



---

**Poseidon House  
Castle Park  
Cambridge CB3 0RD  
United Kingdom**

TELEPHONE:  
INTERNATIONAL:  
FAX:  
E-MAIL:

**Cambridge (01223) 515010  
+44 1223 515010  
+44 1223 359779  
apm@ansa.co.uk**

---

## **ANSA Phase III**

# **DIMMA Progress Status**

**Youcef Laribi**

### **Abstract**

The business problem addressed is...

The technical problem created by that business problem is ...

The solution being offered is....

---

APM.1489.00.01

**Draft**

22nd May 1995

Briefing Note

---

**Distribution:**

**Supersedes:**

**Superseded by:**





# **DIMMA Progress Status**

**youcef Laribi**



## **DIMMA Objectives**

- **Developing a platform suitable for distributed Multimedia and real-time programming.**
- **Using standard APIs to ensure stable and portable applications.**
- **Combining different computational models within the same platform (ODP, CORBA, TINA).**
- **Supporting new models of distributed interaction: Streams & Signals.**
- **Allowing flexible and user-controlled resource management (e.g explicit binding).**



## In a First Stage...

- Define and implement an ODP API runtime for which programmers can write stable applications.
- Develop the set of tools necessary to generate stubs for the above ODP API runtime from different IDL service descriptions.
- Design and implement an engineering engine (DIMMA Nucleus) which will handle distributed multimedia and real-time requirements and interface the upper ODP runtime and the underlying operating systems.



### An ODP-based DPE



## DIMMA Engineering Practices

- **C++ as the programming language.**
- **Self-contained software.**
- **Manage portability by using the GNU autoconf tool.**
- **Manage the building process and the source tree using GUIs.**



## ODP Runtime

- Provide applications with an API based on ODP computational model.
- Support several engineering engines with possibly different capabilities.
- Client/Server Stubs are completely independent from the protocols used (for marshalling/unmarshalling).
- Using a uniform engineering API for stubs, DII and DSI.



## ODP Runtime progress status

- **Defined and implemented interfaces and semantics for a set of ODP abstractions:**
  - Objects
  - Invocation References
  - Interfaces
  - Signatures
  - Terminations
  - Basic Type
- **Non functional utilities:**
  - Local garbage collection.
  - Auditing/tracing/debugging.
- **Tools:**
  - A capsule manager.
  - A handcrafted trader server stub.



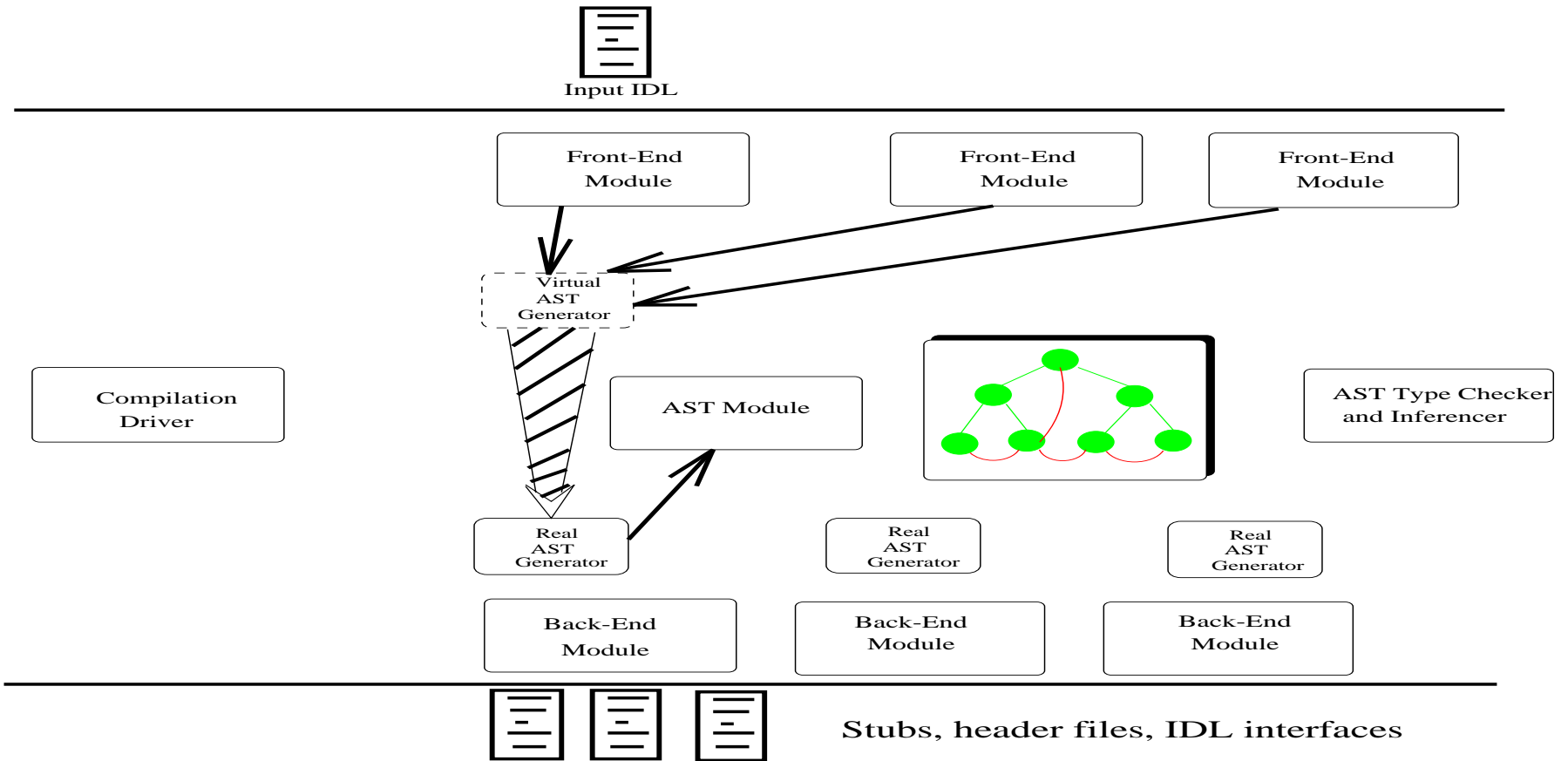


## AST-based tools for stub generation

- **Objectives:**

- 1- Capture the ODP semantics of an interface into an Abstract Syntax Tree (AST).
- 2- Perform the semantic analysis (type inferencing and checking) on an AST tree whatever was the IDL originally used to describe that interface.
- 3- Construct tools that generate ODP ASTs from IDL descriptions or from header files (front-ends).
- 4- Construct tools that generate stubs or alternative IDL descriptions, from an ODP AST (back-ends).

# AST Tools architecture





## Stub Generation Progress Status

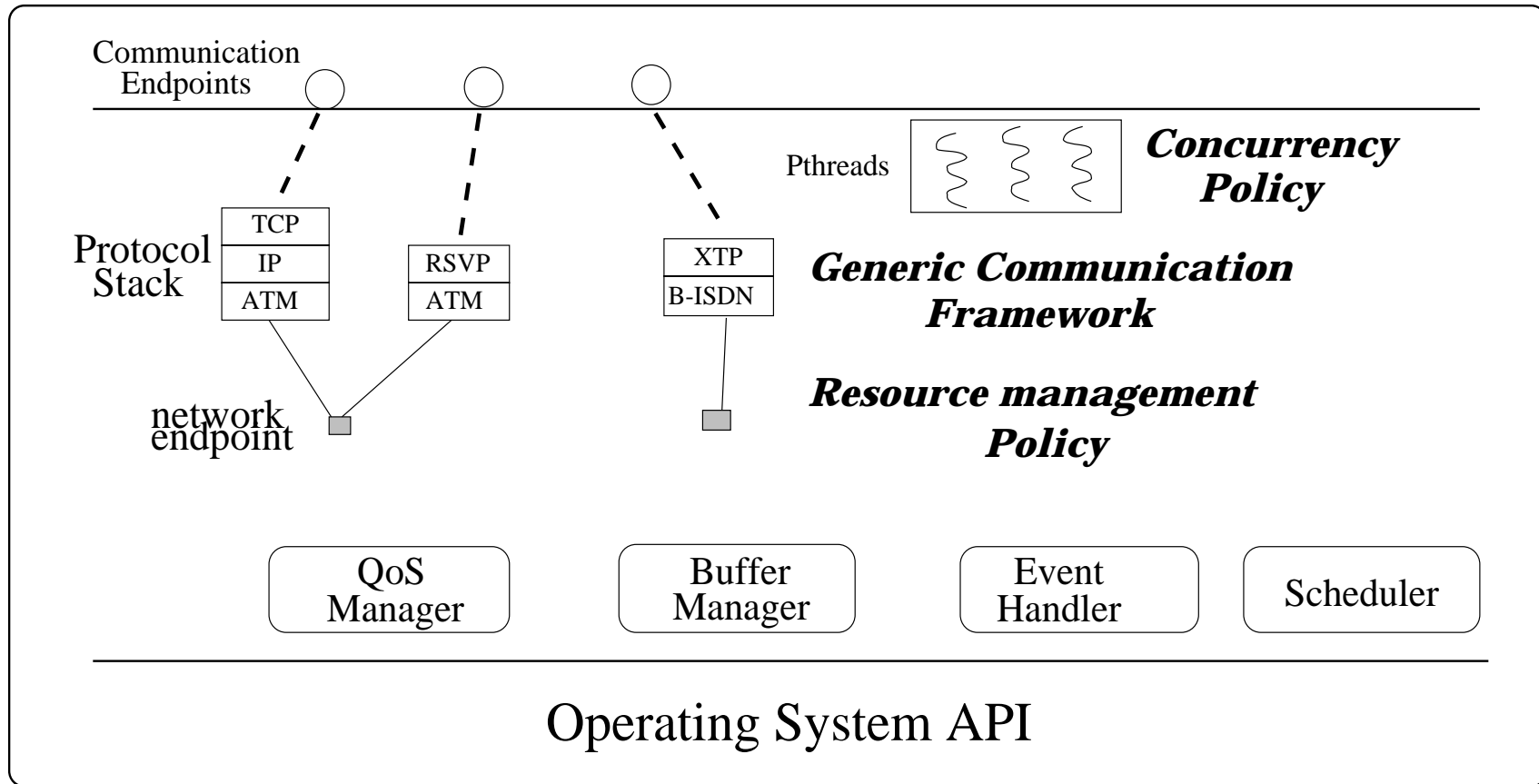
- **Developed a DIMMA IDL front-end and an ODP C++ back-end.**
- **Fixed stable interfaces between different module types.**
- **Integrated with the Type Checker and Inferencer work done within the ANSA C4a workpackage.**
- **Working on the integration of the CORBA IDL front-end into our tools platform.**



## Motivations for a new engineering engine.

- **Not happy with existing ones !**
- **Target a lightweight, highly modular and efficient engine.**
- **Simultaneous Support of several, dynamically configured protocol stacks.**
- **Support of QoS communication.**
- **Use of current emerging network technologies (ATM, B-ISDN, etc.)**

# DIMMA Nucleus Architecture





## Current Progress Status

- **Generic Communication framework.**
  - Protocol module drivers for TCP and UDP.
  - A simple RPC protocol (doesn't understand concurrency yet).
- **Scalable data management scheme (Use an i-node like indexing algorithm for locating interface references from handles).**
- **C++ wrapper patterns for the Posix threads package.**
- **Timing management.**



## Next Step...

- **Finish the ODP runtime by adding the missing bits and connecting it to the DIMMA nucleus.**
- **Understand the translation between CORBA and ODP semantics and incorporate the CORBA IDL front-end into the stub generator.**
- **Working on a nucleus release by the next TC.**