromagnética Simulação.

Na Figura 21, são mostradas algumas das modificações que foram realizadas nas telas como: padronização dos botões no canto direito e a padronização do software com a linguagem do usuário, com palavras familiares a ele.

A imagem de cada um dos botões foi selecionada de acordo a atender sua funcionalidade, sendo todos os ícones padronizados, de forma que uma imagem não possua vários significados, sendo escolhidos os símbolos dos ícones de acordo com conversa entre o analista (projetista) e os usuários.



(a)



(b)

Figura 21: Software após as modificações (a) Interface Cadastrar Novo Material. (b) Interface Inserir Fonte Eletromagnética Simulação

Foi realizada uma segunda avaliação heurística com um novo grupo de 4 usuários, contudo esse grupo de usuários são especialistas em interface.

Após os usuários percorrerem o software e fazer a correlação entre as heurísticas e o grau de severidade, foi realizada a média dos resultados dos quatro avaliadores. Nesta segunda avaliação as heurísticas que apresentavam problemas na primeira avaliação não apresentaram mais problemas, no entanto, a heurística cinco (ajuda e documentação), obteve na média grau de severidade maior que 2, como pode ser observado na Figura 22.

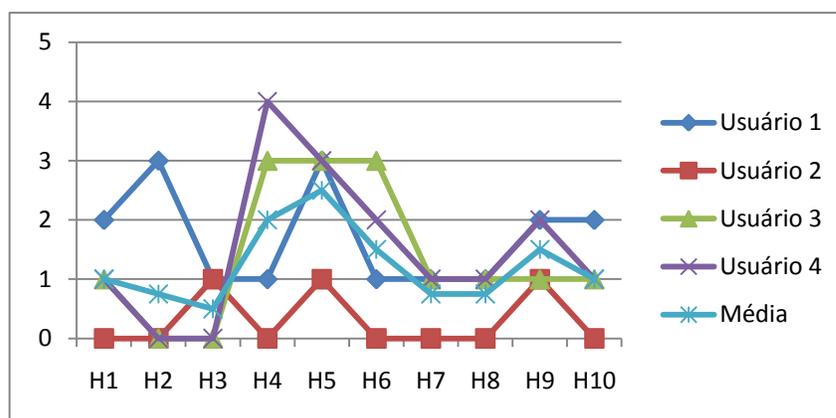


Figura 22: Segunda avaliação heurística

Para atender a segunda avaliação heurística foram inseridas no simulador, telas de Ajuda com a descrição das funcionalidades do software, como pode ser observado na Figura 23. A primeira tela da Figura 23 é a primeira tela do manual de ajuda e a segunda tela são os tópicos da ajuda.

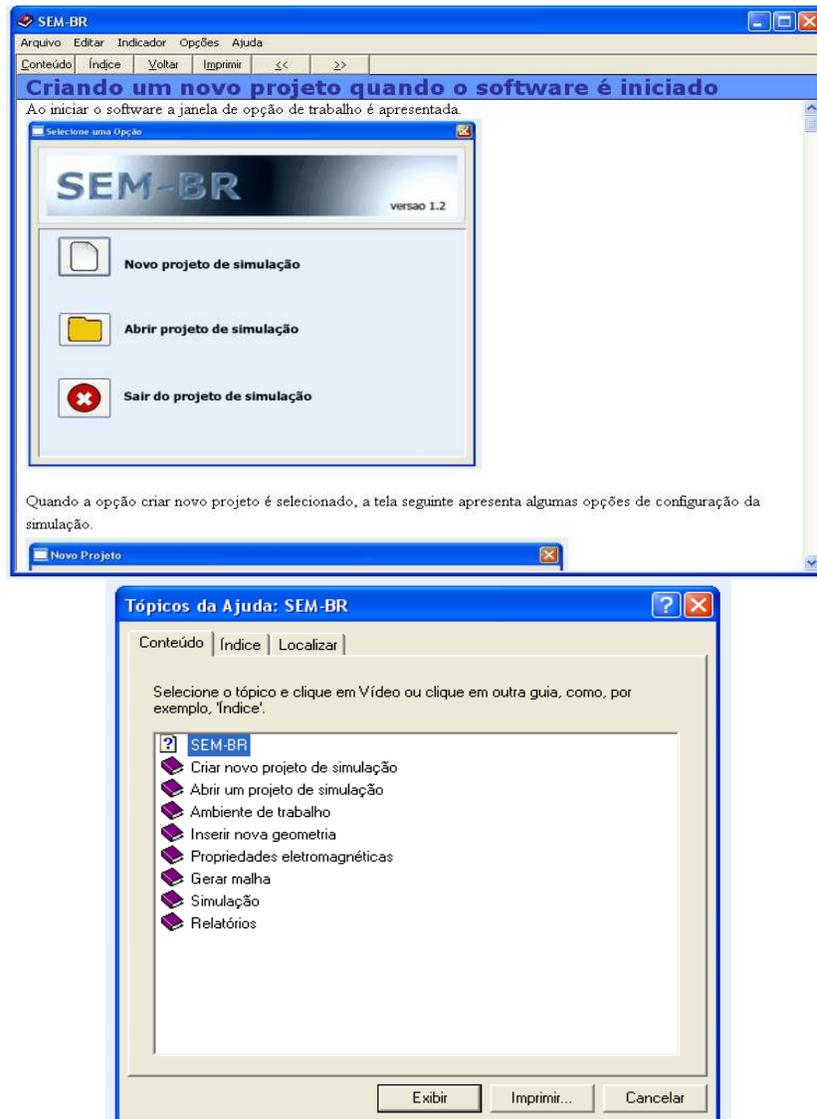


Figura 23: Telas de Ajuda do sistema

Foi verificada algumas inconsistências tela principal do software, como verificado na figura 24 onde é visualizado o menu principal do software em inglês e alguns ícones como mover a direita estavam com sua legenda errada.



Figura 24 - Interface do painel gerenciador

Na figura 25 é mostrado menu principal em português e os ícones com a legenda errada foram alterados como por exemplo mover a direita.



Figura 25 - Interface do painel gerenciador modificada

4.4.2 Resultados da Semiótica Computacional

A semiótica computacional foi realizada com um grupo de cinco usuários juntamente com o avaliador. Foi percorrido do simulador verificando todos os ícones e expressões, sendo alteradas as imagens que não representassem bem a funcionalidade e as expressões cujo conteúdo não estava claro para o usuário. Sendo todos os ícones também padronizados.

Na Figura 26 é apresentada a tela principal do software SEM-BR com a relação dos signos, nesta tela estão presentes os seguintes signos: de interação, controlador e layout. O signo de interação nos botões (novo, abrir e sair), o signo controlador esta presente na borda do software, controlando assim o tamanho da tela e o signo de layout no nome do software, na cor de fundo e logotipo do software.



Figura 26: Aplicação da tipologia por Andersen na tela principal do software SEM-BR

A figura 27 mostra a tela de início do software antes das modificações realizadas pela semiótica computacional.

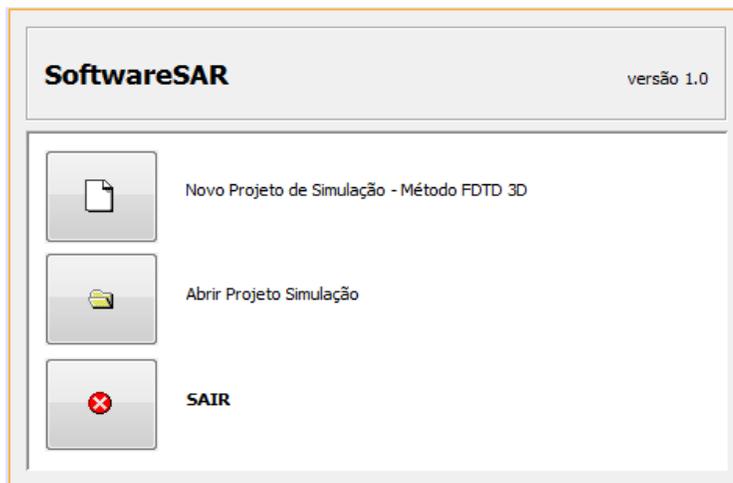


Figura 27 - Tela de início antes das modificações

Na figura 28 é mostrada a tela de cadastrar Novo Material do software SEM-BR com relação dos signos presentes nesta tela, os quais são: signo de interação, objeto, controlador e layout. Signo de interação esta presente nos botões (cor, cancelar e confirmar) sendo a interação entre o usuário e o software, signo controlador presente na borda da janela, para controlar o

dimensionamento da tela, signo objeto presente na caixa de texto, a interação do software com o usuário e o signo de *layout* no título da tela.

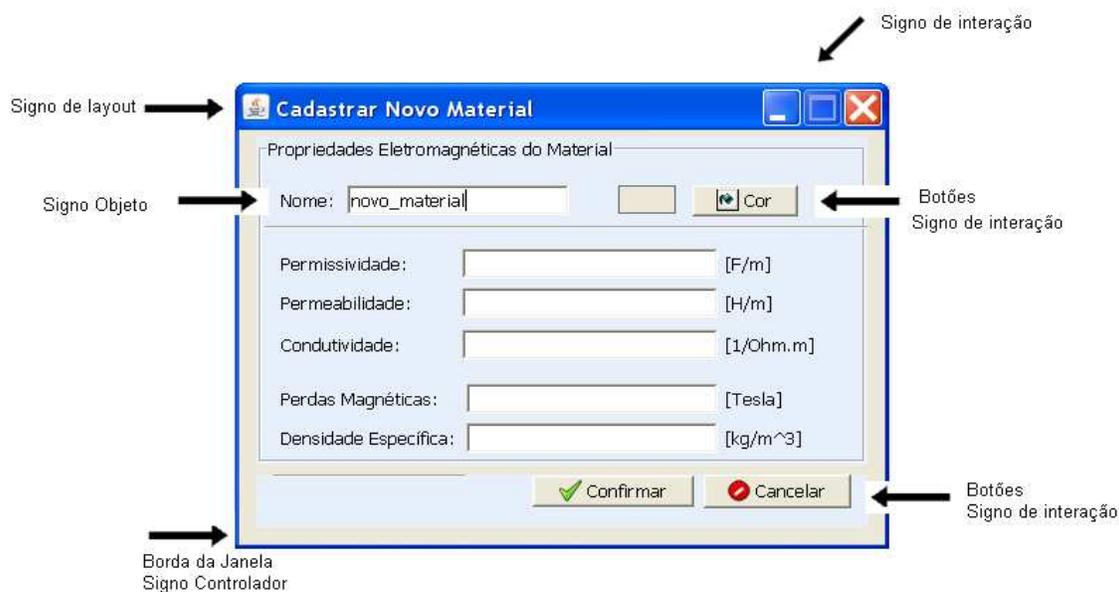


Figura 28 - Aplicação da tipologia por Andersen na Tela Cadastrar Novo Material

4.4.3 Resultados do Percurso Cognitivo

Após realizadas as avaliações heurísticas e a semiótica computacional foi feito o percurso cognitivo.

O percurso cognitivo tem como finalidade percorrer o software verificando se as alterações realizadas foram bem sucedidas e checar se há mais alguma modificação que deve ser realizada.

Diferente da avaliação heurística e da semiótica computacional o percurso cognitivo foi realizado somente pelo avaliador.

Para aplicação do percurso cognitivo foram selecionadas quatro tarefas, as quais são: Criar geometria e modificar suas propriedades, criar uma fonte eletromagnética, modificar o domínio computacional e gerar a malha.

Tarefa 1 – Criar uma geometria e modificar suas propriedades

Cenário – O usuário deve criar uma geometria no campo de desenho do software e modificar suas propriedades como tipo de material.

Sequência de ações:

Passo 1: Clicar no botão nova geometria

Passo 2: Clicar na geometria a ser desenhada

Passo 3: Desenhar na tela a geometria

Passo 4: Clicar com o botão direito sobre a geometria criada na árvore de gerenciamento

Passo 5: Clicar em editar

Passo 6: Mudar o material

Passo 7: Clicar em confirmar

A tabela 6 descreve a análise da tarefa para inserir uma geometria e alterar suas propriedades.

Tabela 6 - Análise percurso cognitivo da tarefa 1

Questões/Passos	Criar geometria	Desenhar geometria na tela	Editar propriedades
A ação desejada pelo usuário está presente no software	Sim	Sim	Sim
O usuário irá interpretar corretamente as respostas do sistema?	Sim	Sim	Sim
O usuário irá associar as ações corretas ao que pretende fazer?	Sim	Sim	Sim

A tarefa 1 então foi realizada com sucesso, não sendo encontrado nenhum problema no seu funcionamento.

Tarefa 2 – Inserir uma fonte eletromagnética

Cenário – O usuário deve conseguir inserir uma ou mais fontes eletromagnéticas no simulador SEM-BR.

Sequência de ações:

Passo 1: Clicar em editar no menu principal

Passo 2: Clicar em Cadastrar fonte eletromagnética

Passo 3: Clicar em Nova fonte

Passo 4: Selecionar a fonte cadastrada

Passo 5: Clicar em confirmar

A tabela 7 descreve a análise da tarefa 2 para inserir uma ou mais fontes eletromagnéticas.

Tabela 7 - Análise percurso cognitivo da tarefa 2

Questões/Passos	Cadastrar uma fonte eletromagnética	Inserir uma fonte eletromagnética
A ação desejada pelo usuário está presente no software	Sim	Sim
O usuário irá interpretar corretamente as respostas do sistema?	Sim	Sim
O usuário irá associar as ações corretas ao que pretende fazer?	Sim	Sim

A tarefa 2 então foi realizada com sucesso, não sendo encontrado nenhum problema no seu funcionamento.

Tarefa 3 – Modificar o domínio computacional

Cenário – Depois de o usuário inserir uma geometria e uma fonte eletromagnética o usuário precisa modificar o domínio computacional, verificando suas dimensões e mudando o tipo de material do domínio computacional, para que seja diferente do tipo de material da geometria.

Sequência de ações:

Passo 1: Entrar na tela de domínio computacional

Passo 2: Alterar o tipo de material do domínio computacional

Passo 3: Clicar em confirmar

A tabela 8 descreve a análise da tarefa para modificar o domínio computacional.

Tabela 8 - Análise percurso cognitivo da tarefa 3.

Questões/Passos	Entrar na tela de domínio computacional	Alterar o tipo do material do domínio computacional
A ação desejada pelo usuário está presente no software	Sim	Sim
O usuário irá interpretar corretamente as respostas do sistema?	Sim	Sim
O usuário irá associar as ações corretas ao que pretende fazer?	Sim	Sim

A tarefa 3 então foi realizada com sucesso, não sendo encontrado nenhum problema no seu funcionamento.

Tarefa 4 – Gerar malha

Cenário – Depois de o usuário inserir uma geometria e uma fonte eletromagnética e modificar o tipo de material do domínio computacional o usuário deve conseguir gerar uma malha

Sequência de ações:

Passo 1: Verificar o domínio computacional

Passo 2: Clicar em gerar malha

Passo 3: Clicar em confirmar

A tabela 9 descreve a análise da tarefa para gerar uma malha FDTD.

Tabela 9: Análise percurso cognitivo da tarefa 4

Questões/Passos	Verificar domínio computacional	Gerar malha
A ação desejada pelo usuário está presente no software	Sim	Sim
O usuário irá interpretar corretamente as respostas do sistema?	Sim	Sim
O usuário irá associar as ações corretas ao que pretende fazer?	Sim	Sim

A tarefa 4 então foi realizada com sucesso, não sendo encontrado nenhum problema no seu funcionamento.

4.5.1 Análise dos resultados obtidos na avaliação heurística e semiótica computacional realizadas no simulador eletromagnético SEM-BR

Com os resultados das duas avaliações heurísticas realizadas com dois grupos de quatro avaliadores no laboratório DMO (Departamento de Micro-ondas e Óptica)/UNICAMP, foi montado o gráfico da Figura 29, onde são mostradas as médias das duas avaliações.

Nota-se com o gráfico da Figura 15 que na primeira avaliação heurística foram apresentados problemas nas heurísticas: 1, 2 e 3, as quais possuíam grau de severidade maior que dois.

Na segunda avaliação heurística foram validadas as alterações realizadas depois de diagnosticados problemas na primeira avaliação. Foi detectado, no entanto que a heurística cinco (ajuda e documentação) apresentava problema com grau de severidade maior que dois, sendo prontamente corrigida.

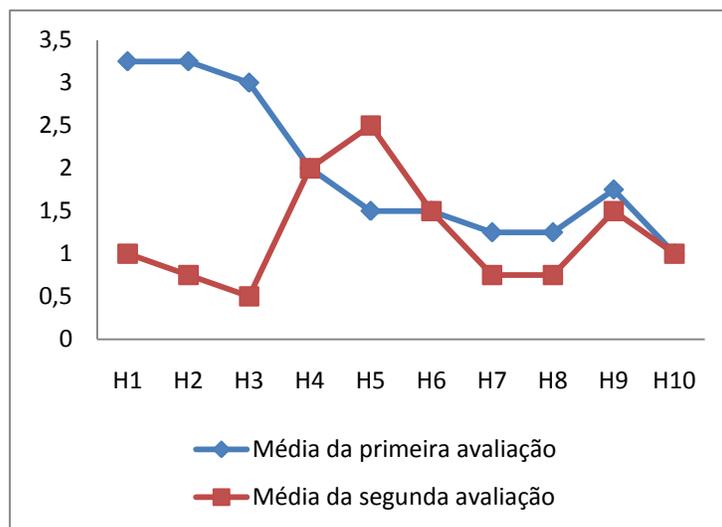


Figura 29 - Média das avaliações heurísticas

Portanto, as heurísticas que apresentaram na média, grau de severidade maior que dois, foram modificadas para atender as necessidades dos usuários, tornando a interface do simulador mais compreensível.

Além das avaliações heurísticas foi utilizada a técnica de semiótica computacional.

Um grupo com cinco usuários, todos com conhecimentos em informática e eletromagnetismo com formação em engenharia elétrica e tecnologia em informática examinaram junto com avaliador, avaliaram as figuras que melhor correspondiam a funcionalidade dos botões e todas as expressões contidas no simulador.

É importante que o usuário participe desse processo porque nem sempre o que o desenvolvedor pensa a respeito de uma imagem corresponde a visão do usuário.

Todas as expressões e as imagens dos botões foram selecionadas de forma a proporcionar clareza, coerência e objetividade.

Com o uso das avaliações heurísticas e da semiótica computacional pode ser notado que a interface continha vários erros graves que comprometiam o funcionamento correto do simulador. Isso é resultado de um processo de testes que não se focava em usuários e sim nas funcionalidades e, algumas vezes, não previa rotinas de testes que cobrissem completamente os recursos do simulador.

Após a realização da avaliação heurística e semiótica computacional foi realizado o percurso cognitivo com intuito de checar se as modificações realizadas não causaram nenhum impacto negativo ao simulador.

Com a realização das avaliações heurísticas, análise de severidade dos problemas e uso da semiótica computacional e percurso cognitivo pode-se corrigir os problemas presentes na interface. Após essas correções uma nova submissão de avaliação foi realizada, não sendo encontradas heurísticas com grau de severidade maior que 2, tornando a interface mais fácil, simples e entendível de ser usada pelos usuários.

4.6. COMPARAÇÃO DAS TÉCNICAS DE AVALIAÇÃO DE INTERFACE

Os parâmetros de comparação entre as avaliações realizadas foram: consumo de tempo e o envolvimento dos participantes.

Primeiro foram aplicadas duas avaliações heurísticas com dois grupos de 4 usuários, tendo a segunda avaliação heurística o intuito de validar as correções realizadas após a primeira avaliação e encontrar possíveis novos defeitos.

Logo após foi aplicada a semiótica computacional com um grupo de 5 usuários, para validar as expressões e imagens dos ícones presentes no simulador.

Por último foi feito o percurso cognitivo com apenas o avaliador, a fim de verificar se todas as tarefas poderiam ser realizadas pelo usuário sem erros, e as modificações realizadas não teriam comprometido o funcionamento do software.

De acordo com a Tabela 10, o envolvimento dos usuários foi presente nas avaliações heurísticas e semiótica computacional, enquanto que apenas no percurso cognitivo somente o avaliador participou.

Todos os métodos de avaliação podem ser executados a partir do protótipo, sendo o percurso cognitivo o método que demanda maior tempo, e as outras duas avaliações demandam maior número de participantes.

Dos itens comparados, a participação do usuário no processo revelou ser muito importante por revelar vários erros ou discordâncias imprevistos ou não detectados pelos avaliadores, revelando ser recomendável a participação dos usuários ao longo do processo de desenvolvimento do software.

Tabela 10 - Comparação das avaliações de interface

Características	Percurso cognitivo	Avaliação heurística	Semiótica computacional
Tempo na aplicação da avaliação	3 horas	1,5 horas	1,5 horas
Nº de avaliadores	1	Dois grupos de 4 usuários	5
Envolvimento de	Não	Sim	Sim

usuários			
Papel do especialista	Avaliador	Mediador	Mediador
Envolvimento do desenvolvedor	Recomendável	Não	Não
Necessidade de especialista	Sim	Não	Não

4.7 Conclusões do capítulo

A interface é um importante recurso responsável por permitir a comunicação entre o homem e o software. Uma interface bem projetada e com boa usabilidade, evita que os usuários percam tempo em descobrir suas funcionalidades e se tornar agradável de serem manuseadas.

Tendo em vista o cenário descrito, foram realizadas avaliações de interface e de percurso de caminho no software com intuito de melhorar a qualidade do mesmo.

Para avaliação da interface foram utilizadas as técnicas de avaliação heurística, semiótica computacional e percurso cognitivo. As duas primeiras técnicas utilizadas com a presença do usuário garantindo, portanto a validação do usuário.

De acordo com as avaliações heurísticas realizadas junto com os usuários foram descobertas algumas violações em algumas heurísticas sendo estas consertadas e com o apoio da semiótica computacional foram selecionados ícones e expressões que transmitissem seu real significado aos usuários.

Foi utilizada a técnica de percurso cognitivo para validar todas as modificações realizadas pelas duas técnicas anteriores e garantir que as funcionalidades estão funcionando e as expressões e imagens acrescentadas faz sentido no contexto do simulador.

CAPÍTULO 5

TESTES UTILIZANDO O UPPAAL

5.1 VALIDAÇÃO E VERIFICAÇÃO

Os testes são executados para mostrar os erros de um sistema, contudo é impossível certificar que um software está “livre de defeitos” – segundo Edsger Dijkstra, testes apenas apontam a presença de erros, e não a sua ausência (Dijkstra, 1972). De acordo com a definição de Myers, testar é analisar um programa com a intenção de descobrir erros e defeitos (Myers, 2004).

Testes também podem ser considerados como um conjunto de procedimentos para verificar a correção de um software em desenvolvimento. Essencialmente, um teste executa o programa com base em dados de entrada pré-fixados e compara os dados de saída com os resultados esperados, para concluir se a execução ocorreu corretamente ou se apresenta algum defeito (Beniti e Zimmermann, 2007).

Os testes podem ser projetados de duas maneiras: caixa preta, quando não é verificado como ocorre o processamento, mas apenas os resultados obtidos, obtendo uma boa cobertura dos testes; e caixa branca, quando determinam defeitos na estrutura interna do programa para garantir sua validade, através de desenhos de testes que exercitem eficientemente os possíveis caminhos de execução.

A etapa de testes faz parte do processo de V & V (verificação e validação) de um software em construção, a qual num âmbito mais geral, procura validar se o software satisfaz as necessidades dos usuários e verifica se o programa atende os seus requisitos computacionais funcionais e não-funcionais.

A validação é muito importante na engenharia de software, pois os erros em um documento podem levar a custos excessivos de trabalho quando é descoberto durante o desenvolvimento ou em maiores custos depois que o sistema está em operação (Sommerville, 2007). Quanto antes o erro é descoberto menor será o custo.

O software que será avaliado será o simulador eletromagnético utilizado em modelagens de dispositivos de telecomunicações, descrito com mais detalhes na seção 2.

A técnica utilizada para modelagem será o teste de software baseado em modelos, sendo descritos abaixo os diferentes tipos de modelos baseados em diferentes abordagens:

- Um modelo de fluxo de dados: mostra como os dados são processados em diferentes momentos do sistema.
- Um modelo de composição: mostra como as entidades no sistema são compostas por outras entidades.
- Um modelo de arquitetura: mostra os principais subsistemas que constituem um sistema.
- Um modelo de classificação: diagramas de classe/herança mostram como as entidades têm características comuns.
- Um modelo de máquina de estado ou diagrama de transição de estado: mostra como o sistema responde aos eventos internos e externos.

Neste trabalho será utilizado o modelo de máquinas de estado, pois supõe que a qualquer instante, o sistema está em um dos vários estados possíveis. Quando um estímulo é recebido, ele pode provocar uma transição para um estado diferente.

Devido ao grande conjunto de aplicação das máquinas de estado para a modelagem existe uma técnica de teste chamado Teste baseado em máquina de estado finito.

Os critérios para realizar tais testes são:

- Seleção aleatória: os caminhos de uma máquina de estado são selecionados aleatoriamente para ser exercitados;

- Todos os estados: devem ser exercitados todos os estados de uma máquina de estado pelo menos uma vez;
- Todas as transições: Devem ser exercitadas todas as transições de uma máquina de estado;
- Todas as combinações: devem ser exercitadas todas as combinações de uma máquina de estado.

Como os softwares estão ficando cada vez mais complexos, os testes foram automatizados, sendo possível a automatização devido ao suporte oferecido pelo modelo da aplicação a partir do qual os casos de teste podem ser automaticamente gerados, executados e os resultados avaliados para obter os resultados.

Além, da possibilidade da automatização, oferece também a flexibilidade. Se o comportamento do sistema for modificado, basta mudar o modelo e gerar novos casos de testes.

No entanto, além dos testes automatizados que utilizam ferramentas, há também os testes manuais, que neste trabalho será utilizado o checklist de testes apresentado na próxima seção.

5.2 CHECKLIST DE TESTE DO SIMULADOR ELETROMAGNÉTICO

Nesta seção é apresentada Tabela 11 com o checklist dos testes, na primeira coluna é descrita a funcionalidade do teste, na segunda coluna é mostrada a da funcionalidade descrição, na terceira coluna são apresentados os resultados esperados e na quarta coluna são verificados os resultados obtidos.

Tabela 11 - Checklist de teste do simulador eletromagnético.

Funcionalidade	Descrição	Resultados esperados	Resultados obtidos
Inserir geometria			

• Cilindro	Criar um cilindro	Seja criado cilindro	OK
• Esfera	Criar esfera	Seja criada esfera	OK
• Cone	Criar cone	Seja criado cone	OK
• Triângulo	Criar triângulo	Seja criado triângulo	OK
Importar arquivo STL	Escolhe um arquivo STL e importa-lo	Arquivo STL seja importado	OK
Configurar domínio	Modificar as dimensões do domínio computacional	Altera domínio computacional	OK
Modificar propriedade material	Modificar o tipo de material	Modificações alteradas do material	OK
Criar fonte eletromagnética			
Fonte: Gaussian	Criar uma fonte gaussiana	Seja criada a fonte gaussiana	OK
Fonte: Onda plana	Criar uma fonte onda plana	Seja criada a fonte onda plana	OK
Inserir fonte eletromagnética	Inserir uma fonte a partir das fontes criadas	Fonte inserida	OK
Inserir relatório			
• Filme 3D	Inserir relatório Filme 3D	Relatório inserido	OK
• Filme 2D	Inserir relatório Filme 2D	Relatório inserido	OK
• SAR	Inserir relatório SAR	Relatório inserido	OK
• Prop. eletromag	Inserir relatório propagação eletromagnética	Relatório inserido	OK
Gerar malha	Após serem inseridos todos os dados de entrada e gerada a malha	Malha gerada	OK

Visualizar relatório de pós-processamento	O resultado da geração da malha seja apresentando no relatório	Relatórios visualizados	OK
---	--	-------------------------	----

O *checklist* faz uma “varredura” pelo software, como um fluxo, tendo todas as tarefas as quais o simulador deve seguir.

Foram feitas 3 varreduras do simulador, ou percorrido o simulador de três formas diferentes para avaliar suas funcionalidades.

No primeiro percurso percorrido, foi criada uma esfera, uma fonte eletromagnética e gerada uma malha.

No segundo percurso percorrido, foram criados um cilindro e um cone, uma fonte eletromagnética e gerada a malha.

No terceiro percurso percorrido, foram criados um quadrado, cone e esfera, uma fonte eletromagnética e gerada a malha.

De acordo com os três percursos percorridos para o *checklist* todas as tarefas foram executadas corretamente, passando desde pré-processamento até a fase de pós-processamento.

Como os softwares estão ficando cada vez mais complexos, os testes foram automatizados, sendo possível a automatização devido ao suporte oferecido pelo modelo da aplicação a partir do qual os casos de teste podem ser automaticamente gerados, executados e os resultados avaliados para obter os resultados.

A automatização oferece a possibilidade da flexibilidade. Se o comportamento do sistema for modificado, basta mudar o modelo e gerar novos casos de testes.

A ferramenta que será utilizada neste trabalho será a UPPAAL, para os testes automatizados.

5.3. FERRAMENTA UPPAAL

Para realizar a verificação automatizada do Simulador Eletromagnético será utilizada a ferramenta UPPAAL.

UPPAAL é uma ferramenta que simula, especifica e verifica as propriedades de modelos abstratos para sistemas de tempo real. Ela foi desenvolvida em colaboração entre o Departamento de Tecnologia da Informação na Universidade de UPPsala, na Suécia e o Departamento de Ciência da Computação na Universidade de AALborg, na Dinamarca (Silva e Melo, 2009).

Sendo utilizada para os sistemas em tempo real modelando autômatos programados e consta de três interfaces:

- Interface de Edição (Modelagem)
- Interface de Simulação
- Interface de Verificação

Na interface de edição (modelagem) é realizada a construção do modelo do sistema.

A interface de Simulação, não é considerada um processo fundamental para a verificação, contudo é muito útil e intuitiva para uma avaliação eficiente da construção de um modelo. A simulação permite realizar ajustes e correções no modelo.

Já na interface de verificação permite confirmar se determinadas propriedades estão presentes na representação do sistema.

Quando todas as propriedades são avaliadas dentro da ferramenta, a mesma fica com o indicador na cor verde ou vermelho. O verde significa que a propriedade foi corretamente verificada pelo algoritmo de verificação e o vermelho que a propriedade não pode ser verificada.

5.4 VERIFICAÇÃO E VALIDAÇÃO DO SIMULADOR ELETROMAGNÉTICO SEM-BR

Para modelar o software SEM-BR em UPPAAL foram consideradas as seguintes opções:

- A criação de um novo projeto, representada como um autômato inicia a execução do sistema.
- A sincronização entre o autômato SEM e o USUÁRIO e mudança de estados é possível através dos diferentes botões do software.

Na Figura 30 é apresentado o processo para o desenvolvimento de um novo projeto de simulação eletromagnética, construído através de diagrama de estados, sendo observado o caminho o qual o software deve percorrer.

O início se dar no pré-processamento onde o usuário pode inserir a geometria que deseja, tendo como opção inserir uma figura pré-definida que pode ser um triângulo, cone, esfera e cilindro ou um arquivo STL. Após inserir a geometria o usuário deve configurar o domínio computacional até que esse esteja de acordo com a geometria inserida.

Com o domínio computacional configurado as propriedades da geometria podem ser editadas, com as opções de cadastrar um novo material, inserir material e modificar as dimensões, neste caso se a variável for *false* retorna para editar propriedades, caso seja *true* fica na opção de modificar as dimensões.

Pode ser inserida também a fonte eletromagnética, se ainda não existir a fonte eletromagnética esta pode ser cadastrada e depois inserida a fonte.

No último estágio são inseridos os relatórios no pré-processamento, chegando a geração de malha no processamento, sendo realizada a simulação FDTD 3D para última fase que a exibição de relatórios fazendo parte do pós-processamento. Todas as etapas descritas são verificadas por meio de lógica de falso e verdadeiro, sendo utilizada a lógica de botões.

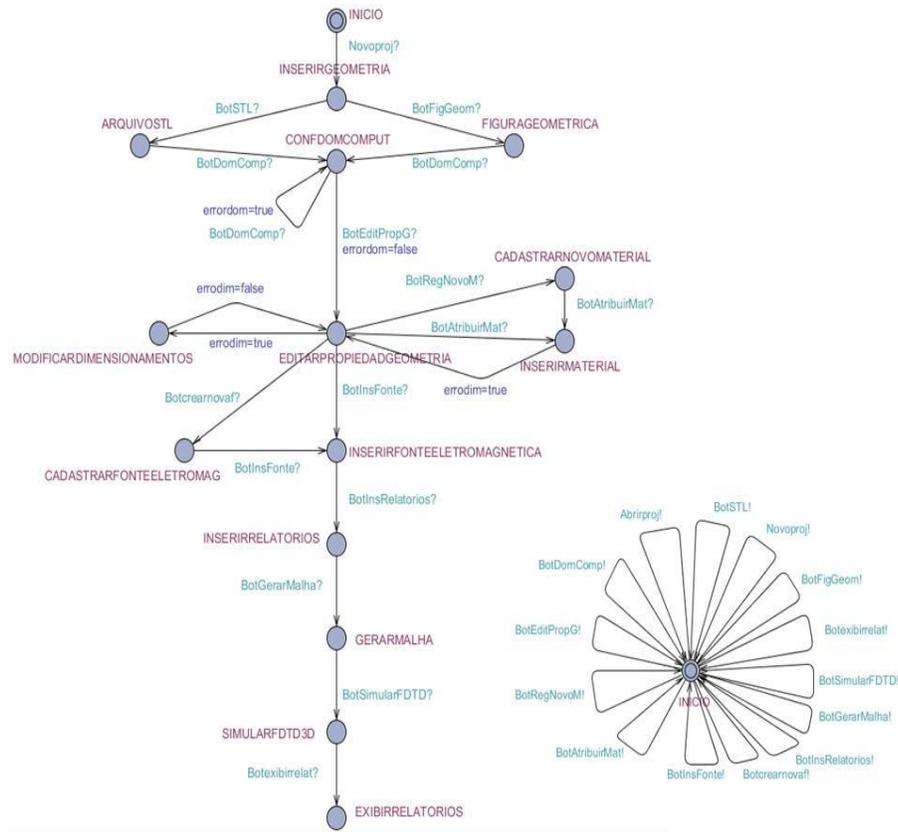


Figura 30 - Teste utilizando o UPPAAL

Após a modelagem, passa-se para a interface de simulação. Nesta é visualizada todas as instâncias e fluxos do processo.

Já na interface de verificação permite confirmar se determinadas propriedades estão presentes na representação do sistema. Os resultados podem ser verificados na Figura 31 interface de verificação, sendo utilizada a lógica *Computational Tree Logical* (CTL), com a seguinte notação:

$E\langle SEM1.INSERIRRELATORIOS$ - É possível que exista um caminho INSERIRRELATORIOS

$E\langle SEM1.EDITARPROPIEDADGEOMETRIA$ - É possível que exista um caminho EDITARPROPIEDADGEOMATRIA

$A[]SEM1.errordom=true$ - Inicialmente existe um caminho $errordom=true$

$A[]SEM1.errordim=true$ - Inicialmente existe um caminho $errordim=true$

A<>SEM1.errodim=true - Eventualmente pode ser errodim=true

E[]SEM1.errodim=true - É possível que exista inicialmente um caminho onde o errodim=true

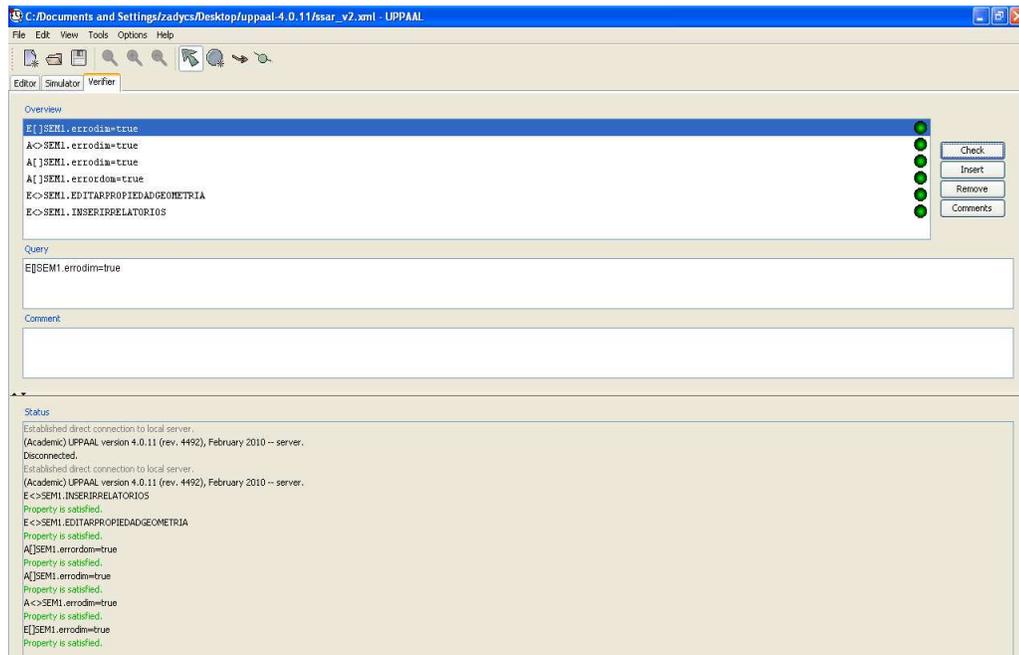


Figura 31 - Resultados dos testes com o UPPAAL

5.5 CONCLUSÕES DO CAPÍTULO

Com o uso da UPPAAL foi desenvolvido um ambiente de verificação de requisitos, no qual é possível estabelecer um relacionamento entre os elementos que compõe os requisitos propostos e, portanto, pode ser verificado formalmente cada um deles, onde para satisfação de todos em um mesmo modelo define a validação do modelo. Como verificado na Figura 31, todos os botões estão verdes o que significa que as propriedades foram corretamente verificadas pelo algoritmo de verificação.

Seguindo o checklist todas as funcionalidades do simulador foram verificadas, sendo utilizada a sequência proposta pelo software SEM-BR.

CAPÍTULO 6

CONCLUSÕES

Documentação, avaliação de interface e testes de validação e verificação de software são assuntos que antes eram deixados de lado, por demandarem tempo e gastos, sendo agora muito discutidos, como formas de agregar qualidade ao software.

Na documentação do simulador SEM-BR foi utilizada a técnica SysML, que tem como vantagem o uso do diagrama de requisitos e diagrama de blocos, além de possuir menos diagramas do que a UML.

Documentar todos os processos durante o desenvolvimento de software mostrou-se muito importante, pois caso seja necessário realizar alguma modificação ou possíveis manutenções no simulador com a documentação gerada será mais fácil.

A avaliação heurística realizada com a técnica de Nielsen mostrou os erros que existiam na interface, de acordo com a avaliação realizada com os dois grupos de quatro avaliadores, com o uso dos graus de severidade, tendo feito a média das heurísticas apresentadas, sendo corrigidas as heurísticas que apresentaram maior grau de severidade. Realizar a avaliação heurística com usuários especialistas e usuários finais foi importante para detectar a visão de cada um desses grupos em relação ao software e verificar o que devia ser corrigido.

Outra técnica de verificação de interface utilizada no simulador foi a semiótica computacional, técnica essa utilizada com o avaliador e os usuários para checar as expressões

presentes no simulador e verificar quais ícones eram mais apropriados a serem utilizados. Foi utilizada também a técnica de percurso cognitivo para comprovar se as imagens dos ícones e as expressões modificadas não distorciam seu sentido em relação ao simulador.

A última etapa da dissertação consistiu em fazer a validação e verificação do simulador, com o uso do checklist e da ferramenta UPPAAL, sendo checados os caminhos que o software deve percorrer, concluindo-se que tanto pelo uso do UPPAAL quanto pelo checklist o simulador percorre corretamente o caminho a ele destinado.

Após todas as técnicas utilizadas pode-se perceber que não são necessários grandes gastos e muito tempo para realizar as avaliações, com poucos usuários uma média de 4 usuários por avaliação e uma média de uma hora e meia são possíveis para realizar testes em geral no software, garantindo assim confiabilidade, usabilidade e satisfação dos usuários.

Por fim, conclui-se que esse conjunto de técnicas reduziu consideravelmente a quantidade de erros de funcionalidade do simulador eletromagnético e proporcionou um procedimento documental eficiente, tanto para os usuários quanto para os desenvolvedores, facilitando o uso e agilizando a manutenção e inserção de novos recursos de software.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT – Associação Brasileira de Normas Técnicas – NBR 9241-11. Requisitos Ergonômicos para trabalho de escritórios com computadores. Parte 11 – Orientações sobre usabilidade, (2002).

ARTISAN STUDIO. Disponível em: <http://www.atego.com/products/artisan-studio/> Acesso em: abr, (2011).

BALMELLI, L. Na overview of the Systems Modeling Language for product and systems development – Part 1: Requirements, use-case, and test-case modeling. IBM - developerWorks, (2006).

BARANAUSKAS, M. C. C., Rossler, F. and Oliveira, O.L. Uma abordagem semiótica à análise de interfaces: um estudo de caso, (1999).

BENITI, F. B. V., Zimmermann, A. P. Testware: ferramenta de planejamento e execução de casos de teste. Seminco, (2007).

BRUNELI, M.V.Q., A utilização de uma metodologia de teste no processo da melhoria da qualidade do software. Trabalho final de mestrado profissional em computação. IC/UNICAMP (2006).

CHAVEZ, C. V. F. G., Um enfoque baseado em modelos para o Design Orientado a Aspectos. Tese de doutorado. Pontifícia Universidade Católica do Rio de Janeiro, (2004).

CORREIA, J. Diagramas de Seqüência. Unified Modeling Language, (2006).

DIJKSTRA, E. The Humble Programmer. In: Communications of the ACM, v. 15, n. 10, pp.859-866, Nova York, NY, (1972).

FAUDOU, R., Marconato, B. Topcased 3.3.0 UML/SysML features tutorial. Topcased, (2011).

FAUST, R. Software como interpretação: uma estratégia de software centrada no registro linguístico dos usuários, (1995).

FIGUEROA, H. E. H., Hernández, M.F.G., Santos, C.H.S., Ambrosio, L.A., Freitas, I.J.F., Ng,J., Buck, R.M. Relatório 2: Avaliação de recursos computacionais a serem utilizados v.14. Simulação de emissão eletromagnética de cortes radiantes de RF acima de 450MHz. CPQD (2007).

FILHO. A. P., Scarpelini. S. Simulação: definição. (2007).

FREITAS, I.J.F, Simulador Eletromagnético em um ambiente de grades computacionais, Tese de mestrado, Departamento de Microonda e Óptica (DMO), Faculdade de Engenharia Elétrica e de Computação (FEEC), Universidade Estadual de Campinas (UNICAMP), (2010).

GAMA, M. Manual do Umbrello UML Modeller. Disponível em: http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-basics.html Acesso em: out, (2011).

GONÇALVES, M. S, Santos, C.H.S., Bordonalli A.C. and Hernández-Figueroa, H.E.; Parallel 3D Full-Time Domain Applied to Photonic Structures, aceito para publicação no IET Optoelectronics Journal, (2010).

GUEDES, G. T. A. UML 2 Guia prático. Novatec, (2004).

HANSEN, F. O. SysML- a modeling language for Systems Engineering, 2010. Disponível em: <http://staff.iha.dk/foh/Foredrag/SysML-SystemEngineering-DSFD-15-03-2010.pdf>. Acesso em: mar, (2011).

IBM. Model Driven Development with SysML, Jun, (2009). Disponível em: <http://www.incose.org/Chicagoland/docs/ModelDrivenDevelopmentwithSysML.pdf> Acesso em: mar, (2011).

IEEE Standards Association. P1528 – IEEE Draft Recommended Practice for Determining the Peak Spatial-Average Specific Absorption Rate (SAR) in the Human Head from Wireless Communications Devices: Measurement Techniques. Disponível em: <http://standards.ieee.org/develop/project/1528.html> Acesso em: nov, (2011).

LIMA, H.S. Cursos: Telecom e telemática. Equações de Maxwell, potencial magnético e equações de Campo. Capítulo 15. (2012).

- MACALAY, L. A., Requirements Engineering. Springer, London. (1996).
- MICHELAZZO, P. A. documentação de software, (2006). Disponível em: http://imasters.com.br/artigo/4371/gerencia/a_documentacao_de_software/ Acesso em: mar, (2011).
- MYERS, G. The Art of Software Testing. Wiley, (2004).
- NETO, R. M. UML 2.0 – Diagrama Caso de uso. Centro Universitário Nilton Lins – Tecnologia na Educação, (2011).
- NIELSEN, J. and Mack, R. L.; Usability inspection methods. John Wiley & Sons Inc. New York, 413 p, (1994).
- NIELSEN, J. Technology Transfer of Heuristic Evaluation and Usability Inspection. INTERACT'95 International Conference on Human-Computer Interaction (Lillehammer, Norway, June 27, 1995). Disponível por [http](http://www.useit.com/papers/heuristic/learning_inspection.html) em: www.useit.com/papers/heuristic/learning_inspection.html (1997).
- NIELSEN, J. Usability Engineering. New York: Academic Press. 26, (1993).
- NORFOLK, D. Artisan Studio. DoDAF – Tutorial version 7.0, (2009).
- OLIVEIRA, V. C., Silva, J. R. Ciência de Serviços na Gestão de Requisitos de Sistemas, (2009).
- OMG. OMG Systems Modeling Language. Disponível em: <http://www.omgsysml.org/> Acesso em: mar, (2011).
- PEIRCE, C.S. Semiótica. Tradução: Neto, J.T.C, (2005).
- ROSA, N. OMG SysML M6661, (2006).
- ROCHA, H. V. and Baranauskas, M. C.C. Design e avaliação de Interfaces Humano Computador. Campinas, SP: NIED-UNICAMP, (2003).
- SALGADO, L. C. C., Bim, S. A. e Souza, C. S. Comparação entre os métodos de avaliação de base cognitiva e semiótica, (2006).

SANTOS, C.H.S., Ambrosio, L.A. and Hernández-Figueroa, H.E.; IEEE-SBMO (2005), International Microwave and Optoelectronics Conference, IMOC Brasília, DF, Brasil, (2005).

SANTOS, C.H.S., Computação Bio-Inspirada e Paralela para a Análise de Estruturas Metamateriais em Microondas e Fotônica, Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC), da Universidade Estadual de Campinas (Unicamp), (2010).

SANTOS, C.H.S., Gonçalves, M. S. and Hernández-Figueroa, H.E.; Designing Novel Photonic Devices By Bio-Inspired Computing, aceito para publicação no IEEE Photonic Technology Letter, (2010).

SANTOS, C.H.S., Klecio, C., Gonçalves, M.S, and Hernández-Figueroa, H.E., Design of Photonic Devices Using Bio-Inspired Algorithms, International Conference on Microwave and Optoelectronics, SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), Belém, Brasil, (2009).

SILVA, C.J.: Melo S. S. "Tutorial de UPPAAL" Universidade da Beira interior, (2009).

SYSML.org. SysML Open Source Specification Project. Disponível em: www.sysml.org
Acesso em: out, 2011.

SOARES, M.S., Vrancken, J. Including SysML in the 4+1 View Model of Architecture for Software-Intensive Systems. 7 Annual Conference on Systems Engineering Research, (2009).

SOMMERVILLE, I.: Software Engineering Eighth Edition, Pearson Education Limited (2007).

TOPCASED. Disponível em: www.topcased.org Acesso em: mar, (2011).

Anexo

Documentação e Avaliação de interface

1. Avaliação heurística

A Avaliação heurística é realizada de duas formas com um grupo de usuários finais e um grupo de usuários especialistas.

Com o grupo de usuários finais foi explicado o que eram cada uma das heurísticas, como por exemplo, compatibilidade do sistema com o mundo real, ou seja, o quanto o sistema condiz com a realidade do usuário.

Associada a cada uma das dez heurísticas foi solicitado ao usuário relacioná-las com graus de severidade, caso a heurística compatibilidade com mundo real possuísse muitos erros seu grau de severidade seria 4, caso possuísse poucos erros 0 ou 1. Todas as heurísticas que possuíssem graus de severidade maior que 2 deveriam ser corrigidas.

Com o grupo de usuários especialistas também foi explicado o que eram cada uma das heurísticas de Nielsen e sua associação com os graus de severidade, sendo corrigidas as heurísticas com grau de severidade maior que 2.

Foram verificadas todas as interfaces, desde fase de entrada de dados até a saída de relatórios, em cada interface foram aplicadas as dez heurísticas de Nielsen.

De acordo com os critérios à direita, foi preenchida a coluna à esquerda com números de 0 a 4, sendo 4 a melhor nota e 0 a pior nota, para cada uma das interfaces. Depois para cada uma foi realizada uma média das avaliações dos usuários.

1

2

3

4

5

6

7

8

9

10

Critérios

- 1 - Compatibilidade do sistema com o mundo real
- 2 - Estética e design minimalista
- 3 - Consistência e padrões
- 4 - Prevenção de erros
- 5 - Ajuda e documentação
- 6 - Visibilidade do status do sistema
- 7 - Reconhecimento ao invés de relembração
- 8 - Flexibilidade e eficiência de uso
- 9 - Liberdade e controle do usuário
- 10 - Ajudar os usuários a reconhecer

2. Semiótica computacional

A semiótica computacional possuiu papel fundamental na escolha dos ícones que melhor se adequassem ao sistema e o entendimento destes ícones pelos usuários.

Foram apresentadas as três opções abaixo de ícone para o botão cancelar sendo escolhido o botão do meio para não ser confundido com o botão sair o primeiro botão das opções abaixo. Este é um importante exemplo de interpretação do significados do ícone e como ele pode induzir os usuários ao erro quando mau escolhidos.



3. Percurso cognitivo

O percurso cognitivo foi realizado apenas com o avaliador, com a finalidade de percorrer todas as tarefas que provavelmente os usuários iriam percorrer, com a preocupação de verificar se as tarefas fossem todas realizadas corretamente.

Ao passo que ia percorrendo as tarefas algumas perguntas deviam ser respondidas como:

- A ação desejada pelo usuário está presente no software?
- O usuário irá interpretar corretamente as respostas do sistema?
- O usuário irá associar as ações corretas ao que pretende fazer?

4. SysML e a modelagem de sistemas

Para realizar a documentação é necessário que sejam levantados todos os requisitos do software, ou seja, os requisitos são os responsáveis por descreverem tudo o que o cliente necessita, descrevendo e especificando o sistema.

Após o levantamento de requisitos junto ao cliente o analista de sistemas irá montar uma tabela com os requisitos e sua descrição e definir os requisitos principais do sistema, ou seja, as principais funcionalidades que o sistema possui.

Depois de selecionar os principais requisitos o analista poderá montar o diagrama de requisitos utilizando ferramentas como a Artisan Studio.

Em relação ao diagrama de blocos, este diagrama busca mostrar uma visão geral do software, todos os módulos que possuem o sistema, sendo descritas todas as relações entre cada subsistema e o fluxo de informação.