

# Integrating Service and Network Management Components for Service Fulfilment

**David Lewis, Chris Malbon, George Pavlou, Costas Stathopoulos, Enric Jaen Villoldo**

**Abstract:** Solutions in the network and service management layers of telecommunications management architectures are currently fragmented both in terms of standards and products. It is often, therefore, difficult for the developers of management systems to reuse and integrate management software from different sources in cost-effective solutions spanning the various TMN layers. This paper presents the analysis, architecture and design of a system that integrates of service and network management building blocks in order to satisfy business process requirements for service order fulfilment. The presented system deals with network planning and provisioning according to network usage predictions drawn from customer subscription requests. It shows how specific components based on existing standards are integrated and how component technology is used to provide a flexible and extensible telecommunication business solution.

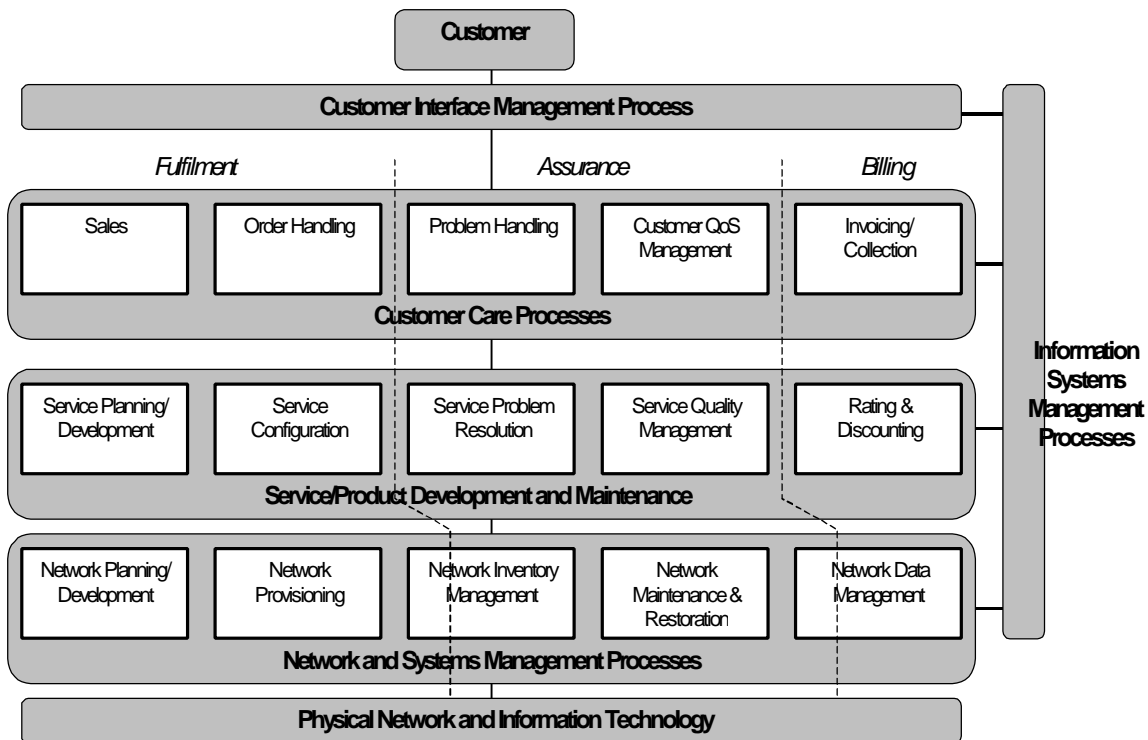
## **Keywords:**

**CORBA Components, TMF Business Processes, TINA NRM, Component Integration**

## **1 INTRODUCTION**

Mature telecommunication management standards exist largely at the element and network management layers of the TMN logical layer architecture. Few standards exist at the service management layer and consequently there is little in the way of standard information models or architectural guidance to those wishing to integrate management software across the network to service management layer boundary.

One industrial body that has attempted to address both service and network management in an open and integrated way is the TeleManagement Forum (TM Forum). The TM Forum aims to build on existing TMN standards with business agreements and procurement guidelines that directly reflect the industry's short term needs. It has developed a business process model [tmf-tom] that, based on surveys of major service providers, gives a detailed breakdown of the set of business processes that typically encompass a service provider's operations management activities. Interactions between processes in different providers are also identified. This model, summarised in Figure 1, is intended for use as the basis for identifying the requirements for specific agreements on common interfaces and information models, the development of which ongoing within the TM Forum. The model is partitioned into processes that relate to customer care, to internal service development and maintenance and to the management of the provider's networks and systems. The processes are also grouped vertically into major service management areas, i.e. the fulfilment/delivery of the service, the assurance/maintenance of the service and the billing/accounting for the service. Individual processes are defined in terms of activities within the process and of input and output triggers to the process.



**Figure 1: TM Forum Business Process Model**

Though this business process model provides developers with a suitable framework for analysing business process requirements that encompass both the network management and service management layer, few of the specific industry agreements and accompanying solution sets that have emerged from the TM Forum have addressed such interaction. Instead, inter-domain management interactions have been prioritised. However without guidance on how to develop open interfaces between management systems operating in different logical layers there will be a tendency for commercial solutions to be proprietary stovepipes, e.g. between network level accounting and service level billing software.

The Telecommunications Information Networking Architecture Consortium (TINA-C) is another industrial body whose work towards open standards has encompassed both service and network management issues. This group has aimed to develop a comprehensive architecture for telecommunications control and management based on Open Distributed Processing principles as defined by ITU-T in [ODP]. It generated detailed models for the integrated control and management of multimedia services and broadband networks based on existing concepts from TMN, IN and ATM. Service Management, principally subscription and accounting management, has been addressed in the TINA Service Architecture [tina-sa]. Network Management has been addressed in the Network Architecture [tina-nrim], which draws heavily on existing network management models, e.g. M.3100, in the structure of its Network Resource Information Model (NRIM). Both these architectures have exploited ODP concepts and component-oriented models to integrate management functionality with service and network control. However, the only area where management interaction between the two architectures is currently well defined is connection management, an area traditionally regarded in B-ISDN as being in the control plane.

A telecommunications management system developer faced with the need to integrate service and network management systems can use the TM Forum business process model to analyse their high level requirements. They can then draw on ITU-T, TM Forum or TINA-C specifications for specific service or network level management interfaces. However, there is little available guidance on the integration of such open solutions across business processes that reside in both the service and network management layers. The development of such guidance is the subject of research in the EU ACTS funded FlowThru project. Here, guidelines are being developed to aid management system developers to construct

systems assembled from reusable components that implement business process interactions. In addition, these guidelines will address how system assembly can be supported by new technologies such as Workflow, Enterprise Java Beans (EJB), CORBA Components and management protocol gateways.

This paper describes the solution to a particular management problem spanning both the service and network management layers, involved the integration of subscription management with network planning and provisioning for a switched ATM service. The approach taken shows how a CORBA component-based approach has been used to integrate existing software components in a loosely coupled fashion and in an easily extensible manner. This solution involved applying TM Forum industry agreements to existing, TINA specified components that were themselves the products of previous research projects.

The next section presents the order fulfilment problem being addressed and puts it in the context of existing TINA and TM Forum specifications. Section 3 reviews the features of the components that were assembled in developing the solution and how they were assembled and modified to satisfy the problem requirements. Section 4 discussed the approach taken before conclusions are drawn.

## **2 PROBLEM ANALYSIS**

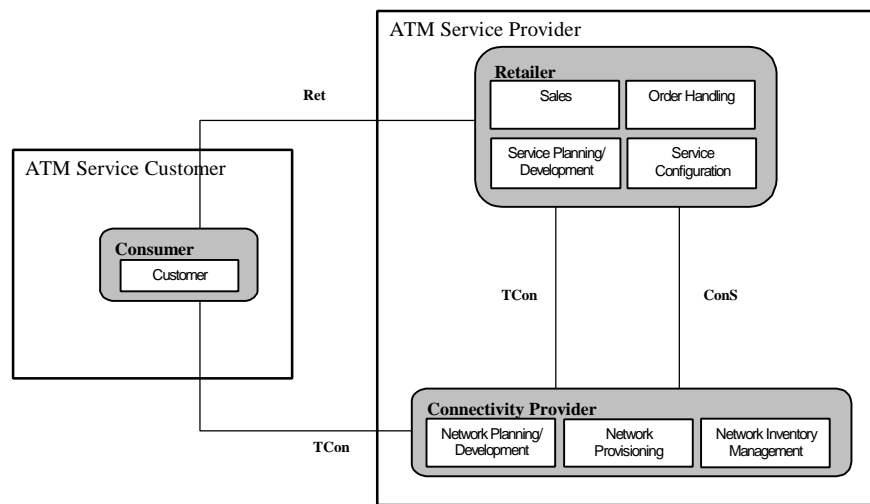
The problem addressed by this work is the integration of management functions in the service and network management layers related to the fulfilment of customer orders for a switched ATM service. A customer could range from a single-terminal, domestic users through to corporate users with switched ATM customer premises equipment, supporting a number of users via a single UNI.

In terms of the TM Forums business process model, the problem domain focuses on the fulfilment process and consists of the following business processes:

- **Customer Care Process:** This involves the management of the interactions between the customer and the provider's management system, in particular translating customer requests and queries into appropriate system operations.
- **Order Handling:** This involved the receipt of orders for ATM services from a customer and the translation of these orders, after appropriate financial checks, into requests for network configuration changes needed to satisfy the customer's order. It then involves tracking the progress of these orders to completion.
- **Service Configuration:** This involves the installation and configuration of a service for a specific customer, including any customer premises equipment.
- **Service Planning and Development:** This involves the design of the technical capability to meet market requirements for services within requirements costs and exhibiting required manageability characteristics.
- **Network Provisioning:** This involved the reconfiguration of network resources so that network capacity is ready for the provisioning of services.
- **Network Planning and Development:** This process is responsible for designing the network capability to meet specified service needs at the desired cost and within operational constraints, determined principally by service level agreement with customers.
- **Network Inventory Management:** This involves the installation of physical equipment in the network.

Within the context of TINA, the problem may be expressed in terms of business roles and reference points between these roles as defined in [tina-bm]. The business roles identified here are the Consumer role, taken by the ATM service customer, and the Retailer role and Connectivity Provider role, taken by the ATM service provider. This mapping aided system assembly as our available components were based on models proposed in the TINA service and network architectures. A mapping had already been

suggested in [lewis99] between the business roles defined by TINA-C and the business processes in the TM Business Process Model. Based on this, Figure 2 shows the overall business context of the problem being addressed in terms of the business roles played by the organisations involved, the business processes undertaken by those roles and the TINA reference points that exist between those roles.



**Figure 2: Business Process to Business role mapping for ATM Service Fulfilment problem.**

As the business problem under examination did not involve multiple providers the need for federation reference points between the roles within the ATM Service Provider (available in the full TINA business model) were dropped. The remaining relevant TINA reference points were:

- Ret: Concerned with: access control between the Consumer and the Retailer; the discovery and commencement of operational and management service offering; the control and management of sessions, stream flow bindings and stream flow content.
- ConS: The control and management of connections between network access points.
- TCon: The negotiation of and control of network level interconnection, typically this would be technology specific.

As the focus of this work was on the development of functions in the management plane, aspects of network and service control were assumed but not analysed or developed any further. The TCon reference point was not therefore examined on the assumption that existing UNI protocols would be used for connection set-up. Instead the focus was placed on the management interactions that occurred across the Ret and ConS reference points. Work on these reference points within TINA-C has mostly been concerned with the control of user sessions, split into access, service and communication concerns. This provided us with further contextual information within which the management of the service could be placed. The Ret reference point was derived from the TINA service architecture that included a detailed model of how customer subscriptions and user's individual profiles could be managed as described in the section on the subscription management component below.

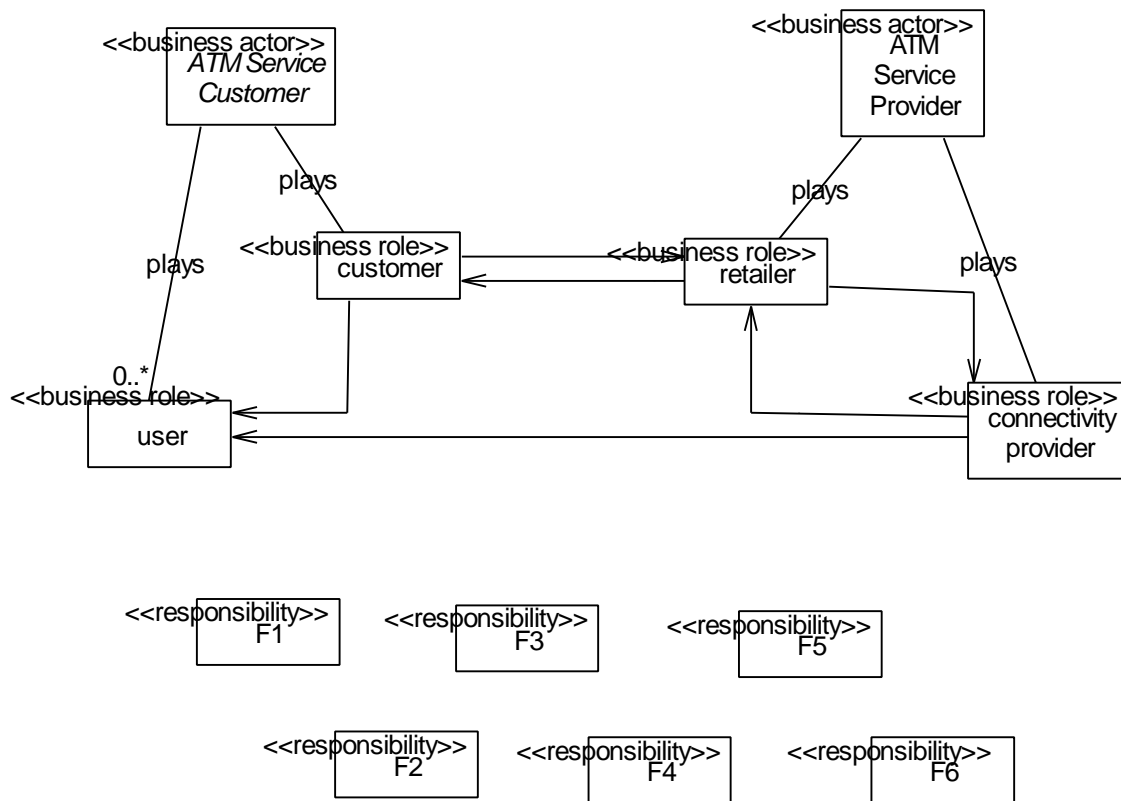
In TINA the ConS reference point addresses connectivity requests, replacing the role of the UNI for switched ATM connections used in the problem addressed by this paper. As currently defined, the ConS is a reference point between service and network control functions, reflecting the TINA preoccupation with control rather than management functionality. This means that there is no TINA equivalent for the management interface between the TMN service and network management layers. This vertical integration between service and network management could be made part of the TINA ConS through a different reference point segment [gmd-isn'99]. Another approach would be to define explicitly another reference point between the Retailer and Connectivity Provider roles in the TINA business model. In the FlowThru project, the former approach was followed, extending ConS with

additional network configuration and planning management features as specified by the REFORM project and described in the sections on Configuration Management and Network Planning.

These existing TINA-specified models provided a core set of concepts that could be used in further analysing the requirements of the ATM service fulfilment business problem and the requirements for interactions between the business processes identified as being relevant to this problem. The key concepts were:

- Network Access Point: This is the representation within the service provider of the point on the network via which the customer uses the switched ATM service. This offers a UNI to the customer and can support multiple simultaneous connections.
- Class of Service: To restrain the complexity of the network dimensioning problem while at the same time offering useful service definitions to the customer, the notion of CoS is introduced to quantise the space made up by ATM QoS types and the parameter ranges they may exhibit.
- Service Usage Group: This is derived from the TINA subscription management model and represents a group of users who may use a particular type of service, defined as a service profile. It also defines the set of network access point from which this group of users can access this service profile.

To analyse the problem further, business roles within the business stakeholders were analysed to determine the responsibilities they have with respect to each other. The set of responsibility relationships is represented as a UML class diagrams in Figure 3.



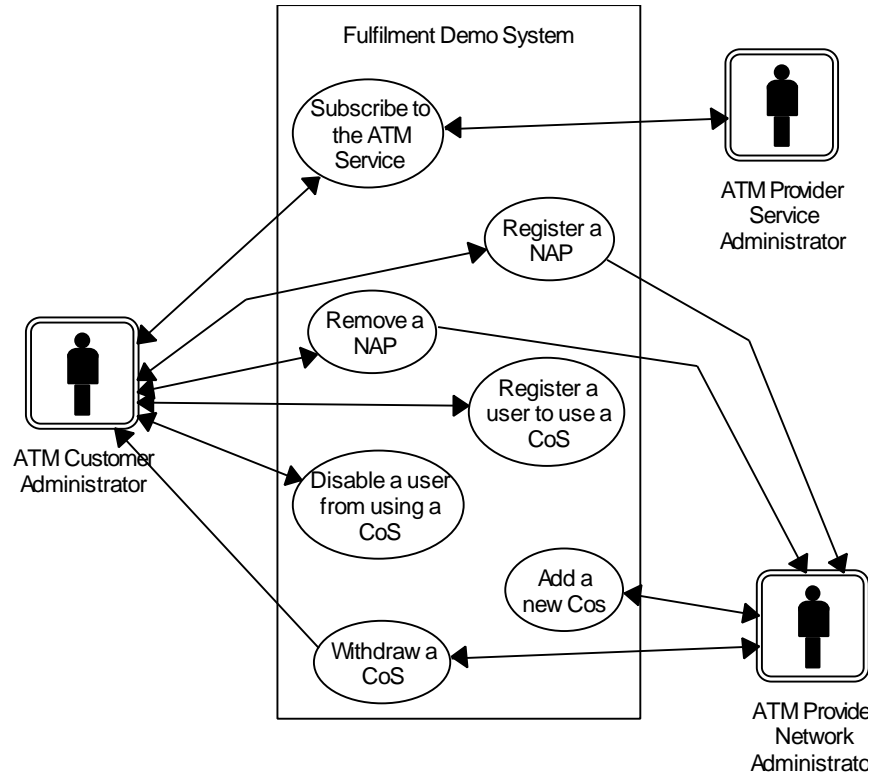
**Figure 3: UML Class Diagram illustrating responsibilities between Business Roles**

The responsibilities from Figure 3 are summarised in Table 1.

ID	Holder	Beneficiary	Description
F1	ATM Service Customer playing <i>Customer</i> role	ATM Service Customer playing <i>User</i> role	To ensure that users have access to the ATM Service.
F3		ATM Service Provider playing <i>Retailer</i> role	To provide accurate information on the number of users intending to make use of the service at different Classes of Service.
F4	ATM Service Provider playing <i>Retailer</i> role	ATM Service Customer playing <i>Customer</i> role	To provide access to the service from specified Network Access Points.
F6		ATM Service Provider playing <i>Connectivity Provider</i> role	To provide information on likely customer usage of the service and on the network access points from where the service will be used.
F2	ATM Service Provider playing <i>Connectivity Provider</i> role	ATM Service Customer playing <i>User</i> role	To ensure that ATM VCs are established according to connection requests with a specified Quality of Service.
F5		ATM Service Provider playing <i>Retailer</i> role	To provide information on the Network Access Points and Classes of Service available for customers.

**Table 1: Descriptions of Business Roles responsibilities**

Based on these responsibilities, use cases were drawn up documenting the functionality of the whole business system from the perspective of the actors who play the the business roles. The use cases are summarised in the UML use case diagram shown below.

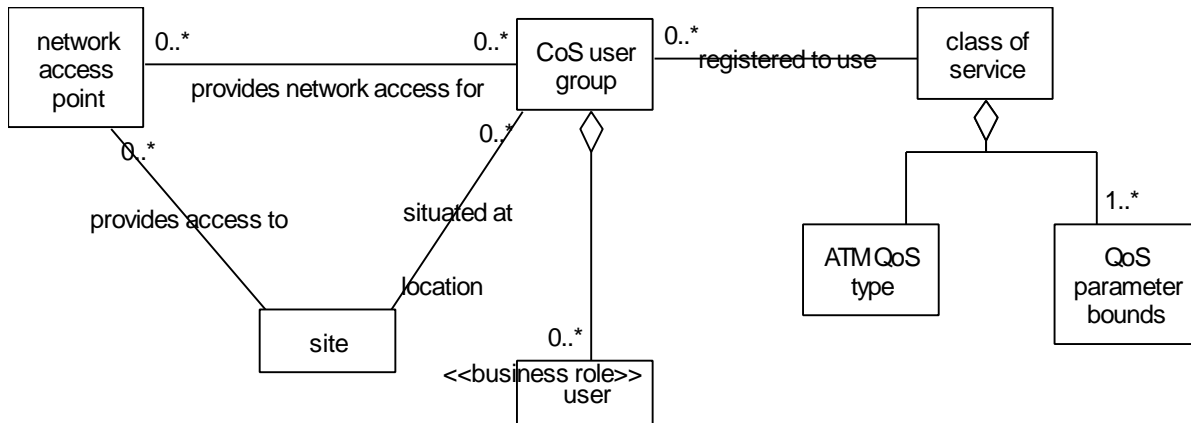


**Figure 4: Use case diagram for ATM Service Fulfilment scenario**

The use cases effectively defined the scope of the scenario being addressed. They were restricted to the following:

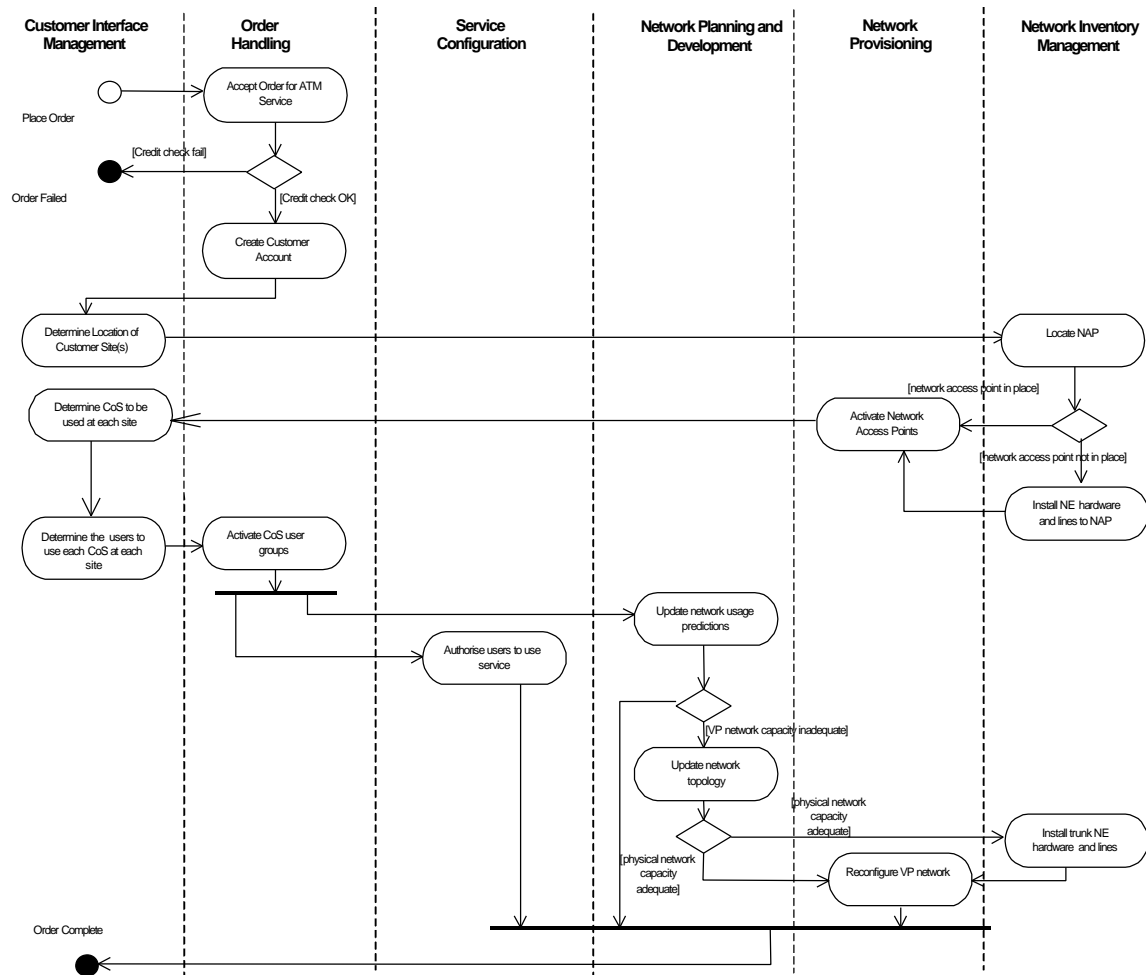
- Adding and withdrawing a class of service to/from those that a customer may use when subscribing to the ATM service.
- Customer ATM service subscription: providing contractual and accounting information on the customer; specifying the physical sites from where the customer will use the service together with lists of users at each site; the classes of service they will use and under which service level agreement parameters they will use them.
- Adding or removing a site from which the customer's users may use the services.
- Authorising or barring a user from being able to use the service.

The use cases allowed us to identify the main entities in the problem domain together with their relationship to each other and are summarised in Figure 5.



**Figure 5: Relationships between main entities identified in the use cases**

They did not, however, yield much direct analysis of the internal business processes, intra-component interactions still had to be identified. This was solved by breaking down individual use cases into separate activities that could be represented in a UML activity diagram, see Figure 6. By representing TM Forum business processes as a set of swim lanes and placing each activity in the appropriate lane, activities could be grouped by business process. As each business process could be encapsulated by a component, intra-component interactions were determined by examining the information exchange required to progress from one activity to the next.



**Figure 6: Activity diagram for subscribe customer to ATM service use case**

The next section presents how this analysis was mapped onto the design of existing management components and the identified interactions with respect to the outlined use-cases.

### 3 INTEGRATED SYSTEM DESIGN

The development of the system analysed in the previous section was driven by the fact that most of the required functionality was already available by existing components, whose re-use was a key concern. These components were the results of previous research projects that had implemented and extended portions of the TINA specification set, implementing systems focusing in either the service or the network management layer.

The need for reuse and integration of existing software led to a component-based development approach. Typically, components can be defined as independently deliverable packages of software services. It is worth noticing that component-based development can co-exist with OO analysis and development. While the later is focusing on analysing and designing systems under a specific business problem (as in the previous section's analysis), at the same time it is useful to perform the packaging of the developed system into a set of self-descriptive, application-independent components using the modelling constructs of a component technology.

This section describes how elements of the emerging CORBA component model [corba-comp] were used for modelling, packaging and seamlessly integrating the components of our system. It shows how the utilisation of a component technology provides easy component assembly and integration in a system designed using the OO development methodology. Three components (cf. Table 2) were



identified, modelled and integrated according to a subset of the CORBA component model.

Before describing how the merits of the component technology were exploited, the functionality of the components is outlined in the next subsection. Based on this high level description of the functionality of the components, Table 2 shows how they mapped onto the TM Forum business processes relevant to the fulfilment problem.

<b>Components</b>	<b>Business Process</b>
Subscription Management	Order Handling Service Configuration
Configuration Management	Network Provisioning Network Inventory Management
Network Planner	Network Planning/Development

**Table 2: Building Blocks and their mapping to relevant Business Processes**

## **3.1 COMPONENTS**

### **3.1.1 SUBSCRIPTION MANAGEMENT**

The Subscription Management component comprises all the management functions needed in order to define service offerings, administer customers and users, and manage the details of service provisioning. For instance, the component allows for authorisation and barring of users' access to specific services, and the addition and removal of network sites from which these users may access the service. The design is based on the subscription model developed in TINA [tina-sa].

The subscription component use cases have been developed based on the simple business model of a service customer and a service provider having a contractual relationship with each other. The subscription component is located in the provider's administrative domain and is used by two types of actors: administrators of the service provider and of the service customer.

The use cases implemented by the component cover the management of services, subscribers, and subscriptions. Management of service offering is concerned with the definition and adaptation of offered telecommunications services, from a management point of view. Management of subscribers consists of the creation and deletion of subscribers, the update of subscriber details, and the definition of subscriber's end-user groups and network sites. Finally, subscription management covers subscribing customers to services, update subscription details, cancel subscriptions, and the authorisation of end-users for specific services.

The component is decomposed into Computational Objects (CO). These define separate functional unit needed to manage subscriber organisations and end-users, different types of services, and subscriptions to these services. The COs includes Service Template Handler, Subscriber Manager, Subscription Registrar, and Subscription Agent.

The Subscriber Manager CO manages information associated to subscribers, e.g., subscriber details and user groups. The Service Template Handler CO is responsible for handling service templates that represent service capabilities provided by the provider. It also provides operations, such as verification of subscription parameters, to the corresponding Subscription Registrar CO. The Subscription Registrar CO acts on behalf of a provider in dealing with subscribers wishing to subscribe to a service. A single Subscription Registrar CO can collect information for one service from one or more subscribers and will handle exactly one service, since each service is likely to have different requirements for subscribing a customer to a service. A Subscription Registrar CO is responsible for managing of subscription contracts, among others. The Subscription Agent CO is closely related to the

User Agent CO that manages the access session of the TINA service architecture, but is not part of the subscription management component. It sends appropriate subscription information to the User Agent CO, in reply to its request. A separate management application is typically used to allow customer administrators or provider administrators to access the subscription component. The implementation of the subscription management component used for this problem was developed in the Prospect project, where it was reused in several different services.

The definition of the component assumed a predominantly synchronous mode of operation in its interactions with users and other service control and management components. The analysis of the problem in the previous section revealed however that the overall fulfilment process involved unbounded delays due to network installations and reconfigurations. The subscription management component therefore had to offer an interface to users that would enable them to manage prolonged order processing by the provider's systems. A further CO, the Order Handler, was added to deal with the reception, tracking and possible cancellation of orders received from customers. The basic abstractions used in this component were taken from the TM Forum's Order Handling business agreement [tmf-order].

### 3.1.2 CONFIGURATION MANAGEMENT

The Configuration Management (CM) component spans the network and element management layers of the TMN hierarchy. It is also part of the Network Resource Architecture (NRA) of TINA. The scope of configuration management is in general two-fold: to maintain a map of network resources both for inventory purposes and also for showing the topological relationships among resources; and to support the activation, deactivation, reservation and release of resources. The CM component presents a "server" interface to other components e.g. to Subscription Management and Network Planning and Dimensioning in the case of the fulfilment system described in this paper. In addition, the CM system acts as a client to network nodes with management interfaces e.g. to ATM switches.

In TMN systems, configuration information is typically held in one or more hierarchically organised network Operation Systems (OSs), whose information model conforms to the Generic Network Element Information Model [m3100] and to the emerging ITU-T Network Information Model. In this case, access to those objects is provided through TMN Q3 interfaces. In the case of TINA, configuration information is split in two parts: dynamic information, concerning on-demand or semi-permanent trails, which is held in the Connection Management (CM) subsystem; and static information, which is held in the Resource Configuration Management (RCM) subsystem. In this case, information is held in CORBA-based computational entities. In both cases, a client of the CM component may request the activation/deactivation of a resource, e.g. of a NAP for a particular subscriber, or the reservation and release of a resource, e.g. an ATM VP trail for planning and provisioning purposes.

The CM component in FlowThru comes from the REFORM and VITAL projects, which used TINA instead of TMN principles in their architecture. Given the fact that TINA concentrates mostly on Connection Management which covers the equivalent control aspects, modification and extension of the relevant models were necessary. TINA prescribes the Network Resource Information Model [tinanrim], which is based on equivalent ITU-T models, but the associated information objects are "hidden" behind computational interfaces. The problem is that there are no mechanisms for navigating the relationships among information objects and build a global network resource view. The NRIM model was thus extended while a CORBA-based computational object with TMN Q3-like IDL interface was designed to provide access to those objects in a flexible fashion [pavlou97]. An additional important architectural consideration was that the distinction between, separately held, static and dynamic resources was not considered appropriate.

As such, the CM component comprises two computational objects: the ConfM Network Map (ConfM-NM) object, which acts in a similar fashion to a TMN configuration OS; and the ConfM Connection Manager (ConfM-CM), which establishes and releases VP trails. The ConfM-NM embodies the functionality of TINA RCM while the ConfM-CM embodies the functionality of TINA CM for semi-

permanent connections i.e. ATM VPs. The difference is that Conf-NM holds also dynamic resources, e.g. VP trails, providing a consistent view of all the necessary information for planning and provisioning purposes. An additional interface to the ConfM-NM object is being developed in FlowThru for presenting a customised facility for NAP management to the Subscription Management component.

The current CM architecture is highly centralised, with two objects taking responsibility for the whole network. In contrast, a pure TINA approach would be fully distributed, with an object administering each subnetwork. Both these approaches have their problems. A hybrid approach which would try to find the best balance between scalability and performance could be the subject of future research.

### **3.1.3 NETWORK PLANNING AND DIMENSIONING**

The Network Planner (NP) component is located at the network management level of the TMN hierarchy. From a TM Forum's viewpoint, it maps to the "Network Planning and Development" process.

More specifically, the Network Planner consists of the following three subsystems:

- The Class Of Service Model (CoSM) subsystem.

The CoSM maintains a repository of the CoSs supported by the network. Each CoS is defined in terms of its bandwidth characteristics, performance targets and restoration characteristics. It should be noted that the CoSM restricts, yet is based on, the services offered to users through the Subscription Management component.

- The Predicted Usage Model (PUM) subsystem.

The functionality of the PUM encompasses the maintenance of a valid model of the anticipated traffic to the network. Based on this model, the PUM supplies the VPC\_TD (see below) with traffic predictions, required for the design of the VP layer. The VP layer (set of VPCs) and the static aspects of the VC layer (set of admissible routes per source-destination and CoS) are constructed so that to satisfy the predicted traffic demand.

Anticipated traffic is modelled in terms of the numbers of connection requests per CoS between source-destination pairs. The anticipated traffic model will be acquired from information regarding the subscribers of the network. This information is given by the Subscription Management component and it describes: the number of users that are subscribed to use the network; the specific sources (i.e. the customer's NAPs) that generate traffic; and the specific CoS characteristics of the traffic. The information regarding the CoS being used is extracted by the Subscription Management component by mapping the QoS characteristics of the services (as in SLAs) to network CoSs. Last, the PUM calculates the destination for the traffic generated by subscribed customers based on the service types that the customers are subscribed to. The following cases are covered:

- For services with specific destinations inside the provider's domain (e.g. a video on demand service, offered by the provider), the possible destinations are distributed among a number of known servers (e.g. the provider's video servers).
- For services with destinations outside of the provider's domain (e.g. an international telephone service), the possible destinations are the provider's egress points to a peer provider (e.g. a PBX connected to a public telephone operator).

Additionally, data regarding historical network traffic will be taken into account in order to make predictions as accurate as possible. Normally, the model needs to be validated and modified against actual network usage. However, this is not considered within the context of FlowThru.

- The VPC Topology Design (VPC\_TD) subsystem.

The objective of VPC\_TD is to design and redesign, whenever necessary, working VPCs and sets of admissible routes taking into account the existing CoSs. In addition, the existing component also designs suitable protection VPCs and allocates the appropriate amount of protection resources, thus

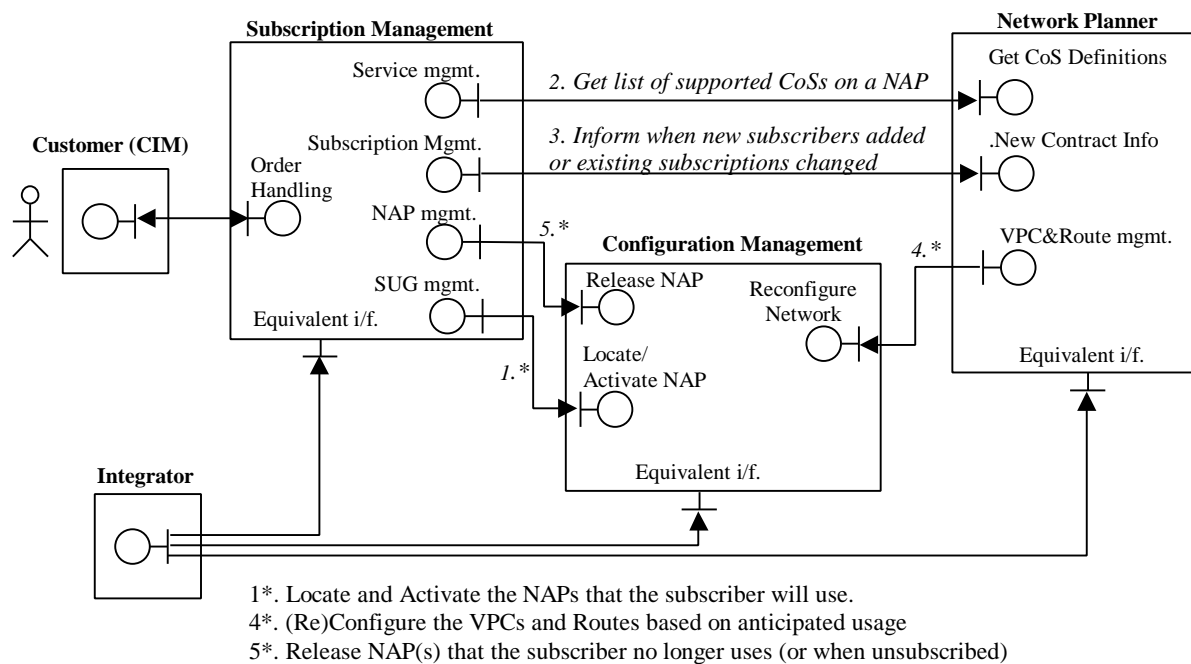
providing for a resilient network design. VPC\_TD's task is based on the predicted traffic and the physical constraints (e.g. connectivity, capacity of ATM links) of the network. These input parameters are subject to changes over the lifetime of a network. VPC\_TD offers a certain level of flexibility to adapt to such changes.

The tasks of designing VPCs and routes based on them should not be seen in isolation as they are tightly coupled; routes are based on VPCs; and VPCs are defined for routing. Along with these tasks, the tasks of designing appropriate protection VPCs and subsequently allocating protection resources across the network is considered. All these tasks are part of an iterative process involving complex optimisation problems aiming at providing cost-effective design solutions. The optimisation targets are set according to the business policy of the network operator. Part of this process will be to identify all possible routes between source and destinations and to choose a certain subset of these, according to the performance targets of the CoSs to be carried.

## **3.2 COMPONENT INTERACTIONS AND MODELLING**

Each component possessed a set of pre-defined IDL interfaces that had to be preserved to maintain access to its core services. Additional component functionality and the information that needed to pass between the components was analysed using activity diagrams detailing each use-case (see Figure 6). The identified component interactions are summarised below:

- The Subscription requires information on the available network-level CoSs to: create service records to which a customer can subscribe; define a set of QoS parameter limits to offer different SLAs for each service
- The Configuration Manager must know what NAPs are needed by a customer so that existing NAPs can be activated or new NAPs installed if none currently existed
- The Network Planner needs to be informed as new NAPs are activated for subscribed customers so that the Predicted Usage Model can be recalculated
- The Network Planner requires information on both the number of users at each NAP and the CoS to which they are subscribed, both influence the Predicted Usage Model
- NAP and user updates must be acknowledged by the Network Planner to indicate to what level each request can be accommodated within an existing SLA
- A user application for the service customer was required. This sends various subscription orders to Subscription Management and needs to be informed of changes in the status of these orders as they go through the various stages of fulfilment.



**Figure 7 Interactions sequence for a subscriber's lifecycle.**

Figure 7 depicts the component interactions and the integration approach followed. The sequence of interactions between the subscription management, the configuration management, and network planning components for a subscription lifecycle are given in figure 7. These interactions and the accompanying IDL definitions could form additional management segments to the TINA ConS reference points. The interactions 3, 4 and 5 can repeatedly happen through the lifetime of a subscription as the customer adds or removes new users and sites.

To promote loose coupling of the components an asynchronous event-based approach to component interaction was taken. An event based approach was also seen as essential for the Order handler CO in the Subscription Manager which had to mediate the progression of orders between the customer application and the other components.

To better support these interactions, the existing components were modelled by means of a CORBA-based component model. As work towards the specification of a CORBA Component Model is still ongoing (cf. [corba-comp]), current CORBA platforms do not provide support for building components as of this writing. However, a subset of the forthcoming component model suitable for our purposes was implemented. More specifically, part of the event model, as well as navigation through supported events, as in [corba-comp], was implemented. The consumer/publisher terminology introduced in the same specification is also used throughout.

The design borrows heavily from the Model-View-Controller classes used to build user interfaces in Smalltalk-80 and described in [gamma] as the Observer design pattern, and more recently deployed in the JDK 1.0+/JavaBeans Delegation Event Model and the CORBA component specification [corba-comp].

An event is described by a simplified version of the Structured Event, specified in the CORBA Notification Service [corba-notif], and consists of two parts:

- A fixed-length header that uniquely identifies the event type.
- A variable-length body consisting of a sequence of name-value pairs, containing the event data depending on the type of the event.

Each component provides the following services

- Navigation through the events that are both published and consume.
- The ability to subscribe an event consumer to an event publisher of a component.
- The ability to send an event to registered components.

The above features are made available through a generic interface (the so-called component equivalent interface) and support a standard set of operations through inheritance from a standard *ComponentBase* interface.

A specialised “integrator” component was used for configuring the runtime relationships between component instances. Initially, all system components are introduced into system being unconnected (that is, no consumer is registered to receive events from a publisher). By means of the navigation capabilities supported by the component equivalent interfaces the integrator identifies the consumer-publisher communication required and undertakes the appropriate event subscriptions within the system. This requires matching the event types that the components publish and consume.

In our prototype component model implementation, the run-time environment is rather simple, providing for no Containers. However, even with this prototype it is possible to easily plug our components in different systems. The separation of the integrator component enables this role to be played in future system by software configuration applications or possibly workflow based systems with knowledge of event types and component features.

### **3.3 DISCUSSION**

In the above example, design coupling between components is better targeted compared to components with synchronous interfaces by employing standardised event registration and message passing interfaces. An event consumer uses a well-known registration interface to notify a publisher of its desire to receive the latter’s events. Events are passed from event publisher to event consumer via a standardised message-passing interface. This enables components to exchange information with only knowledge of the individual event they are interested in, rather than having to bind to an interface definition that may contain many operations that are not of interest.

New functionality in the form of additional event protocols can be created without impacting existing component design couplings. Also, it is possible to introduce components that may be interested in consuming events for reasons other than their original intention, for instance some of the events from our system could be monitored by additional systems measuring the response times to customer order requests.

Components that receive events for which they have no semantic understanding may simply choose to log them or perform some other generic processing on them. We were able to exploit this final point by creating an event listener capable of demonstrating real-time information exchanges occurring in the system. Because the event listener was registered with all event publishers, each publisher notified the listener whenever it emitted an event. Registration was performed so that there was no need for publisher and consumer to enter into the normal event protocol dialogue, the event listener merely recorded and displayed events.

Frameworks can be deployed to expose event interfaces and connect them to consuming components in the same way as our simple integrator component. This allows the flow of information and control between components to be managed more flexibly, being less tightly-coupled to the design of the components themselves. This feature is compatible with the workflow based approach to OSS integration.

Threading ensures that a component is not suspended pending a reply to a given request. This avoids locking problems that may occur in synchronous systems as they are expanded with chains of synchronous requests.

The event-based approach still embodies some design coupling between components. In order to process an emitted event a consumer must know how to process a known set of event types. This is a fact from which no component can escape - a consumed event must trigger some predicted action even if this is to ignore it.

## **4 CONCLUSIONS**

This paper shows how problems spanning both the service and network management layer can be analysed using a combination of business processes (as advocated by the TM Forum) and business roles and reference point segments (as advocated by TINA-C).

The paper further shows, through an example, how such analysis is compatible with the construction of software solutions from existing components. Moreover, it illustrates how the use of UML Activity diagrams mapped to TMF advocated business processes can aid the design of those component extensions needed to effect successful inter-connection.

We have shown that the reuse and integration of off-the-shelf software that spans vertical TMF business process layers is possible through the exploitation of current component technology, particularly that advocated in the CORBA component specification.

The paper stresses how component integration can be more readily achieved by adopting an asynchronous event-messaging model to produce more flexible and extensible software systems. At a practical level, system functionality was extended through the introduction of event logging and monitoring components.

The use of the integrator object demonstrates how dynamic event identification and configuration can also be used to further improve flexibility by reducing component coupling to a runtime configuration

issue. This can be enhanced through the development of component and event repositories: a component repository providing a central location within which component interfaces details are stored; an event repository performing a similar task by enabling events to be defined at runtime. Furthermore, event adapters can be deployed between producers and consumers to further customise the event messaging at the component level.

Event coupling is unavoidable, an event is designed to trigger some action in a recipient. However, by using a non-blocking asynchronous event model, coupling can be reduced to a set of generic message passing and registration interfaces.

The identification of common event types within a given problem domain would be one approach to reduce event coupling. Components can then be developed which by definition are capable of processing some minimum event-protocol subset. We see bodies such as the TM Forum being instrumental in this through further analysis and development of their business process model to encompass the definition of common event types between processes.

## **5 ACKNOWLEDGEMENTS**

The work presented in the paper was conducted with partial funding of the European Commission under the FlowThru project (AC335). The views expressed here are not necessarily those of the FlowThru consortium.

## 6 REFERENCES

- [corba-comp] CORBA Components: Joint Revised Submission. OMG TC Document orbos/99-02-05, Draft, March 1999.
- [corba-notif] Notification Service: Joint Revised Submission. OMG TC Document telecom/98-06-15, June 15 1998.
- [englander] Englander, R., Developing Java Beans, O'Reilly, 1997.
- [gamma] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
- [lewis99] Lewis, D., A Development Framework for Open Management Systems, To be published in a special issues of the Icon Journal. ??.
- [m3100] Generic Network Information Model, ITU-T Recommendation M.3100, 1992.
- [ODP] Information Technology- Open Distributed Processing- Reference Model- part 1: Overview, ITU-T Draft Recommendation X.901/ ISO/IEC Draft International Standard 10746-1, 1995.
- [pavlou97] Pavlou, G., Griffin, D., Realizing TMN-like Management Services in TINA, Journal of Network and System Management (JNSM), Special Issue on TINA, Vol. 5, No. 4, pp. 437-457, Plenum Publishing, 1997.
- [tina-bm] Reference Points and Business Model, TINA-C, Version 3, June 1996.
- [tmf-order] SMART Ordering SP to SP Interface Business Agreement, NMF 504, issue 1.00, September 1997.
- [tina-sa] Service Architecture, ed. L. Kristiansen, TINA-C, version 5.0, 1997.
- [tmf-tom] NMF Telecoms Operations Map, NMF GB910, Stable Draft 0.2b, April 1998.
- [gmd-isn99] P.Schoo, P., Egelhaaf, C., Eckardt, T., Agoulmine, A., Tschichholz, M., Modularisation of TINA Reference Points for Information Networking, ,6th International Confernce on Intelligence in Services and Networks, Barcelona, April 1999
- [tina-nrim] Network Resource Information Model Specification, TINA-C baseline document TB LR.010 2.1 95, 1995