

**UNIVERSIDADE FEDERAL DE PERNAMBUCO**  
**Centro de Informática**  
**Mestrado Profissional em Ciência da Computação**

**Resenha Crítica**

Disciplina: Metodologia de Experimentação Científica

Professora: Teresa Bernarda Ludermir

Mestrando: Artur de Carvalho Alves

Tema: Geração Automática de Interface de Usuário

Artigos analisados:

Tran, V. (2010). UI generation from task, domain and user models: the DB-USE approach. Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, 353–356. doi:10.1145/1822018.1822079

Kennard, R., & Leaney, J. (2010). Towards a general purpose architecture for UI generation. Journal of Systems and Software, 83(10), 1896–1906. doi:10.1016/j.jss.2010.05.079

Kennard, R., & Leaney, J. (2011). Is there convergence in the field of UI generation? Journal of Systems and Software, 84(12), 2079–2087. doi:10.1016/j.jss.2011.05.034

## **Geração Automática de Interface de Usuário**

O tema geral escolhido trata-se da geração automática de Interface de Usuário, uma área que vem apresentando pesquisas significativas desde duas décadas atrás, e que ganhou maior urgência com o surgimento da computação ubíqua e a proliferação de diferentes dispositivos de interface de usuário com capacidades diferentes.

Durante a escolha dos artigos foi possível verificar que cada um dos trabalhos publicados trás uma abordagem diferente. É possível encontrar trabalhos usando a abordagem de Desenvolvimento Dirigido pelo Domínio, ou Desenvolvimento Dirigido pelo Modelo, abordagem de criação de Widgets, abordagem a partir do conceito Naked Objects, entre outras. É possível encontrar, inclusive, abordagens híbridas, usando mais de um desses conceitos.

No trabalho do Tran [1], por exemplo, é apresentado uma abordagem focada em três modelos para dirigir a geração da interface. Os três modelos considerados são: tarefa, domínio e usuário; de forma que, o modelo tarefa define uma interface abstrata, ela expressa como um usuário pode querer interagir com sistema; o modelo domínio prover as características para criar a interface e provê informações a partir dos atributos de objetos do domínio; e o modelo usuário suporta a criação de interfaces que levam em consideração a preferencia dos usuários. O objetivo do trabalho do Tran é definir e usar um Framework para desenvolvimento de Interfaces de Usuários para Sistema de Informação, ou aplicação de Banco de dados, combinando os modelos de tarefa, domínio e usuário.

Já o primeiro trabalho do Kennard [2] apresenta uma abordagem diferente, a partir da perspectiva de criação de Widgets 'ricos e inteligentes'. O trabalho do Kennard, na verdade, não tem o objetivo de apresentar a ferramenta desenvolvida, o MetaWidgets; mas sim de demonstrar que as pesquisas, realizadas junto a uma equipe de profissionais da indústria, foram capazes de indicar uma proposta de arquitetura de geração de Interface de Usuários. O Kennard alcança esse objetivo indicando 5 características fundamentais que um bom gerador de Interface de Usuário deve ter para poder ser adotado pela industria como um modelo padronizado.

Por fim, o último trabalho do Kennard [3] apresenta-se como uma tentativa de validar sua proposta das 5 características fundamentais com uma outra equipe de desenvolvimento de uma ferramenta de geração de interfaces de usuários, a equipe do Naked Objects Framework. A avaliação é feita a partir de uma entrevista com Dan Haywood, o líder de projeto do Apache Isis, que é a iniciativa da Apache Foundation em criar uma ferramenta a partir do padrão Naked Objects. Durante a entrevista, o Kennard faz questionamentos ao Dan no intuito de entender quais abordagens a equipe

do Naked Objects deu para cada um dos problemas, e tenta lincar as soluções implementadas pelo Naked Objects com suas 5 características fundamentais, enfatizando que existe uma convergência na forma de geração de Interface de Usuários, mesmo que a partir de abordagens diferentes.

### **UI generation from task, domain and user models: the DB-USE approach.**

O trabalho do Tran [1], se propõe a apresentar o framework DB-USE como processo metodológico, um meta-modelo e um protótipo de software. Inicialmente ele apresenta a metodologia que, basicamente, se divide em 5 fases: analisar o modelo, elaborar as relações, projetar a interface de usuário, projetar as funções da aplicação e gerar o código. A proposta é que haja um responsável para cada uma dessas fases o que fica bem esclarecido no diagrama apresentado no trabalho. No entanto, por estar apresentando uma metodologia, a impressão que passa é de que todo o processo é manual, e não automatizado como se esperava; mas, ao longo do desenvolvimento do trabalho, percebemos que a metodologia aborda a forma como o desenvolvedor escreverá os artefatos para o gerador de Interfaces de Usuários.

Por fim o autor coloca que depois de documentado todo o processo, o DB-USE irá gerar não só o código da interface, mas também o código da aplicação. No entanto, é importante perceber que o 'código' da aplicação não foi 'gerado', mais 'traduzido' a partir dos artefatos produzidos.

É importante destacar que o diferencial do trabalho foi a forma como o autor uniu as três abordagens distintas, e que se complementam, em uma metodologia simples, e perfeitamente compreensível.

### **Towards a general purpose architecture for UI generation**

O objetivo principal desse trabalho do Kennard [2] é apresentar as cinco características fundamentais que qualquer técnica de geração de Interface de Usuário deve abordar antes que espere ser largamente adotada pela indústria.

Ele começa justificando a importância da geração de Interface de Usuário, citando que cerca de 50% do tempo das equipes de desenvolvimento são gastos escrevendo interfaces, e que apesar de diversos esforços na área, e de alguns casos de sucesso, não existe ainda algo largamente adotado, e para exemplificar cita o exemplo do padrão JPA, que já é uma padrão bem aceito pelas indústrias

para gerenciamento Objeto-Relacional.

Nesse sentido, o Kennard coloca que através de uma metodologia de pesquisa-ação aplicada durante o desenvolvimento da ferramenta MetaWidgets, foi possível identificar as características que um gerador UI precisa ter para ser aceito pelo mercado, que são: Inspeccionar existentes arquiteturas de back-end heterogêneas, apreciar diferentes práticas de pós-processamento do resultado do inspeção, Reconhecer diversas bibliotecas de Widgets(componentes), suportar múltiplos adereços componentes de UI(exemplo: validadores, máscaras), aplicar diferentes layouts.

Durante o desenvolvimento do trabalho, Kennard apresenta os momentos de decisão de projeto que colaboraram para identificar essas 5 características, os momentos principais foram: a necessidade de implementar Inspetores Plugáveis, necessidade de implementar Processadores de Resultados dos Inspetores, necessidade de implementar Construtores de Widgets, necessidade de implementar, Processadores de Widgets e necessidade de implementar Layouts Plugáveis. Enquanto apresenta esses estágios do projeto o Kennard coloca diagramas que detalham claramente o comportamento da solução implementada.

Por fim, o Kennard apresenta que esse conjunto de características permitirá que o gerador de Interface de Usuários seja retro compatível com sistemas que não foram desenvolvidos com frameworks de geração de Interface de Usuário. No trabalho o autor coloca a questão da retrocompatibilidade como um fator de sucesso, o que me fez concordar, tendo em vista que essa característica faça com que seja desnecessário a reescrita de partes de um sistema para que todo ele possa ser adaptado a uma ferramenta de geração de interface.

### **Is there convergence in the field of UI generation?**

No último artigo do Kennard [3] ele trouxe como objetivo pesquisar se existe uma convergência entre diferentes soluções de geração de interface de usuário. Sua metodologia de pesquisa foi realizar uma entrevista com a equipe de desenvolvimento de uma outra ferramenta de geração de UI. No caso ele entrevistou o líder de projeto do Naked Objectc framework. E durante a entrevista ele buscava perceber se a equipe do Naked Objectcs também evoluiu no sentido de implementar as 5 características fundamentais descritas no artigo anterior.

Durante o desenvolvimento do Artigo ele conseguiu perceber que o Naked Objects framework evoluiu desde a sua primeira versão, e nessa evolução eles implementaram soluções para todas as 5

características fundamentais, levando a conclusão que, de fato, mesmo em projetos distintos, com abordagens distintas, se as 5 características foram implementadas nos dois, mostra-se que existe uma convergência nas soluções de geração de Interfaces de Usuários.

Em sua conclusão o autor faz uma autocritica por não ter validado a convergência com outros projetos de sucesso, mas coloca que, caso em trabalhos futuros perceba-se divergências ainda assim, será valido como informação muito instrutiva. Mas, mesmo não tendo entrevista outras equipes, o autor apresenta também um quadro comparativo entre a solução de sua equipe(MetaWidgets), a solução do entrevistado (Naked Objects) e outras duas soluções (UsiXML e OlivaNova); comparando a solução dada para cada uma das características.

Por fim, após a leitura os dois trabalhos do Kennard, foi bastante contundente ver que no segundo trabalho foi possível concluir, de forma científica, a hipótese apresentada; que, na verdade, era fruto de um trabalho anterior, e que havia sido originada de uma aplicação prática. Ou seja, considero o trabalho no Kennard completo. Pois, a partir de uma pesquisa-ação, apresentou um caminho para uma proposta de arquitetura de geração de UI; e em seu segundo trabalho, comprovou cientificamente, por meio de entrevistas, que o caminho que pretende seguir para apresentação da proposta apresenta pontos de convergência em outros trabalhos.

## **Referências**

- [1] Tran, V. (2010). UI generation from task, domain and user models: the DB-USE approach. Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, 353–356. doi:10.1145/1822018.1822079
- [2] Kennard, R., & Leaney, J. (2010). Towards a general purpose architecture for UI generation. Journal of Systems and Software, 83(10), 1896–1906. doi:10.1016/j.jss.2010.05.079
- [3] Kennard, R., & Leaney, J. (2011). Is there convergence in the field of UI generation? Journal of Systems and Software, 84(12), 2079–2087. doi:10.1016/j.jss.2011.05.034