TINA Connection Management implementation over Distributed Processing Platforms

Graziella Spinelli [1], *Wataru Takita* [2], *Giovanni Martini* [1], *Sakae Chikara*[2]

[1] CSELT IRI–STET GROUP
CENTRO STUDI E LABORATORI TELECOMUNICAZIONI SpA
via G.Reiss Romoli, 274
I-10148, Torino , Italy

[2] NTT NIPPON TELEGRAPH AND TELEPHONE CORPORATION
9-11, Midori-Cho 3-Chome
Musashino-Shi, Tokyo 180 Japan

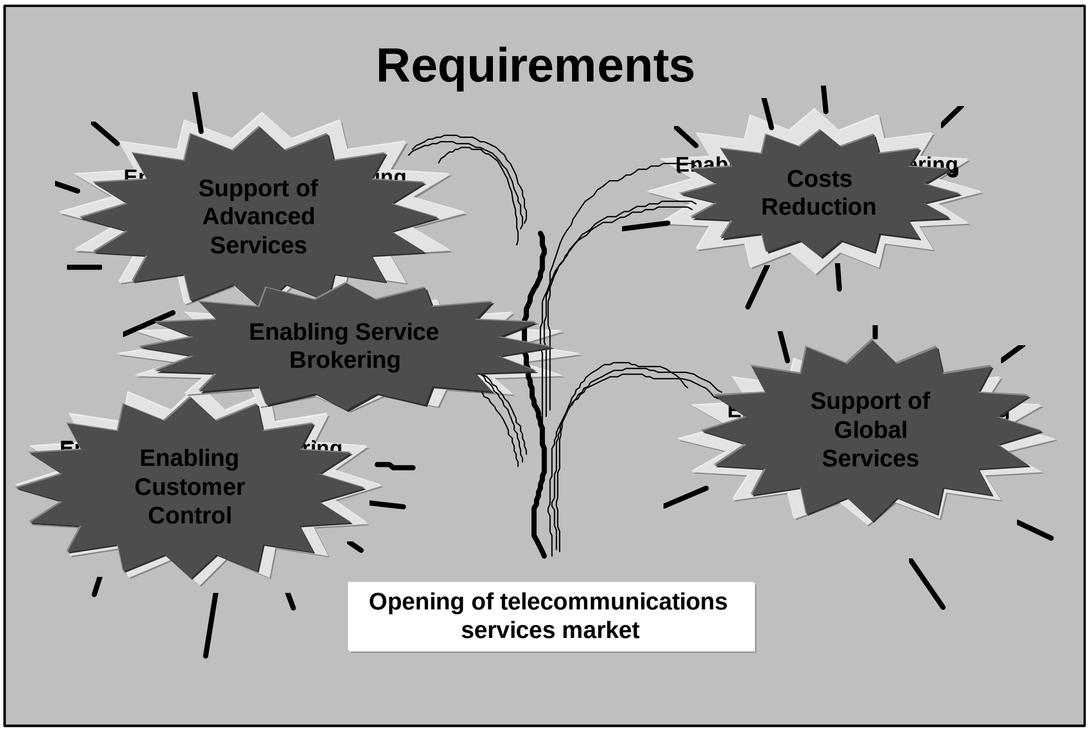
TINA architecture is defined to be applicable to software applications operating in the network, and it is out of question that one of the more straightforward impacts is on management applications.

The object oriented paradigm has been widely accepted to be the mean for structuring management communications. Moreover network management platforms operating on top of general purpose hardware and software computer systems are already available on the market, thus responding to requirements, such as modularity and flexibility, that are part of the objectives of the TINA architecture.

Goal of this paper is to describe the results of the experience of implementing a management application, specifically a Connection Management application, structured according to object-oriented paradigms and running on multiple interoperable distributed processing environments.

Traditionally Connection Management has been always considered as control operations which are view as being different from management. In case of TINA Connection Management we deal with real-time and dynamic management operations.

After a brief introduction on the chosen application scenario and the consequent generic requirements on the Connection Management system, an implementation case is described especially focusing on software platform aspects and interoperability of applications.



Requirements

Support of Advanced Services

Enabling Service Brokering

Enabling Customer Control

Costs Reduction

Support of Global Services

Opening of telecommunications services market

Multiple factors impact on the strategy for defining a new way to manage the communication network:

The opening of the services market will bring several new actors in the telecommunications arena: most notably service brokers will act as one-stop-shopping sites in the case of international communication services.

Market competition impose to service providers minimum costs versus maximum efficiency.

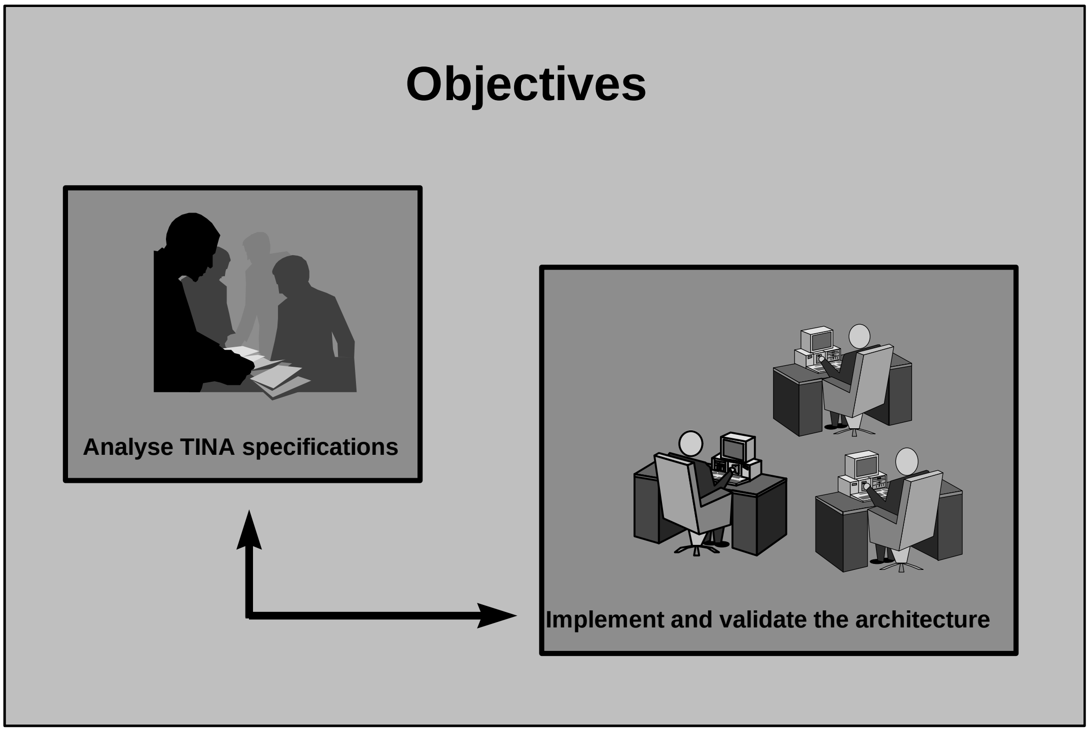
Globalisation and multinetworking

Customer control capabilities

New services characteristics highlight the necessity to have distributed information on the network and consequently to benefit from the realization in these context of principles such as “location transparency”.

Network heterogeneity and specialisation of management applications tied to the necessity of the network operator who manages the network.

TINA tries to provide an answer to the above requirements.



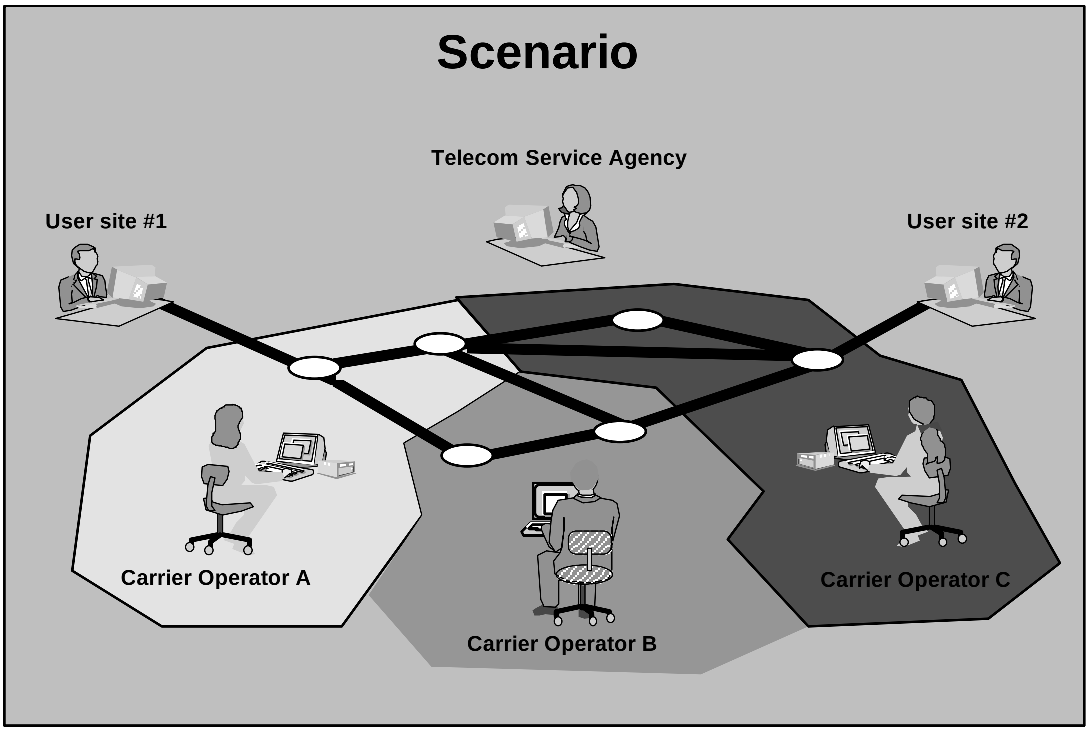
Objectives

Analyse TINA specifications

Implement and validate the architecture

One of the main objectives of our study is to validate the TINA specifications; consequently to provide feedback to the TINA Core Team.

The realization of the prototype offers the opportunity of a close analysis up to implementation details. The specification that have been produced up to now need to be consolidated; and the general approach taken in developing the structure of the architecture is to be verified. Many companies participating the TINA Consortium are willing to see how the TINA concepts can be successfully applied to provide efficiently advanced services.

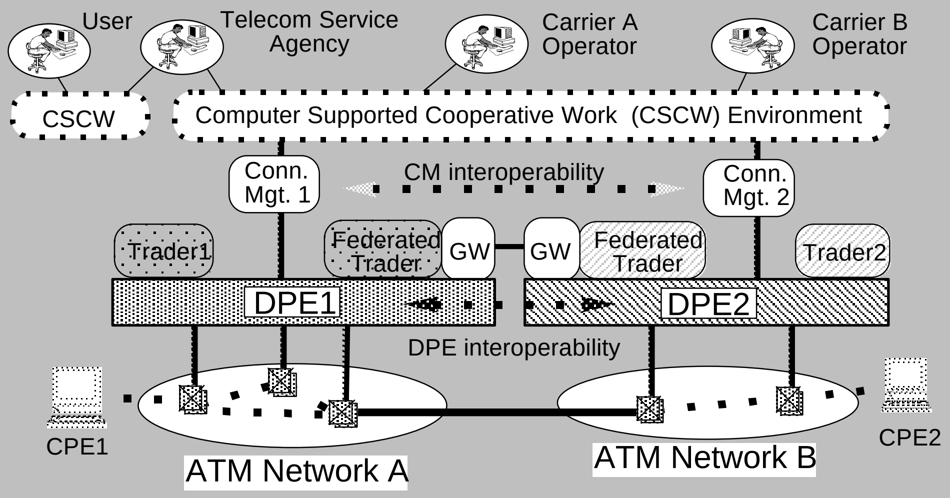


Scenario

The scenario in which the demonstration is placed foresees a TINA system to provide customizable telecommunication services to a company with several premises located worldwide. All company premises make use of a dedicated broadband connection network for their communication pourposes. The capacity/bandwidth requirements for the connections are variable and changing according to the market situation. The company refers for its communications needs to a one-stop-shopping desk called Telecommunications Service Agency. The TSA is in charge of negotiating and allocating the needed network resources as well as the requested bandwidth with the involved Carrier Operators.

All the Customer-TSA and TSA-Carrier Operators Interactions take place by means of Computer Supported Cooperative Work CSCW applications. The TSA is itself allowed to modify the capacity of the connections according to the contract.

Overall System Configuration

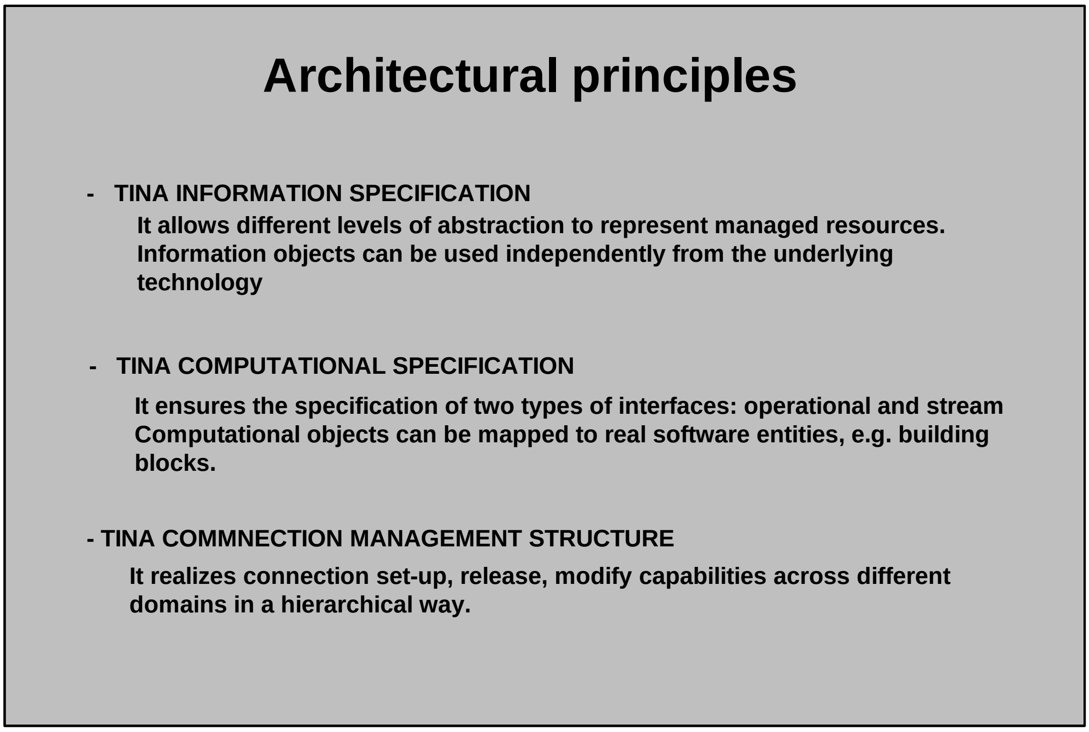


Over the described scenario, the objective of validating and implementing the TINA architecture focuses on the following issues:

- implementation of Connection Management specification. TINA connection management has been designed to be applicable to any type of underlying technology. Computational and information objects, once specialised and istantiated for the specific network used (ATM in our case), provide the connectivity between two end points. Connection management objects run over a distributed software platform. When considering connectivity spanning over several domains, federation capabilities are to be provided.

- platform interoperability. TINA computational objects are implemented over a Distributed Processing Environment (DPE). Services provided by the DPE, such as the Trader service, allow interactions between objects to be transparent. In case the domains that are to be interconnected use different platforms, they have to be interoperable.

The Federated Trader acts as a trader between different domains: it makes the necessary controls over operations invoked from “outside” objects and allows interoperability between DPEs.



Architectual principles

- TINA INFORMATION SPECIFICATION

  It allows different levels of abstraction to represent managed resources. Information objects can be used independently from the underlying technology

- TINA COMPUTATIONAL SPECIFICATION

  It ensures the specification of two types of interfaces: operational and stream Computational objects can be mapped to real software entities, e.g. building blocks.

- TINA COMMNECTION MANAGEMENT STRUCTURE

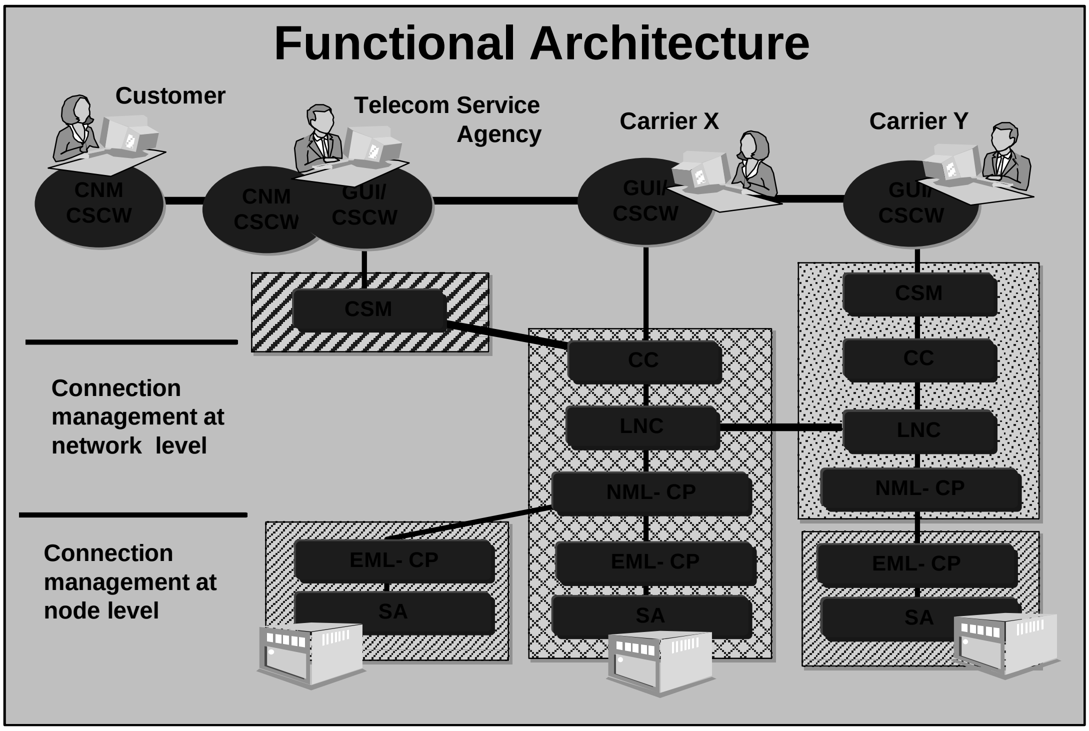
  It realizes connection set-up, release, modify capabilities across different domains in a hierarchical way.

Before going into the detail of the developed applications, the architectural principle on which TINA architecture is based, and therefore our applications, will be shown.

According to the TINA specifications, the problem domain of the system is described through the information specification. The resources of the system, both hardware and software, are then represented by information objects. The TINA information objects represent the telecommunication network independently from the underlying technology; this means that the information objects that have been specified have to be specialised when used for a specific network.

The telecommunication network, as specified by TINA, is seen at several levels of abstraction. At the highest level the network is represented by an information object called layer network. The layer network usually identifies a specific network technology (for example ATM, SDH and so on). The layer network can be subdivided into smaller parts called subnetworks. In turn subnetworks can be further subdivided into smaller and smaller subnetwork until the lowest level of abstraction is reached. At this level the smallest subnetwork represents a single node of the network.

The TINA software applications, such as the connection management applications we have developed for the demo, are described in term of computational specification: a set of software entities, called computational objects, provide the application by means of interactions with each other. Every computational object may provide one or more interfaces, and every objects can send and receive information to/from other objects through the interfaces.



Functional Architecture

Customer

Telecom Service Agency

Carrier X

Carrier Y

CNM CSCW

CNM CSCW

GUI/ CSCW

GUI/ CSCW

GUI/ CSCW

CSM

CSM

CC

CC

LNC

LNC

NML- CP

NML- CP

EML- CP

EML- CP

EML- CP

SA

SA

SA

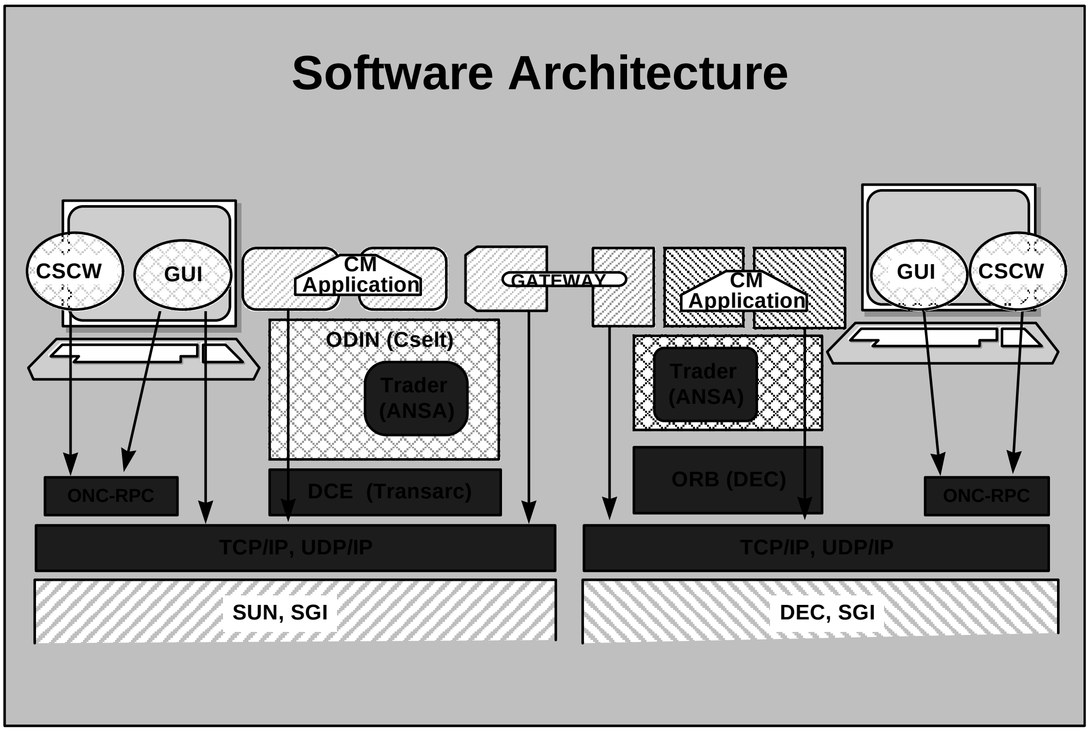
Connection management at network level

Connection management at node level

A set of computational objects defined to support connection management activites have been implemented.

The architectural concepts suggested by the TINA specifications have been followed. The telecommunication network can be composed by several layer networks each of them identifying a specific network technology (ATM, SDH, etc.). There is one LNC (layer Network Coordinator object for each layer network that is in charge of setting up an end-to-end connection within that layer network. This computational objects also takes care of federation capabilities: operation of connection set up and release that come from another domain are checked and possibly processed by this computational object.

The LNC invokes operations on the NML-CP (Network Management Layer Connection Performer). According to the design of the telecommunication network, as far as TINA is concerned, there are several levels of Connection Perfomer objects. Each of them is in turn in charge of managing a set of CPs; the CP at the lowest level of the hierarchy provides connectivity within the smallest subnetwork, that is the node itself. It is also responsible of selecting the specific termination points of the connection.



Software Architecture

In this diagram the global schema of the software architecture is depicted. As it is evident from the diagram itself, the demonstrator application has required the integration of many technologies at different levels of representation of the software.

Without deeping too much into details, we prefer concentrating on some major architectural items.

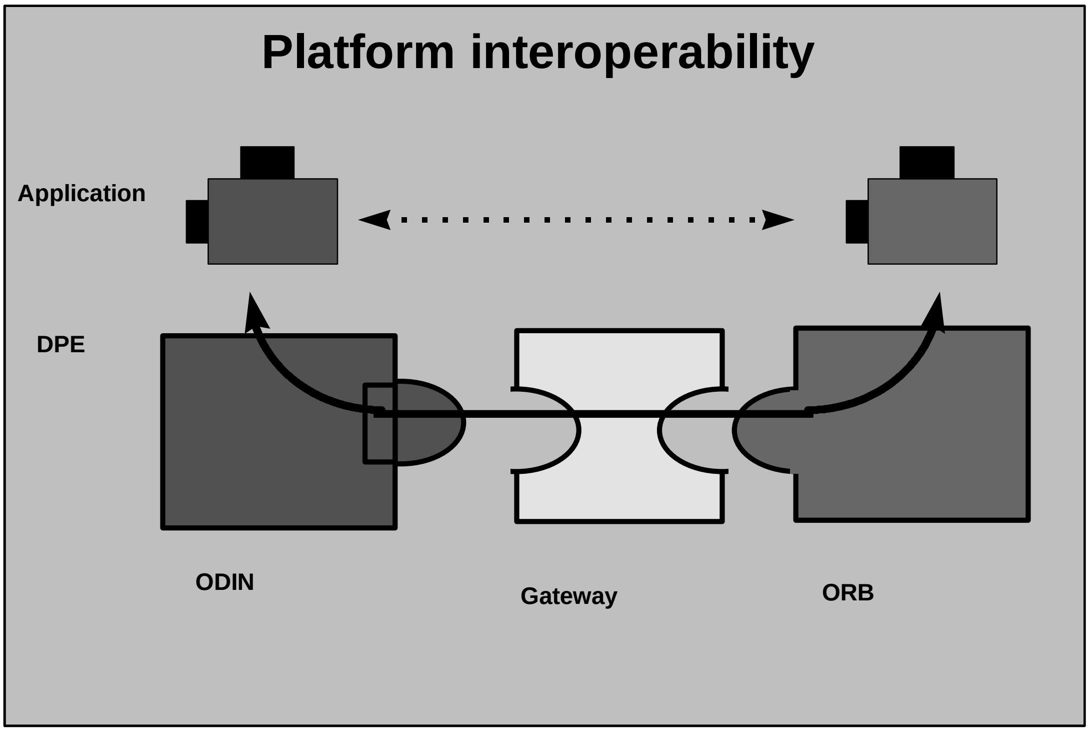
• Two different distributed platforms have been used: ObjectBroker (ORB) and Object-oriented Distributed INterface (ODIN [1]), internally developed in CSELT and based on OSF/DCE.

• The “building-block” and “contract” concepts have been extensively applied in the application definition: the platform makes transparent the physical distribution of the building-blocks on different nodes.

• Two distinct hardware platforms (SUN and DEC) are used to support the distributed platforms.

• An “ad hoc” platforms interoperability mechanism (gateway) has been devised, relying completely on the bare transport level.

[1] P.G.Bosco,E.Grasso,G.Martini,C.Moiso,”A TINA-like computational model and its implementation within an object-oriented transactional platform”, in Proceedings of SDNE’95.



Platform interoperability

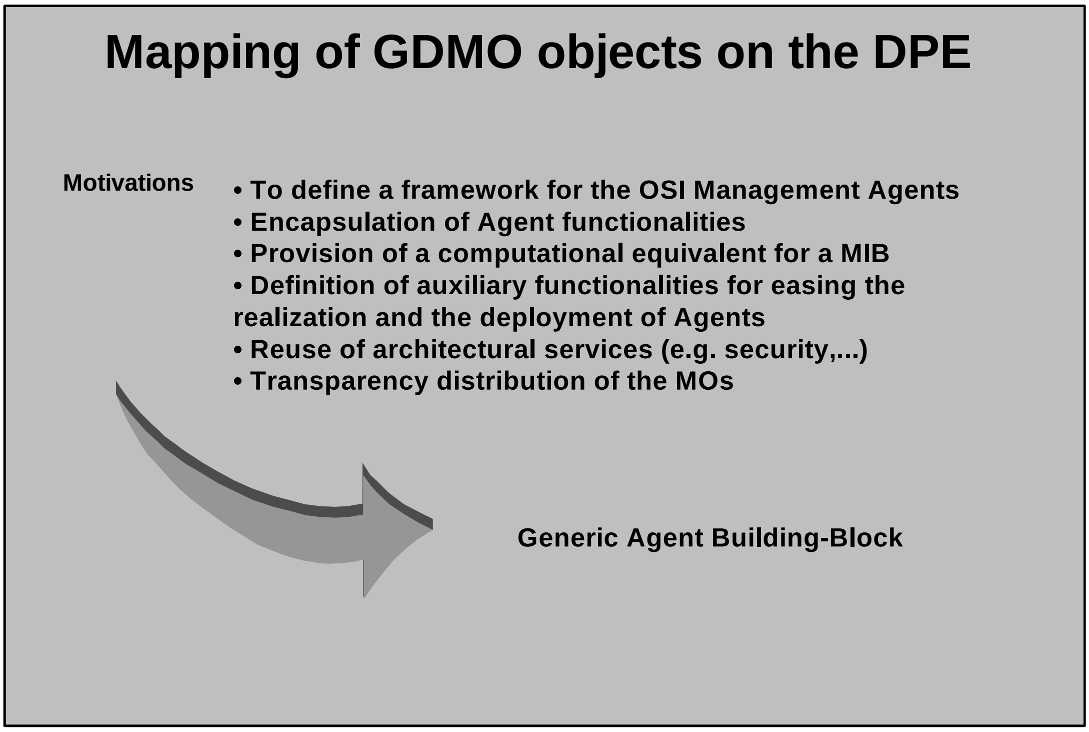
As previously illustrated in the functional architecture, the connection management applications belonging to different operators, at same time in their cooperative activity, have to explicitely federate each one with the other.

The federation activity takes place between the contracts of computational objects, each one running on a separate platform. As illustrated in the previous point, the technology used is different in each carrier’s domain, and it makes impossible for the DPEs to communicate directly.This has posed the problem of interoperability between the platforms: it has solved by devising an ad-hoc gateway mechanism between the DPEs.

The “gateway” allows computational objects running in one domain to address transparently a selected subset of objects of the other domain, as if they were running in the same one. In this way at the application level it is possible having interobject communications preserving access and location transparencies.

In order to let that one object’s interface is visibile by the other domain trading space, it is up to the gateway itself the export of that interface in the appropriate trading space. This sinergy between gateway and trader (the federated trader) tackles completely the interdomain federation problem, both at the level of interface visibility in an external trading space and at the interoperability level.

In each domain, the actual technology used to communicate with the objects running in the other domain (transport mechanism, marshalling/unmarshalling of messages, etc.) it is hidden.



Mapping of GDMO objects on the DPE

Motivations

- To define a framework for the OSI Management Agents
- Encapsulation of Agent functionalities
- Provision of a computational equivalent for a MIB
- Definition of auxiliary functionalities for easing the realization and the deployment of Agents
- Reuse of architectural services (e.g. security,...)
- Transparency distribution of the MOs

Generic Agent Building-Block

The specifications of the cooperative connection management application has been developed by using a quasi-GDMO formalism.

This fact, and the importance of devising a schema for applying and integrating OSI Management concepts onto the TINA architecture, has led to the definition of Generic Agent Building Blocks.

The motivations at the basis of this choice may be summarized in the following points:

• the definition of a framework for the OSI Management Agents that allows their definitioin, realization and deployment on TINA-like environments;

• the encapsulation of Agent functionalities into basic reusable entities as the building-blocks;

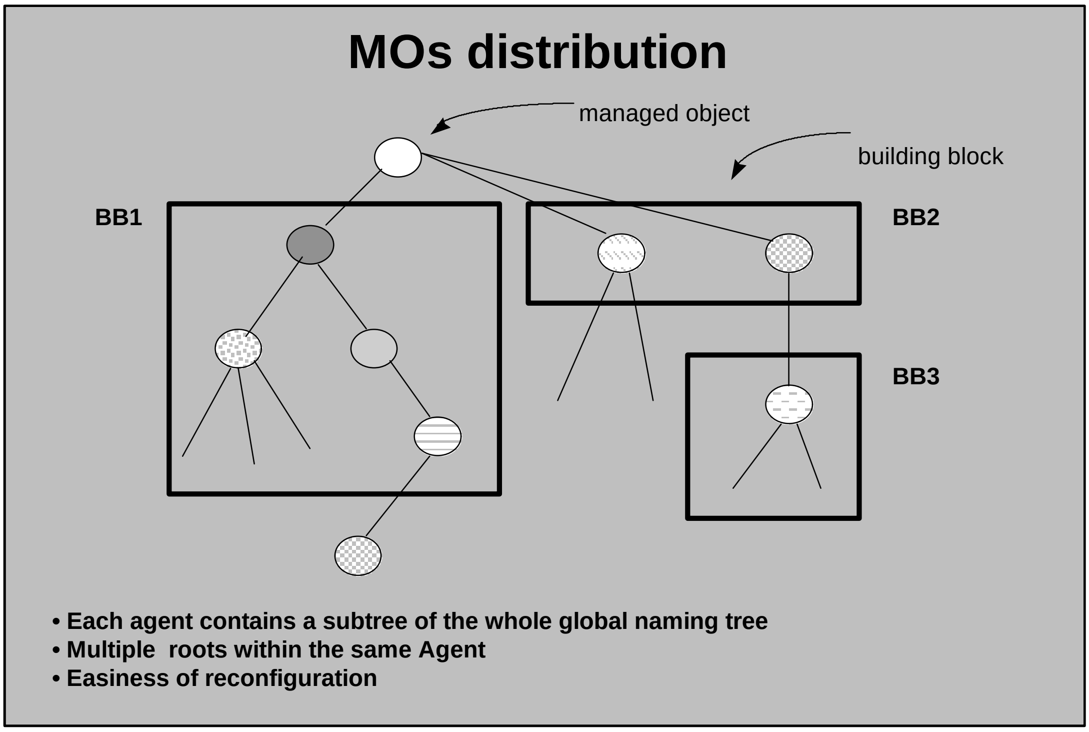
• to express a MIB in terms of an equivalent computational;

• the definition of an auxiliary set of functionalities that ease the definition and the deployment of Agents;

• to provide an effective means for the reuse, not only at the specification time but also at runtime, of architectural services (e.g. security, mapping from distinguished names to real object instances);

• by using the rich expressivness of the computational architecture, devise a mechanism for guaranteeing allocation transparency of the MOs.

It has to be noted that, though the principles and the motivations for the Generic Agent Building Block are the same for each domain, its effective internal structure depends on the underlying infrastructure.

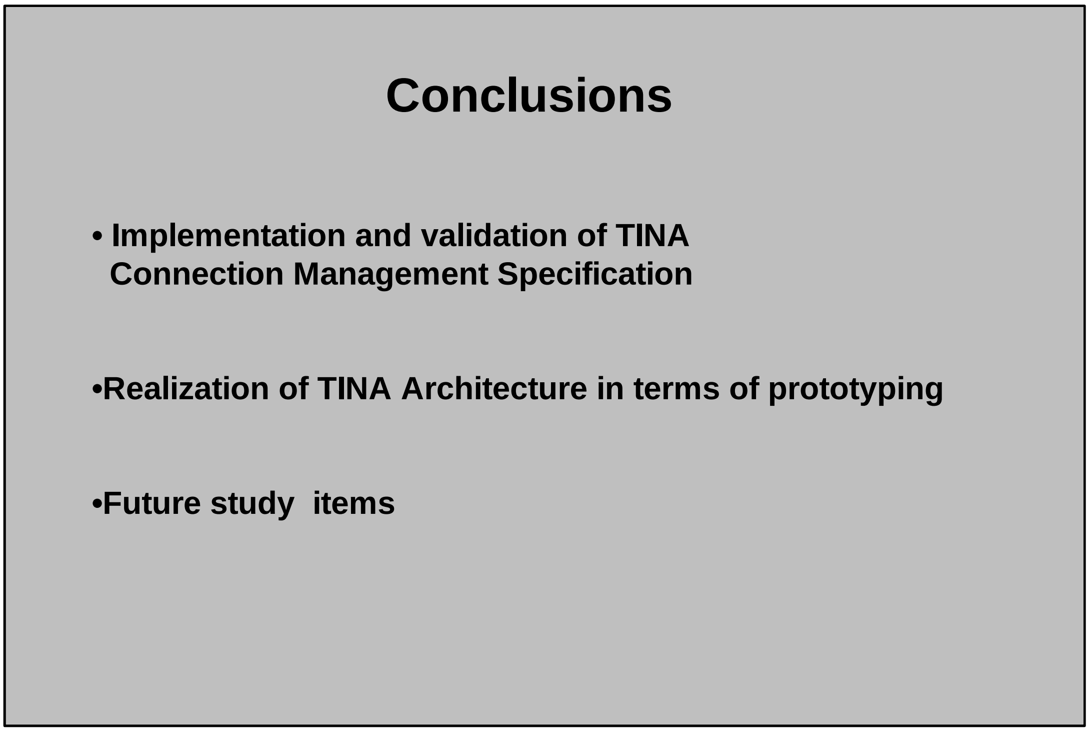


MOs distribution

- **Each agent contains a subtree of the whole global naming tree**
- **Multiple roots within the same Agent**
- **Easiness of reconfiguration**

The main characteristics of such a solution are:

- each generic Agent building-block “contains” and supports MO instances belonging to a subtree of the whole global naming tree;
- the same Agent may contain multiple roots of the naming tree;
- it is possible to change and modify the configuration of each Agent, by customizing only minor internal components and reusing the architectural services it provides.



Conclusions

- Implementation and validation of TINA Connection Management Specification
- Realization of TINA Architecture in terms of prototyping
- Future study items

Implementation and validation of TINA Connection Management specification by use of

- different types of distributed platforms, i.e., ODIN and ORB
- two distinct hardware platforms, SUN and DEC
- an “ad hoc” platforms interoperability mechanism, i.e., gateway
- a Generic Agent Building Block

Realization of TINA Architecture in terms of prototyping

- TINA-C WorldWide Demonstration in Telecom ‘95

Future study items

- Implementation and validation of other services, e.g., VoD services
- Implementation and validation of real-time platforms

The authors would like to thank the Core-Team for all suggestions provided