

# De Requisitos a Protótipos de UI: Uma Abordagem Semi Automatizada

[From requirements to UI prototypes: a semi-automated approach]

Rui Couto, António N. Ribeiro & José C. Campos  
Departamento de Informática/Universidade do Minho & HASLab/INESC TEC, Braga, Portugal

**Resumo**—Software development poses multiple challenges, from requirements specification to final software production. We have developed two approaches, focused on complementary aspects of the problem. On the one hand, SCARP supports obtaining architectural models (hence, business layer code) from requirements models. On the other, MODUS produces UI prototypes from architectural models. This work presents their integration to obtain a development cycle going from requirements to executable prototypes, supported by semi-automated processes.

## I. INTRODUÇÃO

A *Model-Driven Development Architecture* (MDA) surge como uma alternativa aos processos tradicionais de desenvolvimento de software. Movendo o foco do código para os modelos, é possível fazer o desenvolvimento a um mais alto nível de abstracção. Uma grande vantagem destas abordagens é a possibilidade de produzir implementações de uma forma automatizada. Em trabalhos anteriores, desenvolvemos a abordagem SCARP [1], uma abordagem baseada em modelos, que inicia o desenvolvimento ao nível dos modelos de requisitos.

Similar à MDA, a *Model-Based User Interface Development* (MBUID) permite a produção de Interfaces com o Utilizador (IUs) com base em modelos. Neste contexto, desenvolvemos a abordagem MODUS [2], [3], que permite, a partir de modelos arquitecturais, produzir protótipos executáveis de IUs.

Este trabalho explora a combinação das duas abordagens. Os outputs produzidos pelo SCARP são compatíveis com os inputs recebidos pelo MODUS. Desta forma, é então possível obter protótipos de IUs, com base em modelos de requisitos.

## II. SCARP

A geração de software baseada em modelos [4] baseia todo o processo de geração de software na especificação e transformação de modelos. As vantagens deste tipo de abordagens são a resolução de problemas relacionados com produtividade, portabilidade, interoperabilidade e manutenção, uma vez que o foco do processo de desenvolvimento de software passa a ser os modelos. As implementações por sua vez, são obtidas através de ferramentas de transformação de modelos. Uma das lacunas da MDA é o facto de iniciar o processo ao nível arquitectural, havendo um hiato entre os requisitos e as arquitecturas.

A abordagem SCARP (*Scenario Based Rapid Software Prototyping*) permite estender as vantagens das abordagens baseadas em modelos até aos modelos de requisitos. Com

base nestes modelos, formalizações de requisitos previamente identificados, é feito um mapeamento para uma ontologia que, por um lado, fornece estruturação de conhecimento e, por outro, permite a análise desse conhecimento. Com base nessas capacidades, é então efectuada inferência de padrões de requisitos, identificando padrões já documentados para os quais existe uma implementação conhecida, e feita a ponte para os padrões arquitecturais. Estes, por sua vez, podem ser instanciados e compostos de modo a obter uma solução arquitectural. Partindo de um conjunto de modelos de requisitos, informação de domínio, e parametrização por parte do utilizador, o SCARP é capaz de produzir um modelo arquitectural que suporta os requisitos que o representam.

## III. MODUS

A abordagem MODUS surge no contexto do MBUID, onde modelos arquitecturais são utilizados para produzir protótipos executáveis de IUs. A *framework* de referência Cameleon [5] define quatro níveis de abstracção na geração de interfaces: modelos de domínio e tarefas, modelos abstractos da IU, modelos concretos da IU e IUs finais. Apesar de complementares às abordagens da secção anterior, as abordagens MBUID utilizam, assim, um conjunto de modelos distinto, o que dificulta a sua integração. Um ponto comum é a utilização de modelos do domínio.

Ao contrário de outras abordagens MBUID que necessitam de uma especificação completa de todos os modelos envolvidos no processo, MODUS baseia-se na reutilização do diagrama arquitectural da lógica de negócio para um dado domínio conhecido. Este diagrama será o ponto de partida, no qual serão identificados padrões ao nível das classes, a partir de informação sobre o domínio da aplicação. Da mesma forma, informação sobre *layouts* é fornecida pelo domínio, assim como *templates*, que podem ser ajustados pelo utilizador. A conjugação destes *inputs*, permite que o MODUS, a partir de um diagrama arquitectural, produza um protótipo de interface executável [3] e de boa qualidade, tal como demonstrado por estudos com utilizadores [6].

## IV. INTEGRAÇÃO

De modo a obter uma abordagem que permita semi-automatizar todo o ciclo de desenvolvimento, os *outputs* da abordagem SCARP são fornecidos como *input* à abordagem

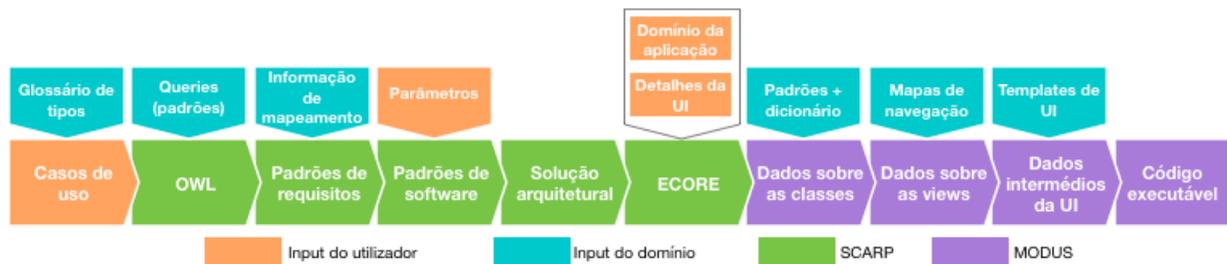


Figura 1. Integração de SCARP com MODUS

MODUS (ver Fig. 1). Isto permite que, com base em modelos de requisitos, sejam produzidos protótipos de IUs executáveis.

Para ilustrar a abordagem resultante, foi realizado um caso de estudo, utilizando como domínio de aplicação a área de comércio electrónico. Foi modelado um conjunto de requisitos, na forma de um modelo de casos de uso, para os seguintes cenários: *Login*, *Register*, *Checkout*, *Add product to cart*, *Browse Products*, *View product*, *Search product*, *Show highlights*, *View operation historic*, *Return home*. Este modelo foi utilizado como ponto de partida para a abordagem SCARP. Nestes requisitos foi possível identificar um conjunto de padrões, os quais foram mapeados para padrões de software. Por exemplo, os padrões arquiteturais *Proxy*, *Singleton* e *Composite* surgiram na solução final em resultado da identificação do padrão de requisitos *Has Shopping Cart*, que denota a existência de um mecanismo de suporte a um carrinho de compras virtual (*shopping cart*). Depois de devidamente instanciados, obteve-se um modelo arquitetural representante dos requisitos previamente especificados. O modelo foi então usado como ponto de entrada para a abordagem MODUS. Foram extraídas informações sobre as classes e relacionamentos, e produzida informação sobre as entidades e as *views*. Neste caso, foi possível identificar a existência de uma representação de um *shopping cart* (tal como identificado na abordagem SCARP). A informação de domínio permitiu obter então as representações de IUs, assim como os respectivos modos de representação. No caso do *shopping cart*, resultou uma página que permite ver o seu conteúdo, e respectivas ligações para esta. Como resultado, foi obtido um protótipo de interface, do qual a página principal é apresentada na Fig. 2.

## V. CONCLUSÕES

Este trabalho explora três dos principais problemas na área do desenvolvimento de software baseado em modelos. Primeiro, o facto das abordagens MDA tipicamente iniciarem o processo de desenvolvimento em modelos arquiteturais, levando a um hiato entre os estes e os requisitos. Segundo, abordagens MBUID tipicamente necessitam da especificação de um grande conjunto de modelos, o que as torna difíceis de usar. Terceiro, foi explorada a viabilidade de integrar abordagens MDA e MBUID, de modo a obter uma cadeia de desenvolvimento completa, desde os modelos de requisitos, até ao software final. Através da abordagem SCARP, foi possível criar um modelo de requisitos onde foi especificado

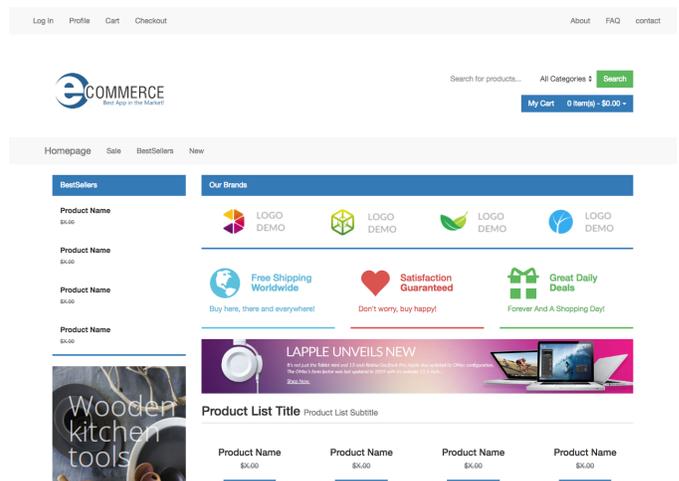


Figura 2. Página principal da interface produzida pelo MODUS

um conjunto de funcionalidades. Utilizando informação sobre o domínio, obteve-se um modelo arquitetural. Este foi depois utilizado como input da abordagem MODUS, que juntamente com detalhes da IU, foi capaz de produzir um protótipo executável, provando assim a viabilidade de conjugar estas duas abordagens.

## AGRADECIMENTOS

Este trabalho é financiado por Fundos FEDER através do Programa Operacional Competitividade e Internacionalização – COMPETE 2020 e por Fundos Nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito do projeto «POCI-01-0145-FEDER-006961»

## REFERÊNCIAS

- [1] R. Couto, “Pattern based software development,” Ph.D. dissertation, University of Minho, 2017.
- [2] M. Machado, J. Campos, and R. Couto, “Modus: uma metodologia de prototipagem de interfaces baseada em modelos,” in *INFORUM 2015*.
- [3] M. Machado, R. Couto, and J. Campos, “MODUS: Model-based user interfaces prototyping,” in *EICS 2017*, 2017.
- [4] A. G. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture: Practice and Promise*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- [5] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, “A unifying reference framework for multi-target user interfaces,” *Interacting with Computers*, vol. 15, pp. 289–308, 2003.
- [6] M. Machado, “MODUS: Generation of interfaces based on models,” MSc dissertation, University of Minho, 2015.