

HTMaster: uma ferramenta para apoio ao ensino e aprendizagem de Modelos Hierárquicos de Tarefas

Márcia de Borba Campos

Faculdade de Informática,
Pontifícia Universidade Católica de Rio Grande do
Sul (PUCRS)
Av. Ipiranga, 6681, Porto Alegre,
Rio Grande do Sul, Brasil
+55 51 3320-3558
marcia.campos@pucrs.br

Jaime Sánchez

Departamento de Ciencias de la Computación y
Centro de Investigación Avanzada en Educación
(CIAE), Universidad de Chile
Blanco Encalada 2120
Santiago, Chile
+56-2 978 0500
jsanchez@dcc.uchile.cl

ABSTRACT

The purpose of this study was to design, develop and evaluate HTMaster, a tool to make hierarchical tasks models that represent what users can do and how they do when using an interactive system. HTMaster, uses the Adapted Hierarchical Task Model to represent the hierarchical structure of tasks as well as sequence structures and iteration, alternative tasks, independent of order, optional and ubiquitous. HTMaster allows for assisting the teaching and learning of hierarchical task models in Human Computer Interaction classes, empowering the user to analyze and reflect on the development of the structures of tasks models by using a diagrams validation module.

RESUMO

O objetivo deste estudo foi projetar, desenvolver e avaliar uma ferramenta para criação de modelos hierárquicos de tarefas, que representam o que os usuários podem fazer ao utilizar um sistema interativo e como o fazem. Denominada de HTMaster, utiliza o *Modelo Hierárquico de Tarefas Adaptado* para representar a estrutura hierárquica das tarefas e as estruturas de sequência e iteração, tarefas alternativas, independentes de ordem, opcionais e ubíquas. Esta ferramenta também foi criada para que pudesse ser utilizada como recurso pedagógico ao ensino e aprendizagem de modelos hierárquicos de tarefas em aulas de Interação Humano-Computador. Para tal, possibilita que usuário analise e reflita sobre a elaboração das estruturas, a partir de um módulo de validação de diagramas.

Descritor de Categorias e Assuntos

H.5.0 [Information Interfaces and Presentation (e.g., HCI)]: General. H.5.2 [Information Interfaces and Presentation (e.g., HCI)]: user Interfaces - Theory and methods.

Termos Gerais

Design, Documentation, Human Factors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

XXXXXXXXXXXXX - As informações serão preenchidas no processo de edição dos Anais.

Palavras Chaves

Interação Humano-Computador (IHC), Educação em IHC, Modelo Hierárquico de Tarefas.

1. INTRODUÇÃO

Interação Humano-Computador (IHC) é uma área de conhecimento interdisciplinar, que se preocupa em entender as características, necessidades e objetivos dos usuários de um sistema computacional interativo para uso humano [2, 8, 13, 14, 16]. Tal foco é necessário para que possam ser projetados modelos de interface e de interação que atendam aos objetivos desses usuários em seu contexto de uso. Sánchez e Sáenz [14] destacam que o objetivo da área de IHC é gerar produtos que sejam usáveis, entendíveis, seguros, funcionais e úteis, dentre outras características. IHC preocupa-se, pois, com a qualidade de uso dos sistemas computacionais e com o impacto que esses sistemas podem ocasionar na vida de seus (futuros) usuários. Por isso, seu estudo é fundamental aos alunos de cursos na área de Computação, que projetam, desenvolvem e avaliam sistemas computacionais.

O Parecer do Conselho Nacional de Educação/Câmara de Educação Superior do Ministério de Educação do Brasil, CNE/CES nº 136/2012, aprovado em fevereiro de 2012, apresenta a área de IHC como um dos conteúdos curriculares da formação tecnológica e básica para os Cursos de Bacharelado em Ciência da Computação, Engenharia de Computação, Engenharia de Software, Sistemas de Informação e para a Licenciatura em Computação [4]. Essa perspectiva de currículo, apesar de já ter sido apresentada em outros documentos [15], não tem sido implantada com facilidade.

O Workshop sobre Ensino de IHC (WEIHC) tem como propósito discutir o ensino de IHC no Brasil no que se refere à perspectiva curricular e às práticas pedagógicas. Trabalhos apresentados em diferentes edições desse evento relatam experiências e discutem, dentre outras questões, a integração de IHC com outras disciplinas, disciplinas avançadas de IHC e a integração de IHC com o mercado de trabalho. Nesses debates, também surgem questões relacionadas às estratégias de ensino de IHC no qual comumente são relatadas dificuldades referentes à ordem dos conteúdos, à baixa carga horária referente às disciplinas de IHC e/ou excesso de conteúdo para a(s) disciplina(s). Bim e Boscaroli [3] discutem sobre uma pesquisa exploratória, realizada em 2009, sobre o ensino de IHC no Brasil. Segundo os autores, essa pesquisa revelou que a maioria das disciplinas de IHC em cursos

da área de Computação é introdutória. O desafio no oferecimento de disciplinas de teor intermediário também é apresentado por [9], que discute estratégias para trabalhar conteúdos de IHC de forma que os alunos possam “pensar” em IHC.

Muitas dessas dificuldades são similares às que vivenciamos na Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul (FACIN/PUCRS) onde é oferecida uma única disciplina na área de IHC. A disciplina, que é intitulada *Projeto de Interfaces*, é oferecida no 4º nível do curso de Sistemas de Informação e no 5º nível do curso de Ciência da Computação. Possui a seguinte ementa:

Introdução aos conceitos básicos de interação homem-computador. Avaliação de sistemas interativos. Estudo de processo de projeto de interfaces. Modelagem de interação. Fundamentação sobre padrões e diretrizes de Projeto. Comparação entre estilos de interação.

A organização dos conteúdos da disciplina se orienta no programa de IHC para os cursos de graduação em Computação, discutido por [3]. Na disciplina em questão, inclui as seguintes unidades de ensino, que devem ser previstas para uma totalidade de 30 horas-aula semestrais:

- Introdução à Interação Humano-Computador
- Avaliação de IHC
- Modelagem em IHC
- Projeto de interação com o usuário

Essa limitação de carga horária resultou no experimento de diferentes estratégias de ensino e, inicialmente, foi priorizada a modalidade de aula expositiva baseada na apresentação de conteúdos pelos professores para que os conteúdos previstos no plano de ensino pudessem ser apresentados no curto espaço de tempo dos currículos. O descontentamento e desmotivação de professores e alunos fez com que se planejassem outras formas de trabalho que fossem condizentes com a área de IHC e buscasse envolver os alunos na apresentação, análise, e discussão de teorias baseadas em trabalho colaborativo, e no planejamento, avaliação e prototipação de sistemas interativos, ou no redesenho de sistemas já existentes. Atualmente, para que se possa balancear adequadamente o ensino de fundamentos de IHC e a prática com tecnologias interativas atuais, os professores e alunos apresentam problemas de interação humano-computador que devem ser atendidos por meio de soluções criativas e inovadoras, que levem a aplicação de conhecimentos aprendidos em aula. Como resultado, se busca uma apropriação do conhecimento pelo aluno e uma melhor relação da teoria com a prática. Neste processo de construção e aplicação de conceitos, os alunos se tornam mais críticos e exigentes até com relação aos instrumentos e ferramentas de apoio que são utilizados nas aulas práticas, e pedem por soluções tecnológicas que possam aperfeiçoar suas atividades.

Dessas questões, surgiu a demanda por uma ferramenta que pudesse ser utilizada para apoiar a construção de Modelos Hierárquicos de Tarefas Adaptado, conteúdo que integra a unidade de ensino sobre Modelagem em IHC.

O propósito deste trabalho é apresentar o projeto, desenvolvimento e avaliação dessa ferramenta para criação de modelos de tarefas para representar o que os usuários podem fazer ao utilizar um sistema interativo, e como o fazem. Intitulada como HTMaster, a ferramenta utiliza o Modelo Hierárquico de Tarefas

Adaptado para representar a estrutura hierárquica das tarefas e estruturas de sequência e iteração, tarefas alternativas, independentes de ordem, opcionais e ubíquas. Foi desenvolvido para auxiliar nesse processo de modelagem porque havia retrabalho na criação desses modelos a partir de editores gerais de diagramas. Esse processo resultava em perda de tempo e em diagramas que nem sempre atendiam à notação do modelo. O HTMaster permite que o usuário analise e reflita sobre a elaboração das estruturas dos modelos hierárquicos de tarefas, a partir de um módulo de validação de diagramas.

2. MODELO DE TAREFAS

A área de IHC possui diferentes métodos e modelos para tratar a complexidade do projeto de sistemas interativos computacionais, ou seja, para representar as atividades envolvidas no design desses sistemas.

O processo de design inclui a fase de elicitação (para identificar os requisitos do usuário), análise e modelagem de tarefas (para compreender as atividades do usuário do ponto de vista dele próprio), modelagem de interação e prototipação, em um processo que é iterativo. A modelagem de tarefas visa à formalização, estruturando cada objetivo do usuário de forma a explorar as diferentes estratégias que o usuário poderá seguir [2].

Existem diferentes tipos de modelos de tarefas, sendo que há conceitos e relacionamentos que são mais comuns, tais como a decomposição de tarefas, tipicamente descritas de forma hierárquica, e os relacionamentos causais ou temporais, que indicam a ordem em que as subtarefas devem ser executadas.

O método de *Hierarchical Task Analysis* (HTA) foi desenvolvido na década de 1960 por Annett e seus colaboradores [1] como um processo para decompor tarefas em subtarefas a qualquer nível desejado de detalhes. Diaper e Johnson [10] descreveram o método *Task Analysis for Knowledge Descriptions* (TAKD) enquanto Payne e Green's, apresentado por [10], descreveram o *Task Action Grammar* (TAG), que focalizava mais a pessoa ao invés das tarefas. Card et al., discutidos em [2] propuseram um conjunto de modelos denominado de família *GOMS – Goals, Operators, Methods, and Selection Rules*, que modela a performance do usuário no sistema.

Conforme já citado, na disciplina *Projeto de Interfaces* é utilizado o *Modelo de Tarefas Adaptado* para representar as funções do sistema, a partir dos requisitos que foram definidos pelos potenciais usuários. Esse modelo foi proposto por [7], conforme segue:

Para o modelo de tarefas contemplar elementos adequados à modelagem das tarefas, sem incluir detalhes da interação ou da interface, foi feito (...) uma simples adaptação à structure chart notation. Esta notação representa a decomposição hierárquica de funções em seus componentes funcionais, indicando algumas relações temporais entre elas (por exemplo, funções sequenciais, alternativas e iterativas).

Desta forma, a partir desse modelo, é possível representar os objetivos do usuário ao utilizar um sistema, definindo as metas, as atividades, que compõem cada objetivo e sua decomposição hierárquica em tarefas. Assim, conforme [7], o modelo de tarefas reflete como o usuário trabalha, evitando-se focalizar o ambiente ou uma plataforma tecnológica específica. Seguem os elementos desse modelo.

Para exemplificar esse método, apresentamos as Figuras 1 e 2, que tratam de um sistema hipotético no qual o usuário pode se logar

ou solicitar que seus dados sejam enviados a ele, bem como para solicitar alteração de seus dados de login. Esses diagramas foram construídos utilizando a ferramenta HTMaster.

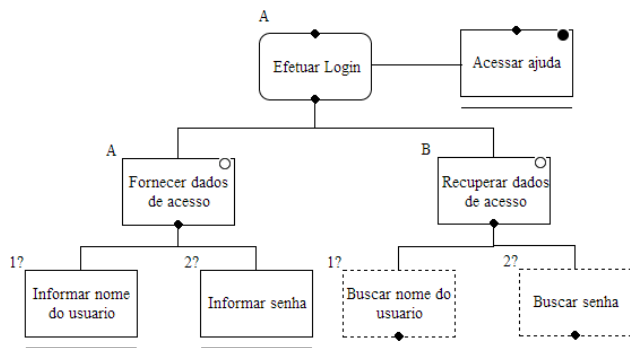


Figura 1. MHT da meta Efetuar Login

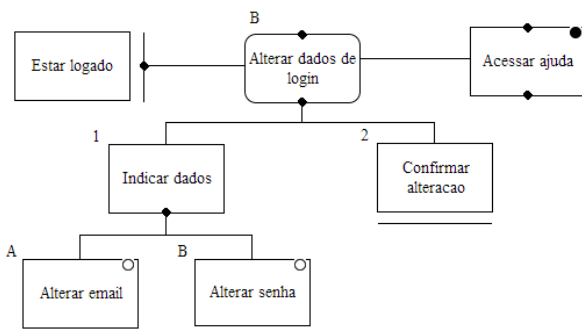


Figura 2. MHT da meta Alterar dados de login

Para um melhor entendimento, segue a descrição dos principais elementos do MHT, e que estão previstos HTMaster.

- Meta: indica o que o usuário deseja fazer. É representada por um retângulo com bordas arredondadas. Na Figura 1 temos a meta A – Efetuar Login, e na Figura 2 a meta B – Alterar dados de login.
- Tarefas e subtarefas: em um modelo hierárquico, as tarefas podem ser subdividas em subtarefas, e assim por diante. São representadas por retângulos contendo o nome da (sub) tarefa com marcações especiais para indicar o tipo de estrutura que estão associadas: estrutura sequencial (na Figura 2, para alterar dados de login, o usuário primeiro deve Indicar dados e depois Confirma alterações, nessa ordem), estrutura alternativa (na Figura 1, Fornecer dados de acesso ou Recuperar dados de acesso; na Figura 2, Alterar email ou Alterar senha), estrutura de tarefas independente de ordem (na Figura 1, o usuário pode Informar nome do usuário e depois Informar senha, ou vice-versa, desde que forneça essas informações), tarefas ubíquas que podem ser executadas a qualquer momento (na Figura 1 e 2, Acessar ajuda), tarefas opcionais (na Figura 1, Buscar nome do usuário e Buscar senha), operadores (todas as tarefas que possuem uma linha horizontal inferior). As pré-condições também são representadas nos modelos (na Figura 2, para alterar seus dados, o usuário precisa Estar logado).

Nos MHT, após a modelagem diagramática das tarefas, podem-se associar signos e rupturas de comunicação, que possam ocorrer

durante a realização da tarefa. Os signos são acompanhados da expressão SIGNOS e quando são dados de entrada devem vir acompanhados de um ponto de interrogação e quando são dados de saída devem utilizar um ponto de exclamação. Ainda, para instâncias de um mesmo tipo pode-se utilizar o construto conjunto, que é representado pelo nome_do_signo e a cardinalidade, por exemplo, 1..n [2, 7].

No que se refere às rupturas de comunicação, em um MHT podem estar descritos o apoio à prevenção de erros e o tratamento de erro propriamente dito. Apresentamos na Tabela 1 algumas sugestões de tratamento e prevenção para os modelos representados na Figura 1 e 2, sendo que as tarefas são identificadas por sua notação. Por exemplo, A.A se refere à tarefa Fornecer dados de acesso (Figura 1), B.1 é Indicar dados (Figura 2) e B.1.A é a tarefa Alterar email (Figura 2).

Tabela 1. Exemplos de prevenção e recuperação de erro

Tratamento apoiado	A.A Fornecer dados de acesso SIGNOS: usuário.nome?, usuário.senha? TRATAMENTO APOIADO: falta de informações obrigatórias TRATAMENTO APOIADO: login inexistente TRATAMENTO APOIADO: senha inválida
Prevenção ativa	B.1 Indicar dados SIGNOS: usuário.email?, usuário.senha? PREVENÇÃO ATIVA: somente o usuário altera seus dados
Prevenção apoiada	B.1.A Alterar email SIGNOS: usuário.email_atual?, usuário.email_novo? PREVENÇÃO APOIADA: os emails digitados são iguais B.1.B Alterar senha SIGNOS: usuário.senha_atual?, usuário.senha_nova? PREVENÇÃO APOIADA: as senhas digitadas são iguais

Essas são as representações e modelo utilizados na Unidade de ensino Modelagem de IHC da respectiva disciplina para especificar o design de sistemas interativos.

3. EDITORES DE DIAGRAMAS

No tópico relacionado à Modelagem de IHC, os alunos devem fazer uso do MHT para descrever as tarefas mais complexas dos sistemas que estão propondo. Para isso, utilizam uma combinação de aplicativos gráficos, que resultam em problemas na notação do modelo e em tempo extra para corrigir/refinar as representações gráficas para que atendam ao formalismo do MHT.

A partir dessa demanda, e de dificuldades dos alunos na criação de modelos de tarefas, é que foi projetado e desenvolvido o HTMaster.

Para definir os requisitos do HTMaster foi realizado um estudo sobre ferramentas de edição utilizadas pelos alunos das turmas de

IHC de 2012/1. Foram analisados os recursos gerais e os esforços necessários para a construção dos modelos. A seguir, uma breve descrição dos editores mais utilizados pelos alunos e de sua (não) adequação à construção de MHT.

- *MS Paint*: é possível representar metas, tarefas, tipos de estruturas das tarefas, tarefas opcionais, tarefas ubíquas, pré-condições e reuso de tarefas e metas. Não há validação quanto à estrutura que está sendo construída e, por isso, podem ser desenvolvidos modelos com erros. Ainda, independente do modelo de tarefa, o trabalho operacional se torna delicado e oneroso, demandando paciência para desenhar os elementos e incluir legendas nas metas e tarefas, e outras informações textuais. Considerando os modelos das Figuras 1 e 2, as ligações entre a meta e as tarefas, e entre as tarefas, devem ser feitas utilizando a opção *Linha*, uma a uma. Ainda, para representar uma tarefa opcional, é necessário utilizar a opção *Retângulo* e depois utilizar o recurso *Borracha* para apagar partes das bordas para simular o efeito de pontilhado. Para representar o símbolo ● de tarefa ubíqua, o usuário precisa observar que pode utilizar o pincel no formato oval. E para cada notação de estrutura (1,2, ..., 1?, 2?, ..., A, B, ..., *) é preciso criar caixas de texto. Não há validação sobre a estrutura construída.
- *Microsoft Word*: também exige trabalho extra para posicionar cada tarefa e subtarefa no diagrama bem como para ligar cada elemento e incluir as legendas e outras informações textuais. Também não há validação na criação do diagrama. Podem ser utilizados recursos de símbolos e formas para as notações.
- *Microsoft Visio*: possui um recurso denominado *Conector*, que ajuda a ligar um componente a outro, evitando conexões mal estruturadas. Podem ser utilizados elementos de *Formas Básicas* para a construção dos Exemplos 1 e 2. Para representar uma tarefa do tipo *Operador* é necessário criar um retângulo e acrescentar uma linha embaixo do retângulo, como também ocorre no uso do *Paint* e do *Word*. No que se refere à notação das estruturas hierárquicas também é necessário utilizar o recurso de caixa de texto sem borda para representar os símbolos e as indicações ○ e ● podem ser feitas com o recurso de *Elipse*.
- *Astah*: exige conhecimento sobre os diferentes diagramas desse software para que se possam combinar os objetos. Tal uso não é facilitado porque, por exemplo, as tarefas precisam ser criadas a partir de caixas de texto com borda, as ligações entre a meta e as tarefas, e entre as tarefas, devem ser feitas com o recurso de linha reta, fazendo uma a uma. Isso é necessário porque, como o aplicativo faz validação dos elementos de origem e destino, o diagrama automático não é aceito. No que se refere à notação das estruturas hierárquicas, também é necessário utilizar o recurso de caixa de texto sem borda para representar cada número, símbolo ou letra. A indicação ○ pode ser feita por meio da letra “O” e ● com o símbolo que representa o nodo inicial do diagrama de atividades.

Além desses editores, os alunos fizeram uso de ferramentas on-line para elaborar os diagramas hierárquicos, mas que igualmente não atenderam aos requisitos do MHT. Foram utilizadas:

- *Text2Mindmap* <http://www.text2mindmap.com/>
- *Giffy* <http://www.giffy.com/giffy/#templateId=blank&signup=1>

- *Grapholite* <https://grapholite.com/Designer>
- *FlockDraw* <http://flockdraw.com/1h1xwq>
- *Diagramly* <http://www.diagram.ly>
- *Creately* <http://creately.com>
- *Bubbl.us* <https://bubbl.us>
- *WiseMapping* <http://www.wisemapping.com/c/try.htm>

A partir dessa descrição, pode-se perceber que os editores comumente utilizados pelos alunos não são adequados e acabam por desmotivá-los a utilizar representações e modelos de IHC.

Por outro lado, sabemos que existem ferramentas específicas de edição e de análise de modelos tais como *CTTE - CTT Environment* [6], e de modelo de interação como *MOLIC* [12]. Entretanto, por demandarem conteúdos além dos previstos no plano da disciplina de IHC em questão, e por não haver carga horária disponível para tal, esses recursos não são apresentados pelos professores bem como, até o momento, seu uso ainda não foi sugerido/solicitado pelos alunos.

4. HTMaster: PARA ALÉM DE UM EDITOR DE CONSTRUÇÃO DE DIAGRAMAS

4.1. Projeto e desenvolvimento

O HTMaster foi construído para apoiar o processo de ensino e de aprendizagem de Modelos Hierárquicos de Tarefas, facilitando a construção de modelos de tarefas e seu entendimento. Esta ferramenta encontra-se disponível na Internet, no endereço <http://htmaster.x10.mx/>

Para o desenvolvimento do HTMaster foram utilizadas as seguintes ferramentas:

- *JavaScript*: foi utilizada a biblioteca *Draw2d*, que permite a criação de desenhos e diagramas. Inclui funcionalidades como inclusão de elementos, conexões entre elementos, desfazer e refazer, entre outras, além de uma *API* estruturada e de fácil adaptação.
- *jQuery*: simplificou a interação entre o documento *HTML* e a manipulação de eventos para o desenvolvimento web em si. É a principal biblioteca utilizada pelo *Draw2d*.
- *PHP*: foi utilizado para gerar os arquivos para impressão e salvamento.
- *HTML*: está presente em todas as páginas, fornecendo a visualização dos modelos, exibição dos menus, botões e mensagens de validação.

Durante a implementação e teste de funcionalidades do HTMaster também foram utilizados:

- *USBWebserver*: combina softwares de servidor Web como o *Apache* (servidor *PHP*), sendo possível desenvolver e executar sites *PHP* sem necessidade de instalação e complexas configurações.
- *TortoiseSVN*: é um cliente de versionamento (*SVN*) de arquivos, implementado como uma extensão de *shell* do Windows. Foi utilizado unido ao *Google Code*.

- *Google Code*: é uma ferramenta on-line que o Google oferece aos desenvolvedores para disponibilizar um ambiente de desenvolvimento para projetos. Nesse ambiente, o “dono” do projeto pode controlar os acessos dos usuários, utilizar versionadores de código e controlar mudanças no projeto, sendo possível visualizá-las na própria página do projeto junto à informação de quem a realizou. Disponibiliza, ainda, um sistema de *wiki* e uma área de *upload* e *downloads*.

Para atender aos critérios de usabilidade, buscou-se uma interface simples, de fácil uso e aprendizado, que permitisse aos usuários acessar os recursos do sistema com o menor número de ações possíveis. No que se refere à simplicidade, foram disponibilizadas ações conforme o contexto de uso em que o usuário se encontra no sistema. Assim, as ações relacionadas a um determinado elemento do MHT são visíveis quando tal elemento é selecionado e os menus *pop-up* são ativados conforme o tipo de elemento que está selecionado. Tal ação auxilia na segurança de uso da ferramenta, e atende ao quesito de prevenção de erros, evitando que alguns problemas de sintaxe na elaboração do diagrama possam ocorrer.

O HTMaster possui um mecanismo de validação que indica, textualmente, problemas que possam ser identificados na construção do diagrama. Mas o modelo não é modificado sem a ação direta do usuário.

No que se refere à interface, a tela inicial do HTMaster possui três áreas principais (área superior: menu, área lateral: validação e área central: construção e edição do diagrama), que estão ilustradas na Figura 3, e uma caixa de ferramentas, que está representada na Figura 4. O menu superior do HTMaster possui 5 áreas funcionais, que, para fins de explicação, estão denominadas de grupos, e podem ser identificadas na Figura 5.

- O Grupo 1 possui recursos de arquivo: fazer um novo diagrama, abrir um diagrama já existente, salvar e imprimir.
- O Grupo 2 está relacionado à validação do modelo de tarefas, que está sendo editado, e que pode indicar erros, alertas, dicas ou sucesso.
- O Grupo 3 permite ao usuário limpar a área de construção e é solicitada confirmação por parte do usuário, como forma de prevenção apoiada.
- O Grupo 4 permite aumentar ou diminuir a área de construção.
- O Grupo 5 acessa o sistema de ajuda e informações sobre a equipe de desenvolvimento do HTMaster. Esse inclui conceitos teóricos relacionados ao MHT e informações de como melhor utilizar a ferramenta.

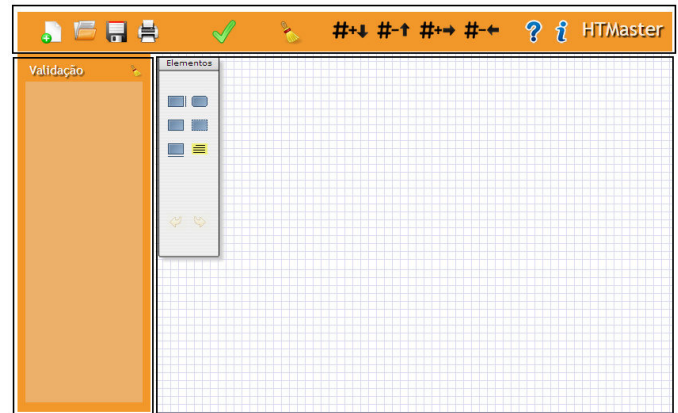


Figura 3. HTMaster: interface principal

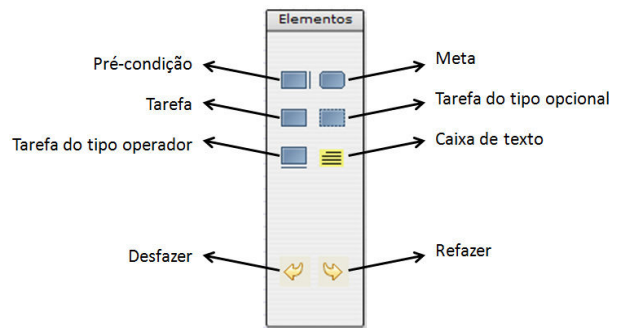


Figura 4. HTMaster: caixa de ferramentas



Figura 5. HTMaster: grupos do menu

O processo de validação no HTMaster pode indicar violação no diagrama ou alertas. As violações estão relacionadas às mensagens de tratamento apoiado enquanto os alertas solicitam que o usuário verifique se uma determinada situação demanda alteração ou não no diagrama. Está mais direcionado à prevenção apoiada. A Tabela 2 ilustra exemplos de mensagens de validação de diagramas.

Tabela 2. HTMaster: exemplos de mensagens do processo de validação

Mensagem	Tipo	Situação de ocorrência
Uma tarefa não pode ter mais de um pai.	Erro	Há mais de uma ligação de chegada. Ligações de entrada de pré-condições são desconsideradas neste caso.
O seu diagrama possui N elementos sem conexões.	Erro	Existe pelo menos um elemento sem ligação. A variável N é substituída pelo número de elementos sem conexão.
O seu diagrama possui N elementos sem texto.	Erro	Existe pelo menos um elemento sem texto informado. A variável N é substituída pelo número de elementos sem texto.
O seu diagrama de tarefas possui um ou mais ciclos.	Erro	Há dois ou mais elementos estão com ligações cíclicas.
O seu diagrama de tarefas deve possuir pelo menos 1 (uma) meta.	Erro	O modelo está sem meta.
Uma meta deve ser pai de tarefa(s).	Erro	Uma tarefa tem como dependente uma meta.
O seu diagrama possui mais de uma meta.	Alerta	É um alerta, pois o usuário pode estar utilizando o HTMaster para construir Diagramas Hierárquicos de Metas que são constituídos por um conjunto de metas.
Na meta, lembre-se de utilizar a mesma letra informada no Diagrama Hierárquico de Metas, caso esse tenha sido construído.	Dica	Geralmente, um Modelo de Tarefa indica a meta a que se refere, mantendo a mesma notação do Diagrama Hierárquico de Metas, caso esse tenha sido construído. Desta forma, essa mensagem é sempre mostrada para lembrar ao usuário que deve utilizar a mesma notação definida no Diagrama de Metas.
Nenhum erro ou alerta encontrado.	Sucesso	Não há erros ou alertas identificados pelo HTMaster.

Um modelo de tarefas é construído a partir dos elementos da caixa de ferramentas (Figura 4), disponível na área de construção. O usuário seleciona o ícone da meta, das tarefas e descreve as informações textuais, deslocando os elementos na área de construção. Se necessário, podem ser utilizados os recursos de desfazer e refazer ações, que, como um mecanismo de prevenção ativa, são habilitados quando tais ações podem ser executadas.

As legendas de meta e de tarefas são incluídas por meio de campos e os tipos de tarefas são selecionados a partir de menus pop-up, que são contextuais (Figuras 6, 7 e 8). Desta forma, suas opções variam conforme o elemento que está selecionado.

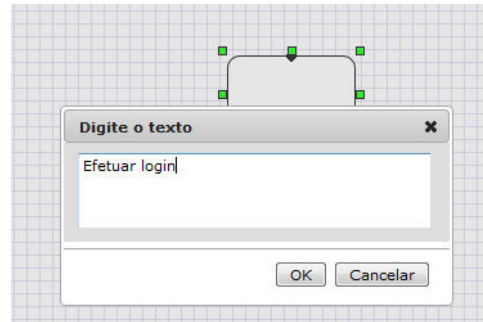


Figura 6. HTMaster: incluir legenda

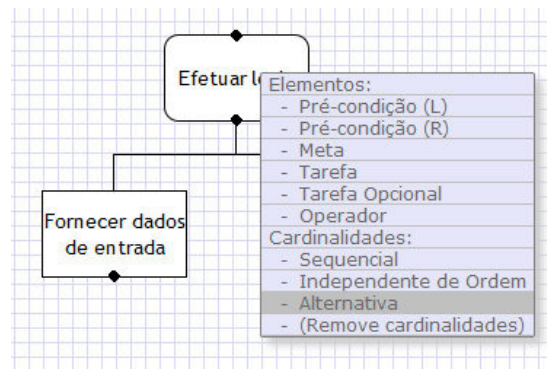


Figura 7. Definir estrutura das tarefas

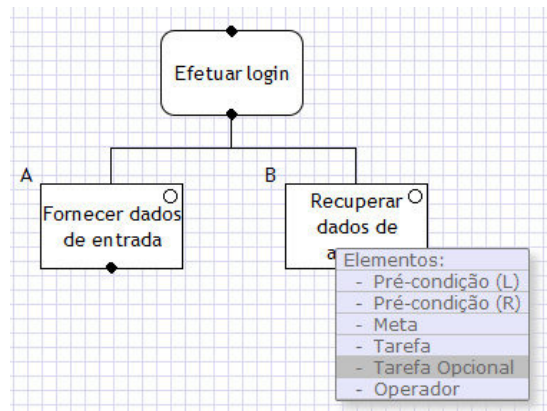


Figura 8. Definir tipo de (sub)tarefa

4.2. Avaliação

Antes do HTMaster ser incluído na referida disciplina de IHC, foram realizados testes com alguns alunos durante 2012/2. Participaram 13 alunos do sexo masculino. Com relação ao conhecimento e uso de MHT, os participantes se consideraram: Iniciantes (5), Intermediário (7), Avançado (1).

Inicialmente, foi dada uma aula teórica sobre MHT na qual se discutiu sobre os tipos de estrutura, definição dos signos e tratamento e prevenção de erro, a partir de exemplos. Na semana seguinte, os alunos fizeram exercícios de construção de diagramas utilizando softwares de sua escolha, e que foram incluídos na seção 3 deste artigo.

Na terceira semana, foi realizado o experimento com o uso do HTMaster. A aula ocorreu no laboratório de Informática, e durou aproximadamente 45 minutos. Os alunos receberam a descrição de uma mesma parte de um sistema que deveria ser representada por meio de modelos de tarefas e que já haviam trabalhado nas aulas anteriores. Após o teste, houve uma discussão sobre os modelos construídos e o preenchimento de um questionário on-line, que continha 13 perguntas abertas. Seguem alguns resultados:

- Você considera que o HTMaster pode ser utilizado para motivar o aprendizado do usuário para a construção de Modelos de Tarefas? Justifique sua resposta.
 - Os 13 participantes (100%) responderam que sim. Dentre as justificativas, destacamos:
 - *Facilita a criação do modelo de tarefas ao mesmo tempo em que ensina o usuário sobre alguns pontos do formalismo.*
 - *Minimiza o trabalho do aluno para o desenvolvimento do diagrama liberando assim mais tempo para o que realmente interessa que é o desenvolvimento das ideias.*
 - *A facilidade de construir modelos usando uma interface gráfica mais simples ajuda a motivar*
 - *A ferramenta é muito útil no auxílio a modelos de tarefa, portanto ao ensinar o modelo aos alunos, a ferramenta deixa a matéria mais palpável.*
 - *A proposta de modelagem via browser e online facilita o processo de construção. Também se pode citar a agilidade e facilidade do HTMaster como pontos fortes de incentivo ao uso.*
 - *O motivo é simples: uma aplicação que proporciona as ferramentas exatas.*
- Você considera que o HTMaster pode fomentar a reflexão sobre os tipos de estruturas das tarefas de um Modelo de Tarefas?
 - Apesar dos depoimentos citados acima, 9 (69.2%) participantes responderam sim, 1 (7.6%) respondeu parcialmente e 3 (23%) ainda não tinham como opinar.
- Você considera que o HTMaster pode ser utilizado como apoio à revisão e/ou ao reforço sobre conteúdos relacionados a Modelos de Tarefas?
 - Das respostas, 12 (92.3%) participantes responderam que sim e 1 (7.6%) respondeu que não, e justificou que *é melhor aprender da forma difícil.*
- O sistema de ajuda ou help é adequado, ou seja, explica as dúvidas do usuário adequadamente?
 - A maioria dos participantes indica que o sistema de ajuda deve ser melhorado. Tivemos 5 (38.4%) participantes que responderam que o sistema de ajuda é adequado, 6

(46.1%) responderam que é parcial e 2 (15.3%) responderam não a esta pergunta.

- Você considera que o tipo de interface utilizada pelo HTMaster é adequado ao público alvo a que se destina?
 - A maioria considera que sim. Houve 11 (84.6%) participantes que responderam que sim e 2 (15.3%) que responderam que é parcial.
- Você considera que é fácil de aprender a usar o HTMaster?
 - Apesar da maioria dos participantes considerar que a interface é adequada, 8 (61.5%) participantes responderam que é fácil de aprender a usar o HTMaster enquanto 5 (38.4%) responderam que é parcial.
- Você considera que o HTMaster é fácil de utilizar?
 - A maioria indica que sim, sendo a resposta de 9 (69.2%) participantes enquanto 4 (30.7%) indicaram que é parcial, e sugeriram melhorias no sistema de ajuda.
- Você recomendaria o uso do HTMaster para a construção de Modelos de Tarefas e/ou para o ensino desse conteúdo?
 - A maioria indica que sim, sendo que foi a resposta de 12 (92.3%) participantes. Um (7.6%) participante respondeu que sua recomendação seria parcial. Dentre as respostas positivas, destacamos:
 - *A facilidade de utilização.*
 - *Pela praticidade e objetividade da ferramenta.*
 - *A ferramenta para uso didático é muito boa e bem simples.*
 - *Aprendi bastante durante o uso do HTMaster.*

Analisando os resultados apresentados é possível observar que o HTMaster obteve uma avaliação com média de 75.9% em respostas positivas, 24.2% em respostas que atenderia parcialmente e 11.4% em respostas negativas nos itens apresentados. Isso demonstra que a ferramenta atendeu positivamente às expectativas, e que pode ser útil para auxiliar professores e alunos no processo de ensino e aprendizagem de modelos de tarefas. Ainda, além do resultado do questionário, foi possível observar que, durante o uso do HTMaster, os alunos realizaram simulações e experimentações de modelos, testando os conhecimentos trabalhados em aula, implicando em uma visualização da teoria abordada, em uma aliança entre a teoria e a prática.

5. CONSIDERAÇÕES FINAIS E TRABALHO FUTURO

O desenvolvimento de ambientes para apoio ao aprendizado de tópicos relacionados à área da Computação tem sido intensificado. Ambiente de apoio ao ensino de programação e teste de Software foram descritos por de Souza et al. [9] enquanto [11] apresentaram um simulador de sistema de arquivos para apoiar o ensino e a aprendizagem de Sistemas Operacionais.

Neste trabalho, apresentamos o design, desenvolvimento e avaliação de uma ferramenta intitulada de HTMaster como uma opção para auxiliar e apoiar o processo de ensino e aprendizagem de Modelos Hierárquicos de Tarefas em classes de IHC, permitindo ao usuário que analise e reflita sobre a elaboração das

estruturas dos modelos de tarefas, a partir de um módulo de validação de diagramas.

Após o experimento com alunos da disciplina de *Projeto de Interfaces*, foi observado que o HTMaster pode auxiliar na representação e no entendimento de modelos de tarefas, e, por isso, essa ferramenta foi incorporada como software de apoio à construção de MHT. Ressalta-se que continuamos acompanhando seu uso em diferentes turmas para analisar o real impacto e o nicho dessa ferramenta no ensino e aprendizagem de Modelos Hierárquicos de Tarefas.

No que se refere a trabalhos futuros, há módulos do HTMaster que ainda podem ser melhorados e redesenhados, em especial o sistema de ajuda que pode ser mais interativo e dinâmico. Além dessa questão, pretende-se indicar os erros e aviso do processo de validação no próprio diagrama que está sendo construído e não somente na área textual de validação. Também está sendo prevista uma versão que permita trabalho colaborativo.

Por fim, é importante destacar que, apesar de haver somente uma disciplina de IHC, alguns alunos começaram a “pensar” mais em IHC e a procura pela realização de Trabalhos de Conclusão nessa área tem aumentado a cada semestre. O HTMaster foi fruto do Trabalho de Conclusão realizado por Diogo Cavalleri, Gustavo Corso Ribeiro e José Paulo Zanchin, [5] sob orientação de autores desse trabalho.

REFERÊNCIAS

- [1] Annett, J. (2003) Hierarchical Tasks Analysis. In: Handbook of cognitive task design. Edited by Erik Hollnagel, Lawrence Erlbaum Associates, (p.17 – 36).
- [2] Barbosa, S. D. J. ; Silva, B. S. (2010) *Interação Humano-Computador*. Editora Campus-Elsevier, 2010.
- [3] Bim, S. A., Boscaroli, C. O que? Quando? E por quem? Uma investigação sobre o ensino de IHC nas universidades públicas do Estado do Paraná. In: *Anais do Workshop sobre Ensino de IHC, WEIHC 2012*, Cuiabá, Mato Grosso, Brasil, 2012.
- [4] Brasil, Parecer CNE/CES nº. 136/2012. (2012) Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, http://portal.mec.gov.br/index.php?option=com_docman&task=doc_download&gid=11205&Itemid=
- [5] Cavalleri, D., Ribeiro, G. C., Zanchin, J. P. HTMaster: Ferramenta web para elaboração de modelos hierárquicos de tarefas. 2012. 121 f. Trabalho de Conclusão (Graduação) – PUCRS, Porto Alegre, 2012.
- [6] CTTE - CTT Environment. (2010) <http://giove.cnuce.cnr.it/ctte.html>
- [7] de Paula, M. G. de. (2003) Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: construção de um modelo de interação. 2003. 76 f. Dissertação (Mestrado) – PUC do Rio de Janeiro, Rio de Janeiro, 2003.
- [8] De Souza, C. S. Ensino de IHC na Computação: Reflexões sobre 14 Anos de Experiência. In: *Anais do Workshop sobre Ensino de IHC, WEIHC 2012*, Cuiabá, Mato Grosso, Brasil, 2012.
- [9] de Souza, D. M.; Maldonado, J. C.; Barbosa, E. F. (2012) Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. In: *Simpósio Brasileiro de Informática na Educação, SBIE*, 2012.
- [10] Diaper, D; Johnson, P. (1989) Tasks analysis for knowledge descriptions: theory and application in training. In: *Cognitive ergonomics and human-computer interaction*. Edited by J. Long & A. Whitefield, Cambridge University Press (p. 191 - 224).
- [11] Gadelha R. N. S.; Azevedo, R. R. de; Oliveira, H. T. A. de; Neves, T. D.; Souza, C. C.; Silva, E. L. de. (2010) OS Simulator: Um Simulador de Sistema de Arquivos para Apoiar o Ensino/Aprendizagem de Sistemas Operacionais. In: SBIE, 2010.
- [12] Molic. (2011) <https://code.google.com/p/molic-designer/>
- [13] Preece, J.; Rogers, Y; Sharp, H. (2005) Design de interação: Além da interação humano-computador. Editora Bookman, 2005.
- [14] Sánchez, J; Sáenz, M. (2009) Enseñar y Aprender HCI: amalgama perfecta entre teoría y praxis. *Revista Bits de Ciencia*, nº 3 (p. 39-43).
- [15] SBC (2005). Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação.
- [16] Silva, A. C. da; Silva, J. C. A.; Silva, S. R. P. da. (2005) Integrando Visões de IHC e de ES por Padrões no Desenvolvimento por Prototipação Descartável. In: *Latin American Conference on Human-Computer Interaction*, 2, 2005, Cuernavaca, México.