

Development of Context-Aware Digital TV Applications using *Contextual Ginga* Authoring Tool¹

Ana Paula B. A. de Carvalho
Centro de Informática - UFPE
Av. Professor Luís Freire s/n
Cidade Universitária, Recife, PE
CEP: 50740-540 +55 81 2126.8430
apba@cin.ufpe.br

Carlos A. G. Ferraz
Centro de Informática - UFPE
Av. Professor Luís Freire s/n
Cidade Universitária, Recife, PE
CEP: 50740-540 +55 81 2126.8430
cagf@cin.ufpe.br

RESUMO

Este artigo apresenta a ferramenta de autoria *Contextual Ginga* e os resultados obtidos com a sua avaliação. Esta ferramenta tem como objetivo permitir o desenvolvimento de aplicações interativas de TV digital sensíveis ao contexto do usuário. Estas aplicações produzidas devem ser executadas na máquina Ginga-NCL. Para o desenvolvimento dessas aplicações através de *Contextual Ginga*, é necessário entender apenas dos conceitos de TV e de sensibilidade a contexto.

ABSTRACT

This paper presents the authoring tool *Contextual Ginga* and the results gathered in its evaluation. This tool's objective is to allow the development of interactive digital TV applications, which are user context-sensitive. These produced applications must be executed in Ginga-NCL machine. It is just necessary to understand about TV and context-sensibility concepts to produce these applications through *Contextual Ginga*.

Categories and Subject Descriptors

D.2.2 [Software Engineering]: Design Tools and Techniques.

General Terms

Design, Experimentation, Human Factors, Languages.

Keywords

Digital TV, Context Sensitivity, Authoring Tool.

1. INTRODUÇÃO

Com o aumento da interação entre usuário e aplicação de TV digital, os usuários podem chegar a fornecer uma informação para uma aplicação e que (a) esta informação já tenha sido fornecida anteriormente para outra aplicação ou (b) que poderia ser capturada sem uma resposta do usuário. Este excesso de informações dadas e recebidas pelos usuários e vice-versa pode diminuir o nível de interesse deles pela aplicação. Com o objetivo de aumentar este interesse, tem sido proposta a inserção

de sensibilidade a contexto [12] nas aplicações. Em [9], recomenda-se a definição do perfil do usuário público-alvo, ou *personas* (representação imaginária do usuário através de suas características principais); e a projeção dos contextos em que a aplicação será utilizada. Através da inserção da característica de sensibilidade a contexto do usuário, os sistemas se tornam mais atrativos, adaptáveis e pró-ativos [12].

Além da sensibilidade a contexto do usuário, outro ponto a ser considerado é que as aplicações interativas para TV digital (TVDi) precisam ter um ciclo de desenvolvimento curto. Isso ocorre, principalmente, em função da rapidez que o conteúdo televisivo deve ser produzido e transmitido [9]. Esse desenvolvimento também deve poder ser feito por pessoas que não são diretamente da área de informática.

Para apoiar o desenvolvimento de aplicações de TV digital, já existe um conjunto de ferramentas de autoria disponíveis. A maioria dessas ferramentas é voltada para *desktop*, com exceção do iTV Project [8]. Além disso, a maior parte é voltada para a plataforma MHP: Icareus iTV Suite Author [4], InteracTV [3], iTV Project [8], JAME Author [5] e SCO Creator Tool [7]. Grande parte, com exceção do *plug-in* NCL Eclipse [6], oferece edição visual com o propósito de diminuir o trabalho de codificação. Além disso, apenas o SCO Creator Tool [7] permite a inserção de regras de contexto, mas ela é específica para aplicações de *t-learning* de TV digital e voltada para o MHP. Das ferramentas encontradas, o NCL Eclipse [6] e o Ginga-NCL Composer [11] são voltados para Ginga-NCL, mas não existe tratamento de contexto do usuário nas aplicações produzidas.

Dessa forma, este artigo apresenta a ferramenta *Contextual Ginga* que visa permitir o desenvolvimento de aplicações interativas de TV digital sensíveis ao contexto do usuário, a serem executadas na máquina Ginga-NCL. A ferramenta está disponível em: <<http://www.cin.ufpe.br/~apba>>. Este artigo está organizado nas seguintes seções: 2 - descreve *Contextual Ginga*; 3 - apresenta algumas informações sobre o experimento de avaliação; e 4 - apresenta a conclusão e os trabalhos futuros.

2. FERRAMENTA CONTEXTUAL GINGA

Nesta seção será apresentada a ferramenta *Contextual Ginga*, através de uma descrição inicial, suas funcionalidades, sua arquitetura e a abordagem de como ocorre a geração de código.

¹ Desenvolvimento de Aplicações Sensíveis a Contexto para TV Digital através da Ferramenta *Contextual Ginga*

Na Figura 1, pode ser vista a interface gráfica do *Contextual Ginga* dividida por áreas. A área (1) exibe o nome do projeto aberto e os nomes das telas associadas. A área (2) exibe através de abas: a representação dos fluxos entre as telas e a representação gráfica de cada tela com seus componentes. A representação dos fluxos varia de acordo com o nome do *persona* selecionado na área (5).

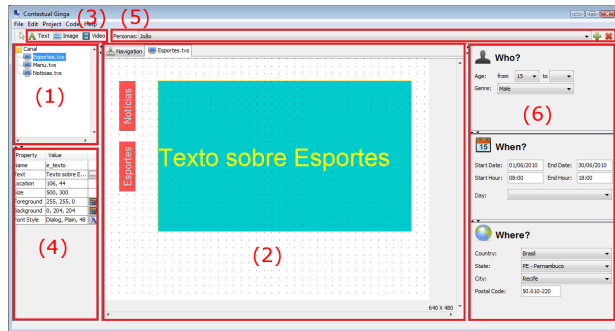


Figura 1: Imagem da tela principal do *Contextual Ginga*.

A área (3) exibe os tipos de componentes que podem ser adicionados: texto, imagem e vídeo. Os componentes do tipo imagem possuem duas propriedades que devem ser destacadas: imagem sem foco e imagem com foco. Com esta possibilidade de permitir que um componente possua duas imagens, juntamente com as transições (explicadas adiante), pode-se fazer com que ele apresente o comportamento de um botão. Apenas o tipo vídeo não pode ser adicionado à representação gráfica da tela. Ele é exibido apenas em tempo de execução. A área (4) exibe o valor corrente de cada propriedade de um componente selecionado na área (2). A área (5) exibe a lista de *personas* associados ao projeto e permite adicionar e excluir um determinado *persona*. A área (6) exibe as características do *persona* selecionado na área (5), divididos nas três categorias contextuais consideradas (*Who*, *When* e *Where*).

As telas de uma aplicação são interligadas através de transições (Figura 2). Uma transição representa a mudança de foco de um componente para outro (do componente C1 para C2), podendo estes componentes estar na mesma tela (do componente C1 para C2, na Tela 1 do *persona* X) ou em telas diferentes (do componente C1 na Tela 1 para C3 na Tela 2 do *persona* Y). Com as mudanças de foco, são construídos os possíveis fluxos de navegação da aplicação. Portanto, uma transição é composta por: telas de origem e destino, componentes de origem e destino, *persona* e um botão que quando acionado dispara a mudança de foco de componente. Os possíveis fluxos dependem do contexto do usuário da aplicação. Na Figura 2, as transições entre os componentes e telas variam de acordo com o *persona*. A categoria *Who* do *persona* possui propriedades de faixa etária e gênero do usuário. A categoria *When* possui propriedades de data e hora inicial e final, e dia da semana. A categoria *Where* possui propriedades de país, estado, cidade e código postal.

2.1 Funcionalidades

As funcionalidades do *Contextual Ginga* foram baseadas nas ferramentas relacionadas a este trabalho, adicionando-se, principalmente, aquelas que tratam do contexto do usuário.

As funcionalidades, agrupadas por conceito, são: (a) **Projeto**: criar, abrir, salvar, fechar e excluir projeto; (b) **Tela**: criar tela,

listar telas do projeto, exibir tela em editor gráfico, renomear e excluir tela; (c) **Componente**: adicionar, excluir, mover componente através de *drag-and-drop*, definir componente e tela iniciais por *persona* e alterar propriedades de um componente; (d) **Persona**: adicionar *persona*, excluir *persona* e alterar contexto do *persona*; (e) **Transição**: adicionar transição, visualizar transições por tela ou componente, excluir transição, alterar transição, visualizar transições graficamente por *persona*; (f) **Geração de Código NCLua**: gerar código NCLua.

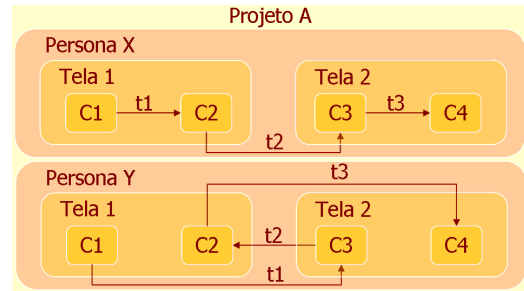


Figura 2: Transições variando por *persona*.

2.2 Arquitetura

O *Contextual Ginga* foi desenvolvido em Java, sem adição de bibliotecas, e para ser executado em ambiente *desktop*. Sua arquitetura está estruturada seguindo o padrão de camadas com: interface, negócio e dados. A camada de interface foi desenvolvida utilizando-se as APIs (*Application Programming Interface*) Swing e AWT (*Abstract Window Toolkit*).

A camada de negócio possui: os cadastros de projeto, tela, *persona* e transição; suas entidades; e a fachada da ferramenta. Das entidades presentes na arquitetura, um projeto possui telas e uma resolução, para o dimensionamento da aplicação. Cada tela possui uma coleção de transições que partem da mesma. Um projeto também possui uma coleção de *personas*, cada uma com um ponto (tela e componente) inicial e as categorias de contexto *Who*, *When* e *Where*.

Na camada de dados, encontram-se a leitura e a escrita dos arquivos salvos pela ferramenta, bem como a geração do código de aplicação. Os dados de um projeto estão armazenados em arquivos XML e, portanto, foi necessária a utilização de uma API para o processamento destes arquivos. A API utilizada foi a *javax.xml.parsers*, também nativa do Java, através do analisador baseado em SAX (*Simple API for XML*). Os arquivos XML criados pelo *Contextual Ginga* para um projeto dividem-se em quatro tipos: um para as características do projeto, um para os componentes de uma tela, um para as transições e um último para os *personas*.

2.3 Geração de Código

A geração de código ocorre com base no conteúdo dos arquivos XML do projeto. Na geração, são criados: um arquivo NCL [1]; um arquivo Lua principal; e para cada arquivo de tela do projeto, um arquivo Lua. O arquivo NCL é responsável, em sua essência, por executar o vídeo principal e invocar a execução do arquivo Lua principal como uma mídia.

No arquivo Lua principal, são coletadas a data e a hora do sistema, como também as informações do usuário que está interagindo com a televisão. As informações do usuário são

originadas do arquivo *Context.properties*, que possui idade, gênero, país, estado, cidade e código postal do usuário. Ainda no arquivo Lua principal, com as informações contextuais coletadas, determina-se a qual *persona* o usuário deve ser associado. Dependendo do *persona*, são exibidos a tela inicial e o primeiro componente a receber foco, sendo invocada a função de desenho desta tela. Se o usuário não se encaixar em nenhum *persona* e existir um *Default*, serão exibidos a tela inicial e o primeiro componente a receber foco para este *persona Default*.

O código Lua específico da aplicação, referente à aplicação criada no Contextual Ginga, é automaticamente gerado pela ferramenta. Nele encontra-se o tratamento das transições que é realizado através do módulo *event* de NCLua. O código independente da aplicação foi separado em uma biblioteca, denominada *Contextual Ginga Lua* que implementa funcionalidades comuns às aplicações geradas pela ferramenta. A biblioteca simplifica o código das aplicações através do reuso de representações de objetos e funções em Lua.

As aplicações geradas invocam a biblioteca para a funcionalidade de desenho dos componentes gráficos, através do módulo *canvas* de NCLua, como também para invocar funções auxiliares que tratam o contexto do usuário, registram as ações tomadas por este e gerenciam os *scripts* Lua. Por isto, os arquivos da biblioteca estão divididos em dois grupos: um para tratamento de interface gráfica, e outro para funções auxiliares.

3. EXPERIMENTO

Para avaliar se a ferramenta proposta atingiu seu objetivo, foi realizado um experimento, cujos dados coletados através de um questionário foram analisados conforme uma abordagem qualitativa. O experimento ocorreu em duas fases: a primeira com 27 alunos e a segunda com 4 alunos, sendo todos do Centro de Informática – UFPE.

Os participantes de ambas as fases realizaram o experimento de acordo com os cinco passos a seguir. **Passo 1:** Com base em um cenário de aplicação, reconhecer quais elementos fornecem sensibilidade a contexto para a aplicação; **Passo 2:** Criar projeto, telas da aplicação e os seus componentes; **Passo 3:** Criar os *personas* e inserir as suas características contextuais; **Passo 4:** Definir qual o ponto inicial (tela e componente iniciais) de cada *persona* e informar suas transições; **Passo 5:** Gerar o código da aplicação e executá-la no Ginga-NCL Virtual STB.

O questionário possuía perguntas sobre: os participantes, a ferramenta e a aplicação gerada automaticamente. Através de respostas sobre os pontos positivos da ferramenta, pode-se constatar que a ferramenta conseguiu atingir o seu objetivo. Exemplos dessas respostas foram: “gerar código sem precisar alterar”, “inclusão de elementos contextuais automaticamente” e “livra o seu usuário de lidar com códigos relacionados ao tratamento do contexto do telespectador”.

4. CONCLUSÃO

A partir do que foi exposto, as duas contribuições mais relevantes conferidas pela ferramenta Contextual Ginga são: (1) o desenvolvimento automático de aplicações para TVDi que tendem a aumentar o interesse do usuário na interação, ou seja, que são sensíveis a contexto. E a segunda é que o desenvolvimento de aplicações mais complexas, por possuírem

sensibilidade a contexto, é mais fácil e mais rápido, por ser auxiliado por uma ferramenta de autoria.

Os trabalhos futuros partem das limitações da ferramenta e das respostas obtidas nos questionários sobre dificuldades encontradas e problemas a serem corrigidos. Para continuidade da ferramenta: deve ser possível a criação de aplicações mais elaboradas; devem ser adicionadas outras categorias contextuais e enriquecido o tratamento do contexto; como também, devem ser realizados testes de usabilidade na ferramenta.

5. REFERÊNCIAS

- [1] ABNT. Associação Brasileira de Normas Técnicas. Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital Parte 2: Ginga-NCL para receptores fixos e móveis - Linguagem de aplicação XML para codificação de aplicações, 2007. ABNT NBR 15606-2:2007.
- [2] D. R. Morse, S. Armstrong, and A. K. Dey. The What, Who, Where, When, and How of Context-Awareness. In *Proceedings of the CHI 2000 Workshop. Georgia Institute of Technology*, 2000. Disponível em: <<http://smartech.gatech.edu/handle/1853/3464>>.
- [3] F. B. M. d. Almeida, R. B. Souza, and M. C. M. Neto. Ferramenta de Autoria Gráfica para a Construção de Aplicações Utilizando o Framework AppTV. In *Sétima Escola Regional de Computação Bahia-Sergipe*, Vitória da Conquista, 2007.
- [4] Icareus. Icareus iTV Suite Author, 2009.
- [5] IMK. Fraunhofer Institute for Media Communication. JAME Author, 2005.
- [6] LAWS. Laboratory of Advanced Web Systems. NCL Eclipse, 2008. Universidade Federal do Maranhão.
- [7] M. Rey-López, R. P. Díaz-Redondo, A. Fernández-Vilas, J. L. Pazos-Arias, M. López-Nores, J. García-Duque, A. Gil-Solla, and M. Ramos-Cabrer. T-MAESTRO and its authoring tool: using adaptation to integrate entertainment into personalized t-learning. *Multimedia Tools and Applications*, 40(3):409-451, Dezembro 2008.
- [8] M. R. Oliveira, C. B. Filho, and A. F. Fer. iTV project: an authoring tool for MHP and Ginga-J based on a web environment. In *UXTV '08: Proceeding of the 1st international conference on Designing interactive user experiences for TV and video*, pages 179-182, New York, NY, USA, 2008. ACM.
- [9] P. C. Gomes. Simplicidade aplicada ao design interativo na TV digital: o desenvolvimento de aplicações com interfaces orientadas ao usuário e seu contexto de uso. Especialização em Produção para TV Digital, Faculdade de Comunicação Multimídia da Universidade Metodista de São Paulo, São Bernardo do Campo, 2008.
- [10] Portal. Portal do Software Público Brasileiro. Ginga, 2009.
- [11] R. L. Guimarães. Composer: um ambiente de autoria de documentos NCL para TV digital interativa. Mestrado em Informática, PUC, RJ, Junho 2007. Disponível em: <<http://www2.dbd.puc-rio.br>>.
- [12] V. Vieira, P. Tedesco, and A. Salgado. A process for the design of Context-Sensitive Systems. In *Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design*, pages 143-148, Santiago, Chile, Abril 2009. IEEE Computer Society.