



---

**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**

---

## **Training**

# **Amber streams presentation**

**Andre Kramer**

### **Abstract**

Presentation for Univ. of Kent MSc Seminar.

---

APM.1713.00.01

**Draft**

6th March 1996

Briefing Note

---

**Distribution:**

**Supersedes:**

**Superseded by:**





# The Amber Project:

**Streams flow on!**

**Andre Kramer**

**ak@ansa.co.uk**



## Communications Applications

- **Telecommunications market is huge**
  - EDI, CSCW, WWW, DAVIC
- ***but:* applications generally lack Architecture**
  - Low level of abstraction
  - => can benefit from CORBA Technology
- **Interactive multi - media:**
  - Many formats and protocols
  - Support application level processing
- **Standardization efforts**
  - **OMG TeISIG: CORBA and Streams**
  - => can apply APM experience (ODP, TINA, DIMMA)

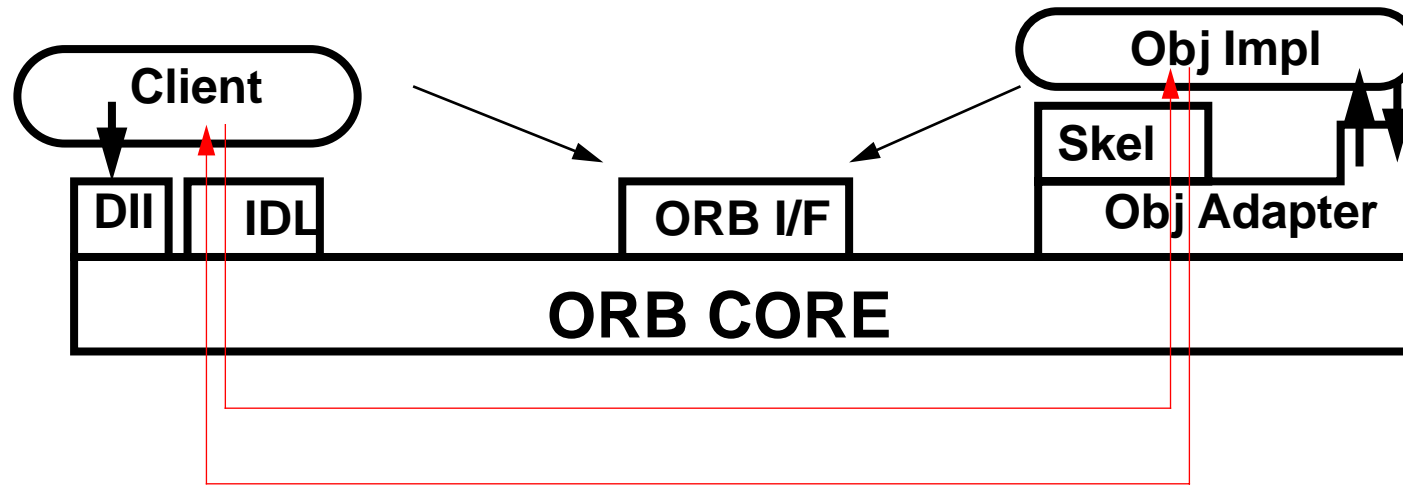


## **An aside on CORBA**

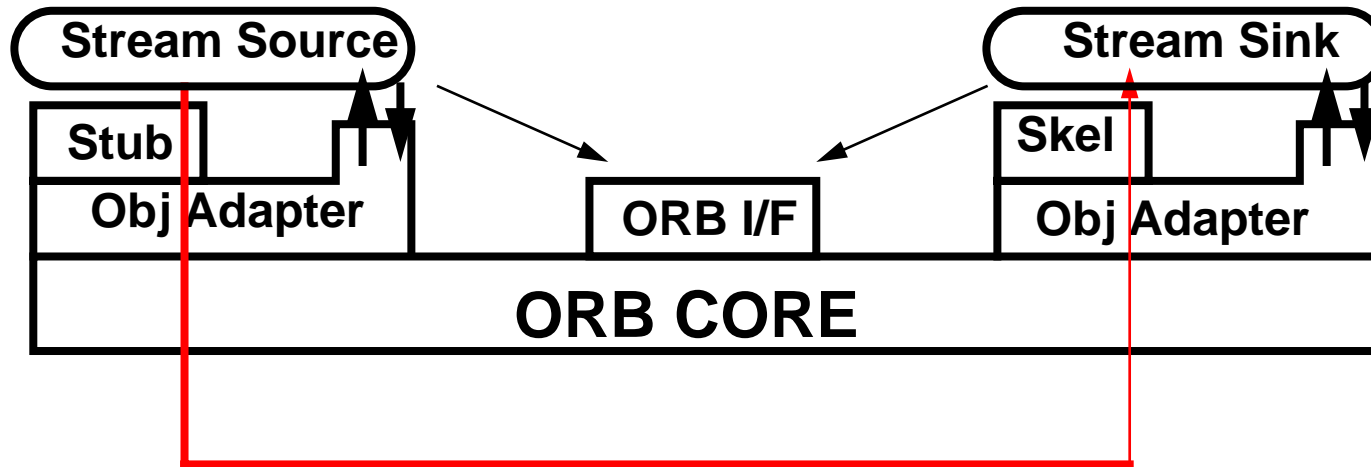
- ***Aim: Portable Distributed Applications***
- **Object - oriented: Objects distributed over ORB**
- **Interfaces: IDL and RPC, operations and object references**
- **Language mappings: C, C++, SmallTalk etc.**
- **ORBs usually engineered for scalability**

## **OMG - Architecture and Interfaces**

- **CORBA 2 - Interoperability between ORBs**
- **Object Services: Life-cycle, Transactions, Security, Events etc.**



## CORBA Architecture



## CORBA and Steams

## Risks



- **Unique opportunity,**
  - to apply CORBA to comms-oriented applications
- **Monolithic solutions**
  - Plug and Play
  - Multi - Media: CBR v.s. Adaptive
- **Complex, conflicting standards:**
  - need incremental approaches
- **Must demonstrate that our approach works:**
  - in face of high heterogeneity
  - has general applicability





# *How to extend **CORBA** for Telcomms?*

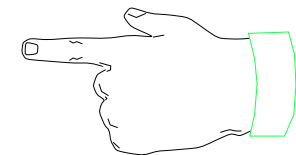


## Extensions

- Streams - directional flows of typed frames
- Explicit binding - stream interfaces & communication end-points

## Options

- **1) Build as Services on existing ORBs**
  - c.f. IMA MSS proposal
- **2) Add connection management**
  - plugs and sockets
- **3) Both connection and interaction support:**
  - application level sources and sinks: s/w stream processing

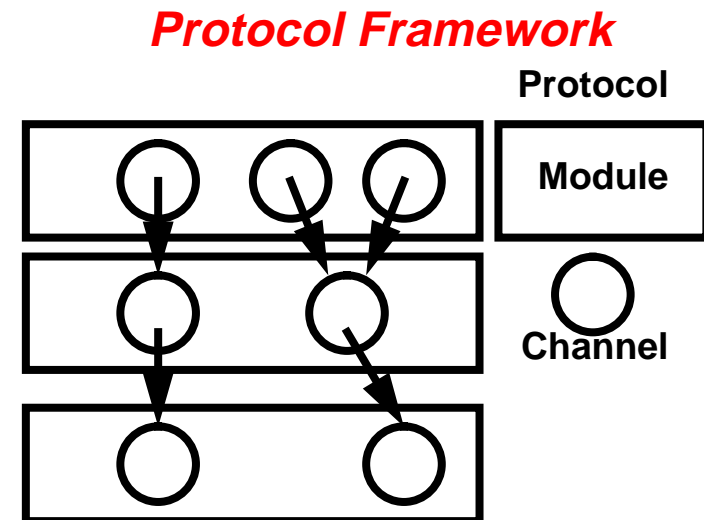


## Recommendation

- Identify set of CORBA extensions
- Demonstrate in ORB and for MM domain:

### *the AMBER Project*

- *Integrate with:*
  - DIMMA Protocol Framework
  - ReTINA Real-time Tele-comms DPE
  - Synchronous programming





## CORBA Implications

- **Minimal IDL extensions**
  - Interfaces with typed data “frames”
- **Overload Object Adapter Concept**
  - **Object Adapters <-> Stream Adapters**
  - **Object Interfaces <-> Stream Interfaces**
  - **Object References <-> Stream References**
- **Add explicit resource management**
  - **Protocols, connections, QoS**
  - **Resource Pools**

**=> Extendable ORB**



## **Benefits**

- **Comms and DPE convergence**
- **Comms services enabled CORBA: use as “Integration Glue”**
- **Demonstrators:**
  - **Amber Vic MBone Tool**
  - **CORBA IOE Extensions**
- **Leads into real-time (synch. programming), QoS Arch.**



## Existing CORBA Connection Model

- **Implicit, late - binding**
  - Engineered for efficiency and scaling
  - Pervasive resource multiplexing
- **Client / Server RPC Model**
  - Asymmetric
  - Best - effort, focus is on inter-operability
- **Need to exercise control:**
  - Multi - party or third - party connections
  - Prioritize and synchronize communications
  - Provide QoS guarantees
  - Manage connection life-times, re-configuration



---

## Application Layer Framing

- **Frames:**

- Sequence of messages from source to a sink
- Multiple frame types
- e.g. video & audio, base / delta coding, in-band cntrl
- Marshalled by stream “stubs”

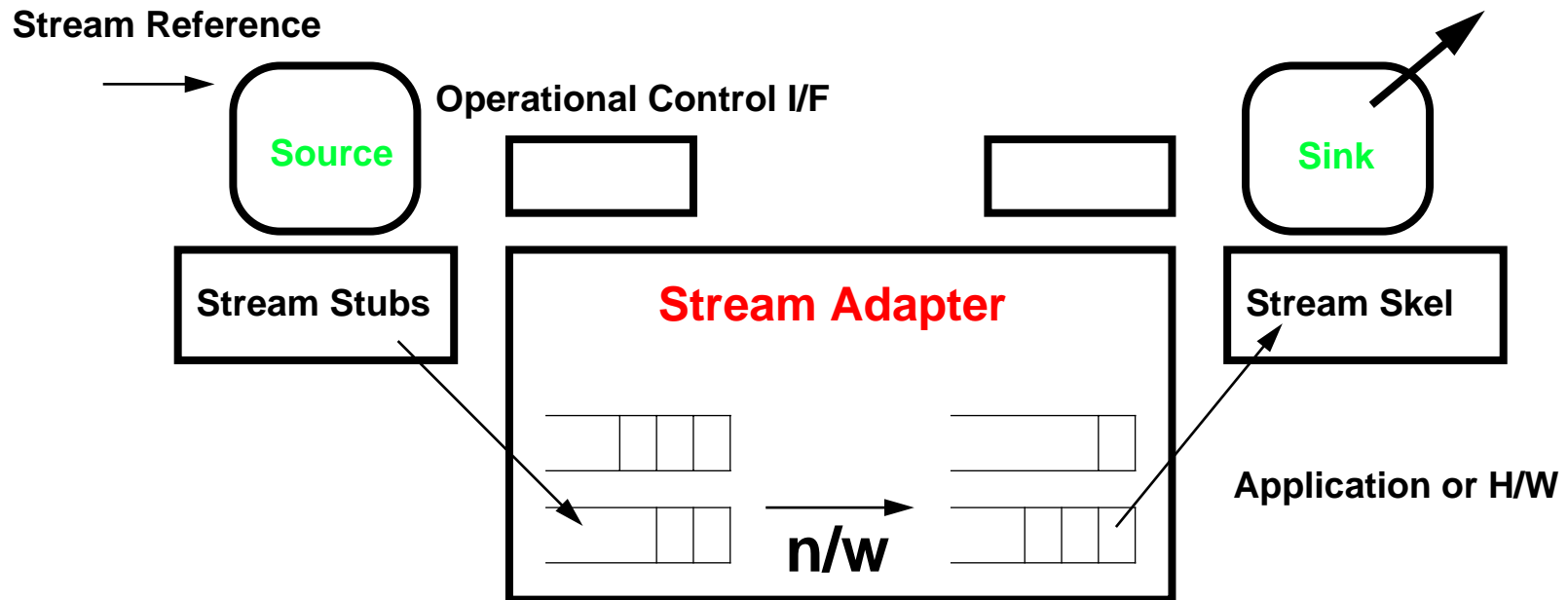
**Frames as “oneway” non-blocking operations**

- **Streams:**

- Bunches of frames - *Flows*
- Explicit connection set-up
- Stream references

**Streams as extended IDL interfaces**

# Streams and Adapters







## Stream IDL

- **Frames and Streams: CORBA IDL Interfaces**

```
stream Conference {  
    video (Participant, VideoFrame);  
    voice (Sound_bite, sequence of Speakers);  
    whiteboardDraw (Item);  
};
```

- **Frame grouping: unidirectional Flows**
  - **Basis for conformance based type checking**
  - **Binder generation -- for sets of interfaces, binding language**
- **JET C++ mapping**



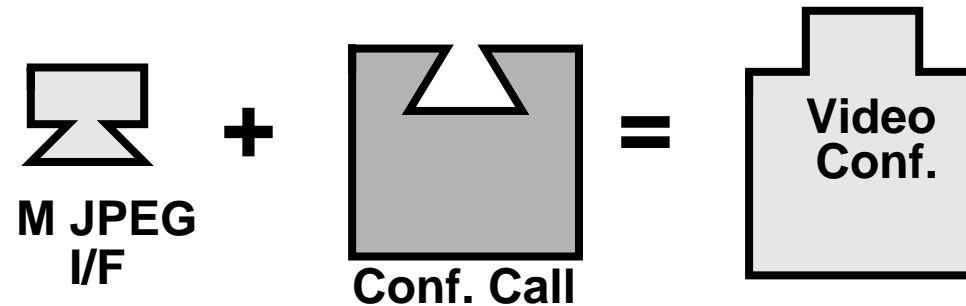
## Flexible Binding

- **Stream Interfaces plug into Stream Adapters**
- **Stream Adapters Types:**
  - **Primitive Adapters:** for std. multi - media formats
  - **Composable Adapters:** for application - level stream processing
  - **Templated Adapters:** for generic connection models

### *Overload Object Adapter Concept!*

- **Forms basis for standardization, vendor differentiation**
- **Examples: Multicast groups, HTTP-ng, MM data-flow**

## Templated Adapters



- **Generic adapters (based on CORBA type ANY)**
- **Adapters parameterized by resources, policy, QoS**
- **Template generation for complex binders; specializedmarshallers**

```
AdapterObject = <AdapterT>.new(Pools, ...);  
boundRef = AdapterObject.bind(unboundRef, EndpointRef, QoS, ...);  
boundRef->message(contents);
```

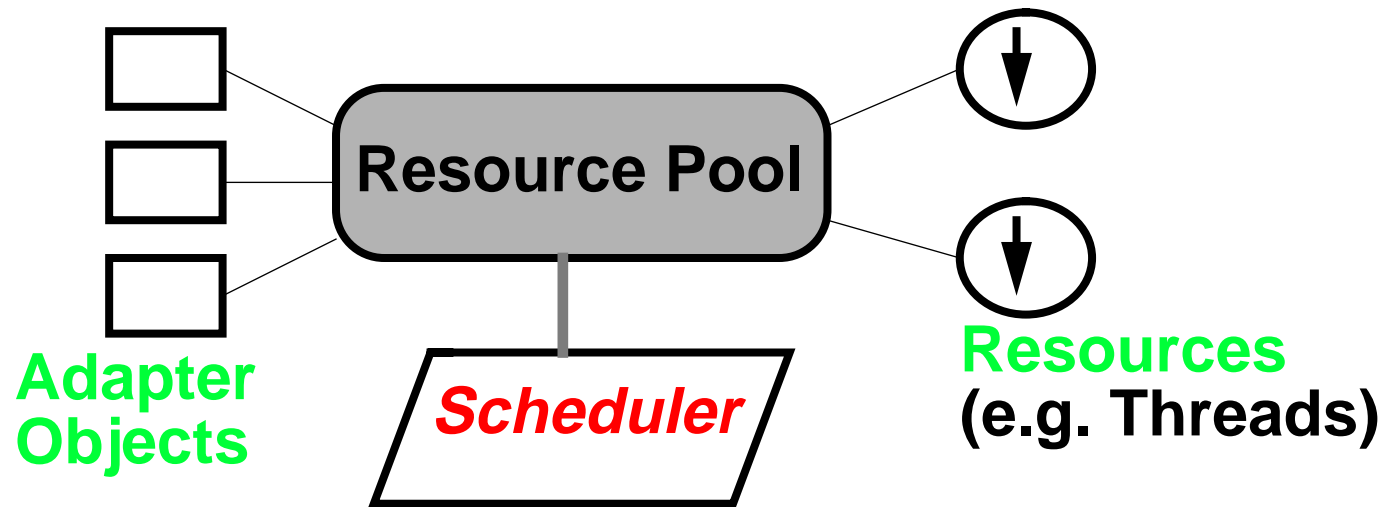


---

## Adapters and Resource Pools

- **Control muxing of all resources**
  - Threads, buffers, protocol endpoints
  - Low-level and system level abstractions: memory, sockets
- **Avoid application **cross-talk** => separate at low level**
- **Adapters parameterized by **policy****
  
- **Upcall based frame delivery at receivers**
  - Thread shepherds message
  - One upcall at a time per interface

## Resourced Stream Adapters





## Synchronous Programming

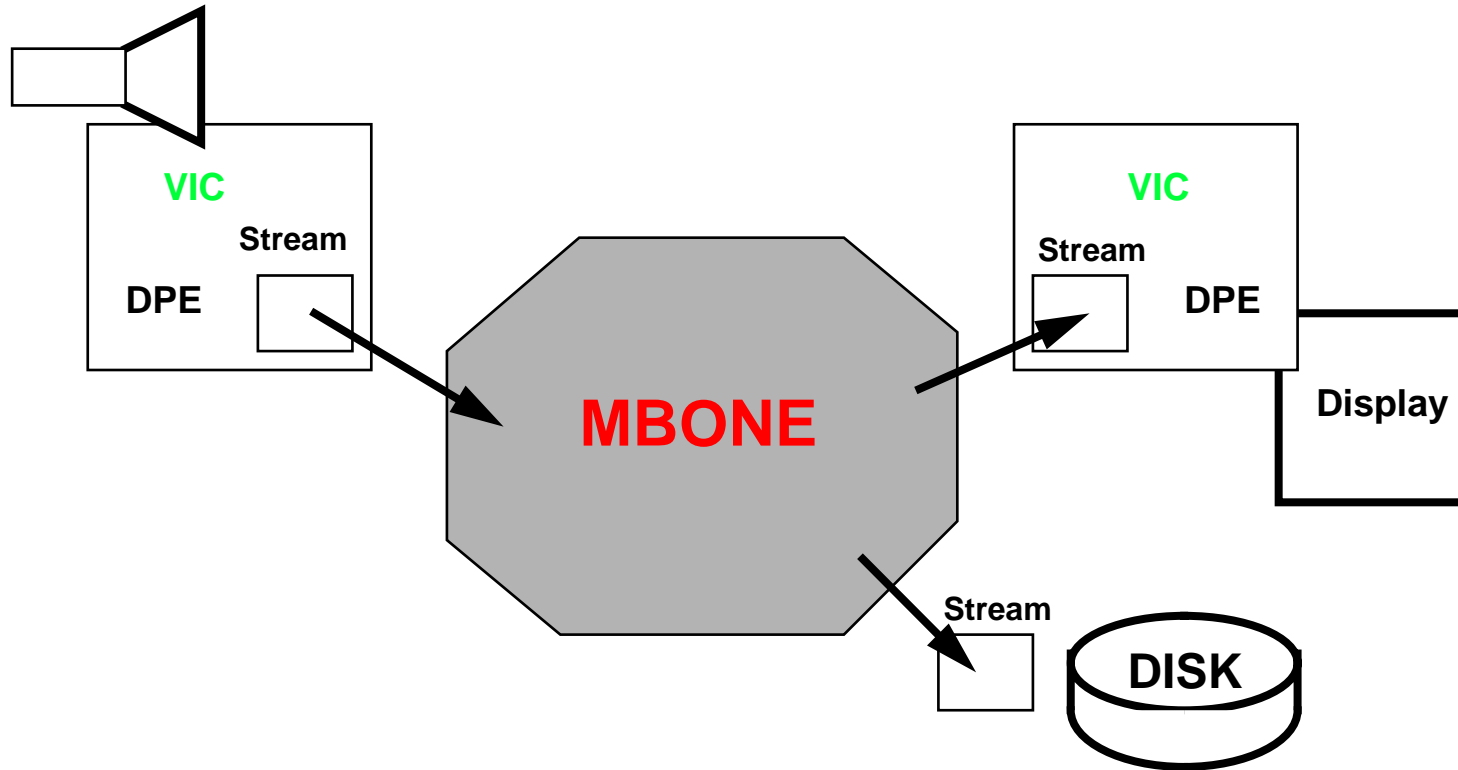
- **Model Controller part of application as a Reactive Object**
  - Interfaces to asynchronous objects (and async. world)
  - At each **tick**: take inputs, process “instantaneously”, emit outputs
  - Resources and real-time synch. guaranteed by OS
  - Predictability: Deterministic (worst case) behaviour
  
- **Synchronous Languages**
  - e.g. **Esterel** from INRIA
  - Clocked like hardware
  - alarm ? chime; bell ! ring || button ? press; bell ! ring
  
- **Synchronous Invocation Interface (SII) for Stream Adapters**
  - c.f. CORBA DII and DSI



## Amber VIC (1)

- **Demonstrate approach applied to Internet MBone**
  - Streams: contrast CBR with adaptive MM approaches and ALF
  - Binding: investigate loose, multi-party conference models
- **MBone (Deering Multicast)**
  - Multicast IP
  - Virtual backbone of multicast routers
- **VIC video tool**
  - Sessions with multiple sources
  - Framework for multiple encodings
  - RTP - transport for multiple encapsulations with time

# Amber VIC







## Amber VIC (2)

- **Results:**
  - MBone multi-media testbed
  - Vic over DIMMA protocols, RTP support
  - General DPE integrated with MBone video tool

### *Combines CORBA and Internet*

- **Future:**
  - Predictability via real-time support
  - Investigate ATM n/ws, reservation protocols (RSVP)



# Wrap Up

- Comms - oriented applications need CORBA
- New concepts: streams, explicit connections, pools, QoS
- Identified CORBA extensions: stream IDL, adapters, pools, SII
- Amber MBone demonstrator
- Firm foundations: ODP, DIMMA, TINA-C / ReTINA

**MM <-> CORBA <-> COMMS**