

5 Conclusões e Trabalhos Futuros

Este trabalho apresenta a primeira implementação de referência do Ginga-NCL para dispositivos portáteis, que foi baseada em sua implementação para terminais fixos apresentada resumidamente na Seção 3.1.

Antes de iniciar a implementação, foi necessário escolher uma plataforma de desenvolvimento para dispositivos portáteis adequada. A Seção 2.3 apresentou um estudo comparativo dos sistemas operacionais e das suas respectivas plataformas de desenvolvimento. Chegou-se à conclusão que o sistema operacional Symbian, juntamente com a sua plataforma de desenvolvimento nativa, era a melhor opção para a realização de uma primeira implementação de referência do Ginga-NCL para os dispositivos portáteis. Durante a realização dessa implementação foram encontrados diversos problemas. Esses problemas são apresentados no Capítulo 3 desta dissertação e têm as suas várias soluções apresentadas, analisadas e, quando havia mais de uma solução para um mesmo problema, comparadas. Por fim, foram descritas as soluções que foram realmente adotadas.

A implementação resultante de todo o trabalho descrito nesta dissertação foi testada em um emulador de dispositivos portáteis Symbian, e em dois dispositivos portáteis diferentes. Em ambos os casos, o funcionamento do Ginga-NCL se deu de modo correto e eficiente, sem a necessidade de se realizar qualquer modificação na especificação. Isso mostra que a especificação do Ginga-NCL é suficientemente leve para ser embarcada em dispositivos portáteis — mesmo que estes não tenham sido desenvolvidos para serem usados especificamente como receptores de TV — e que é, portanto, adequada a esse tipo de dispositivo. Assim, o principal objetivo almejado pelo estudo apresentado nesta dissertação foi alcançado. A demonstração de que a especificação do Ginga-NCL é adequada ao contexto dos dispositivos portáteis também é a maior contribuição oferecida pelo estudo apresentado.

Uma outra contribuição que pode ser citada é decorrente da nova implementação do módulo Conversor. Essa implementação demonstrou que o uso do parser DOM é desnecessário para o processamento de documentos NCL

— tanto no escopo dos dispositivos portáteis quanto para os terminais fixos — e que, por isso, o uso do parser SAX é mais vantajoso.

Salvo as contribuições citadas, algumas outras conclusões podem ser tiradas do trabalho apresentado. Acredita-se que muitos dos problemas observados e apresentados nesta dissertação também podem ocorrer em outras plataformas de desenvolvimento, em especial naquelas que forem fechadas. Os resultados aqui obtidos, portanto, podem ser usados como referência em implementações futuras do Ginga-NCL feitas em outras plataformas de desenvolvimento.

A arquitetura da implementação de referência do Ginga-NCL, apresentada na Seção 3.1, foi desenvolvida com o objetivo de facilitar o procedimento de porte para diversas plataformas. Essa arquitetura foi, portanto, desenvolvida de forma modular e abstrata o suficiente para que as partes específicas de plataforma pudessem ser isoladas das partes gerais e independentes de plataforma. Entretanto, nenhum porte da implementação havia sido realizado, ou seja, a arquitetura em questão ainda não tinha sido validada nessa questão. O resultado do trabalho apresentado nesta dissertação foi, de fato, o primeiro passo para realizar essa validação.

Ao longo de toda a implementação do Ginga para dispositivos portáteis, foram encontradas, na implementação de referência, abstrações no uso de recursos específicos de plataforma. Essas abstrações permitiram que apenas parcelas específicas e pontuais da implementação de referência tivessem que ser modificadas, mantendo intacta a já funcional estrutura genérica do middleware. Isso facilita o procedimento de porte na medida em que o funcionamento geral do middleware não precisa ser modificado, evitando erros e permitindo que um maior foco pudesse ser dado às especificidades da plataforma. A modularização da arquitetura também é um fator que contribuiu para o porte. Na implementação apresentada, por exemplo, o módulo Conversor foi quase que completamente modificado sem que isso interferisse no funcionamento do middleware como um todo. Isso só foi possível graças à boa delimitação dos módulos na arquitetura. Assim, é possível concluir, através do estudo apresentado, que a forma como a arquitetura da implementação de referência do Ginga-NCL foi desenvolvida de fato facilita o procedimento de porte.

Existem alguns possíveis trabalhos futuros que poderão ser desenvolvidos a fim de se incrementar o trabalho apresentado nesta dissertação. O primeiro trabalho sugerido diz respeito ao procedimento de parser do Ginga-NCL. Nesta

dissertação foi apresentada a substituição do parser DOM pelo SAX, levando-se em consideração o fato de que testes realizados em (Violleau, 2002), (Oren, 2002), (DevSphere, 2007) e (Franklin, 2007) entre esses dois tipos de parser indicam, de forma genérica, uma eficiência maior por parte do SAX. Entretanto, nenhum estudo comparativo foi feito entre uma implementação do módulo Conversor que use SAX e outra que use DOM. Seria interessante, portanto, realizar esse estudo a fim de se ter uma comparação entre o DOM e o SAX dentro do contexto do Ginga-NCL. Um outro trabalho que pode ser feito em relação ao parser é implementar o tratamento dos comandos de edição ao vivo, que não foi feito na versão do Ginga-NCL para dispositivos portáteis apresentada.

Um segundo trabalho futuro é a implementação do modelo para a especificação temporal de aplicações hipermídia descrito em (Costa e Soares, 2007). Esse modelo viabiliza, dentre outras coisas, a interrupção da apresentação de uma aplicação NCL como um todo e a sua posterior retomada a partir do mesmo instante temporal. Considerando, ainda, o universo dos dispositivos portáteis, uma vantagem é que esse modelo foi especificado de tal forma que é possível descartar parcelas suas que já tenham sido usadas ou que nunca mais serão utilizadas. Isso pode ser feito em qualquer momento de uma apresentação NCL, o que possibilita a otimização do uso dos recursos de memória, que são muito limitados na maioria dos dispositivos portáteis.

Um outro trabalho sugerido é a execução testes de desempenho na implementação de referência do Ginga-NCL para dispositivos portáteis, verificando o nível do consumo de recursos utilizados, como bateria, memória e CPU. O objetivo é verificar a carga imposta pela implementação ao sistema operacional, determinando se, por exemplo, outras aplicações podem ser executadas ao mesmo tempo em que uma aplicação NCL é exibida.

É sugerida, também, a implementação da funcionalidade de adaptação da NCL em função da disponibilidade de recursos do aparelho, o que não foi feito na implementação apresentada nesta dissertação. Com essa funcionalidade, um aplicação pode ser desenvolvida de maneira que funcione em dispositivos com restrições de recursos diferentes.

Um outro trabalho que pode ser feito diz respeito aos exibidores de mídia. Novos exibidores, o de texto e Lua, podem ser implementados e o exibidor de imagem pode ser melhorado. A API específica de tratamento de texto em Symbian possui documentação insuficiente e, até o término da implementação que é apresentada neste estudo, era pouco utilizada — o que significa que

poucas pessoas da comunidade de desenvolvimento sabiam como utilizá-la. A melhor opção para o início da implementação desse exibidor é começar pela busca de códigos exemplo que possam ser usados no entendimento do uso da API Symbian de texto. Espera-se, portanto, que a implementação desse exibidor ofereça alguns problemas. É incerto, entretanto, se a implementação do exibidor de texto em outras plataformas de desenvolvimento também apresentarão problemas.

No que diz respeito ao exibidor Lua, será necessário portar o interpretador Lua, que é todo implementado em ANSI C, as interfaces do Ginga-NCL com esse interpretador e algumas bibliotecas Lua desenvolvidas especificamente para o Ginga-NCL. Em Symbian, portar o interpretador Lua não deve oferecer muito trabalho por conta da existência das bibliotecas P.I.P.S. e Open C. Já em plataformas fechadas, o ANSI C pode estar apenas parcialmente implementado, ou nem sequer isso. Portar o interpretador Lua para esse tipo de plataforma pode ser mais problemático. As interfaces e as bibliotecas feitas para o Ginga-NCL, porém, possuem algumas características específicas de plataforma que podem introduzir um nível de dificuldade maior, tanto no Symbian quanto em qualquer outra plataforma.

O incremento ao exibidor de imagem é sugerido a fim de que ele possa oferecer suporte à apresentação de outros tipos de imagem além da mbm. Será necessário, entretanto, fazer uso de uma classe que converta outros tipos de imagem, como png, bmp ou gif, em imagens mbm. Isso porque a classe apresentada na Seção 3.7 capaz de carregar imagens no Symbian, a CFbsBitmap, só consegue carregar imagens do tipo mbm. O problema é que a documentação da classe que faz a conversão é confusa, e também existem poucos exemplos do seu uso.

Por fim, o último trabalho futuro sugerido é a implementação dos módulos Sintonizador, Filtro de Seções e Processador de Fluxos de Dados, que são descritos na Seção 3.1, em uma plataforma com recepção SBTVD.