

## **Resumo**

Esta dissertação descreve a implementação de um tradutor de SDL para C++ que tem por base a Gramática Abstracta da linguagem SDL descrita na recomendação Z.100 do ITU-T. A linguagem C++ foi escolhida para efectuar o armazenamento da informação contida nas especificações SDL.

Inicialmente são apresentadas as ferramentas envolvidas no desenvolvimento do tradutor, como sejam a linguagem SDL, o gerador de analisadores léxicos, o gerador de analisadores gramaticais e a linguagem C++.

Descreve-se a seguir a relação entre a Gramática Abstracta e a estrutura para o armazenamento da informação contida nas especificações SDL, denominada árvore de parsing que é posteriormente navegada para a geração dos ficheiros na linguagem C++.

É então abordado o modelo de execução SDL, a partir do qual foi construída uma biblioteca de suporte que acompanha os ficheiros obtidos na tradução; são também analisadas as estratégias de tradução adoptadas para os diferentes construtores da linguagem SDL e paralelamente são discutidas as suas implicações sobre a estrutura da Gramática Abstracta.

Finalmente é descrito um caso prático da aplicação do tradutor na geração de software para equipamentos terminais RDIS e conclui-se sobre as vantagens no uso da Gramática Abstracta para a implementação do tradutor bem como da utilização da linguagem C++ para a criação e navegação da árvore de parsing.

## **Abstract**

This dissertation describes the implementation of a SDL to C++ translator, based on the SDL Abstract Grammar which is described in the ITU-T Z.100 Recommendation. The C++ language has been selected to store the information contained on the SDL specifications.

At first, the tools used for the development of the translator - the SDL language, the lexical analyser generator, the syntactic analyser generator and the C++ language - are briefly presented.

Next, the relationship between the Abstract Grammar and the structure for storing the information contained in the SDL specifications is described; this structure is referred to as the parsing tree, which is then navigated for the generation of the C++ files.

Then the SDL execution model, used to build the support library for the files generated by the translator, is discussed; the strategies selected for the translation of the various SDL constructs are analysed as well as the implications on the structure of the Abstract Grammar.

At last, it is described an application of the translator in the generation of software for ISDN terminal equipment; as a conclusion, the advantages of both using the Abstract Grammar to implement the translator and the C++ language to create and navigate through the parsing tree, are stressed.