Real-Time CORBA MMS for Embedded Systems

OMG-Embedded Systems Workshop

January 2001, Santa Clara

Eric Gressier-Soudan, Erwan Becquet

gressier@cnam.fr, becquet@cnam.fr

Laboratoire Cedric-CNAM

PARIS-France

Real-Time Messaging Service based on CORBA for Manufacturing and Process Control Applications

Distributed System approach applied to embedded systems.

Sources :

- MMS, Manufacturing Message Specification, bring a structuring model

+

- TASE.2, Tele-Control Application Service Element version 2, real-time services

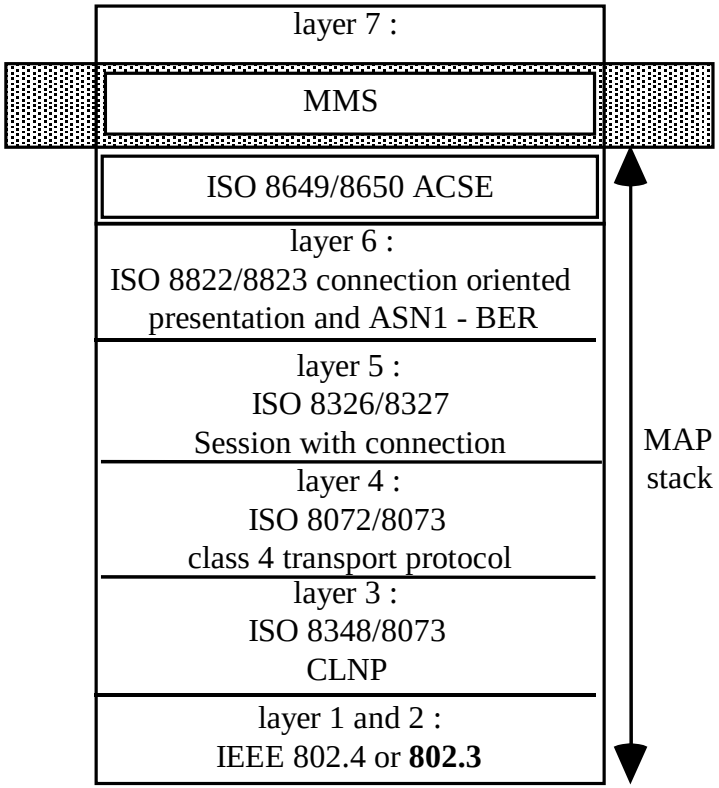
Define

An Industrial Messaging Services + Real-Time behaviour in an ISO environment

Our Goal :

Add Object Oriented design to the previous framework

MMS and MAP from ISO

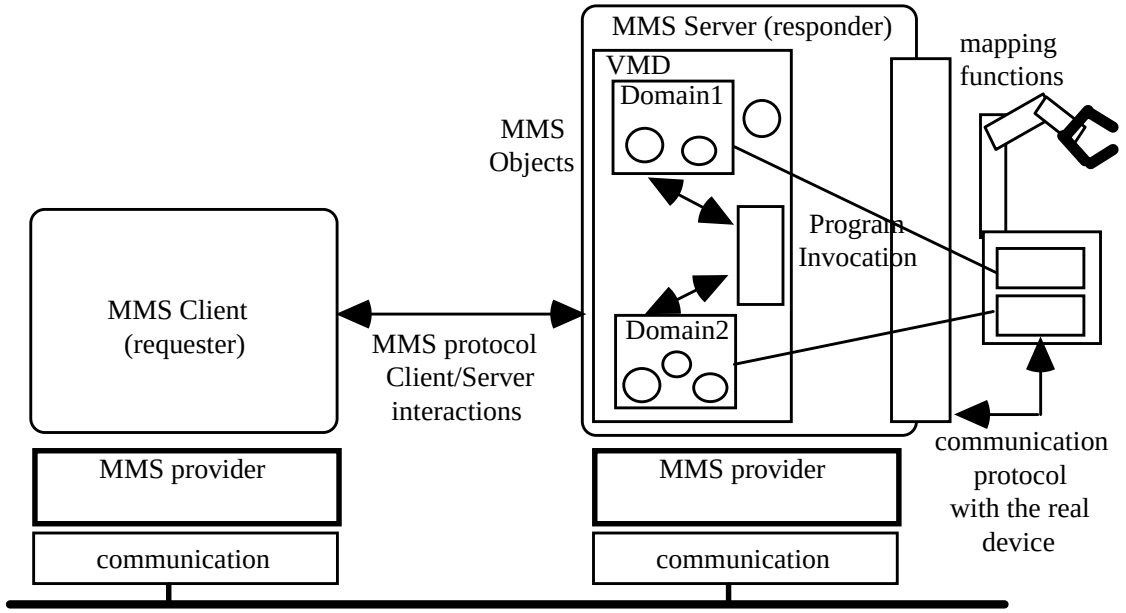


Defines mains functions of an industrial messaging protocol issued in the 90's

can be extracted from MAP

NO REAL TIME, NO OBJECT ORIENTED ISSUES addressed

MMS main features



- Basic Abstractions : **Virtual Manufacturing Device (VMD)**, **Domain**, **Program Invocation**, **Variable**, Semaphore, Event, File...

- OSI services defined for each abstraction

- Client/Server based Interactions + Asynchronous Messages

To be noticed : users' requirements assert "MMS over TCP/IP would be clever"

TASE.2

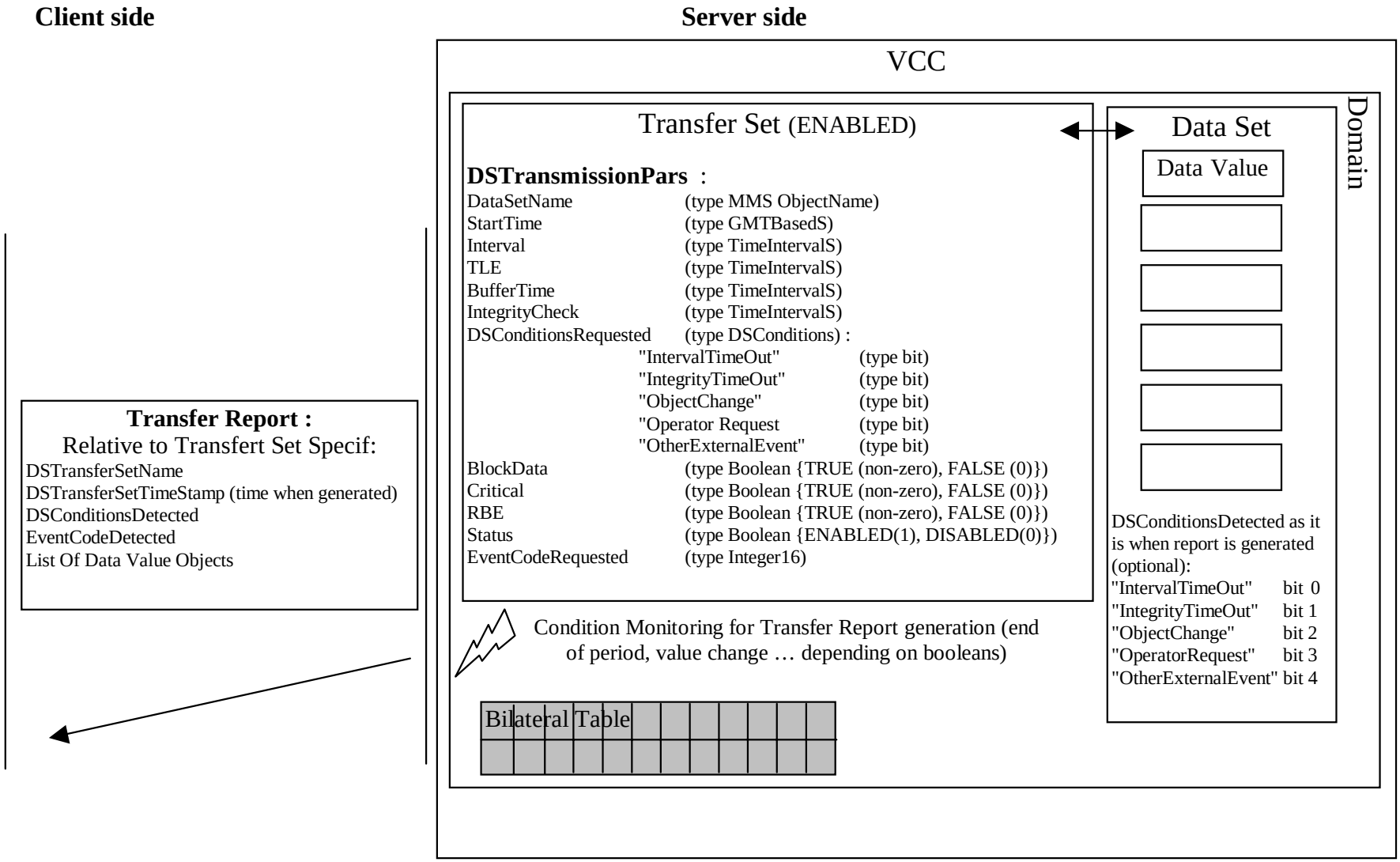
Specifies data exchange for power supply between control centre and production centre:

- 9 functional blocs :
  - bloc 1 : periodic data exchange
  - bloc 2 : event base data exchange
  - bloc 5 : remote control of device

- on top of MMS services and abstractions, for the 3 previous blocs, we need:
  - VMD
  - Association (for access control mainly)
  - Variables
  - List of Variables
  - Domains

  Use Read/Write, and Unsollicited Services mostly

TASE.2 Bloc 1 and 2 abstractions



How can we build a RT-CORBA industrial messaging services

What do we need ?

- Fieldbus like services :
  - Producer/Consumer exchanges
  - Client/Server exchanges
  - Bounded delays
  - Real-Time Communications:
- Object Oriented Abstractions
- Distributed System platform

What do we get ?

- The basis of an object model with MMS and TASE.2 abstractions
- Client/Server and Producer/Consumer interactions with a real-time behaviour
- No Real-Time communications since classical networks are used

Ways to Implement

Add a TASE.2 component on top of CORBA-MMS ?

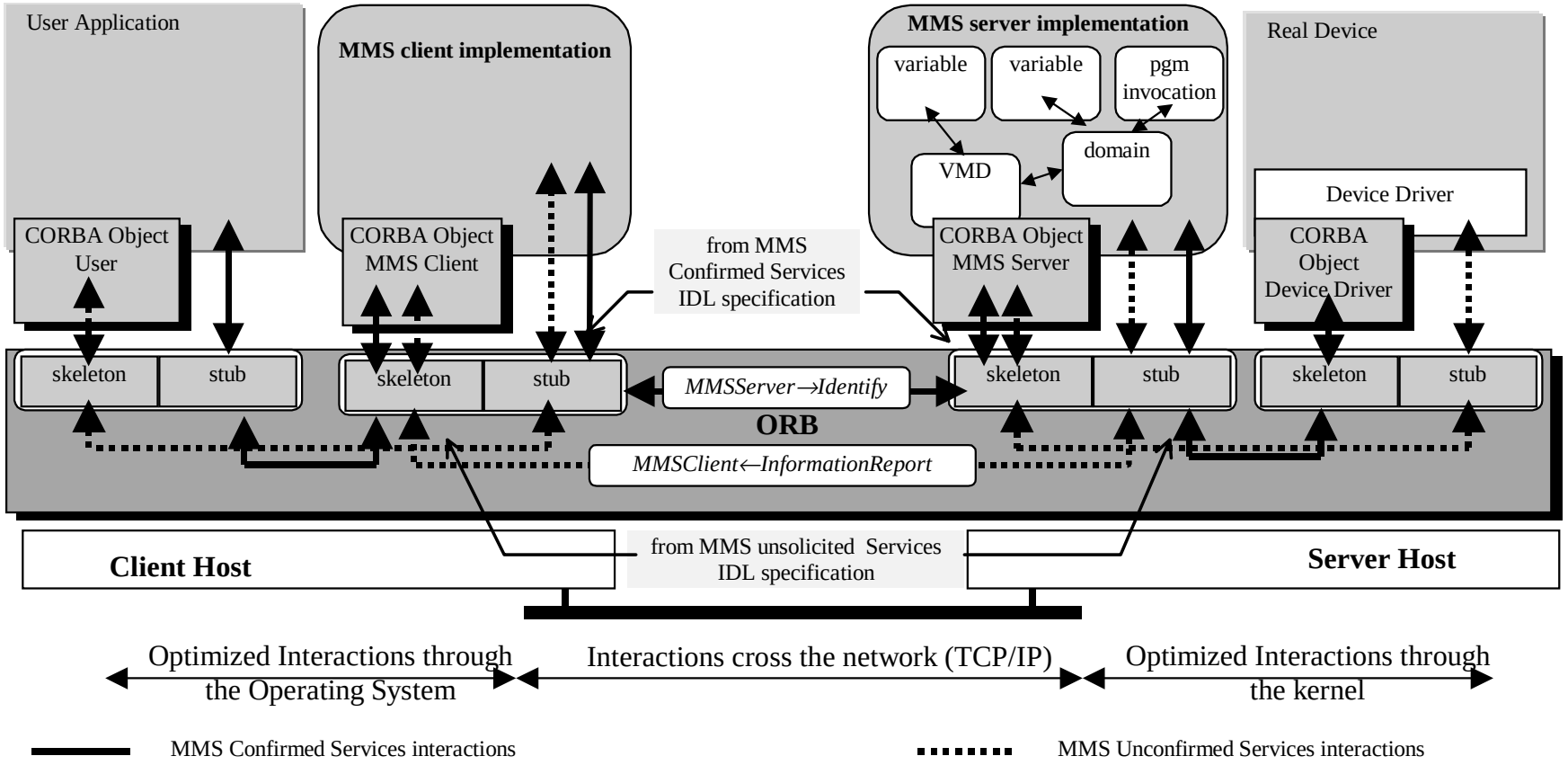
- Similar to the TASE.2 specification
- Heavy to implement

Build new TASE.2 like services from scratch ?

- Lightweight solution
- Nice for prototyping
- CORBA-MMS services are lost

Extend CORBA-MMS with TASE.2 functions !!!

Objectified MMS architecture model



VMD = CORBA object, other MMS abstractions implemented as programming objects

IDL specifications use MMS services and PDUs, method invocations are synchronous

(request PDU => parameters , response PDU => results , error PDU => exceptions)

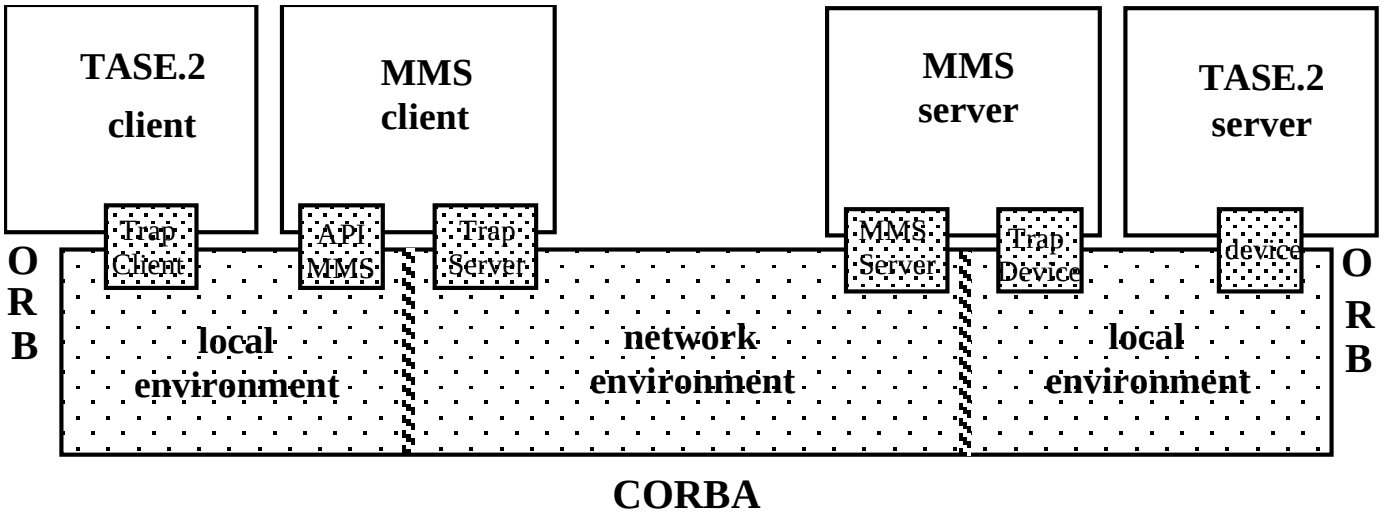
Losses from the ISO standard

- Association vanishes : association scope MMS objects, association parameters negociation, PSAP addressing disappear
  in CORBA : the ORB handles implicitly the relationship between client and server, the VMD IOR becomes its name and gains location transparency

- MMS Requests are concurrent and asynchronous, the ORB solution is synchronous and needs multi-threading support in server and client sides

- Unconfirmed services from the server to the client (*UnsollicitedStatus* and *InformationReport*) are mapped on synchronous methods without result parameters, the MMS server should be a CORBA client and the MMS client should be a CORBA server too … perhaps we could use AMI if available (we used oneways in first experiments) !

CORBA-MMS + TASE.2 components could be done easily



CORBA-MMS runs over Linux and Windows 98, using ORBacus, and Jonathan ORBs and is implemented in Java

An old version written in C++ exists, and runs over Chorus COOL

RT CORBA-MMS implementation project

- pSOS+ real-time micro-kernel
- Java implementation with a RT-JVM : PERC from NewMonics (CNAM is beta tester of PERC)
- Jonathan flexible ORB with a CORBA personality and a RMI personality

Application targeted :
Java Based Embedded Remote Monitoring Tool for Small and Medium Power Plant Units (feasibility demonstrated with partial building blocs)

Addressing Real Time Constraints related to interactions between objects

Real Time Communications :

- a QoS aware network: CORBA-MMS ran over ATM as a prototype, not sufficient, we need end-to-end QoS management

- COTS products : use Ethernet + TCP/IP with DiffServ aware products or domain specific networks (pluggable protocols in ORBs !!!)

Not sufficient :

We need priority management at the ORB level … the distributed scheduling framework from RT-SIG is a first answer !

Software engineering in embedded systems needs proofs … We need a validation tool dealing with distributed real-time constraints.

Conclusion

Our proposal satisfies the requirements of object oriented embedded systems, and, we offer equivalent functionalities as fieldbus systems.

Now :
Implementation of RT-CORBA-MMS to start !

Next:
Find an ORB able to address Real Time Constraints related to interactions between objects !!

More ?
Sponsors ;-)

After ???
Submit an RFP ?!?