SP-TO-SP SERVICE ORDERING SPECIFICATION AND ITS IMPLEMENTATION

GRAHAM CHEN, QINZHENG KONG AND PIERRE-YVES BENAIN[1]

***Abstract*—Network Management Forum (NMF) has defined a set of basic business processes dealing with major TMN service management functions such as performance management, fault management, billing and service provisioning. 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 an overview analysis for the service ordering process including the process definition, object modelling and process interaction.

To verify the process definition and object modelling, we built a proof of concept implementation of the Ordering specification discussed here. The focus of this implementation is to demonstrate the interface defined between different service providers in a transaction of service ordering. It implements the major concepts of the NMF Ordering process such as pre-order, negotiation, order options and service level agreements. The implementation uses the same interface specification for the two interactions in a service ordering scenario: the customer-SP interaction, the main service provider (MSP) and sub-contracted service providers (SSP) interaction. The latter interaction can be propagated in a recursive manner when a supplier requires further order from other service providers as part of the original order.

The prototype is built using Java.

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

***Source of Publication*—Proceedings of the 6th IEEE/IFIP Network Operations and Management Symposium, New Orleans, LA, USA. Feb 1998. ISBN 0-7803-4351-4.**

1 SERVICE ORDERING REQUIREMENTS

Three relationships exist in the interaction between customer and suppliers:

- the customer to service provider relationship
- the service provider to service provider relationship
- the service provider to supplier (network resource) relationship

These relationships can be illustrated as follows:

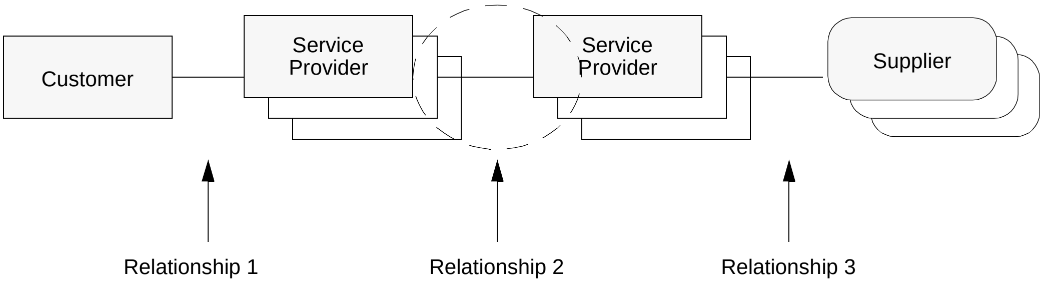

Figure 1: Different Relationships

The NMF SMART Ordering program is focused on the process definition and information modelling for the service provider to service provider (SP-to-SP) relationship. I.e., it focuses to providing a framework for service providers to exchange functions and information of the ordering requests.

The requirements of SP-to-SP Service Ordering has been clearly specified in the [1]. These requirements are summarised as follows:

- **Support for negotiation**—in a competitive environment service orders generally come out of a discussion, comparing different alternatives, with the customer.

[1] SITA, France

- **Support for pre-order request**—a pre-order request allows the service provider to plan for the service and network resource requirements and to issue a proposal to the customer.
- **Support for real-time**—the recommendation must be applicable in a real multi-service provider environment. Having a situation with several dozens of providers involved in an offer will not be unusual. Automation and real time capabilities are paramount when the number of parties involved increases.
- **Support for different roles**—many different types of service provider roles must be supported. New roles will be created in the ordering process. Roles may also change over time regarding different services. All the different players should be able to identify themselves.
- **Support for service ordering**—a service can be ordered. An ordered service can be updated and cancelled.
- **Support for Quality of Services (QoS)**—a Service Level Agreement (SLA) is required between the customer and the service provider with regard to a service. This SLA is used as the measurement of the quality of service parameters specified by the customer.

The information model for SP-to-SP Service Ordering should meet all the requirements listed above.

2 ORDERING SCENARIO

A Service Provider (SP) in today's market differentiates itself from its competitors based on its set of services and the quality of these services. The cooperation between different service providers is often required in order to provide a wider variety of services to customers. It is quite common for a service ordered by a customer to be operated over a network owned and provisioned by more than one service provider. When a customer requires a service, the customer contacts a service provider without knowing whether other service providers will be involved. The service provider makes contacts with all the different service provides required in providing this service. From the customer service point of view the initial service provider is the main service provider while the others are subcontracted service providers. In some cases, some subcontracted service provider may have other service providers as its subcontracted service providers in a recursive manner.

From the end-customer point of view, the customer is ordering services from only one service provider, the main service provider, the involvement of other service providers should be made transparent to the customer. The relationship between different actors in the service ordering process is depicted in the following figure.

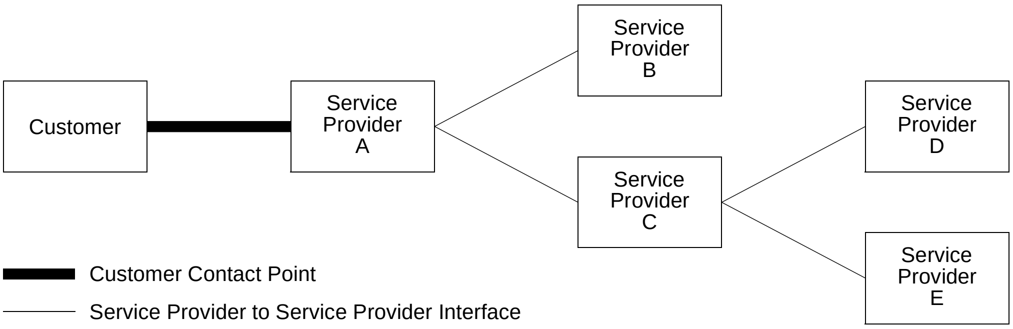

Figure 2: Different Actors in Service Ordering

The service provider A (SPA) is the main service provider to its customer and is the unique customer contact point for the services required. SPA requires services from service providers B (SPB) and C (SPC) and SPC in turn requires services from service provider D (SPD) and E (SPE). From a different point of view, a single service provider may play different roles. For example, SPA plays a service provider role to its customer but it plays a customer role for the services provided by SPB and SPC. The same applies to SPC, it plays service provider role for the services provided to SPA but at the same time it also orders services from SPD and SPE. From the Customer's point of view SPA is the Major Service Provider (MSP) and all other service providers are Subcontracted Service Providers (SSPs). From SPA's point of view SPC is the Major Service Provider while SPD and SPE are Subcontracted Service Providers.

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 the service ordering activities, the MSP is a customer of the SSPs with regard to the services provided.

This paper describes a design specification of the interface objects which are used for SP-to-SP Service Ordering. Interface objects are defined for the pre-ordering phase and the ordering phase. The interface operations for the pre-ordering phase provide operations for an MSP to make pre-order requests for services. While 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.

In this paper, the term MSP is used to refer to a service provider who plays a customer role in service ordering activities and the term SSP is used to refer to the service provider who provides the services. It is similar to the Management Entity (ME) concept in OSI model, where an ME can play a manager role and it can also play an agent role. These roles are only instantiated at the time of an interaction. With this approach, there will be no need to have both MSP and SSP objects.

3 ORDERING OBJECT MODELLING

3.1 CLASS DIAGRAM

Many objects might be involved in the SP-to-SP service ordering processes. This paper discusses only the interface objects required for service ordering.

The following diagram provides a SP-to-SP ordering object model. This model describes SP-to-SP Service Ordering by identifying the interface objects and their relationships to each other. These interface objects shall support all the interactions required between service providers for the purpose of service ordering.

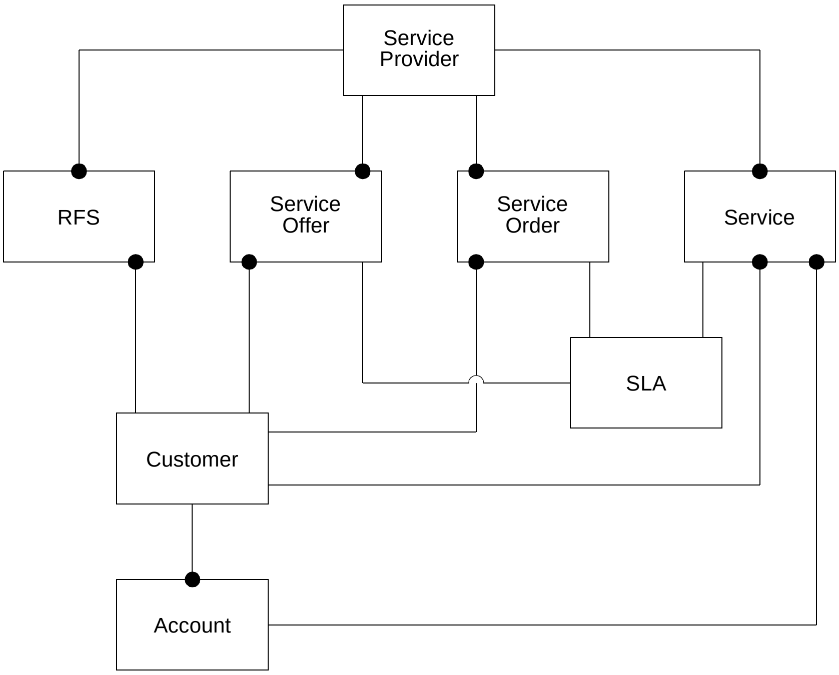

Figure 3: Interface Object Model for Service Ordering

To be able to support service ordering, eight object classes are identified as required at the interface level between different service providers. These object classes support all the service ordering related interactions between different service providers. It is up to each individual service provider to define more internal objects in order to handle service order requests.

The high level object class descriptions are given below:

- **A Service Provider (SP) class**—which represents an organisation which provides telco services. It is the main interface to the service provider. It supports functions including making a pre-order request, issuing a service order, etc. Both MSP and SSP are instances of the SP object class.
- **A Customer class**—which represents an organisation which requires telco services from an SP.
- **A Service class**—which represents a subscribed service provided by the service provider (SSP) to its customers (MSPs).
- **A Service Level Agreement (SLA) class**—which represents the service level agreement between an SSP and its customer (MSP) with regard to a particular service provided.
- **An Account class**—which represents an account created by the SP for billing purposes.
- **A Request For Service (RFS) class**—which represents a pre-order request issued by an MSP to an SSP.
- **A Service Offer class**—which represents a service offer made by an SSP to an MSP in response to the MSP's pre-order request.
- **A Service Order class**—which class represents a service order made by a customer (MSP).

A Service Ordering process contains a number of phases, such as the pre-ordering phase, service ordering phase. Different objects may be used during different phases.

3.2 SERVICE INTERFACE

The two diagrams given below depict the interfaces visible at the SP-to-SP interaction level. One is for the service pre-order and service feature negotiation, the other is for service ordering. These interfaces are the only visible information exchange between two engaging SPs. The specification does not cover any internal process necessary for handling the service ordering requests.

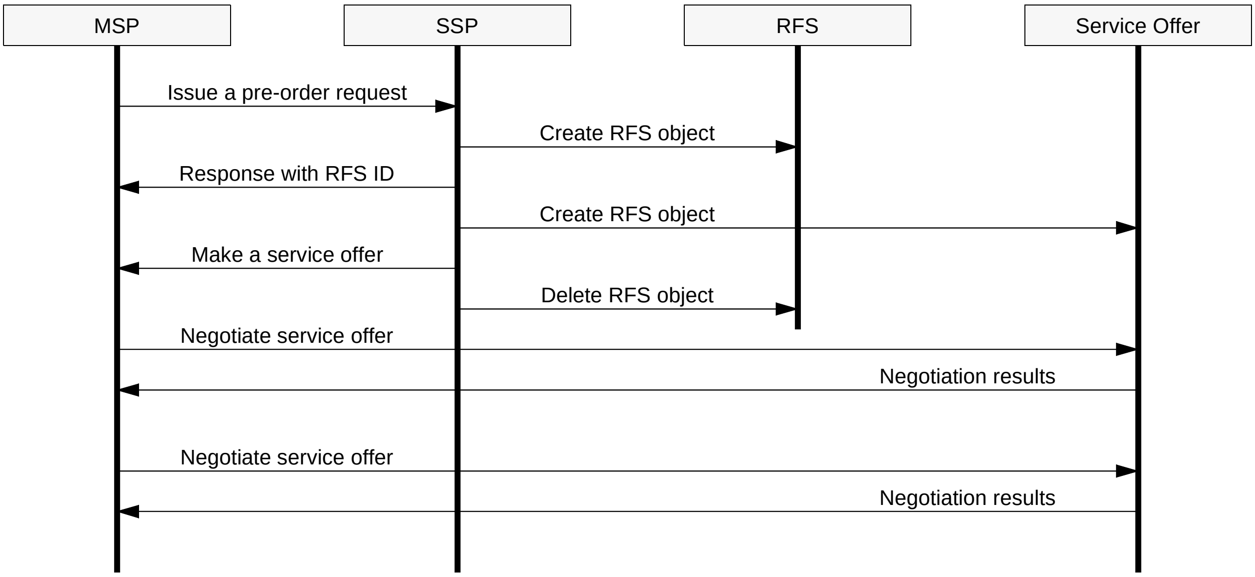

Figure 4: Service Pre-ordering Request and Service Negotiation

The interface specification for the SP-to-SP Service Ordering should be focused on the interface required for the interactions between different service providers for service ordering purposes. How an ordering request is processed internally should not be made visible at the interface level. There are 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 to a standardised internal behaviour.
- The design principle of this specification is to encourage interworking between different SPs, and avoid over-specifying internal system behaviours.

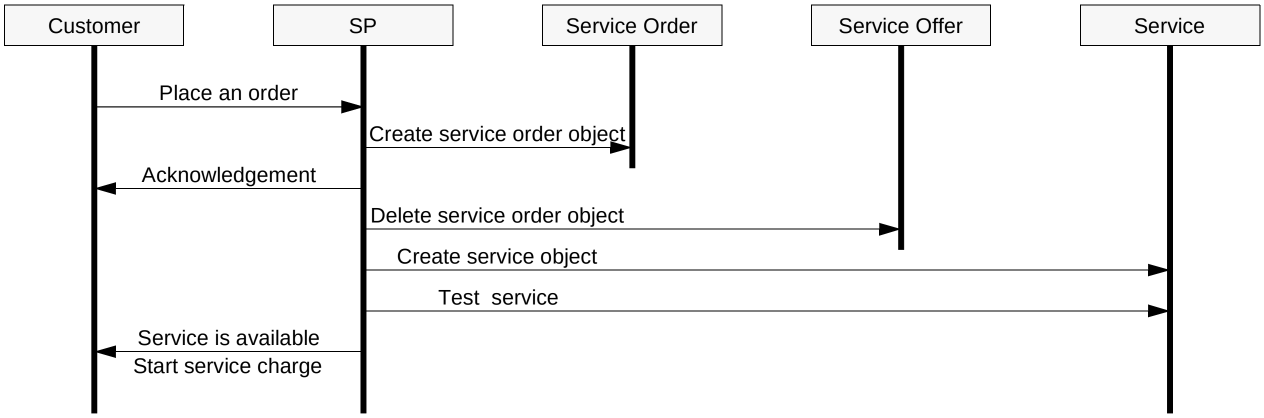

Figure 5: Service Ordering Request

3.3 MESSAGE DIAGRAM

The following message diagram is used to show the sequence of messages that implement an operation or transaction. Since SP-to-SP Service Ordering is focused on the interactions between different service providers, the message diagram is also focused at the MSP-SSP interface level. It is possible that in order to perform a complete service ordering process, each service provider needs to define some extra objects. It is the internal matter for the service provider to define more detailed message diagram according to its internal ordering processes.

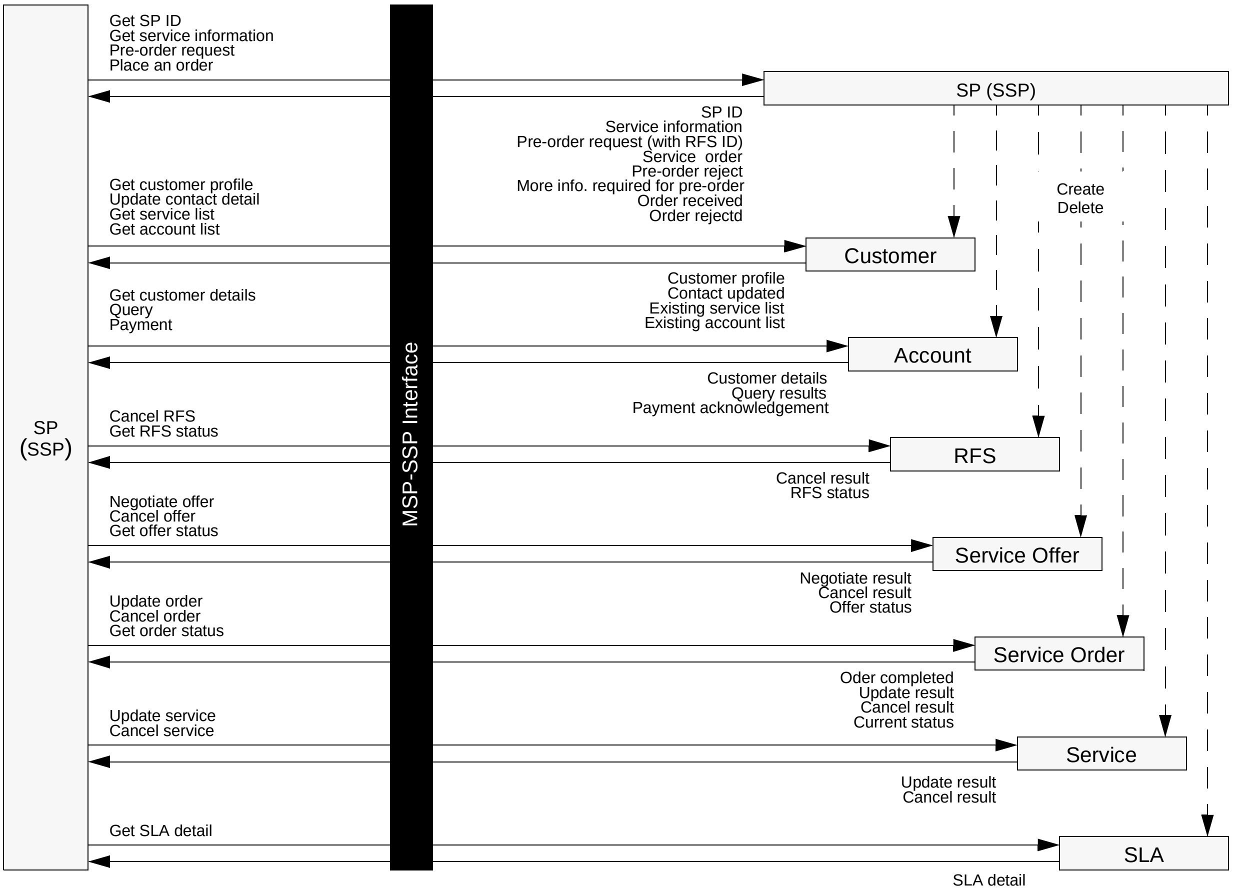

Figure 6: Message Diagram

The object model proposed here is a stateless object model, i.e., the object state is not visible at the interface level. In a distributed computing environment, a stateless object model is safer to implement. The message diagram is closely related to the object life-cycle, i.e., the message diagram triggers the creation/ deletion of the objects and the change of states of the objects.

4 IMPLEMENTATION

To demonstrate the concepts discussed in the Ordering process and information modelling specification, we build a proof of concept of implementation for the Ordering process.

The implementation is based on Java. The implementation supports the following major functional components:

- Service negotiation and service options
- Creation of Service Level Agreements (SLA)
- Server workflow for the process control and automation.

The following sub-sections give some details of the service ordering demonstrator.

4.1 ORDERING SCENARIO

A typical ordering scenario is given below.

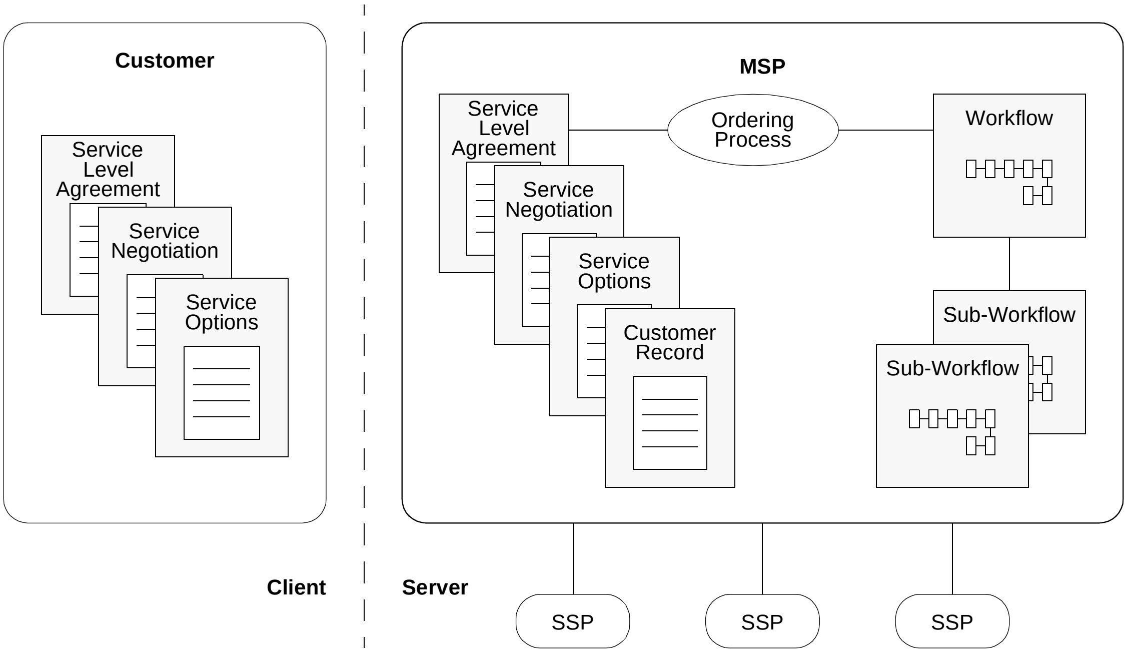

Figure 7: Ordering Scenario

A Customer connects to an MSP, which in turn connects to a number of SSPs to enable the ordering or a subcontracted part of the requested service. This extended service ordering is transparent to the Customer and offers a unique contact point for the service. This scenario results in there being a number of messages being supplied to the Customer by the MSP, such as, a Proposal, an SLA, an Offer, etc.

Workflow is shown here to illustrate the monitoring of the stages associated with the service provisioning process. It is the coordinator of all internal tasks proceeded by a service provider to support the Ordering activity.

Although the Ordering specification focuses on the SP to SP interaction, in a real system the customer to SP interaction is more important for SP's business. Also there should not be fundamental differences between the SP-to-SP relationship and the Customer-to-SP (C-SP) relationship, since in any ordering interaction, one SP is playing the role of customer and one is playing the role of supplier (SP). The differences will appear in the granularity of the ordered services. At the C-SP interface we will have complicated packages of services (e.g. including Voice, data, web etc.) whereas at the SP-SP interfaces we will see generic simple services (such as leased lines, VCs, etc.)

In our implementation, we see the benefit of using the same interface definition between customer to SP and between SP to SP. In a one-stop-shopping environment, this ordering scenario is in fact recursive. Thus the strategy to treat the customer-SP relationship as an instance of SP-to-SP relationship is very useful for increasing the software reuse and interoperability.

4.2 SP-SP INTERFACE

The main components of the demonstrator are the Trader, the Service Provider Server (SP) and the Customer Client, with the focus primarily on the SP and the Client as indicated below.

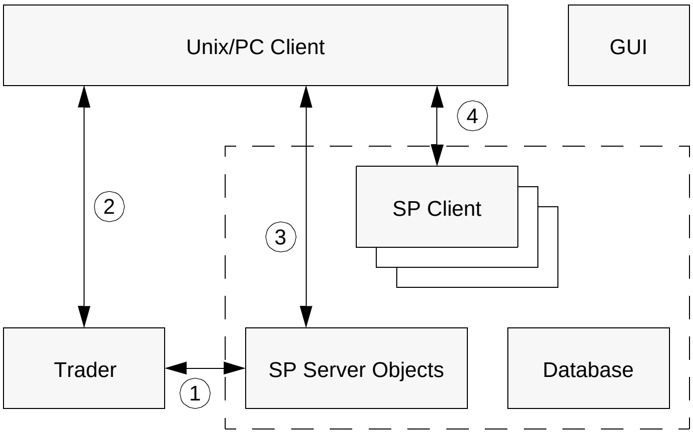

Figure 8: Major Components of the Demonstrator

At start-up the SP Server registers information about itself and its available services with the Trader (1). All connections to the Trader are maintained only for the duration of the registration or request. The Trader stores and collates service information for each service provider which contacts it. The Service Provider requests the appropriate information from the Trader, when a Client has requested a particular service which it is unable to supply from its local resources.

When a Client starts, it contacts the Trader. The Client submits a service information request to the Trader (2) and is supplied with all of the service information currently held by the Trader. The Trader currently has no means of ascertaining whether a registered SP is still executing or not. This means that the Trader may potentially supply stale information to the requesting Client. This feature could be incorporated, which would at the minimum ensure that the registered SP was executing.

The Client presents the trader supplied services information to the Customer for a service selection. The Customer selects a service and the Client connects (3) to this selected SP Server (SPS). The SPS instantiates an SP Client (SPC) to handle this Client instance. The SPS maintains knowledge of this SPC to enable user control of its workflow GUIs. The instance of the SPC sets up the serialised data socket (4) over which all further communication takes place with the Client.

Trader is running as a separate process in every distributed node, providing a federated view for service offerings and registered by all SPs. It is modelled upon the Trader service as exists in a CORBA environment. This Trader is of a limited functionality implementation, having elementary register, de-register and request operations.

4.3 SP SERVER ARCHITECTURE

The SP Server is the kernel component of the demonstrator. It supports the Service Ordering interface and has the basic service order processing capability. The SP Server architecture is given in Figure 9.

At start-up the Service Provider (SP) registers with the Trader service, supplying its unique identification and its available services. The SP object listens for new incoming Client login requests and starts an SP Client (SPC) Interface Object to handle the Client instance

In addition, the SP Object starts the SP User Interface (GUIs), the SP database and maintains visibility control over the SPC Workflow.

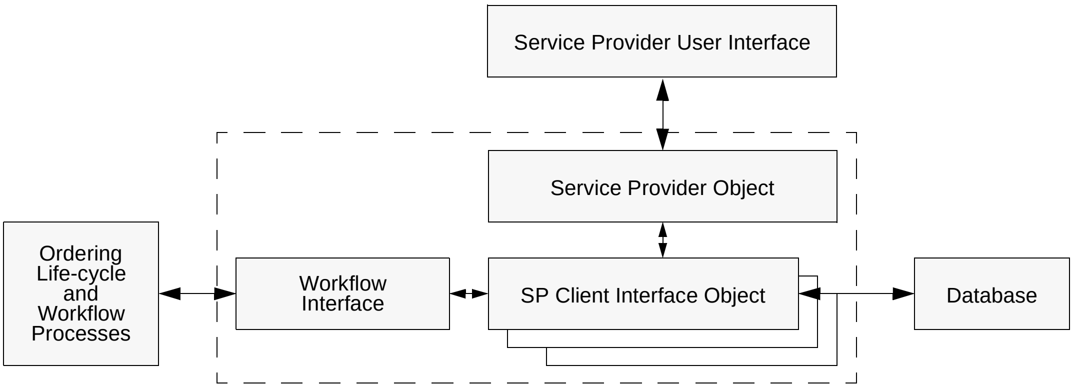
Service Provider User Interface
Service Provider Object
Ordering Life-cycle and Workflow Processes
Workflow Interface
SP Client Interface Object
Database

Figure 9: SP Server Architecture

Each remote customer request is handled in an autonomous asynchronous manner by the SP Client. The request is instantiated in its own thread space and communicates with the database (which stores all objects) and remote Client. This enables the SPC to be constructed in an asynchronous non-blocking model.

When messages and data from the client are received by SPC, the SPC delegates to the various internal sub-processes. The workflow diagram is updated by the SPC at different stages to highlight the ordering life-cycle for a particular client. Each of the sub-processes uses simple control mechanisms to maintain state. This is necessary so that the whole system functions in an asynchronous manner.

The implementation is based on Java. A Video-on-Demand service is used as an example. The following major Java applets and applications are implemented:

- A simple Trader applet
- A customer applet which supports customer ordering GUI including pre-order, SLA negotiation and firm order
- An SP applet which maintains the ordering objects life-cycle, persistence and handles all interactions with the customer.
- An SP workflow applet which controls and displays the workflow associated with the SP supplying the order. The workflow is recursive, i.e., each tasks in the workflow can be further defined as a sub-workflow. The sub-workflows can be managed and displayed in the same manner and by the same applet.

The customer applet and SP applet are reused in the recursive ordering scenario. I.e., the SP can play the customer role and use the same customer applet to request (through trader) other SPs for supplying part of the order.

The demonstrator runs on multiple Java machines. To get the best GUI effect, two machines are minimal. The machine running the server (SP) must be either a Unix workstation or a PC with NT installation.

5 CONCLUSION

In this paper, we have presented a framework for Telco service ordering and its process analysis and information modelling.

The information modelling is focused on the interactions between two service providers. This information model includes the object class diagrams and the interface diagrams and message diagrams. These form a basic framework for Telco service ordering.

We also presented a prototype implementation for the ordering process and the object model. In this implementation, the same ordering interface is used all following three interactions:

- Customer to SP interaction
- MSP to SSP interaction
- Any further SP-to-SP interaction required by any sub-order

A Java-based demonstration is constructed which does not only demonstrate the ordering specification and object model, but also demonstrate the complete service ordering process including:

- customer order and selection

- trading function
- nested order
- negotiation
- creation of Service Level Agreement

ACKNOWLEDGMENT

The authors like to thank all members of the Network Management Forum's SMART Ordering team for their input and all members of CiTR’s R&D team for their effort to build the prototype. In particular, the authors like to thank Paul Foster, Chris Phillips and Darrel Naylor.

REFERENCES

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

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

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

[4] 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.

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