INTEGRATED TMN SERVICE PROVISIONING AND MANAGEMENT ENVIRONMENT

GRAHAM CHEN AND QINZHENG KONG

***Abstract*—The world wide competition and deregulation in the Telco industry has intensified the need to build a service management system, where the Telco service offering environment, service creation environment, service provisioning environment, service management environment, and service implementation (based on the network) environment are all integrated in a single architecture.**

This paper addresses the issues of building such integrated system. We propose an integrated service provisioning and management environment with a Java-based user environment, a CORBA-based distributed service management environment and a TMN-based network management environment. The paper discusses the business requirements and technical requirements for such architecture. It also reports our first phase prototype implementation of the system. It focuses on the experiences of integrating CORBA, Java and OSI technologies to achieve integrated management.

***Keywords*—Service Management, service provisioning, CORBA, TMN, Java**

***Source of Publication*—Integrated Management in a Virtual World, Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management, San Diego, California, USA, Chapman & Hall, May 1997.**

1 INTRODUCTION

The world-wide deregulation of telecommunication industry has forced the telcos to re-organise the business process and adopt the new technology to manage its network and services. As opposed to the network management, which solves the cost reduction side of the equation, the service offering and service management activities deal with the revenue generation side of the equation. The technology framework in which the telco services are offered and managed becomes the major enabler for telcos to compete in the new business environment.

The telco industry has seen the following business trend and technology progression in recent years:

- Following the deregulation and increased competition, the customer care system integrated with the process re-engineering provides the maximum competitive edge to the telco industry.
- The distinction between traditional distributed computing environment and telecommunication environment is blurred following the use of computer technology. For example, some telecommunication service has little difference from a traditional financial service.
- An integrated environment for both service offering and service management is essential to telco's success. There is a chasm between the TMN service management and network management. The challenge facing the industry is to bridge this chasm by providing a reliable, scalable and extensible computing environment upon which telco services can be offered, managed and integrated with the network.
- In the choice of computing technology, the distributed object oriented technology shows its capability to bridge this chasm.

However, the telecommunication industry still needs evidence that the distributed object technology can be integrated seamlessly with the existing TMN environment, OSS environment and the user environment. Without this integration, the technology will have very limited role to play.

The Network Management Forum (NMF) has identified that Customer, Service Provider (SP) and Network are the major players and components in the integrated service management chain [7,14]. The relationship among these players is illustrated in Figure 1:

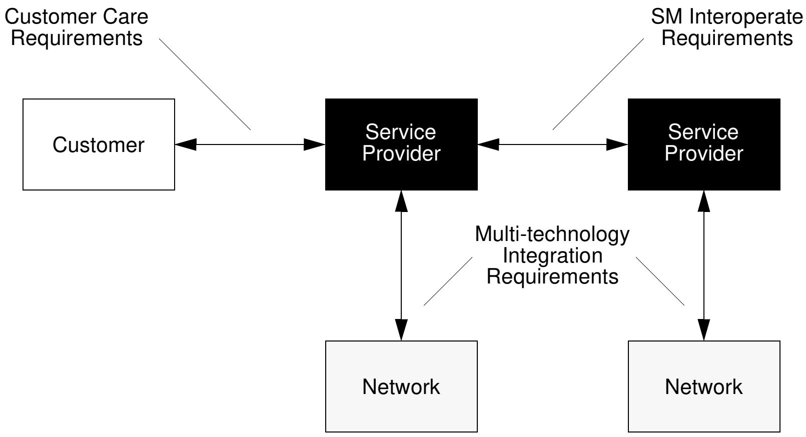

Figure 1: Service Management—Actors and Interactions

NMF is currently investigating the technology requirements, particularly from a customer's point of view, for a complete cycle of service offering and service management. The overriding requirement is the definition of an integrated environment where the telco services can be offered, controlled and managed. In the same environment, the network resources which provide the ultimate support for services can be integrated; the service providers can interact with other service providers and the customer care systems can be integrated to achieve required customer satisfaction and quality of services.

In addition to these, the advanced broadband transmission technology such as ATM has created more opportunities to extend the telecommunication services. This has put more requirements on the environment in which these new services are deployed and managed.

Figure 1 has identified three major requirements of interaction between different players in the value chain. The customer care requirements include the CTI interfaces to the management of customer's ordering requests, quality of service requests and any other access requests. It also includes the customer network management requirements to provide interaction with different service providers. The service management interoperability requirements address issues such as: support of service level agreement; support of cross SP service provisioning and management; and support of customer network management interface.

The multi-technology integration requirements are addressing very important issues of using different technology to provide telco services. These requirements include:

- the support of the integration between technologies used by service management component and network management component
- the support of protocol independent object abstraction
- the support of service designing and creation environment

This paper addresses the issues of the technology integration to provide these support.

2 SERVICE PROVISIONING

2.1 SERVICE MANAGEMENT LIFE-CYCLE SCENARIO

The relationship between different components in Figure 1 can be demonstrated by the following telecommunication service management life-cycle scenario.

Service ordering and provisioning is the basic activity in telecommunications. A simplified service ordering and provisioning process consists of:

- a customer issues a service order to a service provider,
- the service provider collects and verifies customer information and starts the service ordering process,

- service level agreement between the customer and the service provider is created or updated according to the negotiation between the customer and the service provider,
- services are installed and tested—this may involve actions at both service management layer and network management layer.

This process is illustrated in Figure 2:

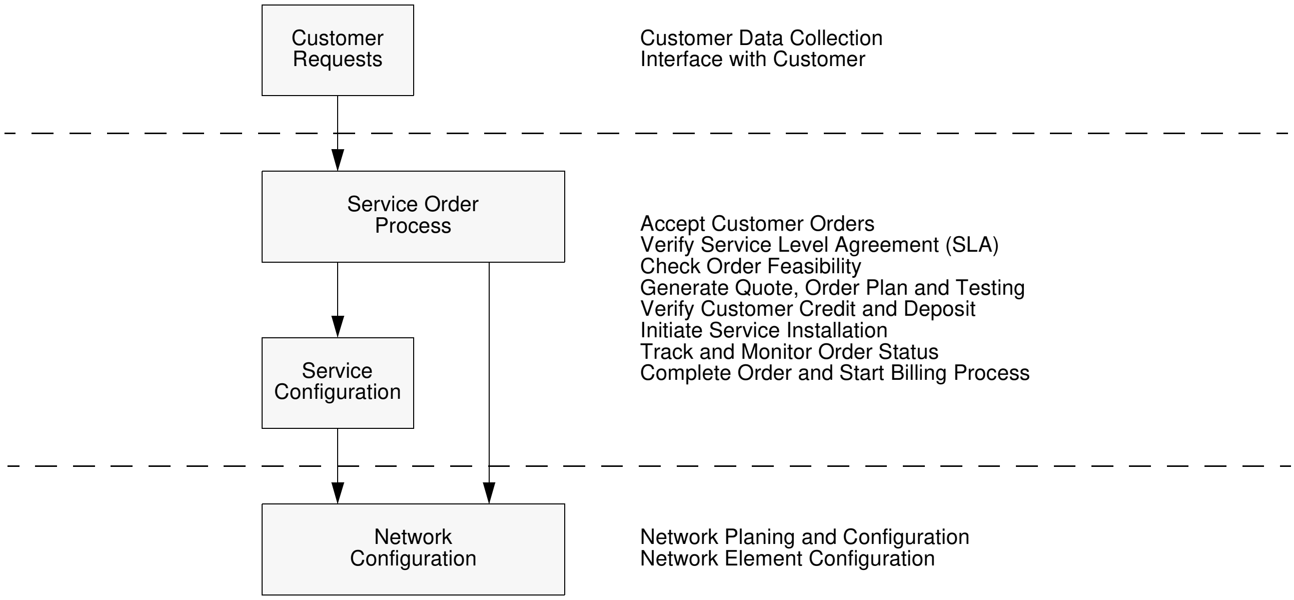

Figure 2: Service Management Life-cycle Scenario

Depending on the type of services ordered, the processes, in particular the installation and test phase of the services, may not be automated.

2.2 AUTOMATION IN SERVICE ORDERING AND PROVISIONING

Automation in service provisioning process includes the following aspects: when a customer requires a new telecommunication service, a call centre, which is responsible for handling customer service requests, is contacted. Customer care system is used to provide a quick and reliable environment for the call centre. A telephony interface or on-line interface such as Web and Java can be used for such access.

To service the request, a process control task (such as a provisioning workflow) may be started, which controls the business rules and policies for the provisioning of such a service. The integration of customer care system and process control system provides the smooth management of the ordering process. This allows the tracking, monitoring, and reporting the customer requests as well as the processing of the requests. The workflow process integration customises the existing generic workflow management system to the specific telco business environment. In the cases where the consistency and correctness is crucial to the business process, the transactional capability can be supported by the workflow engine [2].

The processes involved in a workflow instance can be defined and deployed as distributed objects. They are supported by the distributed object environment. The environment contains a set of well specified generic object implementations which customise the distributed object environment to a specific business domain. These processes require a well-integrated management interfaces so that the service offering aspect and the management aspect are integrated in a single object environment which presents the TMN service management view.

Each service request can be mapped to a set of network requests and implemented on a potion of network and its elements. Network requests are sent to agents involved in the implementation of the service in TMN network management environment. This functionality is well supported by the network management environment.

The automated service ordering and provisioning scenario described in this section requires a great amount of integration between different technologies which offer different functionality. In the next sections, we focus on the integration aspects of technologies.

2.3 TECHNOLOGY SUPPORT

The technology support for service ordering, provisioning and process automation is illustrated below:

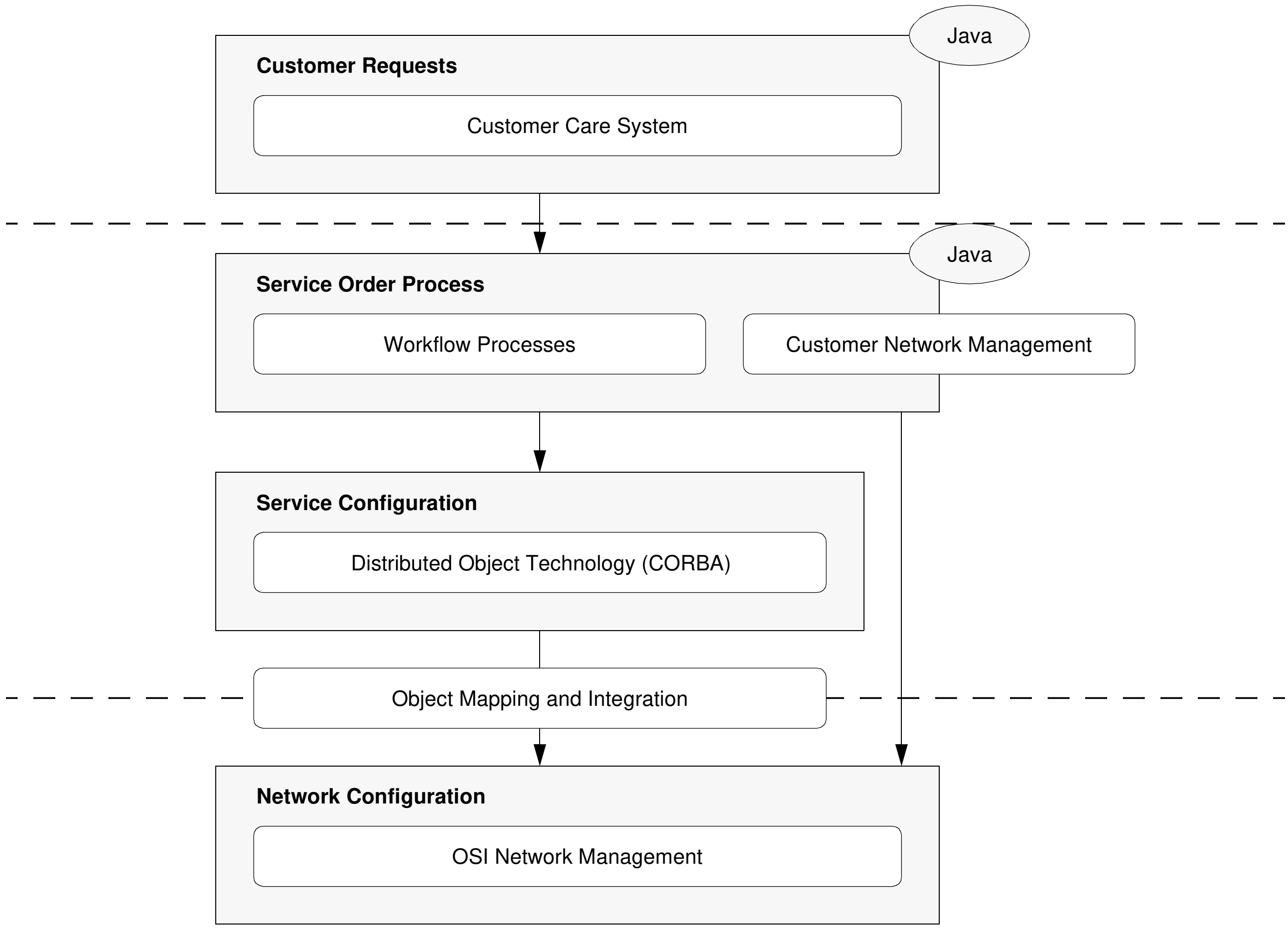

Figure 3: Automated Service Ordering and Provisioning Process

As indicated in Section 2.2, technology required in supporting automated service ordering and provisioning process may include:

- Customer care system—the front-end contact point to customers,
- Customer Network Management—the ability to manage inter-SP service configuration activities,
- Java—a GUI architecture providing standard interface to operators and customers,
- Workflow system—the back-end process control and monitoring system,
- Distributed technology—the distributed platform, such as CORBA, to support distributed service management and to interface with the network environment,
- Object mapping and integration—tools and environment providing static and automatic mapping from service management environment to network elements,
- Network management—the basic technology, such as OSI NM, used to manage telecommunication management networks.

There exist many different technology choices to support the automated service ordering and provisioning process. In the next two sections, we present an architecture and discuss the experience of implementing such architecture.

3 ARCHITECTURE

In this section, we analyse the requirements for supporting the ordering and provisioning process in more detail and propose an architecture based on the integration of different technologies. The aim is to define an architecture that will meet the business requirements, survive the impact of technology evolution and migration and such architecture can be easily deployed to support telco's business processes.

3.1 BUSINESS REQUIREMENTS

Although the trend towards adopting re-engineering using new distributed network and service management technology is very strong, across telecommunication industry, there are well-founded business reasons why people still have reservations of moving too quickly. The following concerns are among the first:

- availability and reliability—the ability to support telco business in a 7 by 24 way. This is overriding requirement that the industry will not use any technology that cannot provide reliable service.
- integration with legacy systems—the strategy to protect the huge investment the industry has made. These systems include all existing network management and element management applications, communication stacks and protocols, and OSS applications which support telcos core business. A clear cut with the past, no matter how desirable it is, is not an option to the telco industry.
- client GUI platform—the strategy to provide flexible and light-weight client GUI technology, which is well integrated with the distributed object paradigm and does not require a huge overhead of traditional client platform. With the popularity of internet and Java technology, telcos need a cheap and flexible way to deploy services to customers quickly and allow easy customer access.

It is essential that the architecture for integrated service provisioning and management satisfy all these requirements.

3.2 THREE LAYERED ARCHITECTURE

The service provisioning and management architecture proposed here has three basic functional layers as illustrated in Figure 4:

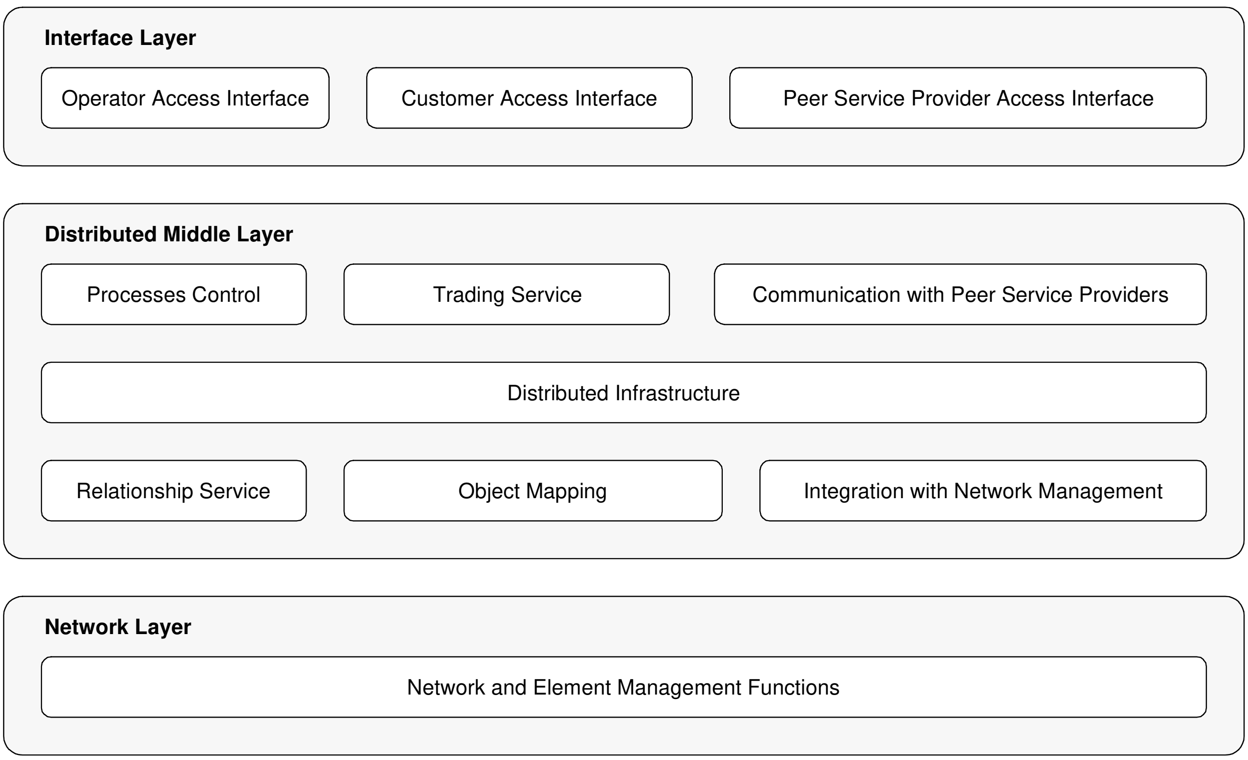

Figure 4: Management Architecture

3.2.1 INTERFACE LAYER

The interface layer supports different type of access to the system. It includes Java-based GUI for both operators and customers. The interface also includes APIs for peer service provider access to archive integrated peer-to-peer service provisioning.

This layer supports the separation of object presentation from their semantics. The semantics of the objects should be encapsulated behind an interface. Access can be permitted to the interface without compromising the semantics.

Java is a platform independent programming language. By using Java-based GUI, the presentation rules are encapsulated in Java objects, while the semantics are encapsulated in other system objects. In this architecture, they are encapsulated in CORBA objects supported by the distributed middle layer. This allows new applications to be developed in a manner that they present the information without interference with the core functionality. The development time for new applications can be reduced and the integrity of core functionality can be enhanced.

3.2.2 DISTRIBUTED MIDDLE LAYER

Distributed middle layer is the kernel part of the architecture. It provides functional support for:

- service management components such as TMN management functions: fault, configuration, accounting, performance and security (FCAPS) [3]
- provisioning process control and automation components
- distribution of application components by supporting distributed infrastructure and services

It also serves as the integration bus between different functional layers, different components and also between different service providers.

CORBA Object Request Broker (ORB) and CORBA services [12,13] are used as the distributed infrastructure. Among many services, the relationship service and the trading service are two important services in providing integrated service ordering and provisioning system. The relationship service offers functionality to model the component structure and topology in a provisioning process and the support for the process automation. The trading service helps the process control to be achieved in a distributed operating environment.

One major responsibility of this layer is to support interaction with the interface layer and the network and element management layer. The Java-CORBA gateway becomes the interaction point between the Java-based interface layer and CORBA-based middle layer. The interaction between the middle layer and the underlying network and element management layer requires the integration between CORBA and network management technology. The components such as object mapping, integration technology and relationship service all play important roles here. Section 4.1 discusses these integration issues in detail and Section 5.2 discusses our experiences of implementing the integrations.

3.2.3 NETWORK AND ELEMENT MANAGEMENT LAYER

The network and element management layer consists of a management platform and a set of network management functions. This environment provides the support for FCAPS functions at network and element level to deliver services. It interacts with the service management environment to ensure that the service requests are mapped to network operations and the quality of service requirements are satisfied.

OSI-based TMN network management and element management is the main component of this layer.

3.3 TECHNICAL FEATURES

The architecture for integrated service management supports the following features:

- Scalability—the scale of the system can be extended when it is required by the business expansion. Distributed systems offer an opportunity to manage systems that are currently too large for a single host to manage by using multiple smaller hosts. One issue in the scalability is to allow hosts to be added into the system without affecting the functionality of the existing system.
- Extensibility—the functionality of the system can also be extended to meet customers' new requirements. The building block concept in distributed systems allows functional blocks to be added when required.

- Reliability—continuous support for service is the major requirement of telecommunication service providers. The architecture therefore must allow construction of systems that can recover from hardware and software failure. Special effort is required to support different level of reliability and fault tolerance in a distributed system. Section 5.3 discusses this feature.

3.4 DEPLOYMENT

The system can be deployed in a wide range of configurations. The primary partitioning of functionality between machines is the user interface and service machine. Typically a PC-based system would provide the user interfaces and a Unix-based system would provide the services and platform support. This configuration offers the advantage of using high performance and reliable systems for the servers that enforce the semantics of the system, while using lower cost systems for the user interfaces. The PC front end also provides an interface that is typically more advanced than Unix-based systems.

Figure 5 depicts a typical deployment scenario of the architecture.

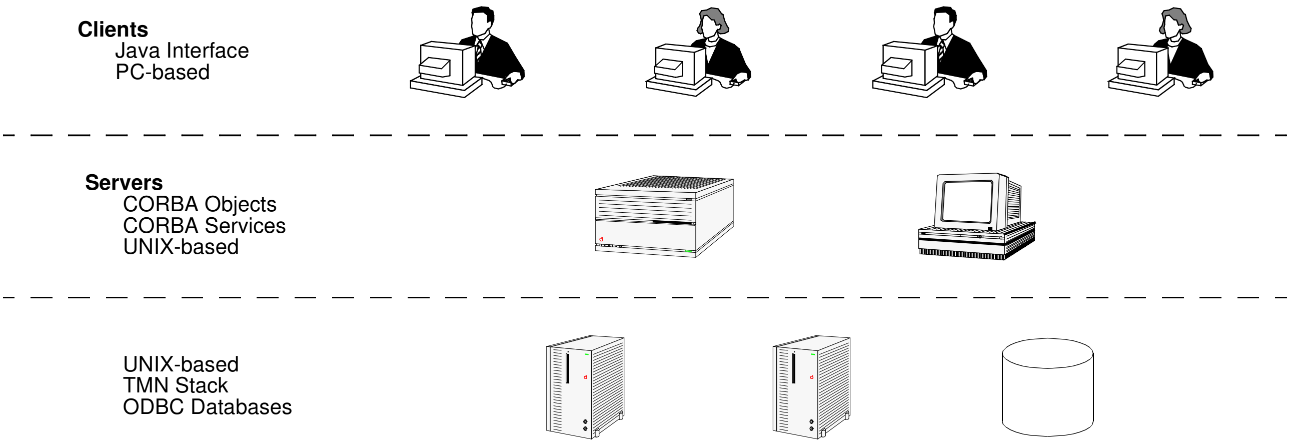

Figure 5: Deployment Architecture

The Java user interface provides a mechanism for rapid development of user interfaces that are portable across multiple platforms.

The CORBA infrastructure provides a basis for distribution and extensibility.

Java and CORBA are both available on a wide range of platforms (unix workstations, PCs, Macs etc.) and the platforms are able to interwork in a heterogeneous environment. The architecture may be implemented on different platforms depending of the performance required for a particular application. For example a low performance application may be implemented on a PC-based system, while a high performance high availability application may be implemented on a Unix-based system.

4 TECHNOLOGY INTEGRATION

The major challenge of providing an integrated telecommunication service management is the integration of existing technologies. The requirements for integration are caused by:

- different technologies are used to provide different functions,
- different technologies are used by different service providers,
- different technologies are used in today's systems which are supporting Telcos business processes. Such examples include the network management systems and OSS systems.

The architecture presented in the previous section requires the seamless integration of different technologies. These include:

- the integration between customer care system and the process control system,
- the integration between GUI technology, such as Java to back-end distributed technology,
- the integration of process control system to distributed environment,
- the integration between CORBA-based service management components and OSI-based TMN network management components.

In this section, we focus our discussion on the integration between CORBA and Java to provide a user environment and the integration between CORBA and TMN to provide a service management environment.

4.1 CORBA/TMN INTEGRATION

The integration between CORBA distributed object technology and the OSI network management technology provides the basis for the integration between telecommunication service management and network management layers [1]. Different object models are used in these two different technologies. Thus, the initial focus of the technology integration is the mappings between object models.

Three different integration approaches are used in our study. These are, *abstract mapping*, *close mapping* and *TMN (GDMO) factory*.

Using the abstract mapping approach, a CORBA object is used to represent a functionality which is usually modelled by a set of GDMO objects or an application entity. Such object presents a CORBA IDL interface and its implementation is based on CMIS operations and uses CMIP protocol. Since the CORBA object represents high level application semantics, the number of these objects, compared with that of the GDMO objects, is relatively small. Also, the GDMO containment structure and associated scope and filter operations can be embedded within the object implementation. This will reduce the complexity of modelling containment relationship within CORBA. These integration objects serve as the base level building blocks and more application semantics level CORBA objects can be defined.

The close mapping approach offers a more direct static mapping between two object models. A CORBA object is used to represent a GDMO object and a one-to-one translation between the attributes and operations can be achieved. Based on these CORBA objects (proxies) more application semantics level CORBA objects can be built. As opposed to the abstract approach, this approach needs to model OSI containment structure and scope and filter operations in CORBA which may present a performance problem.

TMN factory approach allows dynamic creation of CORBA proxies of GDMO objects at run time through a proxy service. This approach allows application to selectively create CORBA representation of GDMO objects and reduces the overhead of promoting large number of GDMO objects to CORBA space.

The experiences we gained from building a prototype of the system suggest a combination of different approaches depending on the way the services are modelled. Section 5.2 details the design of these integration approaches and our experiences of using these approaches.

4.2 CORBA/JAVA INTEGRATION

Java and CORBA complement each other extremely well. Java can be used to develop small machine independent GUIs that may be downloaded across the web to the client's machine from a central server. The GUI can then use a CORBA implementation to make remote method calls on objects elsewhere on the network. This maps well to the real world organisation of distributed data.

This configuration allows the most suitable machines to be used for each component of an application. For example, the user may have a PC which is ideal for the GUI but is not powerful enough to execute some complex computations. Hence the calculations can be implemented as a CORBA object on a high performance resource and remotely invoked by the PC client. Other examples include the situation where the required information can not be downloaded to the client such as a large database or network management system.

A further major advantage of this configuration is that it opens up the application to any machine where the Java client can run. No longer does the user have to be on a certain type of machine to be able to run the application.

However, the Java GUI components need to be well integrated with the CORBA environment in order to perform the GUI functionality. Figure 6 depicts the integration architecture:

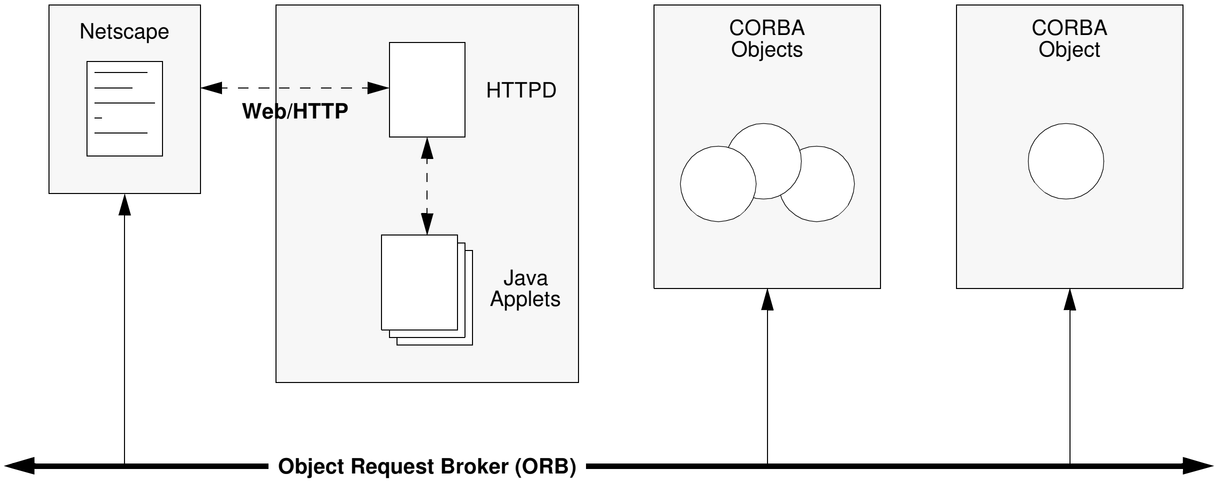

Figure 6: Java/CORBA Integration Architecture

This architecture will enable true distributed computing with Java as the downloadable GUI front end. A Java object can also have a CORBA object interface so that remote objects can make computation requests on it.

5 IMPLEMENTATION

5.1 SERVICE MANAGEMENT ENVIRONMENT

We have implemented a prototype of service management environment based on CORBA integrated with an OSI-based network management platform. In the first phase implementation we used Orbix 1.3 and

OpenView DM 3.3. A Video-on-Demand (VOD) service is used as an example to demonstrate various concepts discussed in this paper. Figure 7 depicts the architecture of this environment:

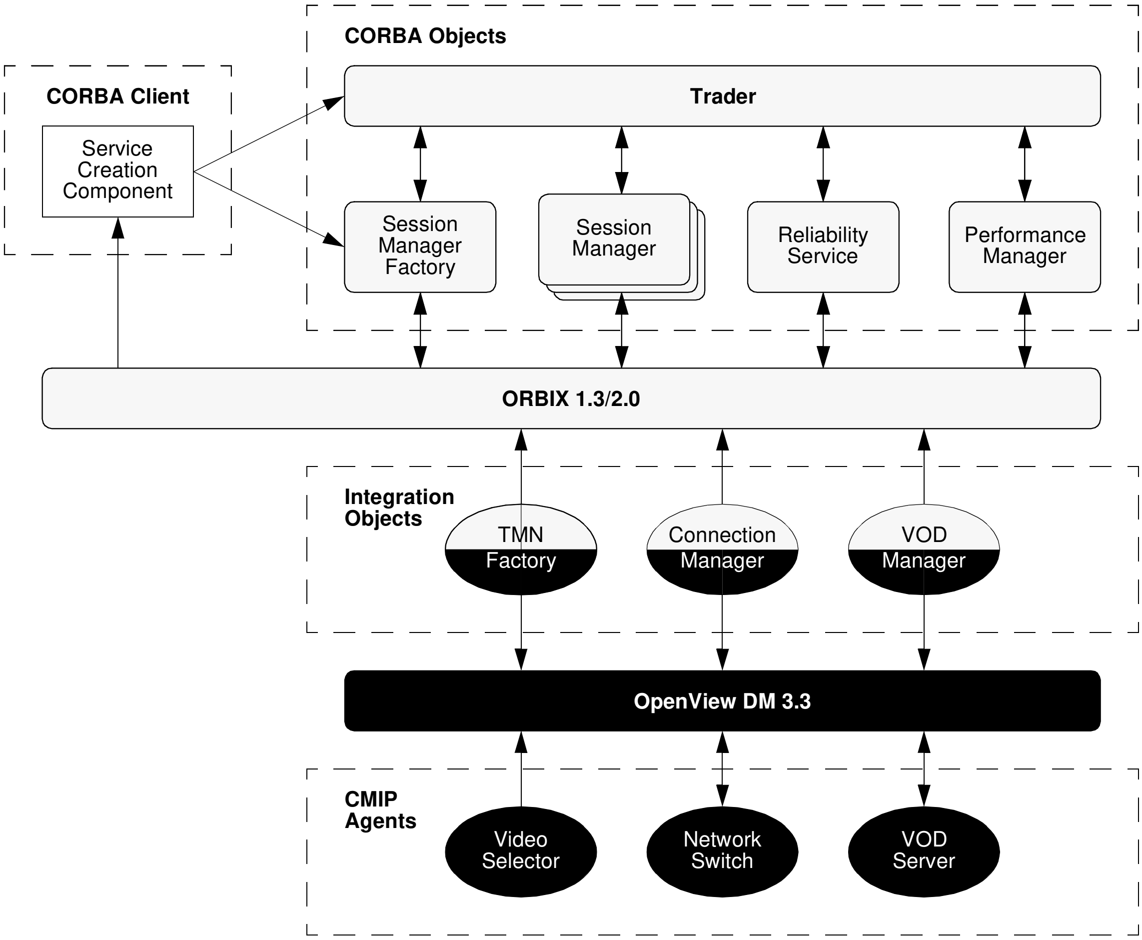

Figure 7: Service Management Environment (VOD Demonstrator)

The demonstrator has three sections—the TMN section, the CORBA section, and the integration section (CORBA and TMN). The TMN section consists of the CMIP agents for the Video Selector, the Network Switches and the VOD servers. The CORBA section has a number of CORBA objects to implement the VOD Service.

The environment is supported by a CORBA-based platform integrated with a CMIP-based platform. The environment offers the existing CORBA distributed infrastructure and distributed services. A set of generic services to support service management and service offering environment is added to this platform.

In the above example, a service is triggered by a CMIP event modelling the video selector at the customer premises to select a video. This event is delivered to the CORBA service creation component which create session manager to provide the service. A trader is used for location transparency as it is vital to achieve scalability and reliability.

5.2 CORBA/TMN INTEGRATION

The CORBA objects `TMN Factory`, `Connection Manager` and `VOD Manager` are objects that integrate with CMIP environment. These objects have defined CORBA IDL interfaces and the implementation of these objects performs CMIP operations on the underline CMIP GDMO objects which model the elements and transmission network. The CMIP environment is implemented using HP OpenView DM.

The CORBA object `Connection Manager` is an abstract representation of the connection in the transmission network. This object does not map into an individual CMIP object but instead it models a connection which is constructed by CMIP operations. This object is a low level building block supporting application semantics. The implementation results indicate that this level of mapping offers the best overall system performance due to its higher level of abstraction [1] and is always preferred integration approach.

The `VOD Manager` is a CORBA object which models the VOD server in the CMIP space. This object represents the close mapping between CORBA and GDMO. With this object the CORBA environment can directly access CMIP objects to control and coordinate network elements.

The `TMN Factory` is a generic CORBA object which dynamically creates CORBA objects acting as proxies for CMIP objects. It provides a generic mechanism to allows access to TMN functionality that has not been encapsulated in an abstract object. It achieves this by providing a fixed interface that can be used to perform the standard CMIP operations on CMIP objects. An IDL object is created on demand and bound to a CMIP object. Remote clients can access the CMIP object via the IDL proxy. The TMN factory allows access to any CMIP object. Clients only need access to the CORBA infrastructure rather than the CMIP stack.

The TMN Factory provides an on-demand 1-to-1 mapping of CMIP objects into the CORBA domain. Due to the 1-to-1 mapping the service is not scalable. It is intended to provide access to rarely used CMIP objects. A typical interaction is depicted in Figure 8:

```
1. Find TMN Factory in Trader: Trader interfaces
2. Bind CORBA object to TMN Factory: bind()
3. Invoke operations on CORBA object: get(), set(), etc.
4. Process events: get_events(), or
     Set event forwarding via OpenView DM’s event service: event_forward()
5. Destroy proxy: destroy()
```

Figure 8: TMN Factory Interaction

5.3 SCALABILITY AND RELIABILITY

Customers require systems that run continuously. The platform must therefore allow construction of systems that can recover from hardware failure. Some form of replication is required to allow another host to provide the services of a failed host.

Distributed systems offer an opportunity to manage systems that are currently too large for a single host to manage. Currently extremely large management hosts are required. Distribution allows multiple smaller hosts to be used in place of the single large host.

The demonstration platform offers scalability inherited from CORBA architecture. By using distributed trader, the service creation process can decide on which host the new service and management objects should run. The system is tested to handle 100,000 objects.

CORBA architecture does not have built-in reliability. In this demonstrator, we designed a generic reliability component which detects the fault in the management system and re-create the management objects by using trader and *virtual data store* (VDS) which is a component based on CORBA’s persistent service. At any time during the running of the system, a host running the service management system can be un-plugged from the system. The reliability service detects the failure of communication with the objects on that host, recovers the state of the objects from the VDS and re-initiates the service from a different host.

5.4 JAVA PERFORMANCE MANAGER

In a system where there could be a large number of users requiring the application services, there are significant problems in configuration management. For example, what happens when a new version is released. By adopting the Java and CORBA approach, the application client can be stored centrally on a few servers and then downloaded on request to the user.

To demonstrate the advantages of using Java as the user environment, a Java applet which monitors the performance of the VOD system was implemented. The applet is a performance monitor based on a simple bar chart display. Performance data from the CORBA and TMN sections of the application are gathered together into a single CORBA object. The Java applet then accesses the CORBA object in order to obtain the performance data and present it to the user. The bar chart consists of a count and a moving bar colour-coded to the current value. The following Figure 9 shows the interaction:

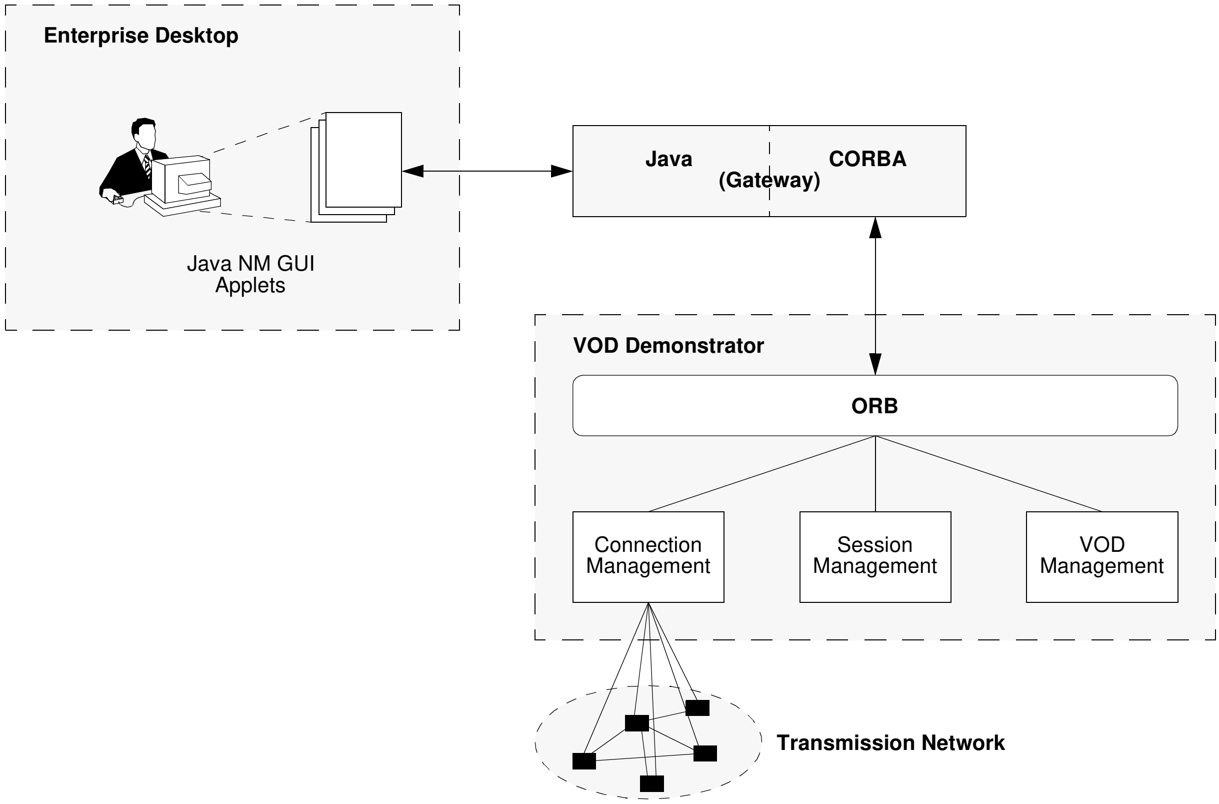

Figure 9: JAVA CORBA Integration

5.5 SERVICE PROVISIONING ENVIRONMENT

The work is continuing to construct a service provisioning environment. The initial effort is to have a GUI-based environment as the single access point for operators to deploy services required by customers to a set of network elements. The major components and activities in this environment include:

- A *Java service order applet* allows operator to enter customer information to a service order form. The ATM service provisioning is used as a demonstration scenario. Information required in an order form include customer name and address, bandwidth required, type of data to be transferred, service type, whether it is bandwidth on demand (pay as you use), etc. Service request is submitted by the operator on behalf of customers. The customer request is stored as the service agreement and is linked to the customer profile.
- A *Java resource monitor applet* allows network operators to view the status of network resource and to find the available resource to meet customer request. The Java/CORBA gateway supports the submission of such search request and returns results to the operator.
- A *Java configuration applet* provides the access point for the service deployment. It includes allocating the available network resource for the service; configuring network resources; associating the service with the set of network resources; testing the service; and finally, activating the billing system. A separate GUI may be used as the network manager's control console to display the connection between end points, or to allow network manager to re-configure the system.

  Java/CORBA gateway allows service configuration request to be performed on service objects in the CORBA environment. The CORBA/TMN integration gateway allows the service requests to be mapped into a set of network configuration requests and implemented by network elements such as switches.

The interaction among system components in the service provisioning environment is illustrated in Figure 10:

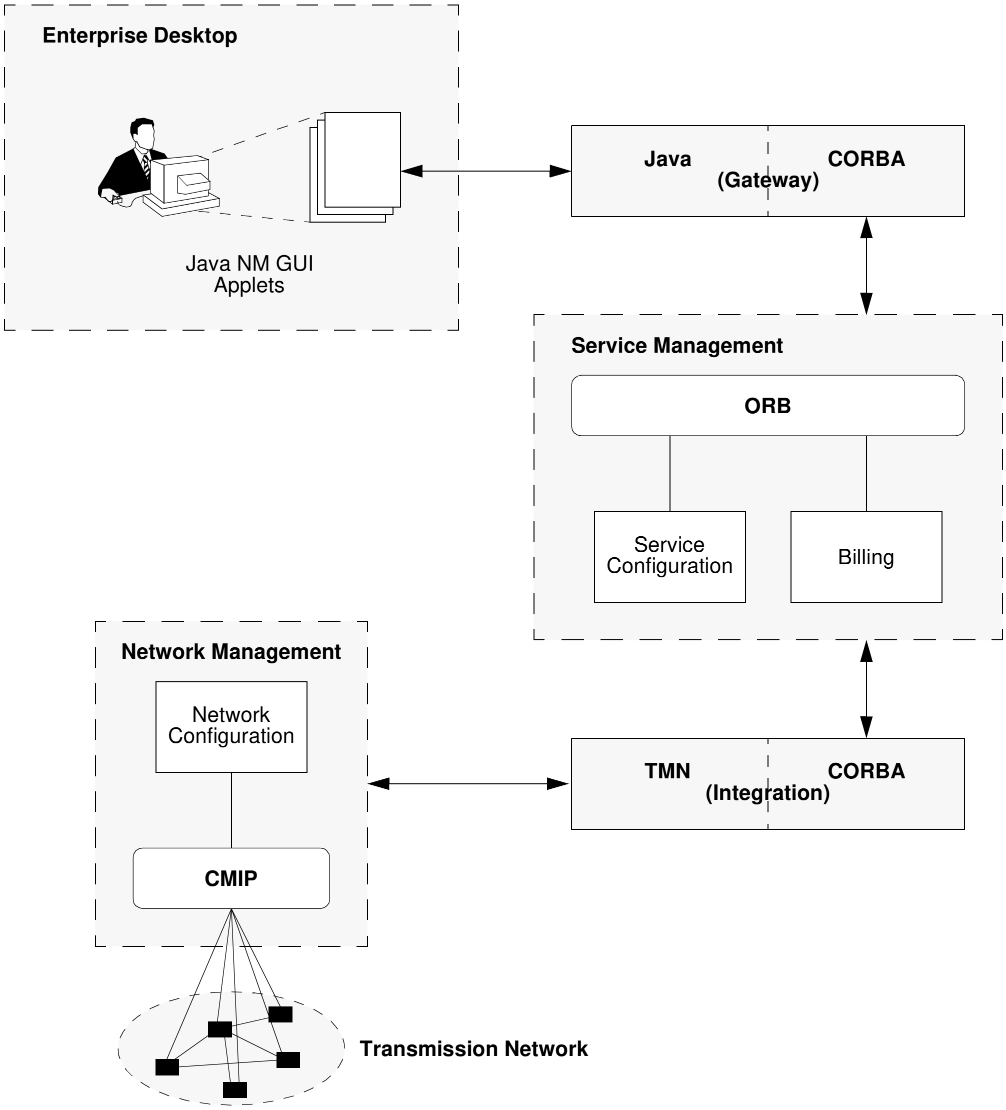

Figure 10: Service Provisioning Interaction

The components are being implemented as extended building blocks to the existing system to form an integrated demonstration.

6 SUMMARY

The paper has demonstrated an integrated service offering and management environment based on the integration of CORBA, Java and TMN network management technology. The prototype we built demonstrated that the architecture offers a very viable solution to telecommunication business processes.

The prototype offers a highly scalable, reliable and extensible environment to support telco business processes. It demonstrates the integration approaches between CORBA, Java and TMN environments are sound and performing well. The resulting system provided very useful platform to evaluate issues relating to TMN service management.

The work is continuing to enhance the system by adding the following functionality:

- implementation of other business processes, such as trouble ticket, fault management, etc.
- the integration of customer network management system to support inter-service provider interactions, such as service provisioning across multiple SPs, management of service level agreements, etc.
- the use of workflow technology for process control and process automation.

ACKNOWLEDGEMENTS

The authors would like to thank the many people who have contributed to this paper. In particular, Michael Neville and Mike Sharrott.

REFERENCES

[1] Q. Kong and G. Chen, Integrating CORBA and TMN environments, *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, Kyoto, Japan, April 1996.

[2] Q. Kong and G. Chen, Transactional workflow in telecommunication service management, *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, Kyoto, Japan, April 1996.

[3] ITU-T Recommendations M.3010 — Principles for a telecommunication management network.

[4] W. Widi, *Standardisation of Telecommunication Management Networks*, Ericsson Review, No. 1, 1988

[5] W. J. Barr, T. Boyd, and Y. Inoue, The TINA initiative, *IEEE Communication Magazine*, March 1993.

[6] Q. Kong, G. Chen, and G. Holliman, Telecommunication service management, *Data Communications OpenView Advisor*, Volume 1, No. 8, August 1995.

[7] Network Management Forum Smart Ordering Team, Ordering white paper, issue 1.1, January 1996.

[8] A. Wittmann, T. Magedanz, and T. Eckardt, Management services design: from TMN interface specifications to ODP computational objects, *Open Distributed Processing—Experiences with distributed environments*, K. Raymond and L. Armstrong (eds.), Chapman & Hall, 1994.

[9] E. K. Adams and K. J. Willetts, *The lean communications provider—Surviving the shakeout through service management excellence*, McGraw-Hill, 1996.

[10] J. P. Chester and K. R. Dickson, Standards for integrated services and networks, *Integrated Network Management IV*, A. S. Sethi, Y. Raynaud and F. Faure-Vincent (eds.), Chapman & Hall, 1995.

[11] M. Flauw and P. Jardin, Designing a distributed management framework—An implementer's perspective, *Integrated Network Management IV*, A. S. Sethi, Y. Raynaud and F. Faure-Vincent (eds.), Chapman & Hall, 1995.

[12] Object Management Group, The Common Object Request Broker: Architecture and Specification, Revision 2.0, July 1995.

[13] Object Management Group, CORBA Services: Common Object Services Specification, Revised edition, OMG document number 95-3-31, 1995.

[14] Network Management Forum, OMNIPOINT V3 Specification, 1996.