

# Suporte ao Formalismo para Análise de Tarefas TAOS através da ferramenta iTAOS – Um Processo Evolutivo

**Francisco Petrônio Alencar de Medeiros**

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba  
 petronio@ifpb.edu.br

**Nielson José Pontes da Silva Júnior**

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba  
 nielsondasilvajr@gmail.com

**Bernardo Lula Júnior**

Universidade Federal de Campina Grande  
 lula@dsc.ufcg.edu.br

**Karolyne Maria Alves de Oliveira**

Universidade Federal de Campina Grande  
 karolyne@dsc.ufcg.edu.br

**RESUMO:** O resultado da análise da tarefa é um documento contendo a descrição detalhada e hierarquizada do trabalho segundo o ponto de vista do usuário e pode ser aplicado para apoiar o projetista em três momentos: especificação das funcionalidades do sistema, projeto da interface com o usuário e elaboração de manuais de treinamento. O surgimento de formalismos, como é o caso do TAOS, foi um passo importante para melhorar a qualidade das descrições das tarefas, principalmente do que diz respeito à colaboração entre profissionais de diferentes áreas que passaram a usar uma linguagem formal comum. Realizar essa análise sem a disponibilidade de uma ferramenta específica implica em diversos problemas, daí a necessidade da automatização desse processo com uma ferramenta (iTAOS). O principal objetivo desse trabalho é apresentar o processo evolutivo pelo qual passou o iTAOS e tornar esse artigo a referência para a terceira release da ferramenta, apresentando suas novas funcionalidades e métodos de desenvolvimento: Overview da árvore, Desfazer e refazer operações, Exportar a árvore e Integração com o FastInterface.

**Palavras chave:** análise da tarefa, TAOS, iTAOS e Projeto de interfaces com o usuário.

**ABSTRACT:** The result of the analysis of the task is a document that contains the detailed and hierarchical description of the work according to point of view of the user and can be applied to support the designer at three moments: specification of the functionalities of the system, user interface design and training manual elaboration. These formalisms, as TAOS, was a step important to improve the quality of the tasks descriptions, mainly of what it says respect to the contribution between professionals of different areas that had started to use a common formal language. To carry through this analysis without the availability of a specific tool implies in diverse problems, from there the necessity of the automatization of this process with a tool (iTAOS). The main goal of this work is to present the evolution of iTAOS and to become this article the reference for third release of the tool, presenting its new functionalities and methods of development: Overview of the tree, Undo and Redo operations, Export the tree and Integration with the FastInterface.

**Keywords:** task analysis, TAOS, iTAOS and User interface design.

## 1. Introdução

O campo da interação homem-computador é o estudo do indivíduo (usuário), da tecnologia computacional (sistema) e do entendimento do trabalho (tarefa) que o indivíduo tenta realizar utilizando essa tecnologia. Para o usuário, o sistema é apenas uma ferramenta e, como tal, exige, para ser utilizado, um conhecimento e habilidade não somente do domínio da tarefa, mas também do uso do sistema, envolvendo as funcionalidades e os meios disponíveis para utilizá-lo. Dessa forma, aprender a usar o sistema torna-se uma tarefa adicional para o usuário, e, portanto, deve exigir o menor esforço possível. Portanto, é de fundamental importância que o sistema seja adaptado às características do usuário, às tarefas a serem executadas e ao ambiente de trabalho, para que o esforço adicional de ter que aprender a usar o sistema seja o menor possível. É nesse sentido que vem se exigindo, cada vez mais, dos projetistas desses produtos e sistemas, uma preocupação com a qualidade da interface.

A partir da necessidade de se desenvolver aplicações utilizáveis e acessíveis a um grande número de usuários, diversas metodologias de concepção e desenvolvimento de interfaces foram propostas. A maioria dessas metodologias tem como elemento base a representação da tarefa que o usuário desempenha ou deve desempenhar com a utilização do sistema correspondente. Elas guiam, a partir da descrição da tarefa, do perfil do usuário e de princípios ergonômicos, a construção de um protótipo da interface levando em conta os objetivos do usuário. As metodologias que se baseiam no modelo da tarefa do usuário constituem um expressivo apoio à concepção de interfaces ergonômicas (MORKOPOULOS e GIKAS, 1997).

Ao longo dos últimos anos, algumas ferramentas foram desenvolvidas para análise e modelagem de tarefas. As primeiras, como por exemplo, Adept e IMAD\*, são rudimentares e com características limitadas, por esse motivo são usadas somente pelos grupos de pesquisas que as desenvolveram. Recentemente, foi proposto na Europa uma ferramenta mais elaborada e que foi disponibilizada na Internet para download: CTTE (CTTE, 2002).

Este artigo apresenta inicialmente uma breve fundamentação teórica sobre análise de tarefas, seguida de uma descrição do formalismo

TAOS e das funcionalidades da ferramenta iTAOS, e por fim, o processo de refatoramento pelo qual passou o iTAOS num esforço conjunto do grupo de pesquisa em Engenharia de Software do IFPB com o grupo de pesquisa de interface Homem Máquina da Universidade Federal de Campina Grande.

## 2. Análise e modelagem da tarefa

A análise da tarefa descreve o conjunto de tarefas com o objetivo de entender o funcionamento cognitivo dos operadores (SEBILLOTE, 1995) e são realizados a partir de entrevistas dirigidas, observações, experimentos ou outros métodos habituais de pesquisa, buscando evidenciar as características do processo de realização da tarefa. O resultado da análise é um documento (relatório) contendo uma descrição detalhada e hierarquizada do trabalho segundo o ponto de vista do usuário, cujo resultado pode ser aplicado para apoiar as ações do projetista em pelo menos três momentos: na especificação do sistema (funcionalidades), no projeto da interface e na elaboração de manuais de treinamento (HEEMANN, 1997).

A informação proveniente do modelo da tarefa é explorada para automaticamente ou interativamente derivar a estrutura de navegação de uma aplicação. O projeto de interface baseado na tarefa incorpora informações relativas à forma como as tarefas do usuário vão ser acionadas nas interfaces do usuário tal como os relacionamentos entre essas tarefas. Esse tipo de informação fornece aspectos de usabilidade como sobreposição de interfaces, agrupamento da apresentação, etc. (OLIVEIRA *et. al.*, 2008)

No entanto, a utilização efetiva da análise da tarefa no processo de concepção de interfaces se deu a partir da introdução de formalismos apropriados e que pudessem ser entendidos tanto pelos ergonomistas quanto pelos projetistas de sistemas computacionais. Estes formalismos surgiram como tentativas de integrar a análise e a descrição da tarefa ao processo de concepção de interfaces (LIMBOURG *et al.*, 2001), melhorando, com isso, a qualidade das descrições e maximizando a completude e a coerência.

Diversos formalismos foram propostos com essa finalidade e destacamos os seguintes pelos seus graus de incidência na literatura a respeito: TKS (JOHNSON *et al.*, 1988) e MAD\*

(GAMBOA, 1998). O Grupo de Interfaces Homem-Máquina (GIHM) da Universidade Federal de Campina Grande (UFCG) tem proposto o formalismo TAOS (*Task and Action Oriented System*) para análise da tarefa no âmbito da concepção e desenvolvimento de interfaces de sistemas computacionais.

### 3. Formalismo TAOS

TAOS (*Task and Action Oriented System*) é um formalismo concebido para a aquisição e representação do conhecimento sobre um domínio (MEDEIROS et al., 2000), para a descrição e análise de tarefas em vista a concepção de interfaces homem-computador de sistemas computacionais.

O modelo de representação proposto por TAOS considera que os conceitos podem ser de

dois tipos: conceitos estáticos e conceitos dinâmicos. Um conceito é considerado estático se ele conserva suas características durante um intervalo de tempo pré-estabelecido. Os conceitos objeto (*Object*), agente (*Agent*), instrumento (*Instrument*), situação (*Situation*) e método (*Method*) são considerados estáticos. Um conceito é considerado dinâmico se ele representa a evolução de uma situação observada num intervalo de tempo pré-estabelecido. Os conceitos processo (*Process*), plano (*Plan*) e ação (*Action*) são considerados dinâmicos.

A Fig. (1) ilustra a taxonomia dos conceitos definidos por TAOS (Meta-modelo). A representação gráfica dos conceitos é uma árvore, cujos nós representam as classes e cujos arcos representam as relações entre as classes.

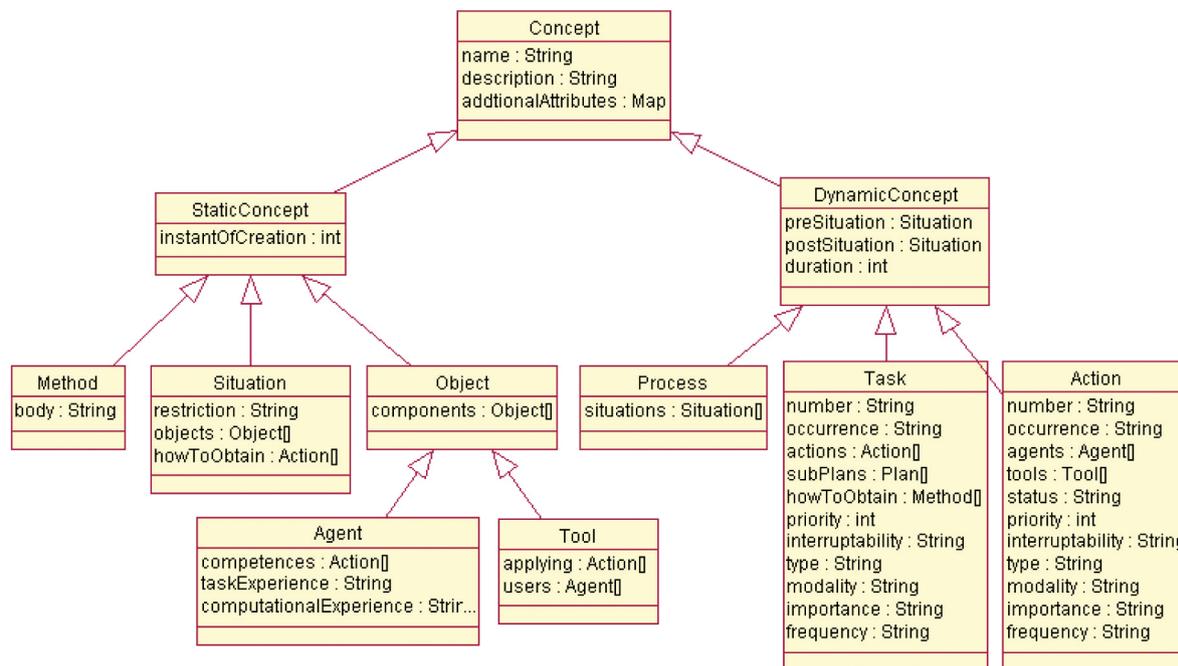


Figura 1. Meta-Modelo em UML do Formalismo TAOS

A descrição hierárquica da tarefa em sub-tarefas e ações (conceitos dinâmicos) utilizando o formalismo TAOS é representada graficamente na forma de uma árvore, cujos nodos representam as tarefas, sub-tarefas e ações e os arcos representam as conexões com as tarefas superiores. Para cada tarefa ou ação há um frame associado, que são representados em

TAOS por descritores (tabelas) que descrevem os atributos relativos aos conceitos estáticos.

O surgimento de formalismos para análise da tarefa foi um passo importante para melhorar a qualidade das descrições das tarefas, principalmente do que diz respeito à colaboração entre ergonomistas e projetistas que passaram a usar uma linguagem formal comum.

Porém, realizar a análise e modelagem de tarefas sem a disponibilidade de uma ferramenta específica que auxilie o projetista implica em problemas (GAMBOA, 1998), tais como:

- Dificuldade em descrever a estrutura de árvores das tarefas e suas notações formais;
- Dificuldade de verificar a consistência e completude das tarefas modeladas;
- Impossibilidade de geração automática ou semi-automática do modelo da interação a partir do modelo da tarefa;

Esses problemas listados são sentidos com uma grande intensidade por projetistas que analisam e modelam tarefas utilizando os formalismos disponíveis citados anteriormente. Pela dificuldade em adquirir uma ferramenta de modelagem, a construção e manipulação das árvores do modelo da tarefa são feitas utilizando lápis e papel ou, no máximo, ferramentas que desenhavam e manipulavam figuras geométricas básicas, e não os conceitos de tarefas envolvidos. Por esse motivo, há uma grande dificuldade em verificar a consistência e completude da tarefa, principalmente pela necessidade de memorização dos atributos, objetos e termos relacionados com a tarefa que está sendo modelada. Além disso, a não existência de uma representação digital apropriada da tarefa implica: (I) na impossibilidade de se ter ferramentas computacionais para auxiliar os projetistas nos passos seguintes do processo; e (II) em grande dificuldade na utilização de conhecimentos ergonômicos para a obtenção da representação da interação (modelo da interação).

#### 4. Ferramenta iTAOS

iTAOS é uma ferramenta gráfica que implementa o formalismo TAOS (*Task and Action Oriented System*) e acompanha o projetista durante a fase de descrição da tarefa. iTAOS foi concebido para ser implementado em dois módulos, de acordo com o princípio da independência do diálogo: o módulo TAME (CORDEIRO, 2003), que inclui toda a parte funcional da ferramenta e o módulo TAOS-Graph (MEDEIROS, 2003), que permite a visualização gráfica do processo de modelagem (módulo interativo).

#### 4.1. Funcionalidades da ferramenta iTAOS

Uma tarefa é descrita em TAOS através de uma metodologia descendente-ascendente, ou seja, deve ser construída uma hierarquia de tarefas (*Plan*) e ações (*Action*), tendo a tarefa mais geral a ser executada no topo e as ações que permitem realizar a execução da tarefa nas folhas da árvore.

iTAOS permite a construção dessa hierarquia de tarefas e ações (árvore TAOS da tarefa) através de um ambiente gráfico, via manipulação direta. Os objetos envolvidos na construção da árvore são os nodos da árvore (tarefas e ações), os arcos que ligam as ações e sub-tarefas as super-tarefas (nodos pai) e os arcos que representam a estrutura das sub-tarefas ou ações. Na Fig. (2) apresentamos um extrato da descrição da tarefa *Efetuar login* modelada utilizando a ferramenta iTAOS.

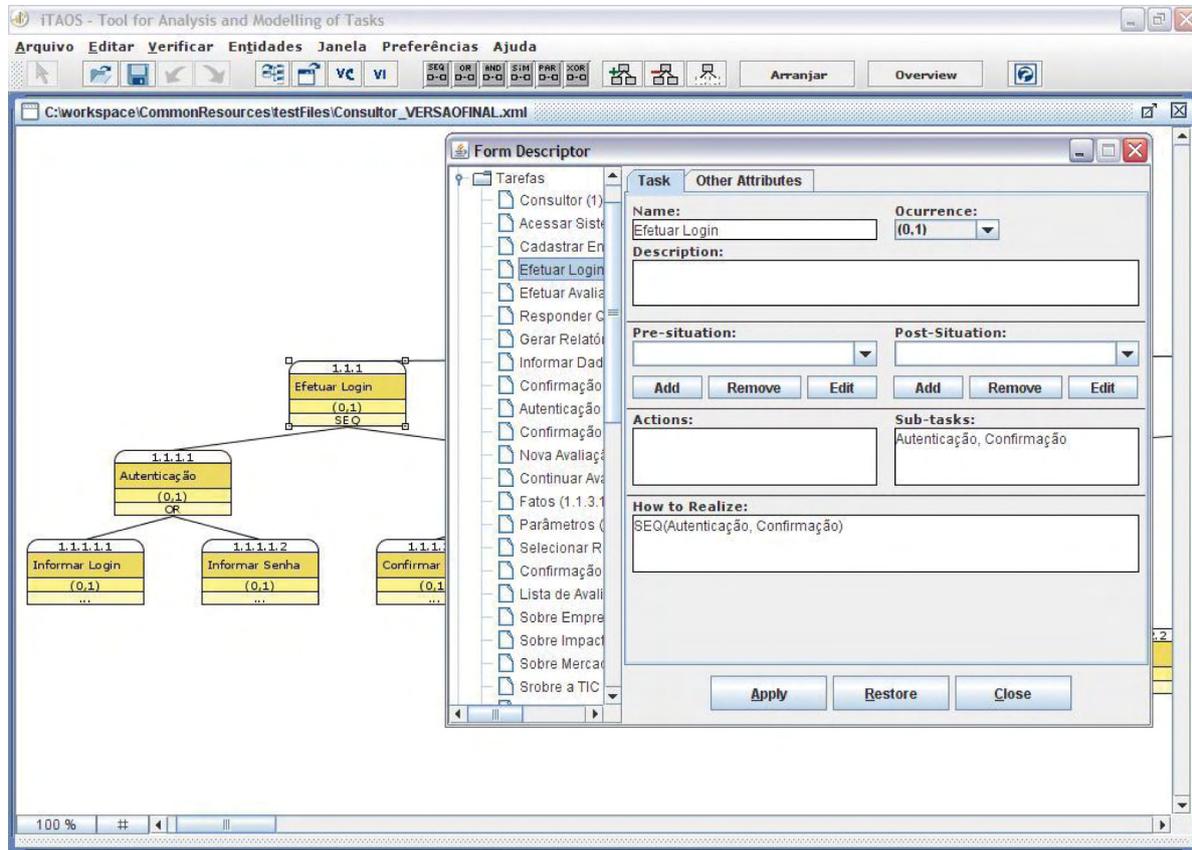


Figura 2. Árvore e Descritor da Tarefa Efetuar Login no iTAOS

Na própria árvore é possível editar o nome e a ocorrência da tarefa ou ação, através de um clique com o botão direito sobre a tarefa e escolhendo a opção no pop-up menu ou através de dois cliques sobre o nome ou ocorrência na tarefa. A numeração das tarefas é atualizada automaticamente, baseada na posição no nível de hierarquia que ela se encontra.

Para cada nodo da árvore (tarefa ou ação), o usuário pode preencher seus descritores. Os descritores das ações (*Action*) e das tarefas (*Task*) possuem atributos referentes à especificação da interface (Tipo, Frequência, Modalidade e Centralidade da tarefa), atributos referentes à simulação do modelo (Interruptibilidade e Prioridade da tarefa) e atributos relacionados aos conceitos estáticos (Pré e pós-situação, agentes e ferramentas envolvidos com a tarefa).

Nos descritores Objeto (*Object*), Ferramenta (*Tool*) e Agente (*Agent*), o usuário pode inserir novos atributos, que são inseridos no formulário em tempo de execução. Na Fig. (2) apre-

sentamos também o descritor da tarefa Efetuar Login.

Além de criar modelos, o usuário pode salvá-los e recuperá-los através de arquivos XML. O modelo pode ser aberto posteriormente e editado sem maiores problemas. O iTAOS também permite que conceitos pertencentes a um modelo de tarefas possam ser utilizados em outros modelos através da funcionalidade de importação de conceitos. Com essa funcionalidade é possível fazer o reuso de conceitos para economizar tempo de trabalho e/ou aproveitar idéias.

O modelo pode ser impresso de várias formas. No caso da árvore, tanto pode ser impressa diretamente ou salvar num formato jpeg para ser exportado, já os descritores podem ser impressos no formato html. A ferramenta iTAOS também permite que seja verificada a consistência e completude do modelo. A verificação da completude se encarrega de identificar conceitos cujos descritores não tenham sido devidamente preenchidos, enquanto que a verificação da consistência se en-

carrega de verificar possíveis inconsistências na descrição TAOS da tarefa, principalmente no que diz respeito às pré e pós-situações dos modelos.

## 5. Refatoração do iTAOS

A refatoração da ferramenta iTAOS atacou dois problemas: o primeiro dizia respeito a má estruturação do código, tornando-o muitas vezes ilegível e a segunda a necessidade de incorporação de funcionalidades que tornassem a ferramenta mais usável e completa.

Para estruturar o código foram aplicados padrões de projetos, entre eles *Façade*, *Factory Method*, *Singleton* e *Observer* (FOWLER, 2004). Concluída essa etapa, o código tornou-se estruturado para receber novas funcionalidades que foram projetadas no sentido de tornar o iTAOS uma referência em ferramenta para análise e modelagem de tarefas, aumentando de forma significativa o uso por instituições de pesquisa e iniciativa privada no Brasil e no mundo.

Com a primeira release da ferramenta iTAOS o usuário foi capaz de modelar uma tarefa seguindo o formalismo TAOS, construindo a árvore da tarefa e os descritores. Na segunda release, a maior preocupação foi deixar a ferramenta em conformidade com o padrão ISO 9241 (ISO, 1997; ISO, 1998; ISO 1999), que preza pela aplicabilidade de recomendações ergonômicas num produto (CYBIS, 2000). Ainda na segunda release foram retirados bugs descobertos com o uso incessante da primeira versão do iTAOS em instituições de pesquisa. Na terceira *release*, disponibilizada em Setembro de 2008, novas funcionalidades foram adicionadas na manipulação da representação gráfica da árvore do modelo da tarefa. Por diversas

vezes essa árvore ficava muito grande, causando dificuldade de manuseamento da mesma pelo usuário. Para que isso fosse facilitado foram implementadas as funcionalidades de:

- *Overview*, que possibilita ao usuário visualizar uma miniatura da árvore;
- O Aumento/Diminuição do espaçamento vertical entre os nodos da árvore;
- Esconder os filhos de um determinado nodo da árvore.

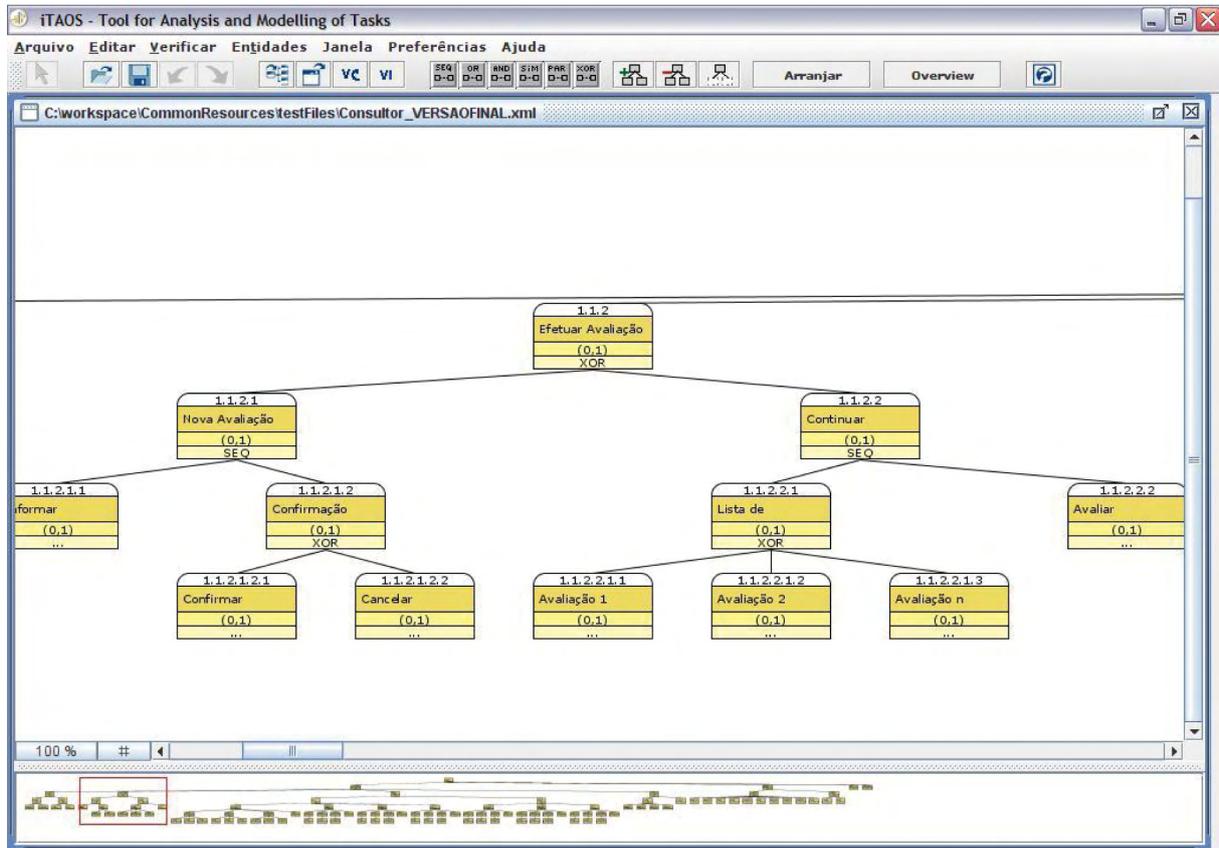
Outra funcionalidade indispensável a uma ferramenta do porte do iTAOS que ainda não estava implementada era o desfazer e o refazer operações realizadas pelo usuário no sistema. Também foi implementada nessa versão.

### 5.1. Overview

A funcionalidade de *Overview* provê ao usuário uma miniatura da árvore de modelagem de tarefa, a partir da qual se pode ter uma melhor visão e manipulação da árvore anteriormente citada.

O usuário com essa nova perspectiva visualiza uma miniatura idêntica da árvore da tarefa e um retângulo vermelho que corresponde à área da árvore, em tamanho real, que está sendo visualizada pelo usuário. Movendo-se esse retângulo a área de visualização da árvore da modelagem também será movida. A recíproca também é válida.

Com a adição dessa funcionalidade, a interface gráfica do iTAOS precisou sofrer uma leve alteração. O projeto de modelagem da tarefa passou a ser inserido em um componente Java Swing chamado *JSplitPane*, para uma melhor visualização de ambas as árvores. O *overview* pode ser visualizado na Fig. (3).



**Figura 3. Overview:** miniatura da árvore de modelagem da tarefa abaixo, área da árvore de modelagem da tarefa correspondente ao quadrado vermelho acima.

Na implementação do Overview foram feitos os cálculos matemáticos para que, na medida em que o retângulo na miniatura da árvore fosse arrastado, a área de visualização da árvore fosse movida na mesma proporção. O desenvolvimento dessa funcionalidade foi lento e trabalhoso, exigiu trabalho com eventos de mouse, evento de compo-

nentes do Java Swing, como a barra de rolagem, e cálculos minuciosos para manter a proporção entre a área de visualização da árvore e a área dada pelo retângulo vermelho na miniatura da mesma. Na Fig. (4) é apresentado o diagrama de classes da funcionalidade Overview.

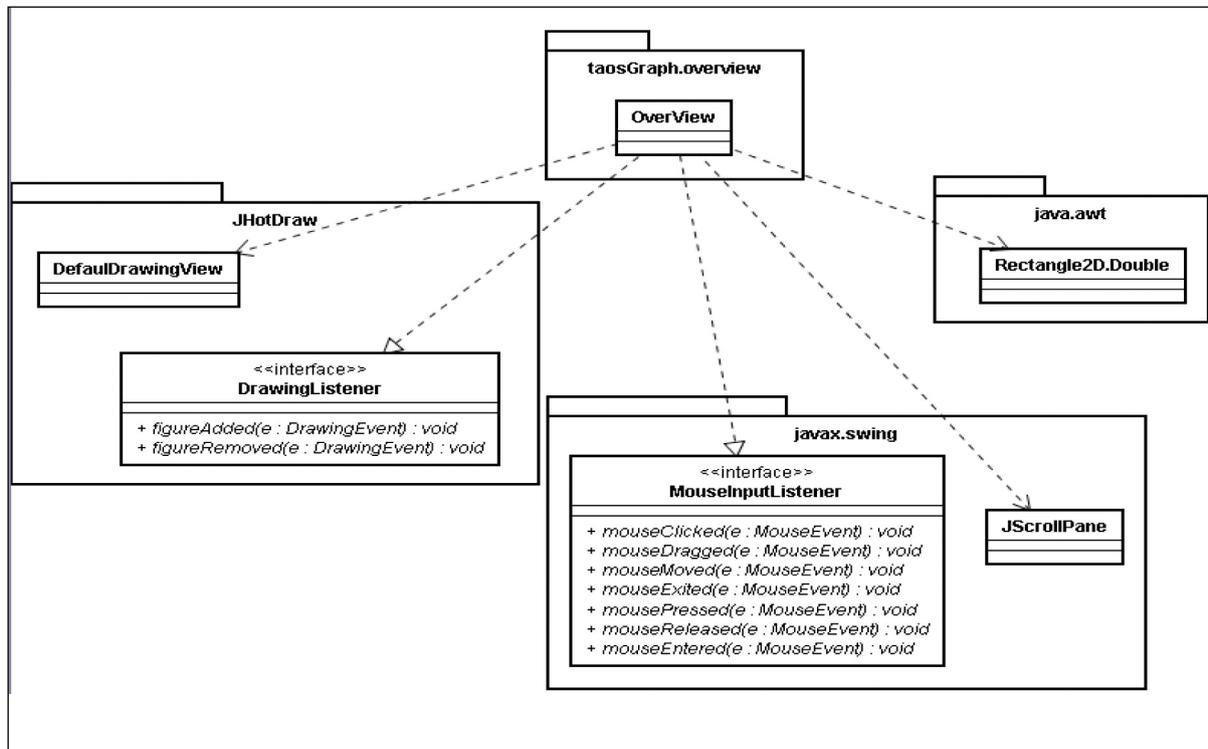


Figura 4 . Diagrama de Classes da Funcionalidade *Overview*

## 5.2. Funcionalidade Desfazer e Refazer

A funcionalidade de Desfazer (Undo) e Refazer, indispensável em sistemas computacionais de manipulação direta, ainda não estava implementada e tornou-se a funcionalidade principal da terceira *release* da ferramenta, justamente devido a excessiva cobrança por parte dos usuários da ferramenta.

Para implementá-la foi necessário o uso do padrão de projeto Command e de uma classe

(ProcessadorDeComandos) responsável pela manipulação das estruturas de dados, que fazem a alteração no estado geral do sistema.

Inicialmente, definiram-se quais as alterações do sistema poderiam ser desfeitas. Concluiu-se que seriam as alterações de deleção e adição de tarefas gráficas, bem como a edição de nome e de ocorrência. Na Fig. (5) é possível visualizar o diagrama de classes dessa funcionalidade.

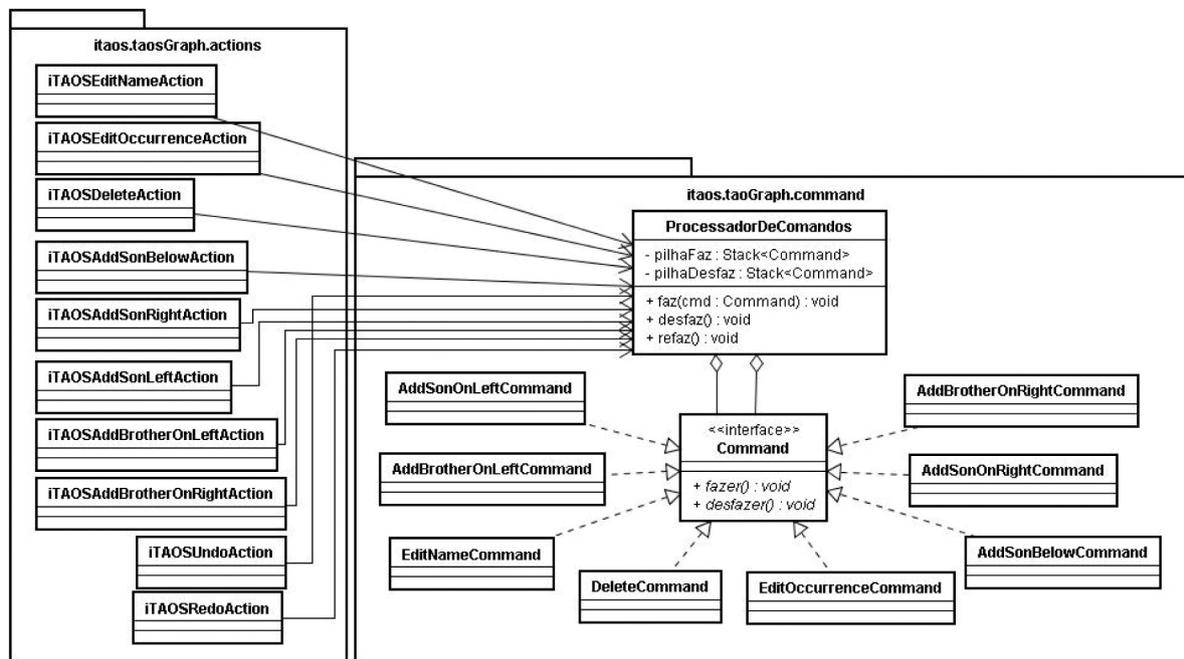


Figura 5. Diagrama de classes da funcionalidade Desfazer e Refazer

A implementação começou com a criação da interface *Command* e de seus métodos fazer e desfazer. Logo após criou-se a classe *ProcessadorDeComandos*, a qual é responsável pela manipulação das estruturas de dados (pilhas), que fazem a alteração no estado geral do sistema. Essa classe possui duas pilhas (*pilhaFaz*, *pilhaDesfaz*) responsáveis por armazenar e manipular os comandos que serão desfeitos e refeitos, e três métodos de manuseamento dessas estruturas: *faz*, *desfaz* e *refaz*. O método *faz*, é responsável pela primeira execução do comando e sua adição na pilha de desfazer. Esse método é chamado pelas actions.

O método *iTAOSUndoAction* desfaz a atividade que acabou de ser realizada, fazendo com que o sistema volte ao estado anterior à mesma. O comando é retirado da pilha de desfazer e adicionado na pilha de refazer. O método *refazer*, é executado a partir da classe *iTAOSRedoAction*, e realiza uma operação inversa ao que o método *desfazer* faz: retorna ao estado anteriormente desfeito. O comando refeito é então retirado da pilha de operações a serem refeitas e acrescentado na pilha de operações aptas a serem desfeitas.

Em seguida, foi feita a implementação dos comandos que poderão ser desfeitos e refeitos, e uma pequena refatoração nas *actions* transferindo

o código de execução das mesmas para as classes *command* relacionadas. As *actions* agora apenas ordenam ao *ProcessadorDeComandos* o que fazer.

### 5.3. Funcionalidade Exportar Árvore da tarefa

Após o término da modelagem da tarefa, o projetista pode querer salvar a árvore da modelagem como uma imagem jpg. Diante dessa necessidade foi desenvolvida a funcionalidade de Exportar Árvore.

Essa funcionalidade já havia sido desenvolvida em uma versão anterior do iTAOS, porém, estava apresentando problemas. A forma como estava implementada não estava sendo satisfatória. Essa forma antiga resultava apenas em um PrintScreen da tela que o usuário estava visualizando e não da árvore da modelagem inteira, sendo portanto insatisfatória.

### 5.4. Integração do iTAOS ao FastInterface

Para dar auxílio aos projetistas, o ambiente *FastInterface* agregou os módulos computacionais iTAOS (MEDEIROS *et. al.* 2002), MAPA (RODRIGUES, 2005), SMILE (LIMA, 2007) e Hi-Fy (OLIVEIRA *et. al.*, 2008), cada um respon-

sável por uma fase da metodologia para projeto de interfaces MEDITE. Essa integração de módulos isenta os projetistas de interface trabalhar com cada ferramenta isoladamente (OLIVEIRA et al.,

2007). A Fig (6) apresenta a ferramenta iTAOS integrada ao FastInterface.

A integração foi uma tarefa complexa que exigiu esforço de toda a equipe de desenvolvimen-

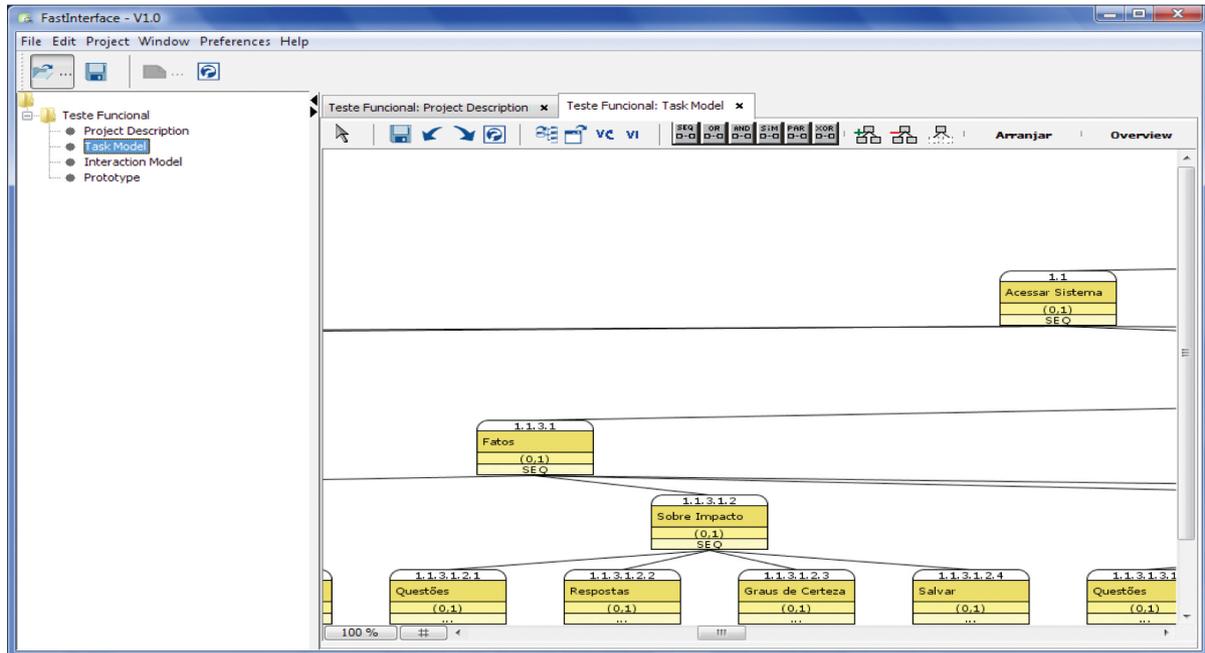


Figura 6. iTAOS integrado ao *FastInterface*

to de todos os módulos do projeto. Foi definida uma arquitetura de integração e algumas diretrizes a serem seguidas.

Uma das exigências foi a criação de uma fachada de integração e ajuste na implementação do iTAOSProject, para que o mesmo não dependesse mais da iTAOSApplication e sim de sua interface mais genérica, Application, possibilitando a exibição do módulo no FastInterface, sem precisar instanciar a aplicação do módulo.

A fachada de integração, que foi chamada de iTAOSGuiFacade, teve que ser modificada para a adição de um conjunto de métodos definidos na documentação de integração. Os métodos implementados são citados abaixo:

- *registerActions()*: registra as ações, uma vez que o método *init()* de Application não pode ser chamado;
- *getToolBar()*: retorna um objeto do tipo JToolBar, que é a barra de ferramen-

tas do módulo adaptada para ser inserida no FastInterface;

- *startGUI()*: responsável por inicializar a Camada Gráfica do módulo;
- *openProject(File arquivo)*: retorna um objeto do tipo Project, que é a representação gráfica do modelo de trabalho do módulo passado no arquivo;
- *saveProject(Project project)*: salva o modelo representado pelo Project passado como parâmetro;

## 6. Conclusões

Analisar tarefas é uma tarefa árdua devido ao tamanho das descrições geradas e às notações formais impostas pelos formalismos, tornando-se muitas vezes impossível a descrição sem a disponibilidade de uma ferramenta computacional específica. A ferramenta iTAOS surgiu para sanar esses problemas devendo: (i) apoiar o projetista

na elaboração da representação da tarefa segundo o formalismo e metodologia TAOS, (ii) verificar a consistência e completude da representação e (iii) possibilitar a geração automática ou semi-automática do modelo da interação a partir do modelo da tarefa através dos arquivos XML gerados ao salvar a descrição TAOS da tarefa.

Algumas características da ferramenta iTAOS devem ser ressaltadas por se destacarem em relação a outras ferramentas do gênero, tais quais: (i) considera tarefas concorrentes, através dos operadores SIM e PAR, (ii) modela tarefas cooperativas envolvendo múltiplos usuários, através da lista de agentes (Agent) das ações (Action) e (iii) permite que projetistas que utilizem diferentes sistemas operacionais possam usufruir da ferramenta iTAOS, por se tratar de um ambiente multiplataforma.

Este trabalho torna-se a referência quanto a terceira release da ferramenta iTAOS, no tocante a refatoração aplicada ao código e a adição de novas funcionalidades. A ferramenta iTAOS cresceu muito durante três ciclos evolutivos pela qual passou, tornando-se uma ferramenta completa para modelagem de tarefas do usuário e está totalmente integrada ao FastInterface, que é uma ferramenta para geração automática, edição e simulação de protótipos de interface de usuário.

O download da ferramenta FastInterface, que inclui a ferramenta iTAOS, bem como sua documentação pode ser encontrada em <http://fi.ifpb.edu.br/ITAOS/>.

## 7. Referências

- CORDEIRO, P. B. **Projeto e Implementação do Módulo TAME da Ferramenta iTAOS para Análise e Modelagem da Tarefa**. Dissertação (Mestrado em Informática) — Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande PB, 2003.
- CTTE, 2002. **The ConcurTaskTrees Environment**. Disponível em: <http://giove.cnuce.cnr.it:8443/CTTE/servlet/DownloadCTTE> Acesso em: Julho de 2007.
- CYBIS, W. A., **Abordagem ergonômica para IHC**, apostila de curso, Laboratório de Utilizabilidade INE/UFSC, Florianópolis, Brasil, 2000.
- FOWLER, M. **Refatoração, aperfeiçoando o projeto de código existente**. Bookman, 2004.
- GAMBOA, F. R. **Spécification et Implémentation d'ALACIE: Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques**, Thèse de Doctorat, Paris XI, Octobre, 1998.
- HEEMANN, V., **Curso de Ergonomia em Sistemas de Informação**, Universidade Federal da Paraíba, Campina Grande, Julho de 1997.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, **Projeto de norma internacional, Menu dialogues - ISO 9241 parte 14**, final draft, Genebra, Suíça, 1997.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Projeto de norma internacional, Form filling dialogues - ISO 9241 parte 17**, final draft, Genebra, Suíça, 1998.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **Projeto de norma internacional, Direct manipulation dialogues - ISO 9241 parte 16**, final draft, Genebra, Suíça, 1999.
- JOHNSON, P. et al. **Task-Related Knowledge Structures: Analysis, Modelling and Application**, Queen Mary College, University of London 1988.
- LIMA, G. A. N. **SMILE: Desenvolvendo uma Ferramenta de Suporte à Prototipagem de Média-fidelidade de Interface do Usuário**. Relatório de Estágio (Graduação) – Tecnologia em Sistemas para Internet, Centro Federal de Educação Tecnológica da Paraíba, João Pessoa PB, 2007.
- LIMBOURG, Q.; PRIBEANU, C.; VANDERDONCKT, J., **Towards Uniformed Task Models in a Model-Based Approach**, GIST Technical Report, Glasgow, Scotland, June 2001.
- MEDEIROS, A. **Projeto e Implementação do módulo TAOS-Graph da ferramenta iTAOS para análise e modelagem da tarefa**. Dissertação (Mestrado em Informática) — Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande PB, 2003.

MEDEIROS, F. P. A., LULA, B. Jr., CORDEIRO, P. B. **A Graphical Tool to Support Task Description Using TAOS Formalism for UI Design.** In Proc. of the 7th ERCIM Workshop, p. 45- 51, 2002.

MEDEIROS, F. P. A.; LULA, B. J., CORDEIRO, P. B. **Suporte à Descrição de Tarefas no Contexto de Interfaces do Usuário.** In: 17º Simpósio Brasileiro de Engenharia de Software, 2003, Manaus. 10ª Sessão de Ferramentas do 17º Simpósio Brasileiro de Engenharia de Software, v. 1. p. 85-90, 2003

MEDEIROS, H.; KAFURE, I. M e LULA, B. Jr., **TAOS: a Task-and Action Oriented Framework for User's Task Analysis in the Context of Human-Computer Interfaces,** Proceeding of SCCC 2000 – XX International Conference of the Chilean Computer Science Society, november 2000, Santiago, Chile, 2000.

MARKOPOULOS P. e GIKAS S., **Formal Specification of a Task Model and Implications for Interface Design,** Cognitive Systems vol 4, pp. 4-3 Queen Mary and Westfield College, University of London, February 1997.

OLIVEIRA, K.; LULA, B.; GOUVEIA, R. **HIFY: Ferramenta para Geração Automática e Edição de Protótipos Alta-Fidelidade em uma Abordagem Baseada na Tarefa do Usuário.** III Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica Fortaleza - CE – 2008

RODRIGUES, C. E. C. L. **Meditate+: utilizando o processo de roteirização para a obtenção do modelo de interação Editor-estendido.** Dissertação (Mestrado em Informática) – Ciência da Computação, Universidade Federal de Campina Grande, Campina Grande PB, Junho de 2005.

SEBILLOTE, S., **Hierarchical Planning as Method for Task Analysis: the example of office task analysis,** Behaviour and Information Technology, vol. 7, No. 3, pp. 275-293. INRIA, Rocquencourt, Le Chesnay-Cedex, France, 1988.

OLIVEIRA, K. M. A.; AGUIAR, Y. P.; LULA, B.; CHAVES, L. C. R.; SOUZA, G. G.; VIEIRA, D. A.; CARVALHO, Y. O.; LIMA, J. G.; ALVES, M. O. **O Uso de Modelos e Múltiplos Protótipos na Concepção de Interface do Usuário.** Principia, João Pessoa, nº 15, p. 15-29, 2007.