ANSA Phase III

TC Stub Generator presentation (09/95)

Youcef Laribi

Abstract

These slides give an overview of the Stub compiler toolset work carried out as part of the DIMMA project. We first analyse the objectives targeted by this work and the functions supported or enabled by the resulting toolset. We then give an outline architecture of the toolset, underlying its important components and their interfacing to each other. Finally, we give an update status of the work achieved and the different components that are operational.

APM.1523.01.02	**Draft**	4th October 1995

Briefing Note

Distribution:

Supersedes:

Superseded by:

DIMMA Stub Generator Toolset
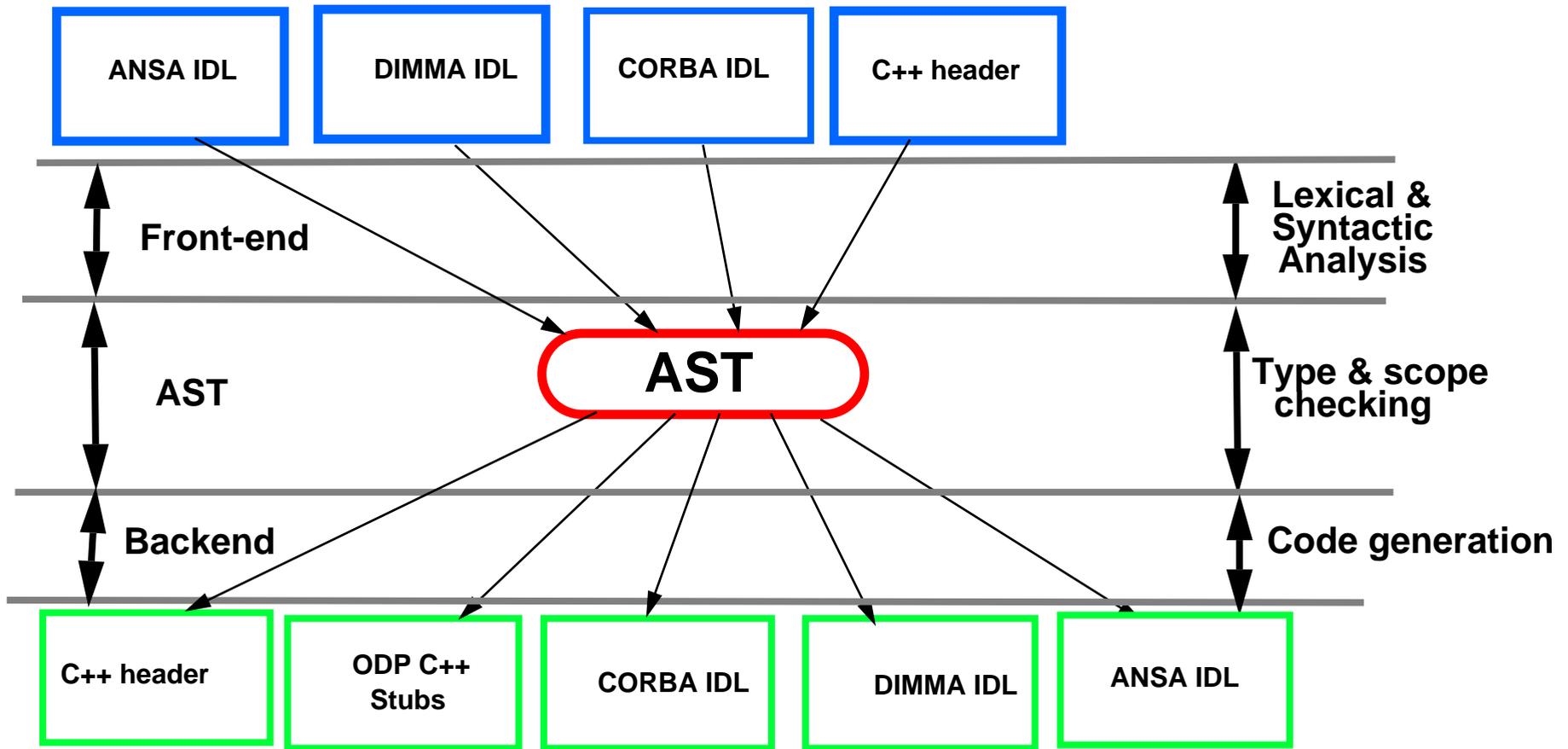
Youcef Laribi (yl@ansa.co.uk)

DIMMA Stub Generator Toolset
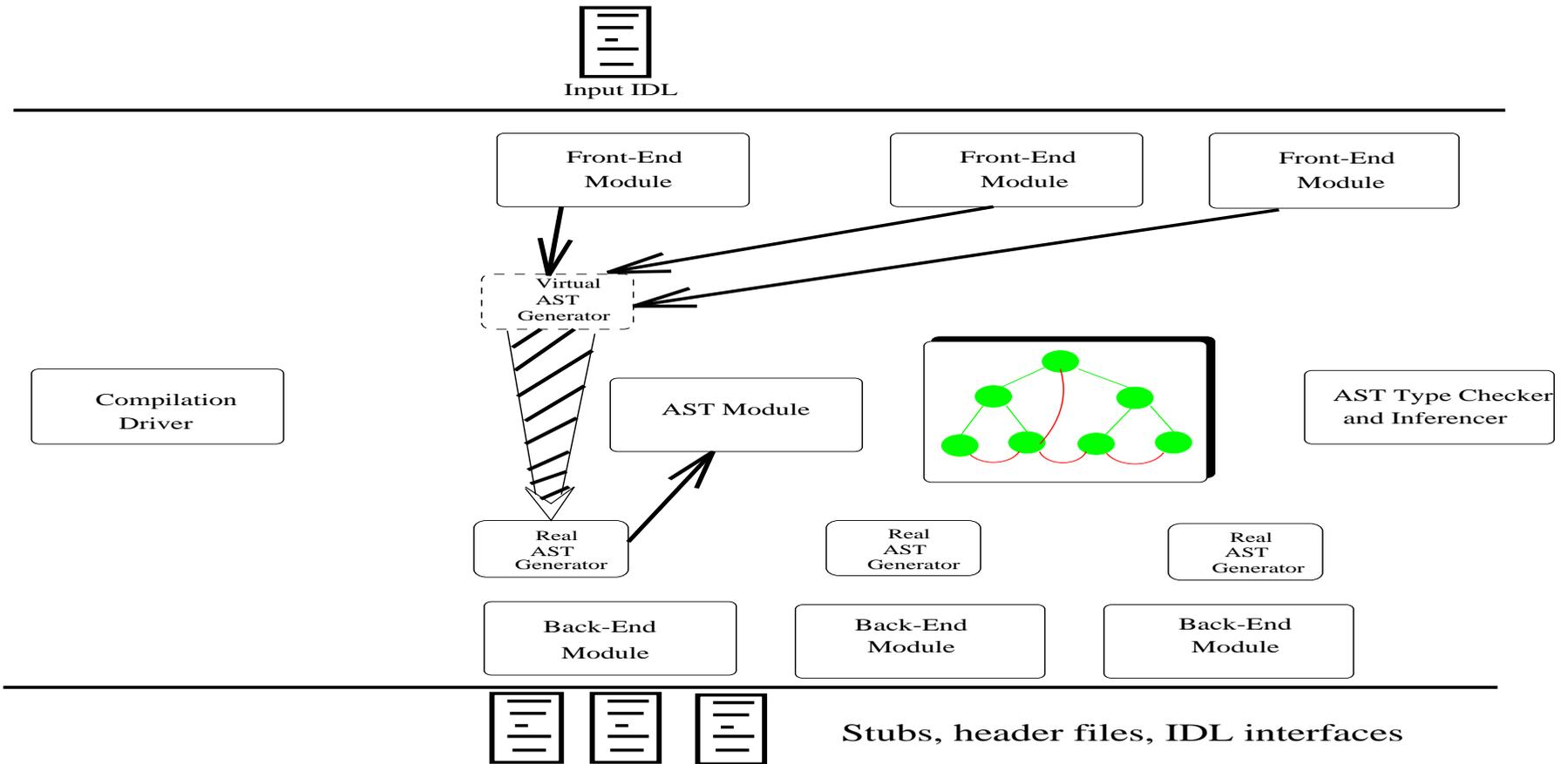
Objectives:

- **Support several input notations to describe service interfaces (CORBA IDL, C++ headers, ANSA IDL, etc).**

- **Support several runtime environments (transport, data formats, etc.).**

- **Generate headers, stubs and skeletons targetting a language mapping (e.g C++) and an runtime environment (e.g ODP runtime).**

- **Translate an input IDL notation to another equivalent IDL notation (e.g DIMMA IDL -> CORBA IDL), flagging incompatibilities.**

- **Perfom Type and Scope checking.**

Stub Generator Master Plan

ANSA IDL	DIMMA IDL	CORBA IDL	C++ header

Front-end — Lexical & Syntactic Analysis

AST

AST — Type & scope checking

Backend — Code generation

C++ header	ODP C++ Stubs	CORBA IDL	DIMMA IDL	ANSA IDL

AST Tools architecture

Input IDL

| Front-End Module | Front-End Module | Front-End Module |

Virtual AST Generator

Compilation Driver

AST Module

AST Type Checker and Inferencer

Real AST Generator

Real AST Generator

Real AST Generator

| Back-End Module | Back-End Module | Back-End Module |

Stubs, header files, IDL interfaces

Benefits

- **The AST is a generic syntax-free form on which tools in the development environment can be based, and reused for different runtime environments.**

- **It allows compile-time and if necessary runtime type checking to be performed whatever the IDL used to describe the interface.**

- **Generated stubs for DIMMA are transport-independent, allowing great flexibility for marshalling/unmarshalling, communication buffer management, etc.**

- **Enables type-conformance based trading.**

- **There may be more than one backend for some environment, allowing the generation of stubs for different purposes (security, transaction, replication, etc.)**

Progress

- **Built a DIMMA IDL front-end (lexical, syntactic analysis and AST generation).**

- **Completed a C++ back-end which generates stubs for the DIMMA nucleus.**

- **Adding structured types to the DIMMA IDL.**

- **In the process of adding a CORBA IDL front-end to the stub generator toolset.**

References

[APM.1554] Y. Laribi
  "DIMMA Stub Generator Design and Implementation"