

5

Conclusão

O formatador NCL é um programa complexo. Em parte, essa complexidade está associada a forma como as implementações atuais tratam as redundâncias presentes na linguagem de autoria. O objetivo deste trabalho foi delimitar com precisão o menor subconjunto da linguagem definida pelo perfil EDTV que é capaz de representar, de forma equivalente, qualquer documento EDTV válido. Dessa forma, as principais redundâncias no perfil EDTV foram identificadas e procedimentos de eliminação correspondentes foram definidos. A partir desses procedimentos foi definido um perfil mínimo, praticamente livre de redundâncias, chamado NCL *Raw*. Por ser mais simples e consistente, esse perfil simplifica o modelo de apresentação do formatador NCL. De fato, através do uso de conversores, é provável que este perfil permita uma melhor distribuição da complexidade entre os diversos subsistemas do formatador. Nesta dissertação, também foram propostas duas arquiteturas para um conversor de documentos EDTV em documentos *Raw*. A primeira considera a conversão de documentos propriamente ditos, i.e. a conversão entre arquivos XML. A segunda arquitetura trata da conversão de comandos de edição entre os dois perfis. A dissertação apresentou ainda uma implementação de um pacote para conversão entre perfis NCL, chamado NCC (*NCL Converter Collection*).

Podemos destacar as seguintes possibilidades de trabalhos futuros. No que diz respeito a linguagem, é possível pensar em perfis ainda menores compatíveis ou não o EDTV. Por exemplo, pode ser possível remover o reuso de instância (“instSame” e “gradSame”) através da criação de portas e elos. Uma outra linha possível integrar outras linguagens de propósito similar (e.g. SMIL, HTML, etc.) no sistema, por exemplo definindo novos procedimentos de tradução. Dentre as principais características a serem incorporadas ao conversor NCC podemos destacar: a inclusão de chamadas que permitam monitorar o progresso da conversão, possibilidade de seleção da codificação de do documento gerado, implementação de *bindings* para outras linguagens de programação (Libncc), e inclusão de mecanismos que permitam adicionar dinamicamente novos *parsers*, operações, e geradores de código. Outra possibilidade é a incorporação de novas APIs para conversão de comandos de edição.