THE BUSINESS PROCESS AND OBJECT MODELLING FOR SERVICE ORDERING

GRAHAM CHEN AND QINZHENG KONG

Abstract—**This paper examines two major issues in the TMN service management arena: the specification of service management activities and processes; and the interactions between different business processes and different service providers in order to provision and manage telecommunication services.**

Network Management Forum has defined a set of basic business processes dealing with five major TMN service management functions (FCAPS). In addition to these, a service ordering process is defined as an important interaction with the end customers requiring services. The service order process provides an integration point for all major TMN functions and business processes.

This paper presents a detailed analysis for the service ordering process. It covers the process definition, object modelling and process interaction and integration. It also discusses the technical issues related to the implementation of the ordering process and its object model.

Keywords—**Service Management, Service Ordering, Object Modelling**

Source of Publication—**Proceedings of the 8th International Workshop on Distributed Systems: Operation and Management (DSOM 1997), Sydney, Australia, Oct. 1997, ISBN 1 86365 344 9.**

1 INTRODUCTION

Due to the deregulation and liberalisation in the telecommunication industry, and the increasing demand of customers for global services, telecommunication service providers are less likely to be able to satisfy their customers requirements using only their own networks and resources. Therefore, more and more service providers start to rely on other service providers for providing complete service solutions in order to meet their customers' requirements. In many cases, how to increase the satisfaction of customers is no longer within the total control of a single service provider. This new trend in telecommunication industry requires inter-working between different service providers.

Apart from the demand for global services, customers also require fast service provisioning, better service management and better quality of services. In other words, the inter-working at different levels between multiple service providers is required. However, the current telecommunication service provisioning and management environment heavily relies on manual interactions between service providers. It has the following problems:

- It is hard to have consistent service delivery and quality of services, such as solving a customer's problem within stated time.
- It lacks the visibility and control of the partial services provided by other service providers. Hence, it is difficult to meet the end customer's requirement and provide a complete service view to end customers.
- In order to provide a service to an end-customer the type and the amount of information required to be handled and managed is complex and huge.

All the above problems require better coordination between different service providers and demand automatic integrated service provisioning and service management systems.

Network Management Forum (NMF) has embarked on a SMART (Service Management Automation and Re-engineering Team) program to define the business processes [1,5] for the most common telecommunication business activities and the integration of these processes. One important business process considered in NMF is the service ordering process [3,6,7,8].

This paper explores the major issues in the integrated service management environment and uses the business process for service ordering between different service providers as an example to examine these key integration issues. Section 2 discusses the major issues common to all the integrated service provisioning and management. It also addresses different types of integration required in providing a complete solution. Section 3 uses the service ordering as an example to explore the information model required to support interactions between different service providers. It focuses on the object model at the interface level rather than at the internal ordering process level. Section 4 discusses some detailed technical issues which are crucial to the deployment of the integrated service provisioning and management environment.

2 INTEGRATED SERVICE MANAGEMENT

2.1 MAJOR PLAYERS IN SERVICE MANAGEMENT

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

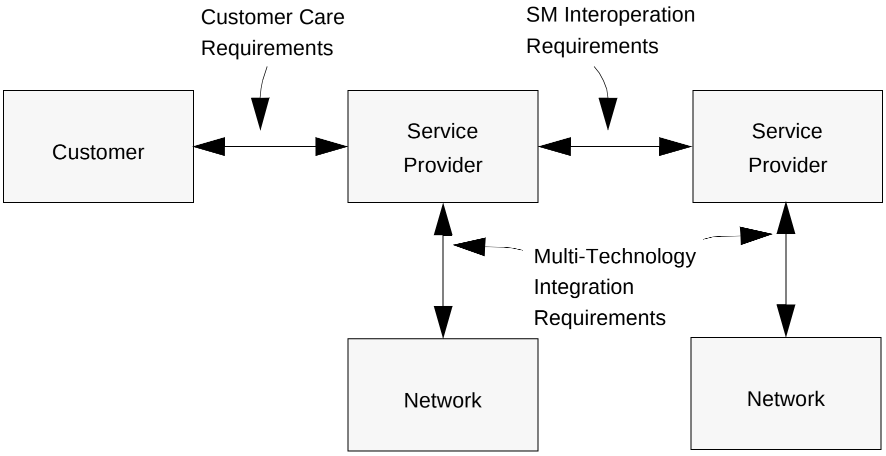

Figure 1: Service Management—Actors and Interactions

NMF business processes focus on the interaction and integration between different service providers.

2.2 MULTIPLE SERVICE PROVIDERS

When a customer requires a new service, the customer will contact a service provider. In many cases, the provision of the required service will involve more than one service provider, for example, local telephone carriers and long distance carriers. From the customer's point of view, it is convenient that the customer deals with only one service provider while the service provider will order from the other service providers if required.

The following list of terms are often used in the service ordering, provisioning and management processes.

- Customer—a customer is the user of a service provided by a service provider. In some cases a service provider may also play a customer role when it requires services from another service provider.
- Service Provider (SP)—an organisation which provides services for others.
- Major Service Provider (MSP)—This is a special instance of the SP used to distinguish the SPs who provide services directly to an end-customer.
- Subcontracted Service Provider (SSP)—a subcontracted service provider is a special instance of SP who, instead of providing a service directly to an end-customer, provides services to other SPs.

If a service provider is directly contacted by the customer, it is then regarded as a main service provider (MSP) and is the unique customer contact point with regard to the service required. Other service providers are subcontracted service providers (SSP). It is also possible that some of the subcontract service providers (SSPs) may require services from other service providers.

This relationship can be illustrated in the Figure 2, in which SP-A is the Major Service Provider, while all the others are Sub-contracted Service Providers (SSPs):

2.3 ROLE CONCEPT

The concepts of the Major Service Provider (MSP) and Sub-contracted Service Provider (SSP) are only relative to a specific service ordering and provision. An MSP in one transaction may become an SSP in another transaction. It is possible that another customer may select SP-B as its main contact point and SP-B requires services from SP-A. In this case, SP-B is the MSP and SP-A is the SSP with regard to the service required by the second customer.

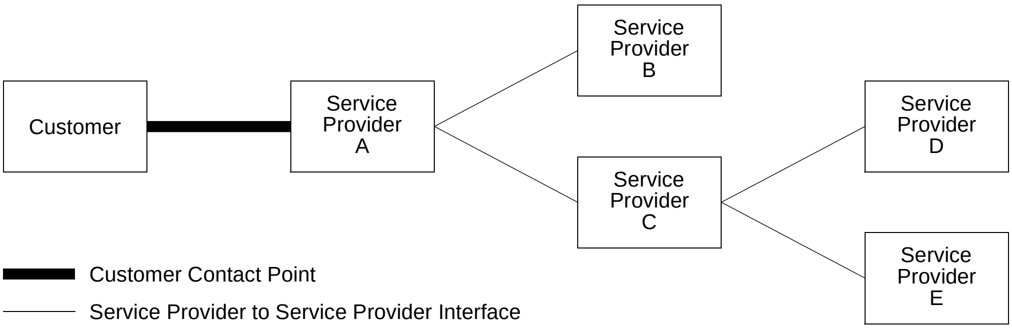

Figure 2: Multiple Service Providers in Service Provisioning

In a single service request, a service provider may also play different roles. For example, SP-A plays a service provider's role to its customer but it plays a customer role for the services provided by SP-B and SP-C. Same applies to SP-C, it plays service provider role for the services provided to SP-A but at the same time it also orders services from SP-D and SP-E. From SP-A's point of view SP-C is the Major Service Provider while SP-D and SP-E are Sub-contracted Service Providers.

Hence, the concept of role is introduced in the process model. A service provider can play different roles with respect to different services. If a service provider requires a service from another service provider, it plays a customer role with respect to the services it orders. If its services are required by another service provider, then it plays a service provider role.

This concept is very similar to the concept of Management Entity (ME) in OSI. In OSI, an ME is defined to be an entity responsible for certain management functions. It may act as a Manager to issue management requests to other MEs (acting as Agents), and at the same time it may also act as an Agent to receive management requests from other MEs (acting as Managers).

No matter which role a service provider plays in a service provisioning, the basic functionality is the same to all the service providers.

2.4 MAJOR ISSUES IN SERVICE MANAGEMENT

Service management is a complex management application. It covers many different management functions and introduces many interaction points between its functional components. An example of a set of service management processes across two service providers can be illustrated in Figure 3.

The issues related to the management functions such as trouble ticketing, fault management, billing, etc. are important to the service management. However, a more important issue is how to provide an integrated solution to the service management. There are two major aspects of the integration issue:

- Integrations between different processes
  Each business process offers its own functionality. But in many cases, one process requires information from other processes, or requires co-ordination of other processes, in order to carry out its own functionality. For example, when a customer reports a service problem, the fault management system is required to provide information in order to identify the possible cause of the fault; the configuration management system may be involved in changing the service configuration; the performance management system is consulted to check whether the fault affects the quality of service agreed between the service provider and the customer; and finally, the consequence of the fault may affect the service charge to the customer.
- Inter-working between different service providers
  In order to provide different services, a service provider often requires to order parts of services from other service providers. The degree of automation of the interworking processes between different service providers greatly affects the efficiency of the service provisioning and the quality of the service.

In the remaining sections, the Service Ordering process between different Service Providers (also referred to as the SP-to-SP Service Ordering) is used as an example to focus on the integration issue between different service providers.

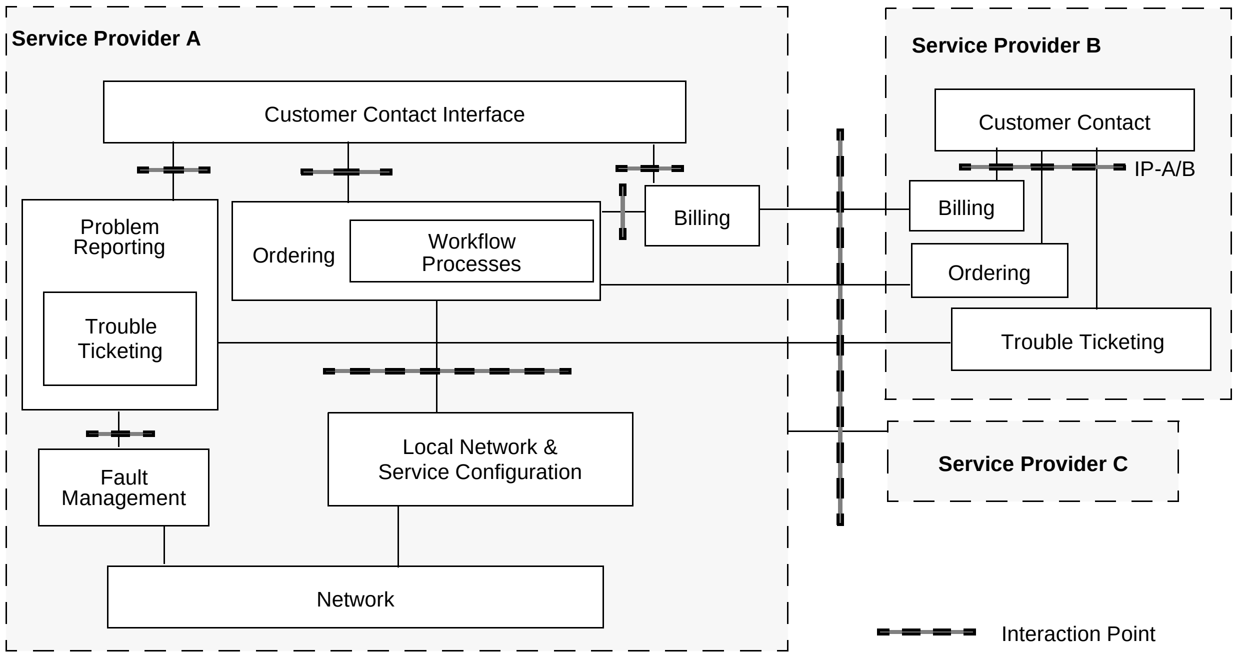

Figure 3: A Scenario of Service Process Interactions

3 SP-TO-SP SERVICE ORDERING

SP-to-SP service ordering is concerned with the ordering of services among telco service providers. In other words, SP-to-SP service ordering is used by an MSP to order services from SSPs. In all service ordering activities, the MSP is a customer of the SSPs with regard to the services provided.

To support automatic processing of the SP-to-SP service ordering, the interface between different SPs needs to be clearly specified. This section discusses the object model used for SP-to-SP service ordering and how this model can be used by examining a number of service ordering scenarios.

There are two major phases in a service ordering process: pre-order phase and ordering phase. Interface objects are defined for both phases. The interface operations for the pre-ordering phase provide operations for an MSP to make pre-order requests for services. The interface operations for the ordering phase provide operations for an MSP to make service orders. The interface can also be used by the MSP to query the status of both pre-order requests and service ordering requests.

3.1 SP-TO-SP SERVICE ORDERING INTERFACE

Since the object model is to be used by different service providers to enable automatic service ordering processing, the following issues need to be addressed:

- Interface and internal processes
- State-oriented or stateless object model
- Lifecycle of Objects

3.1.1 INTERFACE AND INTERNAL PROCESSES

The interface specification for the SP-to-SP service ordering is focused on the interactions between different service providers for service ordering purposes. How an ordering request is processed internally is not made visible at the interface level. This is due to the following reasons:

- Each service provider may have its own business rules. The handling of an ordering request should conform to these rules.
- There may be some existing systems used by different service providers which provide basic capabilities of handling service ordering requests. It is not practical to require all the existing systems to conform with a standardised internal behaviour.

- As a principle, the object definition should encourage interworking between different SPs, and avoid over-specifying internal system behaviours.

3.1.2 STATE-ORIENTED OR STATELESS OBJECT MODEL

In this specification, the stateless object model is used. This is based on the following two reasons:

- In a distributed system that supports distributed applications across multiple entities, a stateless object modelling of the application is generally safer to implement. It requires less synchronisation after communication failure or failures of the distributed processes. On the other hand, a state-based object model relies on the synchronisation between two state machines to maintain the consistency between two distributed and communicating components. It normally requires a more sophisticated integrity service to handle communication or process failure.
- It is more efficient to define an ordering process as a stateless process. For instance, a pre-order may or may not be required before an order can take place; an MSP may withdraw at any time before an order is made by simply forget it (not continue any further exchanges with the SSP); etc. In this model, any invocation of the interface does not rely on the previous sequence of invocations thus resulting in a much simple implementation.

3.1.3 LIFECYCLE OF OBJECTS

The lifecycle of each object requires to provide a common understanding between MSP and SSP of when and which object can be accessed. Based on a stateless object model, lifecycle offers better ordering process control than using finite state machine between two communication entities.

Object creation and deletion are the only object lifecycle states that are visible using the interface. Any other required lifecycle activities are considered implementation specific and thus not part of the interface specification.

3.2 BASIC OBJECT MODEL

The object model describes SP-to-SP service ordering by identifying the interface objects and their relationships to each other. These interface objects support all the interactions required between service providers for the purpose of service ordering.

The following object classes are defined for SP-to-SP service ordering.

- A Service Provider (SP) class—The SP class represents an organisation which provides telco services. It supports operations for making a pre-order request, issuing a service order, etc. Both MSP and SSP are instances of the SP object class.
- A Customer class—The Customer class represents an organisation which requires telco services from an SP. This class contains all information about the customer. It also contains references to a set of services the customer has ordered from the service provider and related account information. When an MSP requests a service from an SSP, a Customer object is created for the MSP.
- A Service class—The Service class represents a subscribed service provided by the service provider (SSP) to its customers (MSPs). An MSP may order more than one service from an SSP and an SSP may provide more than one service to an MSP.
- A Service Level Agreement (SLA) class—The SLA class represents the service level agreement between an SSP and its customer (MSP) with regard to a particular service provided. It includes the objectives (e.g., performance objectives) and other parameters regarding the provision of a service and the quality of services.
- An Account class—The Account class represents an account created by the SP for billing purposes. Between an SSP and an MSP, a number of Account objects can be maintained. More than one service of the same MSP can be charged to the same Account.
- A Request For Service (RFS) class—The RFS class represents a pre-order request issued by an MSP to an SSP. The MSP can issue a query requesting the current status of the pre-order request. At any stage the MSP can cancel the pre-order request.
- A Service Offer class—The Service Offer class represents a service offer made by an SSP to an MSP in response to the MSP's pre-order request. For each service pre-ordered by an MSP, only one service offer is created. However, a service offer may contain a list of options detailing how the service can be

provided in different configurations. These options are used by the MSP as the basis for the service offer negotiation.

- A Service Order class—The Service Order class represents a service order made by a customer (MSP). This is different from a service class as no service requests can be issued to the service order object, the only valid requests that can be issued to the service order object are those to update the order, to cancel the order and to query the progress status of the order request.

In the object model, an object class SP is defined to represent the service provider. However, MSP and SSP are not defined as derived classes from the SP class. The reason for this is that no special attributes and operations are required for MSP and SSP. Both MSP and SSP are instances of the SP class and the only difference is that they play different roles when a service ordering activity is taking place between them.

The object model is depicted in Figure 4:

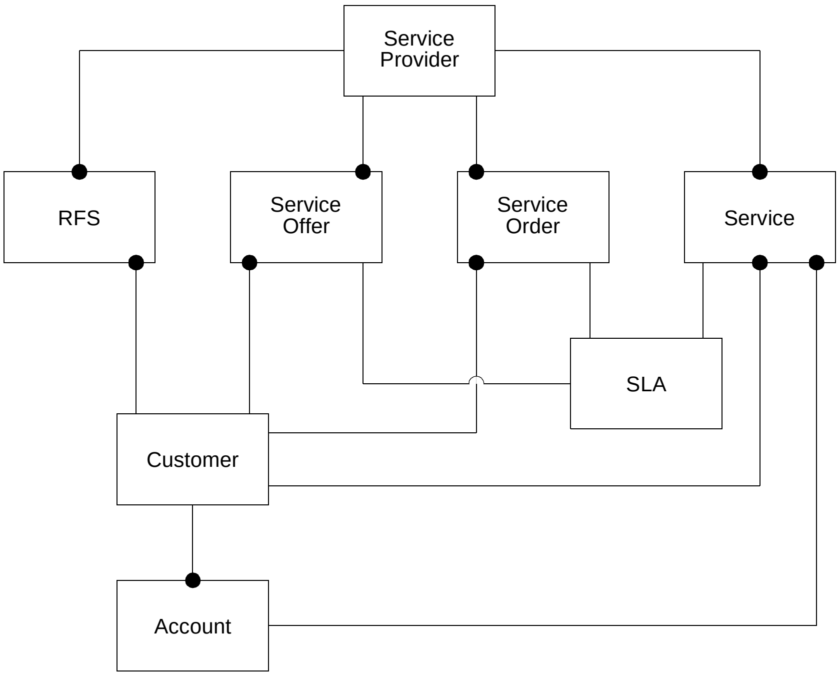

Figure 4: Object Model for SP-to-SP Service Ordering

The relationships among the service ordering object classes are:

- Each service provider is represented by a single SP object. Both MSP and SSP are instances of the SP object class.
- Each customer is represented by a customer object. An MSP is a service provider. However when it orders services from an SSP, it is treated as one of the SSP's customers.
- Each customer can have one or more accounts with a service provider. Each account is represented by an Account object.
- Each customer can have more than one RFS instances associated with it. Each RFS represents a pre-order request made by the customer.
- A service offer is made by a service provider to a customer that satisfies the conditions of the customer request and the negotiated service parameters. A service offer can be made by the service provider for each service requested by a customer. Each service offer is represented by a Service Offer object.
- Each customer can have more than one Service Offer instance associated with it.
- A service order is made by a customer based on the negotiated result of a service offer. Each service order request is represented by a Service Order object.
- Each customer can have more than one Service Order instance associated with it.
- A service level agreement is made between a customer and a service provider with respect to a service. Each service level agreement is represented by an SLA object. During pre-ordering, service offer negotiation and the service ordering phase an SLA object instance can be associated with an RFS, a Service Offer, or a Service Order object instance.

- For each service offer, an order can be made.
- A Service object represents a service provided by a service provider to a customer based on the agreed SLA. Each service will be charged according to the account related to the service. Different services may be charged to a single account.

3.3 SERVICE ORDERING SCENARIOS

A number of service ordering scenarios are presented here to demonstrate how the object model can be used.

3.3.1 PRE-ORDERING

A pre-order request is issued by a customer (MSP) to a service provider (SSP) to find out all the information about a service required by the customer. The process is started by the MSP issuing a pre-order request, but can be completed in a number of different ways. A successful pre-order scenario is described as:

1. MSP issues a pre-order request to an SSP about a certain type of service.
2. Upon receiving the pre-order request, the SSP creates an RFS object instance for the pre-order request and sends a response back to the MSP with the RFS identifier. This identifier can be used by the MSP for status tracking and request cancellation.
3. The SSP decides to make a service offer, and it sends a reply to the MSP with some details of the offer including the initial SLA associated with the proposed service.
4. The MSP and SSP may enter into further negotiation to finalise the choice of the options and the service features.
5. The MSP is satisfied with the negotiation result and is willing to proceed by making a service order within the valid period. The pre-order request is regarded as complete when a service order is placed by the MSP.

At any stage during the handling of the pre-order request and before a service offer is made, the MSP may query about the progress of the request. The SSP should send back a response with the status of the request.

At any stage after a service offer is made, the MSP may query the progress of the negotiation by getting the status of the service offer object. The SSP should send back a response with the status of the negotiation.

3.3.2 SERVICE ORDERING

A service ordering request is used by a customer (MSP) to a service provider (SSP) to place an order for the identified service. The service ordering can either be based on the result of a pre-order request or on information obtained from other sources.

All the service order scenarios are started with an MSP service order request. A successful service ordering scenario can be given as:

- The MSP issues an order request.
- Upon receiving the order request, the SSP creates a service order object and starts its internal ordering processes. It then sends a reply back to the customer with the reference to the order object and the expected processing time for the order.

  The service ordering processes are internal to each SSP. It may include some of the following activities:
  - develop an order plan
  - configure the network resource required for the service
  - place sub-order(s) if required
  - inform other processes, such as Billing, Performance Management about the service order
  - create a service object when the order has been implemented
  - perform internal service testing before informing the customer
- Once the order process is completed, the SSP informs the MSP of the availability of the service.
- Any interactions with the Billing system and the Performance Management system is internal to each service provider.

4 TECHNICAL ISSUES OF SERVICE ORDERING INTEGRATION

4.1 TECHNOLOGY CHOICES

Different service providers may select different technology as its underlying platform and implement the service ordering interface based on this technology. Examples of such technology include:

- Object oriented management technology, such as CORBA
- RPC based client/server system, such as DCE
- Peer-to-peer communication paradigm and its extensions, such as CMIP based TMN systems
- Message oriented middleware (MOM)

Also, the ordering processes and other business processes are supported by element and network management technology include:

- OSI based TMN technology
- SNMP based management technology
- Some CORBA based proposals and implementations for network and element management.

Given that the industry trend is already towards the co-existence of multi-technology, there are two major issues need to be addressed:

- What does it mean by conformance to the interface specification? In other words, at what level do we expect two different ordering systems to interoperate?
- How can different systems be integrated to provide a technology framework for SP-to-SP service ordering?

These two issues are common to all the telecommunication service management functionality concerned with SP-to-SP communications.

4.2 CONFORMANCE TO THE SPECIFICATION

There exist two levels of conformance to a specification: the protocol level and the interface level.

Each of the above mentioned technologies normally has a well defined communication protocol. For example in CORBA based systems IIOP is used for inter-ORB communication and in OSI-based systems CMIP is used for this purpose. One approach is to define the conformance at this level.

However, this level of conformance greatly reduces the level of abstraction of interface specification and its operations, thus reduces the degree of interworking between two different ordering handling systems.

The interface level conformance has different meaning in different systems. In object oriented systems or systems using object modelling techniques, it means to establish a mapping between the interface specification and the target object model. In non object oriented systems, it means to define a set of data structures and functions which can be invoked in that system and support all the capabilities specified in this document.

The following table lists the conformance level for some example target systems:

Target System	Method of Conformance
CORBA	IDL specification for objects and interfaces
OSI	GDMO object modeling and/or SMF functions
PRC	Remote procedures
MOM	Format of messages

4.2.1 CORBA BASED SYSTEMS

The Interface Definition Language (IDL) is used in CORBA based system to specify the interfaces of all the objects which can be accessed by external users. In the case of the SP-to-SP service ordering, the interface objects specified in this specification document need to be mapped into the IDL.

There are two special issues which need to be addressed. The first issue is how the relationship of objects can be modelled. OMG has defined a relationship service to represent different relationships between objects. However, few implementations of the relationship service are currently available on the market.

Another way of specify relationships is to build explicit links into the object specification. That is in each object, there are some attributes which give the references to other objects. This is the method currently used in the interface specification document. For example, for each service instance, one attribute points to the account to be charged for using the service and another attribute points to the service level agreement for the service.

One problem with this method is that whenever an object is created or deleted, all the links to the object must be updated. It is required that the object server has the ability to maintain the information and be able to make the update accordingly.

In general, it is a better practice to specify object relationships separately using some relationship service. This will encourage more object reuse and modularity.

The second issue is the communication between the CORBA Client and the CORBA Server. CORBA uses an RPC based communication and all the object invocations are initialised by a CORBA Client. In service ordering between MSP and SSP, different types of communications are required. For example, asynchronous communication is more suitable for placing a service order. Another example will be the requirement for an SSP (server) to initialise a contact to an MSP (client). There are a number of ways to allow a server to pass information to a client. The event service is an example.

Thus when a CORBA platform is selected, it is important to evaluate whether the platform supports asynchronous communication, and/or whether it supports the event service which allows a CORBA Server to send message to a CORBA Client. Support for different styles of communication may cause problem for the interoperability between different CORBA platforms.

4.2.2 CMIP BASED TMN SYSTEMS

The GDMO, ASN.1 and OSI General Relationship Model (GRM) are used in the CMIP based TMN system as the object modelling languages. They can be used to define object classes, data structures and general object relationships. However, there still remains an issue of lack of tools to support GRM.

The approach taken by NMF is to define all interface specifications using a general object model and map it into different technology domains: CMIP, SNMP and CORBA.

4.3 INTEGRATION ISSUES

4.3.1 CORBA/TMN INTEGRATION

Many efforts have been made in the area of integration between distributed middleware and the OSI network management technology [4]. CORBA/TMN integration work by the NMF and X/Open (JIDM work [9]) is one such example.

When mapping the interface specification into different object models, the integration issue has to be addressed. It is not hard to provide an IDL specification and a GDMO+ASN.1+GRM specification directly from the interface specification discussed in this paper. But it may well be the case that these specifications may not be consistent with the mappings supported by the JIDM approach. The following figure depicts this potential inconsistency:

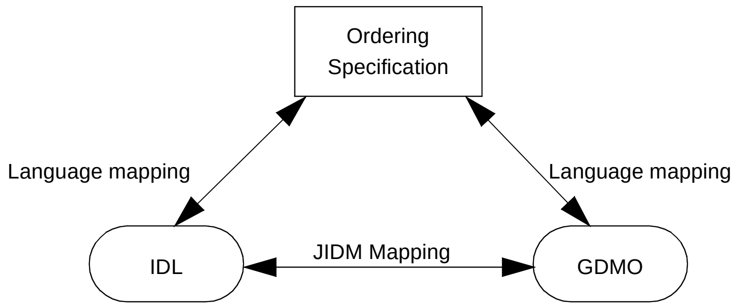

Figure 5: Language Mapping of the Object Model

In addition to this problem, there also exist some other mapping strategies. How these mapping strategies co-exist with the language mapping of the interface specification is also a problem.

Thus when defining the mapping from the interface specification to any implementation object models, it is important to keep in mind the integration issues between different object models.

4.3.2 OTHER INTEGRATIONS

More considerations are required on the issues regarding the integration with other systems, such as DCE and Message Oriented Middleware.

5 DISCUSSIONS

This paper presented an object model and interface specification for the service ordering business process between different service providers. It also analysed some implementation issues related to this interface in the areas of technology choice and integration.

The research team at CiTR is now continuing its contribution to the NMF in this area. The following issues are the target of further investigation:

- the implementation of this object model using different technologies, in particular, the TMN OSI-based technology and the CORBA-based distributed object technology
- the further modelling of service level agreement as the focal point between customer and service provider interaction
- the integration of ordering process and other business processes, such as performance, trouble handling, billing, etc.

ACKNOWLEDGEMENTS

An early version of this paper was submitted to the NMF SMART Ordering team as a discussion paper. The authors would like to thank the NMF Ordering team for their comments. The authors would also like to thank Paul Foster and the many people who have contributed to the information model and the integration issues discussed in this paper.

REFERENCES

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

[2] G. Chen, SMART Ordering Interface Specification Discussion Paper, NMF member's contribution, ORD/MIA006, January 1997.

[3] Y. B. Choi, O. Kang and H. F. Weng, A generic service ordering interface for the customer-service provider environment, Proceedings of the IEEE/IFIP 1996 Network Operations and Management Symposium (NOMS 96), Kyoto, Japan, April 1996.

[4] Q. Kong and G. Chen, Integrating CORBA and TMN Environments, Proceedings of IEEE/IFIP 1996 Network Operations and Management Symposium, Kyoto, April 1996.

[5] NMF, A Service Management Business Process Model, 1995.

[6] NMF SMART Ordering Team, SP-to-SP Service Ordering White Paper, version 4, January 1997.

[7] NMF SMART Ordering Team, SP-to-SP Service Ordering Interface Requirements, Issue 1.02, March 1997.

[8] NMF SMART Ordering Team, SP-to-SP Service Ordering Interface Specification, Issue 0.1, January 1997.

[9] X/Open, Inter-Domain Management Specifications: Specification Translation, X/Open Preliminary Specification P509, Nov. 1996.