

6 Conclusões

O objetivo principal deste trabalho foi elaborar um ambiente de execução declarativo para sistemas que implementem a especificação do GEM. O ambiente desenvolvido foi baseado na implementação Java do Formatador NCL criada pelo Laboratório TeleMídia da PUC-Rio e permite que os programas declarativos produzidos para o Sistema Brasileiro de TV Digital Terrestre sejam apresentados em qualquer sistema que implemente o *framework* GEM.

A fim de melhor compreender as tecnologias envolvidas na elaboração desse ambiente, foram apresentados no Capítulo 2 os *middleware* para TV Digital Terrestre que implementam a especificação do GEM e foi feita uma breve descrição deste padrão.

No Capítulo 3, foram descritos alguns trabalhos relacionados com o tema proposto nesta dissertação. Foram analisadas duas implementações de ambientes declarativos para sistemas GEM tendo como base a linguagem SMIL. A partir dos trabalhos realizados na linguagem SMIL, foi possível compreender a grande dificuldade encontrada no desenvolvimento de aplicações para sistemas de TV Digital: o tempo de iniciação. Ou seja, o tempo decorrido desde a recepção do documento declarativo até sua exibição.

Ao final do Capítulo 3, foi apresentada a implementação Java do Formatador NCL, que serviu como base para o ambiente proposto nesta dissertação. Essa implementação foi alterada visando atender à API diferenciada do GEM e aos requisitos inerentes a sistemas de TV Digital (como, por exemplo, o tempo de iniciação) Seus módulos foram transformados em componentes de software, sendo criadas diferentes implementações desses componentes, atendendo tanto a *desktops* como sistemas GEM.

Foi desenvolvida uma estratégia diferente de implantação do Formatador NCL para receptores com recursos de memória limitados. Essa estratégia faz uso do mecanismo de serialização, oferecido pela linguagem Java, e realiza todo o processo de conversão do documento NCL em seu distribuidor.

Outra otimização (mostrada no Capítulo 4) pode ser realizada para sistemas DVB-MHP que ofereçam suporte às facilidades de plug-ins interoperáveis e armazenamento de aplicações. Dessa forma, o ambiente declarativo proposto poderá se comportar como aplicação nativa nos receptores.

As otimizações propostas levam a uma implantação enxuta do Formatador, permitindo uma redução do seu tamanho e tornando-o, inclusive, menor que o exibidor SMIL apresentado no Capítulo 3. Em sistemas com suporte à persistência de dados e de mídia empacotada (como o *Blue-Ray Disc*) as otimizações permitem um desempenho ainda maior em termos de tempo de iniciação.

6.1. Trabalhos Futuros

Alguns pontos dos trabalhos realizados nesta dissertação podem ser explorados como trabalhos futuros.

A configuração de máquina virtual CDC e seus perfis possuem implementações disponíveis para as mais diversas plataformas (mostradas na Tabela 3). Novos testes em novas plataformas são valiosos para a validação da implementação proposta nesta dissertação.

Darwin/PowerPC
Linux/ARM
Linux/MIPS
Linux/PowerPC
Linux/Sparc
Linux/x86
Solaris/Sparc
Solaris/x86
Symbian/ARM
Symbian/x86
Vxworks/Sparc
Vxworks/x86
Win32/ARM/PocketPC
Win32/ARM/WindowsMobile5
Win32/MIPS/WindowsCE
Win32/x86
Win32/x86/PocketPC

Tabela 3 – Plataformas que implementam o CDC.

Outro aspecto mostrado neste trabalho que pode render trabalhos futuros é a modelagem do Formatador NCL baseada em componentes. Nem todas as facilidades proporcionadas por essa modelagem puderam ser aproveitadas neste trabalho por motivos de requisitos do sistema (como mostrado na Seção 4.3.1). Em ambientes que ofereçam suporte a diferentes linguagens de programação, por exemplo, pode haver uma cooperação entre componentes implementados nessas diferentes linguagens.

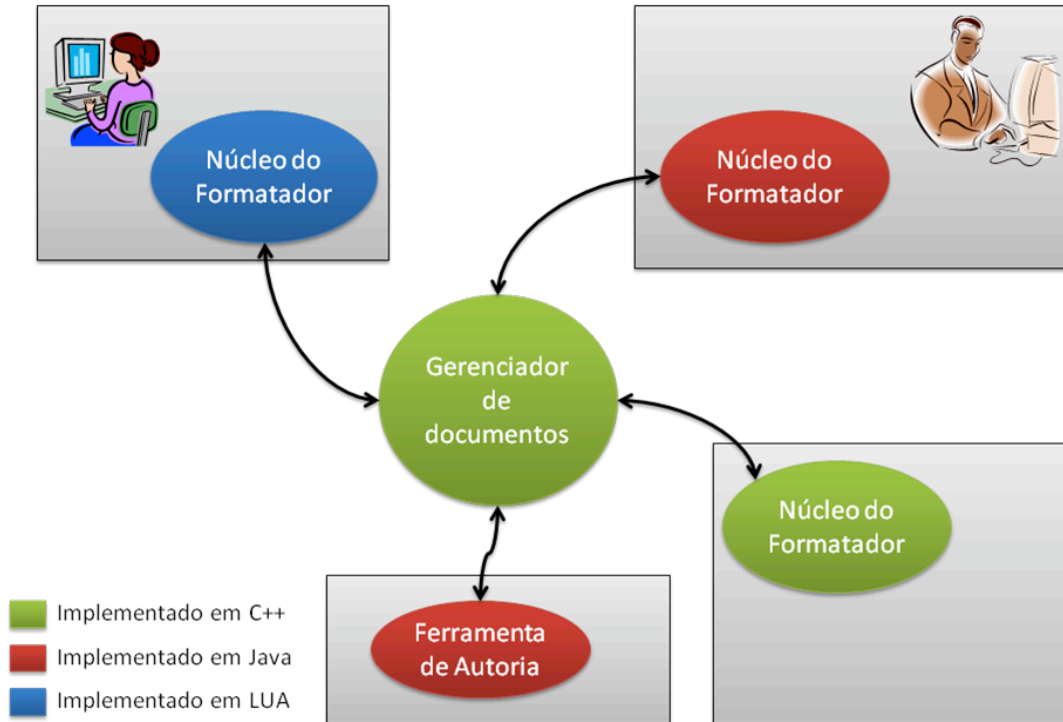


Figura 36 – Arquitetura de execução distribuída do Formatador NCL.

Além disso, pode-se explorar a distribuição desses componentes criando arquiteturas de execução como a mostrada na Figura 36. Nessa Figura, existe um componente Gerenciador de Documentos centralizado de onde vários Núcleos de Formatares obtêm os documentos NCL a serem apresentados.

Utilizando esse mesmo componente podem existir, ainda, diversas ferramentas de autoria. Dessa forma seria possibilitada uma autoria colaborativa de documentos NCL ao vivo.