

# Adaptação nativa de interfaces com o utilizador em dispositivos móveis

Sandra Rodrigues<sup>1</sup>, José Creissac Campos<sup>2</sup>, António Nestor Ribeiro<sup>2</sup>

DI/CCTC

Universidade do Minho – Braga

<sup>1</sup> [sandra.maria.rodrigues@gmail.com](mailto:sandra.maria.rodrigues@gmail.com)

<sup>2</sup> [{jose.campos,anr}@di.uminho.pt](mailto:{jose.campos,anr}@di.uminho.pt)

<http://sim.di.uminho.pt>

**Abstract.** Neste artigo apresentamos o trabalho que tem vindo a decorrer no desenvolvimento de uma *framework* que permita a adaptação de interfaces com o utilizador directamente no ambiente nativo dos dispositivos, fornecendo assim ao utilizador interfaces de aplicações adequadas ao dispositivo, e ao estilo de interacção a que o utilizador está habituado.

## 1 Introdução

No mundo móvel e ubíquo uma das realidades incontornáveis é a heterogeneidade dos dispositivos que os utilizadores podem utilizar para aceder à mais diversificada informação. Essa heterogeneidade dificulta a criação de interfaces a disponibilizar ao utilizador e tem vindo a obrigar a que a concepção e desenvolvimento das aplicações evolua, por forma a acompanhar a evolução do *desktop* para os dispositivos móveis.

Neste contexto, as duas principais famílias de soluções existentes são:

- Soluções *browser based*, onde existe um componente servidor, que entrega conteúdos aos dispositivos de forma mais ou menos acertada;
- Soluções baseadas no desenvolvimento de um conjunto de interfaces nativas que são fornecidas a famílias de dispositivos, conforme apropriado.

Estas soluções apresentam algumas desvantagens. A primeira solução é limitada no grau de adaptação que possibilita quando comparada com soluções nativas ao dispositivo. A segunda solução permite fornecer uma melhor adaptação, mas tem problemas de manutenção, desenvolvimento e, conseqüentemente, custos.

Face às limitações e desvantagens das soluções existentes, é objectivo deste trabalho a criação de uma *framework* que permita aceitar aplicações e adapte as suas interfaces a cada dispositivo de uma forma eficiente. Com este intuito a *framework* permite fornecer UIs (*User Interfaces*) tendo em conta as características do dispositivo e personalizações realizadas pelo utilizador. A *framework* tira benefício de ser executada no ambiente nativo do dispositivo, utilizando as técnicas de interacção e subseqüentes metáforas de apresentação nele existentes. Assim, será capaz de fornecer interfaces melhor adaptadas ao dispositivo, bem como mais consistentes com as que o utilizador está habituado a ter, sem que para isso seja necessário criar múltiplas interfaces para diferentes dispositivos.

A mais valia da abordagem defendida nesta *framework* é o facto de tornar a adaptação ao dispositivo transparente para o servidor, passando-a de algo que representa uma alteração significativa do contexto de utilização - resolvida pelo servidor -, para um aspecto que pode ser tratado de forma implícita (pelo cliente). Assim, retira-se complexidade à componente servidor à custa da introdução de alguma complexidade adicional nos clientes. Considerando que cada cliente só terá que se conhecer a si próprio, enquanto o servidor teria que conhecer todos os tipos de cliente, em termos líquidos a complexidade global do sistema deverá diminuir.

A adaptação da interface ao dispositivo considera características como por exemplo: técnicas de interacção e estilos de interacção – num telemóvel a interface mais adequada pode ser baseada em navegação por *softkeys*, num PDA essa mesma interface pode recorrer à manipulação directa. Genericamente o modelo e as entidades utilizadas para a *framework* são as apresentadas na Figura 1. As entidades principais da *framework* são a aplicação, o dispositivo (algumas características) e estilos que o utilizador define no dispositivo. O produto entre a aplicação e o tuplo - dispositivo e contexto do utilizador -, é utilizado para fornecer uma interface adaptada ao dispositivo tendo em conta a aplicação.

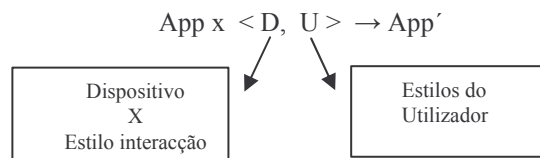


Figura 1. Entidades genéricas da *framework*

## 2 Trabalhos relacionados

A questão da adaptação de UIs tem sido abordada por diversos investigadores a diferentes níveis. A noção apresentada por Thevenin e Coutaz em [1], designada por plasticidade, é das noções que mais se ajusta aos objectivos deste trabalho. Os autores definem plasticidade como “*a capacidade de um sistema interactivo suportar variações de contexto enquanto mantém a usabilidade*”. Foi desenvolvida uma *framework* que suporta o desenvolvimento de UIs plásticas, cuja principal utilidade é permitir produzir múltiplas versões da aplicação, que servirão os requisitos dos dispositivos e dos sistemas, sendo que a aplicação é especificada somente uma vez, o que minimiza os custos de desenvolvimento e manutenção.

Sendín e Lorés, em [6] e [7], apresentam uma extensão ao termo plasticidade definido por Thevenin e Coutaz em [1]. A extensão apresentada consiste na divisão do termo plasticidade em dois conceitos, que deram origem a dois mecanismos com a mesma designação:

- Plasticidade Implícita, mecanismo capaz de detectar mudanças de ambiente/dispositivo e realizar ajustes à UI – tratado do lado do cliente;
- Plasticidade Explícita, mecanismo que permite gerar automaticamente novas UIs em resposta a alterações no contexto e/ou ambiente - tratado do lado do servidor.

A divisão de plasticidade em dois mecanismos complementares permite gerir a adaptação de forma mais eficiente.

Outra abordagem para se obter a plasticidade é a proposta apresentada por Gilroy e Harrison em [5] designada por estilos de interacção. Os autores afirmam que esta é a melhor abordagem para fornecer a interface mais adequada ao dispositivo. O sistema de estilos de interacção apresentado contém suporte para três estilos – *form-fill*, *dialogue* e *menu*. Para utilizar as especificações desenvolvidas para os três estilos foi desenvolvida uma *framework* – *SIS*, que fornece *UIs* aos dispositivos. A especificação da *framework SIS*, separa a especificação de funcionalidades de uma aplicação da especificação da sua interface, e fornece uma selecção de diferentes estilos de interacção, para com isso a interface estar mais próxima das capacidades e limites dos dispositivos.

### 3 A Framework

A *framework* deste trabalho pode ser considerada uma especialização da noção de plasticidade apresentada em [1] e [6].

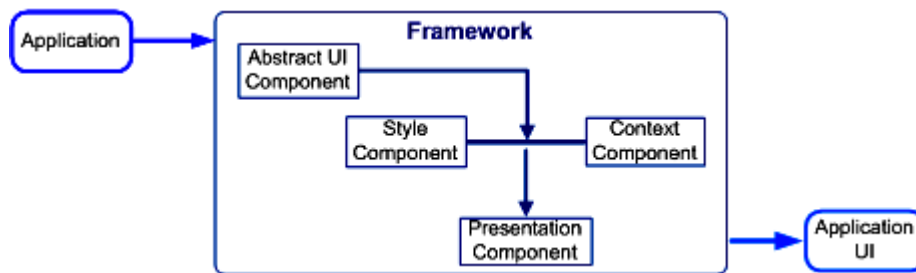


Figura 2. Arquitectura da *framework*

Como mecanismo de estruturação da adaptação da interface utilizamos estilos de interacção. No entanto, a nossa utilização deste mecanismo difere do originalmente proposto em [5]. Nesse trabalho o objectivo é fornecer o mesmo estilo de interacção em diferentes dispositivos, enquanto no nosso caso o objectivo é fornecer a cada dispositivo o estilo de interacção que mais se lhe adequa. O estilo de interacção captura características como softkeys, métodos de acesso e navegação, entre outros. Assim sendo, uma interface utiliza a informação contida no estilo para se adaptar ao dispositivo.

Na Figura 2 é apresentada a arquitectura da *framework* em desenvolvimento. Os componentes que a compõem são:

- *Abstract UI Component* - é o modelo da interface a nível abstracto, os modelos são definidos recorrendo a classes que capturam os diferentes componentes da interface sem recorrer a técnicas de interacção concretas;
- *Style Component* - este componente instância as classes abstractas utilizando as capacidades concretas de cada dispositivo, isto poderá passar por uma reestruturação da interacção (p.ex.: dividindo um formulário em sub-formulários);

- *Context Component* – parametriza o *Style Component*, identificando o dispositivo e as preferências do utilizador;
- *Presentation Component* – este componente conjuga toda a informação e adapta a interface ao dispositivo fornecendo-a ao utilizador.

## 4 Conclusão e Trabalho Futuro

Num contexto de computação móvel, as aplicações necessitam de adaptar as suas interfaces tendo em conta o dispositivo, os requisitos e necessidades das aplicações e utilizadores.

Neste artigo apresenta-se o trabalho em curso no desenvolvimento de uma *framework* que visa aceitar aplicações e adaptá-las ao dispositivo. A interface fornecida ao utilizador da aplicação, será gerada no dispositivo à qual a aplicação tem destino, agregando todos os benefícios que daí advêm. A *framework* tira partido de executar as interfaces no ambiente nativo do dispositivo, fornecendo aos utilizadores interfaces semelhantes às que eles estão habituados a ver no mesmo.

O trabalho futuro passa pela implementação da *framework*, especificando os componentes e modelos necessários. Para realização deste trabalho será considerado um caso de estudo para servir de protótipo. Planeamos também realizar uma avaliação comparativa das interfaces geradas a nível de usabilidade.

## Referências

1. Thevenin, D., Coutaz, J.: Plasticity of User Interfaces: Framework and Research Agenda. In Proceedings Interact 99, IFIP, IOS Press (1999), 110-117
2. Thevenin, D., Coutaz, J., Calvary, G.: A Reference Framework for the Development of Plastic User Interfaces. Proc. Eng. Human-Computer Interaction Conf. (2001) 173-192.
3. Calvary, G., Coutaz, J., Thevenin, D.: Embedding Plasticity in the Development Process of Interactive Systems. HUC, First workshop on Resource Sensitive Mobile HCI, Conference on Handheld and Ubiquitous Computing, HU2K, Bristol, September, (2000)
4. Calvary, G., Coutaz, J., Thevenin, D. et al: Plasticity of User Interfaces: A Revised Reference Framework. Bucharest, Romania, July 18-19 (2002), 127-134
5. Gilroy, S. W., Harrison, M. D.: Using Interaction Style to Match the Ubiquitous User Interface to the Device-to-Hand. Engineering Human Computer Interaction and Interactive Systems (Joint Working Conferences EHCI-DSVIS 2004, Hamburg, Germany, July 11-13, 2004) Springer Lecture Notes in Computer Science Vol. 3425 (2005), 325-345
6. Sendín, M., Lorés, J.: Plasticity in Mobile Devices: a Dichotomic and Semantic View. Proceedings of Workshop on Engineering Adaptive Web, supported by AH (2004), 58-67
7. Sendín, M., Lorés, J., Toni, G. et al: Implicit Plasticity Framework: a Client-Side Reusable Software Architecture for Context-Awareness. Actes del III Taller de *Sistemas Hipermedia Colaborativos y Adaptativos* (SIHICA'2005). CEDI'2005 (2005)