

## 6 Conclusões e trabalhos futuros

Prover mecanismos para facilitar o trabalho de autores com pouca ou nenhuma experiência em programação é necessário para estimular profissionais com esse perfil a produzirem não somente conteúdo, mas também aplicações para TV Digital. Por isso, a solução de arquitetura proposta nesta dissertação vai ao encontro dessa necessidade, à medida que promove reúso de especificações de aplicações – por meio da generalização apresentada na Seção 4.2 – e interfaces gráficas para autoria de aplicações – por meio da ferramenta apresentada no Capítulo 5.

As abordagens de autoria orientadas à *templates* permitem que haja reúso das especificações de *templates* por parte de uma família de documentos. Nesse contexto, ao utilizar um método de autoria orientado a *templates*, pode-se concluir que o trabalho realizado nesta dissertação traz o benefício adicional de uma solução de arquitetura que facilita a construção de aplicações que se recriam em tempo de apresentação e que mantêm seus *templates* após a recriação. Essa é a maior contribuição deste trabalho. No ambiente de desenvolvimento de aplicações interativas para TV Digital, essa solução de arquitetura pode ser útil na especificação de outros programas que mantêm um formato de aplicação. Espera-se que, com a proposta de arquitetura apresentada, surjam novas aplicações que se recriam dinamicamente. A implementação de novas aplicações é importante para que a arquitetura evolua e atenda outros requisitos.

Com base na solução de arquitetura apresentada, foi implementada uma ferramenta que dá suporte à autoria de aplicações que se recriam dinamicamente. Essa ferramenta provê uma linguagem gráfica, que representa uma interface mais amigável entre o autor e a obtenção de sua aplicação final. Para autores que não se sentem confortáveis para trabalhar com códigos-fonte, essa linguagem gráfica é mais atraente do que uma linguagem textual.

Além dos produtos que foram mencionados como contribuições concretas, este trabalho também contribui com uma análise detalhada de como se especificar

aplicações NCL dinâmicas. A experiência reportada nesta dissertação com a especificação da arquitetura para aplicações que se recriam dinamicamente e com a ferramenta **SAGGA** constitui-se em uma importante base de conhecimento, podendo servir como base para outros projetos com foco nessa área.

A pesquisa realizada nesta dissertação também traz uma contribuição secundária: a implementação de um exibidor de legendas no formato SRT. Esse exibidor foi necessário para que as aplicações **SAGGA** pudessem exibir legendas nesse formato, considerando-se que a Norma ABNT NBR 15606-2 do SBTVD não prevê na implementação do Ginga-NCL a existência de um exibidor para o tipo de mídia SRT.

O projeto **SAGGA** apresentado na Seção 4.1, aliás, não se restringe ao módulo **SAGGA1**. Outros módulos desse projeto propõem a implementação de outros tipos de aplicações que se recriam dinamicamente. Por exemplo, o módulo **SAGGA2** propõe que, além de vídeo principal e exibição de legendas, as aplicações incorporem *widjets* com o intuito de exibir propagandas, prover jogos interativos etc. Como trabalho futuro, pretende-se criar *templates* para esses módulos e adaptá-los à arquitetura para aplicações que se recriam dinamicamente. A implementação de tais módulos é importante para demonstrar a utilidade dessa arquitetura.

Como consequência da implementação de outros módulos **SAGGA**, há a necessidade da ferramenta gráfica apresentada na Seção 5.2 contemplar a autoria de aplicações conforme os *templates* desses módulos. Portanto, um trabalho futuro é a ampliação do suporte gráfico da ferramenta **SAGGA** aos outros módulos do projeto.

Outro trabalho futuro é a adaptação da ferramenta **SAGGA** – que foi apresentada no Capítulo 5 – como um plug-in do *Composer*. *Composer* provê diferentes tipos de visões sobre os documentos NCL por meio de *plug-ins*. Como **SAGGA** pode ser encarada como uma visão sobre um tipo específico de documento NCL, ela poderia ser anexada ao *Composer*. Dessa forma, o usuário de **SAGGA** disporia de facilidades providas por outros *plug-ins*, como os de visão estrutural, leiaute, etc.

Um quarto trabalho é a implementação de um gerador de documentos de preenchimento para o **SAGGA1** que faça buscas nos *sites* do projeto e analise semanticamente quais são os vídeos relacionados a um determinado vídeo. Nesta

dissertação foi dado ênfase à autoria dos documentos e não às técnicas para associações semânticas entre conteúdos. Contudo, essas técnicas precisam ser investigadas e implementadas futuramente para atender aos requisitos que **SAGGA** demanda. Os *sites* que estão associados ao projeto precisam prover metadados de seus conteúdos que sejam suficientes para permitir a implementação de uma máquina de busca capaz de inferir relacionamentos semânticos entre os conteúdos.

Por fim, um aspecto mostrado na Seção 5.1.3 e que pode render trabalhos futuros é a possibilidade de implementar geradores de conteúdo no ambiente de um servidor *proxy*. A implementação de geradores em *proxies* demanda uma infra-estrutura que permita aos objetos que auxiliam a geração de conteúdo e que são executados nos *proxies*, terem seus métodos invocados remotamente a fim de devolverem composições que serão adicionadas por comandos de edição em documentos NCL.