

1

Introdução

A TV Digital Interativa permite a criação de aplicações interativas que funcionam em um aparelho de TV provido de um set-top box, fato que não é possível no sistema de TV tradicional. Para possibilitar esse novo recurso, diversas tecnologias foram criadas a fim de facilitar o desenvolvimento dessas aplicações interativas. Uma dessas tecnologias é Nested Context Language (NCL) (NCLSpec 2007), uma linguagem declarativa de marcação, com base no XML, que possibilita a criação de aplicações multimídia para a TV Digital Interativa. Como exemplos de outras linguagens baseadas no XML, temos: o XHTML, para criação de aplicações para web; SVG (SVG 2009), para criação de gráficos vetoriais e SMIL (SMIL 2008), que possui o mesmo propósito do NCL.

1.1

Motivação

Apesar da semelhança em termos do paradigma da linguagem de implementação, o suporte de ferramentas de autoria para NCL ainda está muito aquém do XHTML e de outros padrões públicos e proprietários de documentos hipertexto e hipermídia. Com isso, escrever aplicações NCL atualmente é uma tarefa que demanda muito esforço e é muito propícia a erros.

No contexto atual, as ferramentas de apoio à autoria de aplicações NCL disponíveis são o NCL Eclipse (NCLEclipse 2008) e Composer (Guimaraes 07). Ambas, por fazerem uso do NCL-Validator (Araújo et al. 2008), fornecem algum suporte ao usuário em termos da qualidade do código. Porém, o NCL-Validator não possui qualquer mecanismo de extensão, dificultando a adaptação da ferramenta as necessidades específicas de cada usuário. Isso também dificulta o desenvolvimento colaborativo da ferramenta, limitando o desenvolvimento da ferramenta ao grupo de desenvolvedores que a criou.

Um sistema ideal para elevar a qualidade do código escrito em NCL restringiria o código produzido, impedindo que o desenvolvedor crie um código de baixa qualidade. Entretanto existem motivos que dificultam ou inviabilizam isso, que são: (1) nem todo aspecto de código pode ser previsto ou avaliado de forma precisa; (2) diferentes equipes de desenvolvimento podem ter critérios de qualidade diferentes; (3) não é possível prever todo tipo de problema que possa ocorrer.

Em relação ao primeiro item, cabe à ferramenta apenas detectar problemas potenciais e estabelecer diferentes graus de confiabilidade para suas detecções. Nos casos onde as detecções sejam de baixa confiabilidade, caberá ao autor ou desenvolvedor da aplicação NCL decidir qual solução será adotada durante a implementação.

Já os itens 2 e 3 nos remetem ao desenvolvimento de um sistema de críticas (Miller 1983, Miller 1986, Fischer 1993), que deve ser facilmente extensível. Esse sistema é um ponto de partida para facilitar a avaliação formativa (Hix et al. 93) de aplicações NCL. Utilizando o sistema será possível prever problemas de implementação e más práticas no uso da NCL (e.g. construções que atrapalhem o reuso e a legibilidade de código), reduzindo assim o esforço necessário para o seu desenvolvimento.

1.2 Objetivos

Esse trabalho teve como objetivo principal aumentar o suporte ao desenvolvimento de aplicações NCL. Para alcançar esse objetivo, construímos um sistema de críticas que pode ser facilmente estendido. O sistema recebe uma aplicação NCL como entrada e efetua críticas no código fonte dessa aplicação, a partir de um conjunto de regras que ele possui.

O requisito-chave é que a ferramenta seja facilmente extensível, facilitando o seu desenvolvimento descentralizado e, com isso, não mais limitando o desenvolvimento ao grupo de desenvolvedores que a criou. Dessa forma, atacaremos as dificuldades 2 e 3 que foram mencionadas anteriormente. Sendo um sistema fácil de estender, sempre que uma nova regra for conceitualmente definida, ela poderá ser incorporada à ferramenta.

1.3

Estrutura da dissertação

Para uma maior clareza desta dissertação, uma breve descrição da estrutura deste documento é dada nos parágrafos a seguir.

O Capítulo 2 mostra a fundamentação teórica dos princípios que foram utilizados para criar a ferramenta e também fazer a avaliação dos programas escritos em NCL. São descritos problemas de código conhecidos como maus cheiros. Também é apresentada a ideia principal usada para desenvolver o sistema.

O Capítulo 3 apresenta uma breve descrição das ferramentas de verificação e crítica de código fonte para Java e NCL, que foram usadas como inspiração para este trabalho.

O Capítulo 4 contém a principal contribuição deste trabalho. São apresentados os requisitos desejáveis de um sistema de críticas, a especificação, arquitetura e decisões de implementação do sistema desenvolvido.

O Capítulo 5 mostra a avaliação qualitativa feita com o sistema, a fim de identificar melhorias e avaliar como os usuários se comportam ao estender a ferramenta.

E finalmente, o Capítulo 6 apresenta uma discussão sobre o trabalho realizado, propostas de melhorias futuras e outras oportunidades que surgiram com a sua realização.