

# Um modelo para avaliação de usabilidade em um cenário de convergência digital

Artur Henrique Kronbauer  
DMCC – UFBA  
Av. Adhemar de Barros s/n sala138  
Salvador – BA, Brasil  
CEP 40.170-110  
arturhk@gmail.com

Celso A. S. Santos  
DMCC – UFBA  
Av. Adhemar de Barros s/n sala138  
Salvador – BA, Brasil  
CEP 40.170-110  
saibel@dcc.ufba.br

## ABSTRACT

Converging devices, such as cell phones and Interactive Digital TV, are on the list of the most sold equipment nowadays. Besides the great search for those equipment, the diversity of users, differences on the form of interaction, dimension of equipment, aspects of mobility and diversified use scenarios, demand usability assessments for the investigation of possible improvement of the interaction involving users and the interfaces of the available applications. In this paper we present a model to evaluate the usability of applications for this kind of device. The proposed model is divided in three modules: (i) mapping of the intentions of the application's designer, (ii) capture of the user's interactions with the application's interface, (iii) module of comparison between the intentions of the designer and the usability of the final user.

## Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems.

## General Terms

Design, Performance, Measurement.

## Keywords

Automatic usability evaluation, Aspect Oriented Programming, Metrics, Digital Convergence.

## 1. INTRODUÇÃO

Ao longo das últimas décadas, diversas tecnologias que revolucionaram a forma de comunicação, entretenimento e a execução de atividades foram apresentadas à sociedade. Durante a década de 90, o processo de convergência gerou uma gama de dispositivos (*smartphones*, PDAs e a TV Digital móvel) capazes de reunir todas estas formas de interação usuário computador de forma integrada, com mobilidade e razoável poder de

processamento. Essa integração é consequência da evolução do hardware e do software que estão embarcados nestes dispositivos. O problema é que, em muitas situações, os aplicativos que são executados nesses dispositivos podem apresentar interfaces rígidas e nem sempre adequadas para o seu propósito, o que traz prejuízos em termos da usabilidade das aplicações.

Neste artigo, o termo “cenário de convergência digital” se refere a um cenário onde diferentes equipamentos são capazes de tratar diversos tipos de mídias em um mesmo contexto, realizarem algum tipo de processamento, além terem suporte à mobilidade. Um dos principais desafios para o sucesso do uso desses dispositivos num cenário de convergência é a produção de aplicativos com boa usabilidade, que possibilitem uma fácil interação dos usuários com a aplicação na realização de tarefas.

A área que estuda o projeto, a avaliação e a implementação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos associados ao uso desses sistemas, é chamada de Interação Humano-Computador (IHC). O foco da IHC está na construção de interfaces com alta qualidade de uso [1].

O conceito de interação usuário-sistema abrange a forma com que os usuários formulam suas intenções, planejam suas ações, atuam sobre a interface, percebem e interpretam as respostas e avaliam se seus objetivos foram alcançados [12]. O grande desafio aqui é mensurar se as interfaces disponíveis nas aplicações atingem seus objetivos de forma a contemplar as necessidades dos usuários finais na realização de tarefas. Mesmo com a utilização de metodologias clássicas para a criação de interfaces, o sucesso não é garantido [18]. Sendo assim, somente com avaliações de usabilidade é possível comprovar se uma interface atende às necessidades dos usuários na realização das tarefas.

A maioria das abordagens de avaliação de usabilidade requer investimentos elevados, tanto de recursos financeiros como de pessoal e, na maioria das vezes, não conseguem reproduzir uma contextualização de uso para aplicações convergentes por serem realizadas em laboratório e descontextualizadas do cenário real dos usuários [2]. Por esse motivo, muitos gerentes de projetos ignoram os benefícios da avaliação de usabilidade durante a construção dos sistemas ou após estarem em produção, mesmo correndo o risco do fracasso do projeto [1].

A contribuição principal deste artigo é a proposta de um modelo de avaliação da usabilidade de aplicações e a sua validação num cenário de convergência. Para isso, duas implementações de

aplicações incluindo métricas de usabilidade para a plataforma de TV digital e para um dispositivo móvel, são implementadas e analisadas.

O modelo é a base para o processo de avaliação apresentado no decorrer do texto e que abrange as seguintes etapas:

1. O mapeamento das intenções do projetista de uma aplicação, com relação à execução das tarefas que a aplicação poderá prover a seus usuários.
2. A instrumentação do código da aplicação com métricas de usabilidade para possibilitar a coleta automática de dados de interação do usuário final com a aplicação.
3. A comparação dos resultados obtidos por meio das métricas de usabilidade com a especificação prevista pelo projetista para a execução das tarefas em sua aplicação.

A intenção do modelo é permitir avaliar a usabilidade de aplicações confrontando a ótica do projetista com o comportamento efetivo do usuário em situações reais de uso da aplicação, utilizando uma técnica conhecida como Avaliação Automática de Usabilidade [6]. O processo de avaliação associado ao modelo permite a tomada de ações de melhoria da aplicação fundamentadas em dados concretos de intenções e usabilidade.

O artigo está estruturado da seguinte forma: a seção 2 apresentada uma contextualização dos métodos de avaliação de usabilidade; a seção 3 aborda em detalhes o processo de avaliação baseado na captura automática diretamente da aplicação; a seção 4 descreve algumas metodologias para a concepção de interfaces; a seção 5 apresenta o modelo proposto; a seção 6 ilustra a aplicação do modelo para a construção de aplicações piloto, nas quais medidas de usabilidade podem ser efetuadas e, finalmente, a seção 7 apresenta as conclusões e perspectivas do trabalho.

## 2. AVALIAÇÃO DE USABILIDADE

A usabilidade dos dispositivos convergentes e dos seus aplicativos associados é um fator determinante do sucesso deste segmento. As necessidades e características do usuário, o contexto de uso, as características dos aplicativos e as limitações físicas são fatores que influenciam a interação e devem ser considerados tanto no projeto das interfaces quanto na avaliação de usabilidade [2].

A grande penetração desses dispositivos convergentes no mercado significa que, cada vez mais pessoas de diferentes perfis são potenciais usuários dessas tecnologias. Aliado a este fato, a penetração destes dispositivos nas empresas como instrumentos de trabalho, tem levado vários fabricantes e institutos de pesquisa a se preocuparem cada vez mais com aspectos de usabilidade de produtos desta natureza [14].

Para uma contextualização, a Tabela 1 apresenta os diferentes aspectos envolvidos nos métodos de avaliação de usabilidade, conforme a categorização de Preece *et al* [13].

No mesmo sentido, os principais métodos de avaliação existentes, seguindo a estruturação de Barbosa e Silva [1], são apresentados na Tabela 2. Essas metodologias são largamente utilizadas em avaliações de usabilidade para cenários convergentes e suas descrições servirão para posicionar esse trabalho quanto à metodologia adotada e suas peculiaridades.

Mesmo com vários métodos e aspectos de avaliação existentes, a maior parte das avaliações é executada em laboratório,

descontextualizado do ambiente onde o aplicativo será usado, o que pode influenciar diretamente nos resultados obtidos [2]. Por exemplo, cenários que envolvem a utilização de aplicativos para dispositivos convergentes normalmente envolvem pessoas em movimento, interagindo simultaneamente com outras pessoas e realizando tarefas diversificadas em paralelo, o que dificilmente pode ser reproduzido com perfeição em laboratório [14].

**Tabela 1. Diferenças entre os métodos de avaliação.**

Aspectos	Diferenças
Momento da Avaliação	Durante o ciclo de desenvolvimento
	Após o produto estar pronto
Técnica para Coleta de Dados	Opinião dos usuários
	Observação dos usuários em ambiente real
	Observação dos usuários em laboratório
Tipo de Dado Coletado	Coleta da opinião de especialistas
	Quantitativo
	Qualitativo
Tipo de Avaliação	Análise preditiva feita por especialistas de IHC
	Análise interpretativa obtida durante a interação do usuário.
	Análise experimental de dados coletados em ambientes controlados (laboratórios).

**Tabela 2. Descrição dos métodos de avaliação.**

Método de Avaliação	Descrição
<b>Analíticos – Não utilizam a participação de usuários</b>	
Avaliação Heurística [11]	Visa identificar problemas de usabilidade conforme um conjunto de heurísticas.
Percurso Cognitivo [11]	Avalia uma proposta de projeto de IHC no contexto de tarefas específicas.
Inspeção Semiótica [18]	Avalia a qualidade de comunicabilidade de uma interface de acordo com o ponto de vista do designer.
<b>Empíricos – Utilizam dados coletados junto aos usuários</b>	
Teste de Usabilidade [15]	O objetivo é avaliar a interação do usuário de acordo com critérios previamente definidos. Normalmente, os dados para comparação são capturados durante a interação do usuário com o sistema.
Avaliação de Comunicabilidade [11]	Avalia a qualidade de comunicabilidade de uma interface de acordo com o ponto de vista do usuário.
Prototipação em Papel [17]	Avalia a usabilidade de um protótipo representado em papel, por simulações de uso com a participação de usuários.

Por outro lado, avaliações realizadas em campo, embora contextualizem os experimentos ao cenário real de uso das aplicações, normalmente requerem disponibilidade de tempo para serem realizadas, além de demandarem grande parte dos recursos financeiros e de pessoal alocados ao projeto [20].

Para contribuir com a solução do problema, este artigo propõe um modelo baseado no método de captura automática das ações do usuário diretamente da aplicação. Com isso, a avaliação pode ser realizada após o produto estar pronto ou em fase de testes,

provendo a coleta de dados no ambiente natural do usuário final, com a capacidade de capturar dados quantitativos e possibilitar avaliações interpretativas.

### 3. CAPTURA AUTOMÁTICA DE DADOS DIRETAMENTE DA APLICAÇÃO

A Captura Automática Diretamente da Aplicação, também denominada captura automática de uso real do sistema por Nielsen e Mack [11] ou coleta contínua de dados sobre o desempenho do usuário por Shneiderman e Plaisant [16], é uma estratégia de avaliação de usabilidade que consiste na monitoração e coleta automática de estatísticas relativas ao uso de aplicativos.

Segundo Downtown [4], os dados monitorados consistem de eventos disparados pelos usuários ao interagir com um aplicativo, implicando respostas correspondentes do sistema, bem como a captura automática do relógio do sistema, visando à coleta de informações sobre a temporização dos eventos. Esses eventos permitem identificar padrões de comportamento de usuários, tais como, velocidade de conclusão de tarefas, taxas de erros, e frequência de uso de mecanismos de ajuda. Este tipo de dado é extremamente valioso para alterações em procedimentos operacionais, criação de planos de expansão, otimizações de tarefas e personalizações para determinados grupos de usuários.

A captura de informações sobre o uso real de um aplicativo se mostra particularmente útil para fins de usabilidade porque reflete o desempenho dos usuários ao executarem tarefas cotidianas. Além disso, é possível coletar dados relativos a um grande número de usuários, atuando em diferentes circunstâncias, de um modo discreto e "transparente" [11].

Os benefícios advindos da monitoração automática não se estendem apenas aos comandos e outros recursos do sistema de uso corrente. Recursos não utilizados ou raramente acessados poderão ser analisados com propósitos de otimização e/ou de implementação de mecanismos que os tornem mais acessíveis ao usuário [11][16]. Em última instância, as análises feitas sobre tais recursos poderão conduzir à sua substituição por outros mais eficazes ou à sua remoção definitiva.

Um grande inconveniente desta estratégia é que apenas ações diretas podem ser monitoradas, já que nenhuma informação subsidiária é registrada para respaldar a análise de outras atividades de trabalho do contexto avaliado [4]. Em outras palavras, o método pode mostrar o que os usuários fizeram, mas não porque o fizeram [11]. Este é um dos problemas que a inteligência artificial e áreas afins poderiam ajudar a resolver.

Outra dificuldade do método se refere ao volume de dados a serem capturados, transmitidos e analisados. Entretanto, com a evolução das ferramentas OLAP e ferramentas de mineração de dados, o problema pode ser contornado [5].

O último problema reportado se refere à necessidade de incorporar ao código fonte das aplicações componentes de medida que possibilitem quantificar as ações dos usuários. Normalmente, este código se torna intrusivo no contexto das aplicações [9]. Esse problema pode ser solucionado com a utilização da Programação Orientada a Aspectos [7][9]. Com a instrumentação do sistema para captura automática, torna-se fácil manter coletas de dados por extensos períodos de tempo. Se facilidades de análise estatística também forem automatizadas, torna-se possível o processo de captura automática de dados como uma estratégia de

alerta à equipe de usabilidade sobre quaisquer alterações nas necessidades da comunidade usuária do sistema, traduzidas por mudanças no modo como utilizam o aplicativo [11].

A implementação da captura automática e análise dos dados de interatividade exige ainda que métricas de usabilidade para quantificar as interações dos usuários sejam especificadas. Mais ainda, é importante que sejam disponibilizadas APIs que facilitem a implementação de códigos específicos para a captura dos eventos de interação [11].

O modelo proposto neste artigo está baseado nesta forma de avaliação, a partir de métricas de usabilidade criadas com Programação Orientada a Aspectos (POA). A POA é uma boa alternativa para que o código funcional do aplicativo a ser avaliado não seja poluído por códigos que dizem respeito à captura de informações de usabilidade, possibilitando ainda, que essas medições possam ser disponibilizadas ou retiradas da aplicação de acordo com os interesses da equipe de avaliação. Essa característica é o principal fator motivador para a utilização de POA já que outras metodologias advindas da Engenharia de Software como, por exemplo, a utilização de padrões de projetos, não cumpre esse papel com a eficácia desejada [18].

### 4. METODOLOGIAS PARA DESCRIÇÃO DE TAREFAS EM INTERFACES

A partir da necessidade de se desenvolver aplicações utilizáveis e acessíveis a um grande número de usuários, a proposta de metodologias de concepção e desenvolvimento de interfaces tornou-se premente. Um ponto em comum da maioria destas metodologias é a representação de tarefas que o usuário desempenha ou deve desempenhar ao interagir com um aplicativo. Balbo *et al* [3] apresentam uma taxonomia das metodologias de descrição de tarefas, analisando e comparando detalhadamente as metodologias MAD, DIANE+, UAN, FLOW e GOMS. Em geral, estas metodologias realizam a decomposição hierárquica de uma tarefa principal em subtarefas, até atingir tarefas elementares. Estas tarefas permitem entender o funcionamento cognitivo dos operadores e objetos relacionados com a realização de uma determinada atividade na interface, cujo resultado pode ser aplicado para apoiar as ações do projetista em pelo menos quatro momentos: (i) na especificação do sistema; (ii) no projeto da interface; (iii) na elaboração de manuais de treinamento e (iv) para a avaliação da execução de tarefas [10].

Em muitos casos as metodologias são usadas como apoio para o desenvolvimento de ferramentas que geram automaticamente interfaces a partir da especificação das tarefas. Entretanto, elas podem ser aproveitadas como base para prover a avaliação da usabilidade de aplicativos.

No modelo proposto neste trabalho, utilizam-se descrições semelhantes à metodologia MAD para prover uma unidade de mapeamento das intenções do projetista. Esta unidade tem a finalidade de possibilitar que sejam descritos os percursos disponíveis para completar as tarefas existentes no aplicativo. Desta forma, os métodos que são executados para a realização das tarefas podem ser mapeados e posteriormente usados como pontos de atuação de métricas de avaliação de usabilidade.

Na metodologia MAD as tarefas são decompostas hierarquicamente e ligadas por operadores que representam as relações temporais e lógicas. As tarefas são mapeadas em árvores, onde os nós representam as subtarefas. As árvores são associadas

a tabelas que representam atributos indicando: o estado inicial, atual e final das tarefas; se a tarefa é interativa ou não; se é opcional ou obrigatória; se pode interromper ou ser interrompida por outras tarefas.

## 5. O MODELO PROPOSTO

Com base nas discussões das seções anteriores, o modelo aqui apresentado, usa as ideias advindas dos formalismos para a descrição de tarefas, com o objetivo de construir uma unidade responsável em mapear as tarefas disponíveis em um aplicativo, de acordo com a visão de seu projetista. Por outro lado, para obter os dados referentes à captura da usabilidade por parte do usuário final, utiliza-se o método de Captura Automática de Dados Diretamente da Aplicação. O objetivo é confrontar se um usuário interage com a interface da aplicação de maneira conforme com as intenções do seu projetista. Isto possibilita que avaliações quantitativas sejam realizadas de forma a melhorar a interface, tendo como base dados reais de usabilidade, contextualizadas ao cenário natural do usuário.

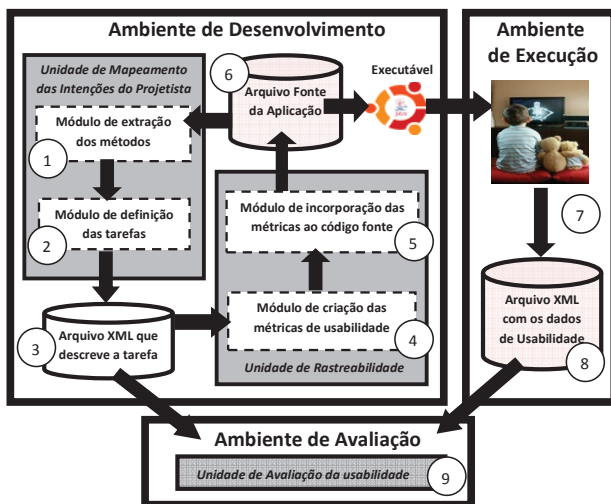


Figura 1. Modelo de avaliação de usabilidade proposto.

Como pode ser observado na Figura 1, o modelo está estruturado em três ambientes distintos:

- Ambiente de Desenvolvimento:** é composto pela unidade de mapeamento das intenções do projetista e da unidade de rastreabilidade. São essas unidades que possuem a lógica do modelo para prover o tratamento das intenções do projetista e captura das informações de usabilidade.
- Ambiente de Execução:** corresponde ao cenário e plataforma onde o aplicativo irá executar. Nesse ambiente, temos a interação do usuário final e a captura das suas ações através de métricas de usabilidade incorporadas ao código executável do aplicativo. O resultado da interação é armazenado em um arquivo XML para posterior utilização no Ambiente de Avaliação.
- Ambiente de Avaliação:** é o ambiente responsável em receber, através de arquivos XML, as intenções do projetista e os dados de usabilidade referentes à interação do usuário. A partir desses arquivos, é possível realizar comparações para identificar se o aplicativo está sendo usado em conformidade com as expectativas do projetista.

## 5.1. Unidade de Mapeamento das Intenções do Projetista

Nessa unidade foi previsto que o projetista de uma aplicação deve dispor de duas funcionalidades básicas, a primeira é identificar o caminho para a execução de uma determinada tarefa; a segunda se refere à possibilidade de definição dos valores para os parâmetros de usabilidade, para serem comparados posteriormente com os valores capturados na unidade de rastreabilidade.

Para contemplar a primeira funcionalidade foi proposto um módulo de extração de métodos das classes relacionadas com a interface das aplicações, possibilitando que o projetista possa indicar, através da seleção dos métodos, o caminho para a execução de uma tarefa. Além disso, foram previstos operadores que podem prover, para as ações, relação de ordem, sincronismo e funções especiais. Estes operadores estão descritos na Tabela 3 e exemplificados na Figura 2.



Figura 2. Representação do mapeamento de uma tarefa.

Além dos operadores apresentados na Tabela 3, o modelo prevê a especificação de níveis que indicam a profundidade atual no caminho da execução de uma tarefa.

Tabela 3. Operadores usados na descrição de ações para o mapeamento de tarefas.

Operador	Descrição
<b>Operadores relacionais</b>	
OR	Pelo menos uma ação deve ser executada.
XOR	Apenas uma das ações deve ser executada.
AND	Todas as ações devem ser executadas não importando a ordem.
<b>Operadores de sincronismo</b>	
SIM	Ações que podem ser executadas simultaneamente de forma independente.
PAR	Ações que podem ser executadas concorrentemente, com pontos de sincronismo.
<b>Operadores especiais</b>	
HLP	Usado para a identificação de ação referente a ajuda.
BCK	Usado para a identificar ações de retrocesso.
EXT	Usado para identificar ações de abandono da tarefa.

O mapeamento de uma suposta tarefa de visualização de fotos em um álbum será utilizado como exemplo para facilitar o entendimento da proposta (ver Figura 2). O Nível 0 contém a ação inicial da tarefa (Álbum de fotos). Para ir ao Nível 1, basta executar a única ação mapeada no Nível 0. No Nível 1 são apresentadas duas opções de álbuns: o de família e o de famosos. Para a exibição do álbum desejado, é necessário escolher uma das duas opções, o que é caracterizado pelo operador OR. Quando uma das duas opções é escolhida, a tarefa é completada. Em



paralelo aos Níveis 1 e 2, é permitida a execução de ações especiais, representadas por operadores como o de ajuda (HELP, operador HLP), retorno ao nível anterior (BACK, operador BCK) ou abandono da tarefa (EXIT, operador EXT).

Nesta unidade está previsto que os métodos escolhidos para a execução da tarefa, os operadores usados, a representação dos níveis de profundidade e os dados referentes ao desempenho esperado pelo projetista com relação à execução da tarefa sejam gravados em um arquivo XML (ver Figura 3), para o uso nas outras unidades do modelo.

```
<?xml version="1.0" encoding="UTF-8" ?>
<task>
  <level> <method>
    <name>private void album()</name>
    </method>
  </level>
  <level> <method>
    <name>private void familyAlbum()</name>
    <operator>OR</operator>
    <name>private void famousAlbum()</name>
    </method>
    <back>private void backLevel(int)</back>
    <help>private void help(String)</help>
    <exit>private void exit()</exit>
  </level>
  <number>
    <field>Percentage_of_non_conformities</field>
    <value>3</value>
    <field>Percentage_of_dropouts</field>
    <value>3</value>
  </number>
</task>
```

Figura 3. Trecho de Arquivo XML referente a uma tarefa.

Para dar suporte às atividades relacionadas a esta unidade, foi criada a ferramenta *Mapping Task*, que contempla todas as funcionalidades descritas anteriormente, permitindo que o projetista possa mapear as tarefas e determinar os valores de usabilidade previstos para a execução da tarefa, de forma fácil e eficiente. A interface da ferramenta é apresentada na Figura 4.

O uso da *Mapping Task* começa com a escolha do caminho e o nome da classe que será mapeada. A ferramenta se encarregada de carregar todos os métodos utilizando a biblioteca *java.lang.reflect* da linguagem Java. Posteriormente, basta escolher os métodos e operadores relacionados a cada nível para realizar o mapeamento da tarefa. O mapeamento termina com a definição dos valores de usabilidade esperados para a tarefa e a geração do arquivo.

### 5.2.Unidade de Rastreabilidade

A unidade de rastreabilidade tem como propósito investigar as ações de interação realizadas pelo usuário final para a realização das tarefas disponibilizadas pelo aplicativo. Para oferecer essa funcionalidade, a abordagem seguida neste trabalho propõe a utilização da metodologia de Captura Automática Diretamente da Aplicação, tomando como base a utilização de métricas de usabilidade, que inicialmente foram propostas por Nielsen [11], adaptadas para a utilização em plataformas WIMP por Moldovan e Tarta [9] e estendidas para a utilização em plataformas convergentes por Kronbauer e Santos [7][8]. As métricas utilizadas estão descritas na Tabela 4, a seguir.

Tabela 4. Métricas de Avaliação.

Métricas	Objetivos
<b>Métricas para avaliação da assertividade (%)</b>	
Ações em conformidade	Quantificar a dificuldade de interação do usuário na tarefa usando a interface.
Ações sem conformidade	
<b>Métricas relacionadas à busca e utilização de tarefas (%)</b>	
Desistências de tarefas	A desistência pode significar que o usuário está perdido ou à procura da tarefa desejada.
Tarefas concluídas	
Tarefas concluídas e utilizadas	
Tarefas concluídas e não utilizadas	A utilização ou não das tarefas indica se o usuário encontrou o que estava procurando.
<b>Métricas relacionadas com o desempenho (%)</b>	
Tempo médio para completar uma tarefa	Indicar possíveis necessidades de melhorias que minimizem o tempo e ações para a execução de tarefas.
No. médio de ações para completar a tarefa	
No. de tarefas concluídas com erros no caminho	
<b>Métricas relacionadas à utilização de ajuda (%)</b>	
Frequência de uso do help	Mensurar o nível de dificuldade enfrentada pelo usuário.
Inconformidades cometidas após o acionamento do help	Identificar se a ajuda disponível proporciona o efeito esperado.

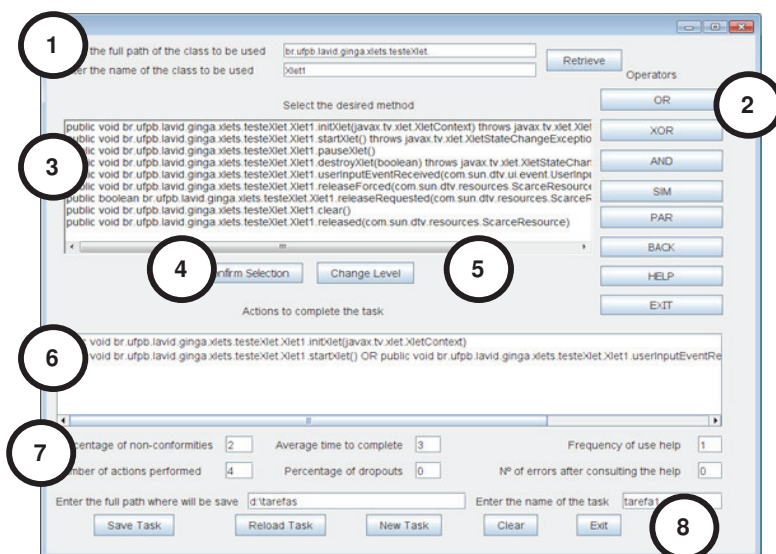
O modelo prevê que as métricas sejam implementadas utilizando o paradigma POA, considerando três fatores principais:

- Extensibilidade: possibilidade de adicionar métricas de rastreabilidade, sem grande esforço de programação.
- Reusabilidade: facilidade para que métricas existentes possam ser usadas em diferentes contextos e aplicações.
- Separação de interesses: em especial, separação do interesse de “medição” (código não funcional) das funcionalidades do aplicativo (código funcional), de modo a facilitar a manutenção.

Para dar suporte a essa unidade, foi construída a ferramenta *Creation Usability Aspect*, que possibilita a criação de métricas de usabilidade de forma automática. Para isso, a ferramenta utiliza como entrada de dados o arquivo XML criado na Unidade de Mapeamento das Intenções do Projetista, com o objetivo de capturar os métodos, operadores e níveis relacionados a execução da tarefa. De posse dessas informações, são usadas estruturas desenvolvidas na concepção da ferramenta para gerar os aspectos de usabilidade.

O resultado final do uso dessa unidade é a geração de um arquivo com o aspecto de usabilidade, contendo os pointcuts e advices relacionados à tarefa que será rastreada. Além disso, os métodos responsáveis em tratar a persistência dos dados são incorporados a esse aspecto, permitindo que as informações capturadas possam ser lidas e gravadas em um arquivo XML. Esse arquivo é utilizado pela unidade de Avaliação para comparação com o arquivo XML resultante da Unidade de Mapeamento das Intenções do Projetista.

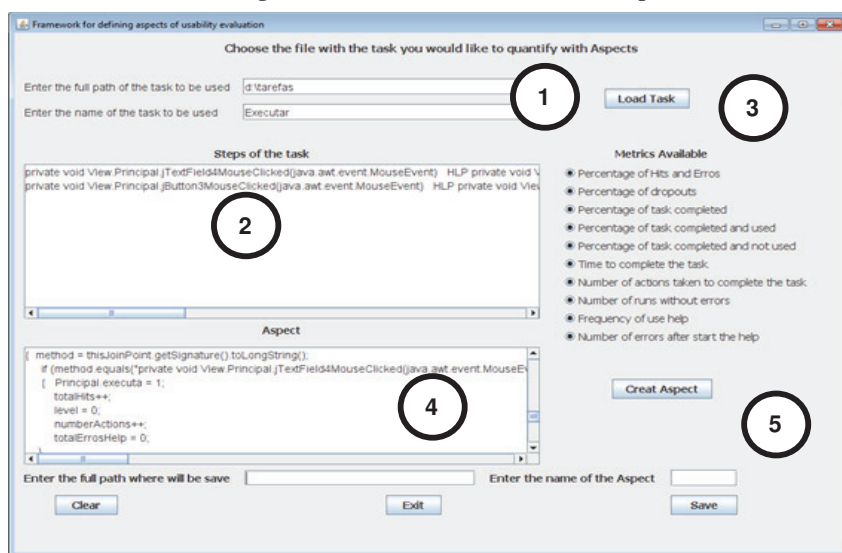
Para a utilização do aspecto gerado, basta incorporar esse aspecto ao projeto do aplicativo que será rastreado e recompilá-lo utilizando o AspectJ. A interface da ferramenta *Creation Usability Aspect* pode ser visualizada na Figura 5.



### Legenda

- 1) Localização da classe a ser usada
- 2) Operadores de descrição de tarefas
- 3) Apresentação dos métodos da classe
- 4) Opção de seleção de métodos
- 5) Opção de troca de nível
- 6) Apresentação do mapeamento da tarefa
- 7) Definição dos valores de usabilidade esperados
- 8) Funcionalidades da ferramenta

Figura 4. Interface da ferramenta de mapeamento de tarefas (*Mapping Task Tool*).



### Legenda

- 1) Localização da tarefa a ser usada
- 2) Apresentação do mapeamento entre tarefa e método
- 3) Métricas disponíveis
- 4) Apresentação do Aspecto de usabilidade gerado
- 5) Funcionalidades da ferramenta

Figura 5. Ferramenta de geração de aspectos (*Creation Usability Aspect Tool*).

## 6. ESTUDOS DE CASO

Dois estudos de caso são apresentados para verificar a viabilidade do uso do modelo proposto e avaliar as ferramentas que dão suporte à sua implementação.

O primeiro caso corresponde a um aplicativo desenvolvido para TV interativa, utilizando as APIs do Ginga-J e foi desenvolvido exclusivamente para permitir as validações preliminares do modelo proposto nesse trabalho. A aplicação interativa consiste basicamente de um menu que oferece uma série de serviços complementares ao programa de TV. Esses serviços são acessados a partir da interação do usuário com o menu (Figura 6).

O segundo caso se refere a um aplicativo que é executado em dispositivos sobre a plataforma *Android* e cujo código está

disponível na Web<sup>1</sup>. A finalidade do aplicativo é dar suporte aos usuários no controle de gastos com combustíveis de automóveis. A interface do aplicativo, ilustrada na Figura 7, consiste de um menu com 4 botões, cada um deles permitindo acessar as funcionalidades (tarefas do usuário) associadas ao controle de consumo/gasto, histórico, estatística e gráficos.

A adoção desse segundo aplicativo está fundamentada em dois aspectos importantes para o processo de validação do modelo:

- (i) Mostrar que o modelo pode ser utilizado num cenário de convergência, variando de outra plataforma.
- (ii) Mostrar que, apesar do processo de concepção do aplicativo não possuir qualquer ligação com o modelo de avaliação de

<sup>1</sup> Disponível em <http://evancharlton.com/projects/mileage>

usabilidade proposto, métricas de usabilidade podem ser integradas ao código.



Figura 6. Aplicativo para Plataforma GINGA-J.

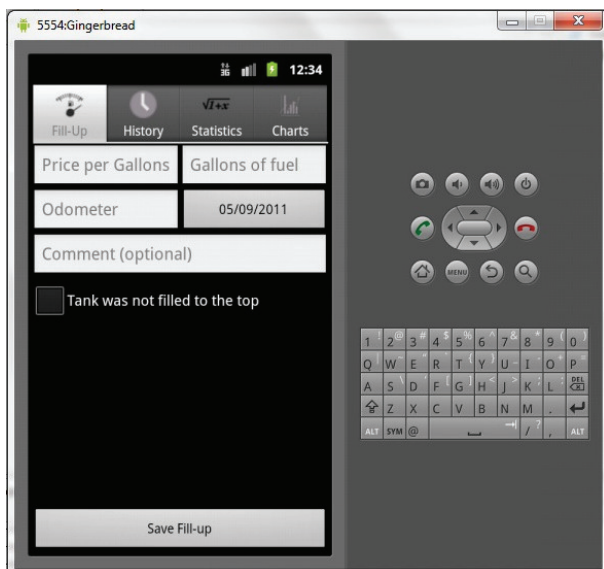


Figura 7. Aplicativo para plataforma Android.

As avaliações preliminares mostraram que o modelo é aderente às plataformas testadas, uma vez que foi possível tanto realizar o mapeamento das tarefas, bem como acoplar métricas de avaliação à aplicação, permitindo a realização de testes preliminares de interações de usuários sobre a interface. Um dos trunfos do modelo é a possibilidade de contextualização das métricas a uma determinada plataforma e a diferentes perfis de usuários.

O mapeamento das tarefas pelo uso de métodos desenvolvidos no código fonte não é simples como se esperava, obrigando que o responsável pela avaliação de usabilidade conheça o código do aplicativo a ser avaliado. Isso ficou claro no caso do aplicativo para celular, já que o código fonte era totalmente desconhecido até a avaliação. Entretanto, vale destacar que o processo proposto prevê que o próprio projetista do aplicativo realize o mapeamento.

Com relação à inclusão automática de métricas no código, foram necessários alguns ajustes até que as métricas geradas fossem consistentes e gerais o suficiente para serem acopladas a qualquer aplicação. O ponto positivo é que a ferramenta *Creation Usability Aspect* faz a geração automática sem qualquer tipo de intervenção do usuário no código gerado, o que viabiliza o uso do modelo por pessoas que não possuem familiaridade com a POA.

O acoplamento dos aspectos gerados ao código dos aplicativos a serem avaliados é feita por meio de uma nova compilação do projeto utilizando o AspectJ e agora incluindo os aspectos gerados. Além disso, todas as bibliotecas inerentes à plataforma que executará o aplicativo devem fazer parte do projeto.

Não foram observadas variações no desempenho dos aplicativos, apesar da necessidade de processamentos adicionais para a manipulação de arquivos e a execução dos aspectos de rastreabilidade. Entretanto, devem ser realizados estudos mais aprofundados a fim de verificar a influência da instrumentação realizada com os aspectos no código final da aplicação avaliada.

Com relação à confrontação das informações das intenções do projetista com os dados capturados, fica claro que, apesar dos arquivos XML permitirem uma fácil interação e visualização por ferramentas apropriadas, a dificuldade de avaliação será proporcional ao número de tarefas mapeadas e ao tamanho da aplicação. Nesse sentido, o uso de ferramentas de processamento e visualização dos dados obtidos é fundamental para análises de aplicações de maior complexidade.

Os estudos de caso foram importantes para identificar possíveis ajustes e correções nas ferramentas desenvolvidas. Além disso, ficou constatado que a forma proposta para o mapeamento das intenções do projetista é de difícil execução, mesmo com o suporte da ferramenta *Mapping Task*. Esta observação provocou uma ligeira mudança do processo de avaliação apresentado neste trabalho. Em [8], os autores propõem um novo processo de mapeamento de tarefas baseado na captura dos passos realizados pelos engenheiros de testes ou especialistas (*experts*), ao interagirem diretamente com a aplicação a ser avaliada. Esta captura permite o mapeamento automático da sequência de ações a serem realizadas para completar uma tarefa, que seria supostamente ideal, uma vez que foi definida por um especialista.

## 7. CONCLUSÕES E PERSPECTIVAS

Dispositivos convergentes requerem que as interfaces de seus aplicativos estejam adaptadas às suas características, potencializando alta qualidade de uso. Apesar da existência de metodologias que auxiliam os projetistas a criarem seus aplicativos, somente pela avaliação efetiva de usabilidade é que se pode concluir que realmente o *layout* da interface alcança as expectativas do projetista e as necessidades dos usuários.

Para que seja possível realizar testes coerentes e contextualizados ao uso cotidiano dos aplicativos, é necessário submetê-los a avaliações em seu ambiente natural de uso pelos usuários. Isso porque, normalmente, os usuários estão inseridos em cenários com diversas interferências do ambiente que dificilmente podem ser reproduzidas em laboratório. Como alternativa para contornar o problema, este artigo apresentou um modelo para a avaliação de usabilidade das aplicações disponibilizadas em plataformas convergentes. O modelo é baseado no mapeamento de tarefas, possibilitando que o projetista possa descrever suas intenções em termos de usabilidade. Por outro lado, são realizadas medições,

através de métricas de usabilidade, para capturar a interação do usuário final com a aplicação. De posse dessas informações, pode ser avaliada a efetiva qualidade da interface.

As principais contribuições que podem ser destacadas nesse trabalho são: (i) a especificação de um módulo de mapeamento das tarefas, permitindo descrever as intenções dos projetistas, e a criação de uma ferramenta para prover esse serviço; (ii) a definição de métricas, que quantificam a interação de um usuário com a aplicação; (iii) a utilização de POA, que possibilita a separação do código funcional da aplicação do código referente à captura de informações de usabilidade; (iv) a criação de uma ferramenta para a geração automática de aspectos de usabilidade, permitindo que o modelo possa ser usado sem a necessidade do usuário conhecer POA; (v) a comprovação da aderência do modelo a duas plataformas diferentes, TVDI usando as APIs do Ginga-J e dispositivos móveis usando as APIs do *Android*.

Os conceitos e resultados apresentados neste artigo devem ser utilizados em trabalhos posteriores que envolvam as seguintes atividades: (i) aprimorar a ferramenta de mapeamento das tarefas para que se torne mais interativa (ii) desenvolver novas métricas, que possam ser contextualizadas a diferentes plataformas; (iii) criar uma ferramenta de apoio a interpretação dos dados obtidos após a utilização do modelo; (iv) usar os resultados da avaliação de usabilidade para realizar ajustes na interface automaticamente e; (v) efetuar testes de usabilidade de aplicações reais com um número suficientemente grande de usuários, com diferentes perfis e em diferentes plataformas convergentes, com a finalidade de validar a aplicabilidade do modelo.

## 8. AGRADECIMENTOS

Os autores agradecem à CAPES pelo apoio através dos Projetos “CODES – Cooperação Acadêmica em Engenharia de Software e Sistemas para TV Digital” (PROCAD-NF Edital 21/2009) e “xTVD – Explorando novas fronteiras na TV Digital Interativa” (PROCAD-NF Edital 08/2008).

## 9. REFERÊNCIAS

- [1] Barbosa, S. D. J.; Silva, B. S. *Interação Humano-Computador*, 1 ed. Elsevier: Rio de Janeiro, 2010.
- [2] Betiol, A. H.; Cybis, W. A. Usability Testing of Mobile Devices: A Comparison of Three Approaches. *INTERACT 2005, Lecture Notes in Computer Science*, 2005.
- [3] Balbo, S.; Ozkan, N.; Paris, C. Choosing the right task modelling notation: A Taxonomy. *In: The Handbook of Task Analysis for Human-Computer Interaction*, D. Diaper and N. Stanton (Eds.), Lawrence Erlbaum Associates (LEA), 2004.
- [4] Downtown, A. C. *Evaluation techniques for human – computer systems design*. London: McGraw-Hill, 1991.
- [5] González, M.P., Lorés, J., Granollers, T. Enhancing usability testing through data mining techniques. *Information Software Technology*, 2007.
- [6] Ivory, M. Y.; Hearst, M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Computer Survey*, v.33, n.4, Dec. 2001, p. 470-516.
- [7] Kronbauer, A. H.; Santos, C. A. S. Um Modelo para Avaliação de Usabilidade de Aplicações Imperativas em GingaJ. *In: InfoBrasil 2011, Fortaleza*, 2011.
- [8] Kronbauer, A. H.; Santos, C. A. S. Um modelo de avaliação da usabilidade baseado na captura automática de dados de interação do usuário em ambientes reais. *In: X IHC & V CLIHC, Porto de Galinhas, PE - BR, Oct. 2011*. 8p. (to be published).
- [9] Moldovan, G. S.; Tarta, A., Automatic Usability Evaluation using AOP. *In: Proc. 2006 IEEE-TTTC Int. Conf. on Automation, Quality and Testing, Robotics, TOME II*, pp. 84-89, Cluj-Napoca, May 25-28, 2006.
- [10] Mori, G.; Paternò, F.; Santoto, C. CTTE: Support for Developing and Analysing Task Models for Interactive System Design. *IEEE Transactions on Soft. Eng.* 2002.
- [11] Nielsen, J.; Mack, R. L. *Usability Inspection Methods*. John Wiley & Sons, New York, 1994.
- [12] Norman, D. A. *Cognitive Engineering – User-Centered System Design*. NJ: Lawrence Erlbaum Associates, 1986.
- [13] Preece J.; Rogers, Y.; Sharp, E. *Interaction Design: Beyond HCI*. John Wiley & Sons, New York, 2002.
- [14] Queiroz, J. E. R.; Ferreira, D. S. A Multidimensional Approach for the Evaluation of Mobile Application User Interfaces. *Human-Computer Interaction, New Trends. Lecture Notes in Computer Science*, 2009.
- [15] Rubin, J. and Chisnell, D. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Test*, 2ed. Indianapolis, IN:Wiley Publishing, Inc., 2008.
- [16] Shneiderman, B.; Plaisant, C. Designing the User Interface: Strategies for Effective Human-Computer Interaction. *In: 4th Ed.*, Addison-Wesley Publ. Co., MA, 2004.
- [17] Snyder, C. *Paper Prototyping: the fast and easy way to design and refine user interfaces*. Morgan Kaufmann Publishers, 2003. ISBN 1558608702.
- [18] Souza, C. S. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, MA: The MIT Press, 2005.
- [19] Tarby, J. C.; Ezzedine, H.; Rouillard, J.; Tran, C. D., Laporte, P. and Kolski, C. Traces using aspect oriented programming and inter active agent based architecture for early usability evaluation: basic principles and comparison. *In: Proc. 12th Inter. Conf. on HCI interaction: Interaction design and usability*, Springer-Verlag, Berlin, 2007.
- [20] Weiss, S. *Handheld Usability*. England: John Wiley & Sons Ltd, 2002.