Managing Dynamic Inter-Domain Services

Fernando Matos, Alexandre Matos, Paulo Simões, Edmundo Monteiro
CISUC, University of Coimbra, Portugal
Pólo II, Dep. Eng. Informártica
3030-290 Coimbra, Portugal
{fmmatos, aveloso, psimoes, edmundo}@dei.uc.pt

Abstract

Because the world wide range of the Internet and infrastructure advance, customers can access any service over the entire world despite their location. However, the vending strategy current used in the Internet is no longer desirable to most of the providers. They no more wish to offer infrastructure facilities and charge based on how much traffic clients generate. Providers want to offer services and charge for them, thus providing types of services for clients according to their needs.

In this paper we propose an architecture for service management in inter-domain environments, which supports the service life cycle. Despite it deals with several service management aspects, we plan to focus in the service composition and service negotiation functionalities.

This architecture is able to offer a common infrastructure to provide different types of service and allows dynamic configuration and provisioning of services based on an end-to-end connection. It was built based on NGN principles and is composed of three layers: business layer for negotiation and service orchestration between providers; policy layer used for resource management and; infrastructure layer for network configuration. To validate the architecture we implemented a scenario comprising the service life cycle of an inter-domain VPN establishment, using RFC 4364 mechanisms for configuration.

Keywords — *Service management, SLA, NGN.*

1 Introduction

Since the establishment of the Internet, providers have charged their clients based on how much traffic they consume. In other words, their main revenue come from the infrastructure used to provide access and to transport the content requested by consumers.

This business view was not a mistake. Actually, it was a natural strategy to implement since in the early stages Internet did not have the range of services which it has today. Because the infrastructure advance it became possible to use the Internet as the main marketplace in the world, providing an enormous diversity of service for virtually any client in the globe.

This situation led providers to review their business strategies, forcing them to become more service providers instead network providers only. Besides, with the advent of next generation networks (NGN), it became clear the need for new models of service provisioning to deal with the aspects introduced by NGN [1, 2], like the possibility to offer services beyond provider domains and consequently integrate their services.

However, this change in perception brings other problems such as the tight relation between service management and network management [3]. Usually, when a service developer wishes to create a new service to offer, he must be aware of network details (available resources, topology, and other network manageable features) and consider them on the service specification. This can hamper the service provisioning since changes in the network could affect the service management.

Dynamic configuration is another challenge that emerges with the inter-domain service provisioning. Due to this heterogeneous environment it is assumed that changes in the topology may occur, occasioning a dynamic configuration on the services [4]. Furthermore, it is necessary to provide mechanisms to guarantee that services delivered by providers satisfy the agreed requirements, since they probably have different policies to deal with service provisioning.

In this paper we propose an architecture for service management independent from the underlying network technical information. This architecture gives support to the framework presented in [5]. It allows providers to offer on-demand services to their clients and to

clients beyond its boundaries through dynamic configuration, conforming new NGN characteristics. The service provisioning takes place in an end-to-end connection composed by intermediary services providers along the path. This composition uses service technical details from each provider to verify the appropriated connection to provide the service, according to the customer requirements.

In addition, it allows providers to perform dynamic agreements exchanging SLAs during service negotiation. The rest of the paper is organized as follow: Section 2 discusses some related works in service management. Section 3 gives an overview of the architecture. Section 4 shows how the architecture deals with service management aspects. Section 5 presents some implementation aspects, showing an inter-domain VPN establishment scenario used to validate the architecture and Section 5 concludes the paper.

2 Related Works

Inter-domain service management is one of the cornerstones in new generation networks. However, to put this into practice some of the today service management obstacles need to be addressed, like the decoupling between service management and network management and a flexible model to provide services despite their nature. Several studies have been done in this context and some of them are promising initiatives. This section reviews some important and current projects about service management.

NGOSS (New Generation Operations System and Software) [6, 7] is a model proposed by TMF [8] to help providers to develop and deploy operation support systems (OSS) and business support systems (BSS), aiming better efficiency in the interaction between providers, regarding service management and the associated business implications. It is composed of eTom (enhanced Telecom Operation Map) [9], SID (Shared Information Data and Model), TAM (Telecom Operations Map) and TNA (Technology Neutral Architecture). eTOM provides a common language of business process and helps to guide the development of theses process required by a service provider. SID is an object model created to provide a common vocabulary in the telecommunications industry, enabling the exchange of information in the business and system contexts.

Another initiative proposed by TMF is MTOSI (Multi-Technology Operations System Interface) [10]. It is a model to integrate providers' OSSs using XML/Web Services-based interfaces. It was developed as an extension of TMF MTNM (Multi Technology Network Management) and aims to provide a mapping from business specific activities to well defined contracts between OSSs. The MTOSI drawback is its lacking for a more complete service management strategy, regarding service publishing and service composition, for instance.

TISPAN [11] is another major project, held by ETSI, which goal is to provide standards to NGN architectures, concerning the aspects on several points, such services, protocols and QoS. It is composed of two layers:

- Transport layer: Provides IP connectivity for users;
- Service layer: Provides control functions over services.

One more effort towards service management is the OMA Service Environment (OSE) [12] initiative, from the Open Mobile Alliance (OMA). The OSE architecture is based on enablers, which are reusable codes that provide intrinsic functions (essential function to perform a specific task). This architecture intends to facilitate the service development process, by using a combination of enablers to perform a service.

The IPsphere Forum proposes a framework (Figure 1) where service providers can create and expose their services without the limitations of the classical IP model [13, 14]. The idea is to create a new layer above the IP architecture: SSS (Service Structuring Stratum). This new layer supports all business negotiation necessary to locate, compose, initiate, and terminate a service.

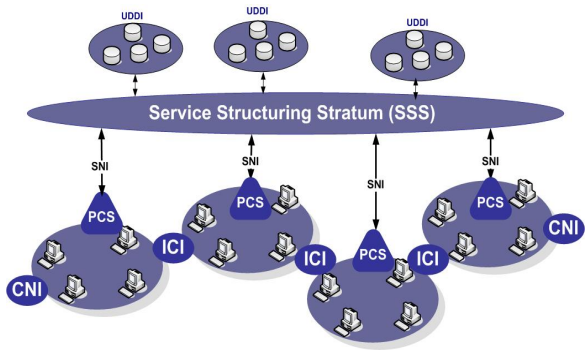

Figure 1: IPsphere framework overview

Customers connect to the framework through the CNI (Customer Network Interface) to perform service requisitions while providers use the ICI (Inter-Carrier Interface) to connect to each other. In order to translate business requirements from SSS, the Signaling Network Interface (SNI) sends instructions to the lower Policy and Control Stratum (PCS) which translates

high level statements in configuration instructions of its own domain.

The IPsphere framework aims to provide an integrated platform for service management, where customers can request services and providers are able to interact with each other to fulfill the customer requisition. However, more extensive support is still desirable for issues such as trust models and service level management strategies.

In [3] the authors propose a model for a service management layer (USMM) supported by two main concepts: SMFM (Service Management Function Model) and SMIM (Service Management Implementation Model). The former acts like a guide to specify what the service management should do, while the later how it should do. It claims that this model provides an independent layer, simplifying network details, which differ from other works. However, it focuses only in the model, lacking of a discussion on implementation details or a prototype presentation to validate the model.

Considering the mentioned works, we want to present an architecture that addresses the NGN requirements for service management, showing a functional validation of this architecture through an implemented prototype.

3 Proposed architecture

The proposed architecture was conceived as a mechanism to automate and facilitates the dynamic service provisioning in a multi-domain environment. It is intended to handle the entire life cycle of the service, since its creation and publishing until its execution and termination. Besides, it is modeled as a generic service provisioning mechanism which could support different types of services.

This architecture supports the framework depicted in [5], which is composed of three layers: Business Layer (BL), Policy Layer (PL) and Network Infrastructure Layer (NIL) (Figure 2).

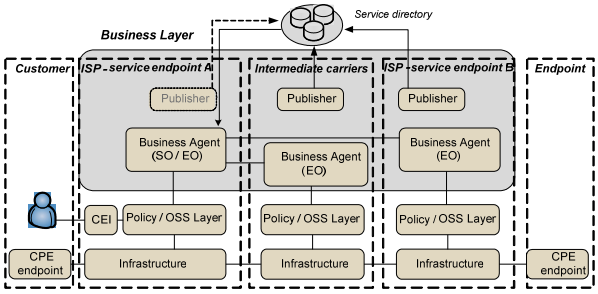

Figure 2: Global framework overview

BL is used as a collaborative environment where providers offer and negotiate services. In the architecture a service is offered by a service provider (Service Owner: SO) and is constituted by one or more service elements. These service elements are offered by providers (Element Owner: EO) and have similar behavior than services, except they are negotiated only between providers (SO/EO) and never between the customer and the SO. In order to offer a service or a service element in the BL, each provider needs to publish their offer in a service directory using the Publisher component. The Business Agent is responsible to locate services and negotiate their parameters on behalf each provider and can plays the role of SO or EO.

At the customer side, an interface named Customer Entry Interface (CEI) is used as front end to perform requisitions to the architecture. This interface can lighten the requisition task, since it hides technical details of the service, letting the customer to inform only indispensable parameters.

When SO and EOs reach an agreement, it is necessary to translate the service parameters into configuration instructions. This is done by the PL, considering internal policies and available resources of the provider. Finally, the NIL is responsible to configure the network equipments using the configuration instructions received from the PL.

The purpose of this architecture is to enable providers offer their services in a more dynamic fashion and enabling them to reach customers beyond their own domains. This new approach increases the service management complexity, since it is necessary to handle the service life cycle across different domains with probably different policies. The following section presents how the architecture deals with service management aspects.

4 Architecture Service Management Functions

Figure 3 illustrates how the main architectural entities (Customer, SO, EO and Service directory) interact with each other to provide a service.

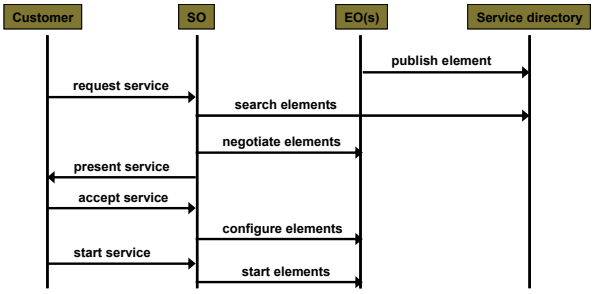

Figure 3: Service provisioning overview

When a provider wants to offer a new service, it starts by creating a Service Specification Template (SST) instance. This template is a form to correctly describe the service and at same time allowing providers to exchange this information in a standardized manner. The same occurs to service elements, thus creating Service Element Specification Templates (SEST) instance. Those templates include a general description section and a technical description section, which is inspired by the Tequila Service Level Specification (SLS) [15]. Figure 4 shows a fragment of a Service Specification Template.

After template creation, the provider must publish the element in a service directory, making it available for searches and requisitions. In the service directory, providers are able to look for element specifications which best suit their customer requirements. It is possible that only one element could be enough to provide the entire service. However, in most of the cases it is necessary several elements to accomplish that, especially if the service execution runs between domains.

Once the element providers are chosen, the SO initiates the negotiations with each EO. These negotiations consider element parameters exposed in the Service Element Specification templates and are made based on internal policies of each provider. For instance, a SO could choose the EOs that offer the best prices or had agreed to offer minimal packet loss. The result of this negotiation is a SLA between SO and each EO.

If the customer accepts the settled conditions between SO and EOs, a SLA between customer and SO is created. After that, the customer can allow the SO to initiate the service configuration. Henceforth, SO sends the service requirements for each EO in order to configure the network equipment. When an EO receives these requirements, it tries to reserve resources based on internal policies. It is entirely up to the EO which resources are configured as long as they fulfill the agreed conditions.

It is worth to mention that the information described in the templates is related only to services (service requirements, pricing details, warranty conditions) and this is everything providers need to exchange between them in order to negotiate and provide the service. This brings a significant benefit, since the service provider is not obliged to know network details to create/deploy a service offer, turning these tasks network independents. For instance, whether a provider wishes to offer a video streaming service it does not need to know:

- The network topology: in the service context, does not matter if the provisioning occurs through one or five providers, as long as the requirements are fulfilled;
- The network resources: the resources are dynamically allocated, thus it is not important to be aware of the resources capacity at service deployment. If some resource could not support the service in a specific moment, alternative providers are contacted.

```
- <ServiceOffer>
  + <ServiceOwner></ServiceOwner>
  - <ServiceDescription>
      <ServiceGlobalId>001122</ServiceGlobalId>
      <LastUpdate>2007-05-29</LastUpdate>
      <ServiceName>PortugalTelecomInterdomainVPN</ServiceName>
      <PublicationDate>2006-10-06</PublicationDate>
      <ValidFor>2007-10-06</ValidFor>
      <Category>VPN</Category>
      <SubCat>Interdomain VPN</SubCat>
    - <Scope>
        <ammountOfEndpointsSupported>10</ammountOfEndpointsSupported>
      - <multiplicity>
          <multiplicitySupported>(m,n)</multiplicitySupported>
          <multiplicitySupported>(1,n)</multiplicitySupported>
          <multiplicitySupported>(n,1)</multiplicitySupported>
        </multiplicity>
      </Scope>
    - <Monetary>
      - <BasicConfiguration>
          <minEndpoints>2</minEndpoints>
          <minTimeExpected>5 hours</minTimeExpected>
          <price>89</price>
          <currency>Euros</currency>
        </BasicConfiguration>
      - <AdditionalPrice>
          <valuePerHour>7.50</valuePerHour>
          <valuePerSite>12.45</valuePerSite>
        </AdditionalPrice>
      </Monetary>
    - <Performance>
        <worstBandwidth>512 Kb/s</worstBandwidth>
        <betterBandwidth>10 Mb/s</betterBandwidth>
        <worstDelay>30 ms</worstDelay>
        <betterDelay>0</betterDelay>
      </Performance>
    </ServiceDescription>
  </ServiceOffer>
```
Figure 4: Service specification template (fragment)

At the end of the configuration phase, the service is ready to be executed under the customer consent. In this case the SO starts the service and, monitors its execution to guarantee the SLA accomplishment. Similar behavior occurs in each EO, where they monitor the elements execution to guarantee the agreement accorded with the SO. At last, the service can be terminated at customer requisition, due to its regular flow or because technical problems, which

leads the SO to start a new service instance or to compensate the customer according the agreed SLA.

Despite the proposed architecture were built to deal with all those service management aspects, our focus in this paper is to discuss the service composition and service negotiation facets.

4.1 Service composition

One of the premises of the service management in NGN is the possibility to offer new kinds of service as soon as they are created and deployed. Nevertheless, in most of the cases this is a complex task, since the service management systems are built in a stovepipe fashion [1], what prevent them to be as adaptive and scalable as needed to support the growth in service diversity.

When SO searches for possible service elements in the EOs to satisfy the customer requisition, actually it is looking for providers (endpoints and transport providers) that, once their service elements are combined, it is possible to offer the service with the accorded requirements. In other words, it is looking for an adequate end-to-end connection.

To choose the appropriate connection, SO considers common services information like price, jitter, and bandwidth. At the moment a customer initiates a request, the SO verifies which service parameters have to be considered and uses them to choose the appropriate elements to compose the service. For instance, a requested service has a maximum limit of 3 percent for packet loss. From now on, SO searches for EOs who offers service elements which do not exceed this limit and discards everyone else.

Once SO selects the possible EOs, it must build the connection between the endpoints taking into account those EOs. To build this connection, SO verifies the reachability of each service element from each EO. This information is found in the SEST and at an initial phase we are considering the domains as the granularity for service element accessible points. In other words, a SEST specifies the domains which that service element can reach.

Once that information is extracted from the templates, SO concatenates the EOs to form a path between two endpoints. Suppose we want to establish a connection between the endpoints *A* and *B* using the providers *x*, *y*, *z*, and *w*. Figure 6 shows a graph representing how these providers are physically interconnected.

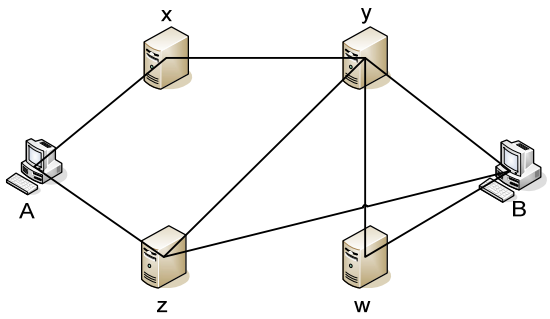

Figure 6 – Connection graph

SO creates a list with all possible paths between *A* and *B*. However, the creation of this list could follow some rules. For this example we applied a constraint in the number of hops (maximum of two hops). Thus, the created list has the paths (*A*, *x*, *y*, *B*), (*A*, *z*, *y*, *B*), and (*A*, *z*, *B*). Neither the path (*A*, *x*, *y*, *w*, *B*) nor the path (*A*, *z*, *y*, *w*, *B*) appear in the list because they have more than two hops.

After SO creates the possible paths, it applies the customer requirements to sort the list and selects the best path to provide the service. For instance, SO could choose the path with the lowest price. In case some EO along the path can no more guarantee the service requirements, SO chooses the next path in the list where that EO does not appear.

This method for selecting elements brings a significant advantage because SO does not consider specific service details when composing the end-to-end provisioning channel. This process allows that any type of end-to-end service could be provisioned on the architecture, independent on the specialized service characteristics. Besides, the storage of the possible paths speeds up the service reconfiguration time.

4.2 Service negotiation

Services ordered must be described in SLAs. These agreement documents are the basis where a SO guarantees the service provisioning. However, according to [16] and [17], there are some issues in SLA management that must be especially handled at a NGN perspective.

First of all it is important to consider that in a multi-domain perspective SLAs also must be accomplished between peers that collaborate in a service. In this architecture the SO indirectly intermediates SLA establishment between customer and all involved providers (Figure 5). The first step starts when BL decomposes the whole SST on p-SSTs in order to attend only of a specific domain provider. Between SO and EOs occurs a negotiation based on each

corresponding p-SST. If the parameters stated on p-SST are in accordance with the respective element domain policies, it must be automatically established a SLA between SO and EO, here stated as p-SLA. As indicated at Figure 5, after all p-SLAs established, the SO is prepared to request customer approval of the composed SLA.

The technical information representing SLS (Service Specification Level) from SLA is initially produced in the SO during service negotiation, and completed at Policy Layer (PL). This information will be used by Network Infra-Structure Layer (NIL) in order to configure domain resources.

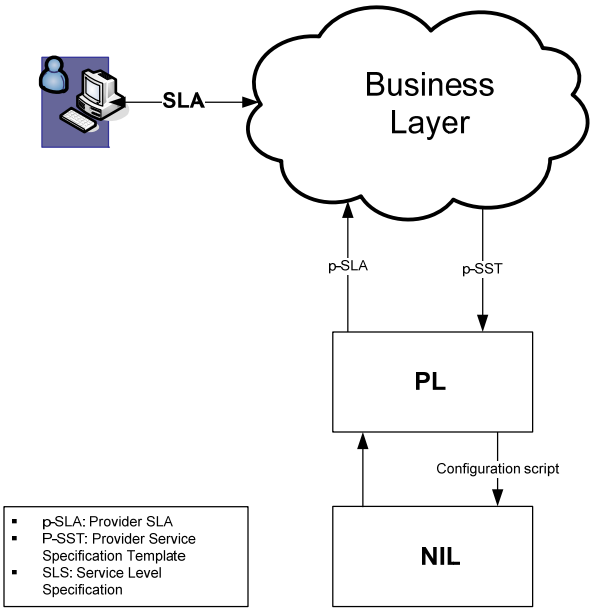

Figure 5 – SLA establishment flow

5 Inter-domain VPN use case with implementation aspects

We implemented a prototype to validate our architecture and it was used to request and create an inter-domain VPN service. UDDI [18] was used as service directory and the mechanism to manage the entire service life cycle was built, considering the following workflows:

- Service publishing. Element/Service providers publish service offers using an UDDI-based federation of service directories.
- Service discovery. Element/Service offers can be searched at UDDI directories according to specific criteria.
- Service creation (activation), by means of Service Owner to Element Owner negotiation.
- Service ordering by the customer.
- Service management, including monitoring of the service provisioning to guarantee its requirements.
- Service termination, after execution is completed or due to unrecoverable failures.

To enable the customer to request the VPN service, a web-based application (B2C Portal) was designed. This application hides service technical details and only asks for high level requirement information. Figure 7 shows a screen of the developed application. Web Services [19] were used as the technology for service invocation. It can guarantee interoperability between providers, since it is based on standard protocols and patterns, like XML, WSDL and SOAP,

A simple Policy/OSS Layer was implemented to receive service requisitions from Business layer and translates this requisition in router configuration commands, forwarding them to the Network Infrastructure Layer. The policies used to generate the appropriate configuration, based on equipments availability, are internal to the providers. Hence, they are most likely to use their own OSS application to make this role. However, it is important to consider the interactions with the business layer.

Concerning the network infrastructure layer, we used the Dynamips Cisco router emulator [20] and the Dynagen front-end [21] to emulate routers to be configured at VPN provisioning. When the network infrastructure layer receives the configuration commands came from the PL, it opens a Telnet/SSH connection with each router in the emulator, thus allowing to configure the VPN. This VPN configuration follows the RFC-4364 (BGP/MPLS VPNs) [22] recommendations.

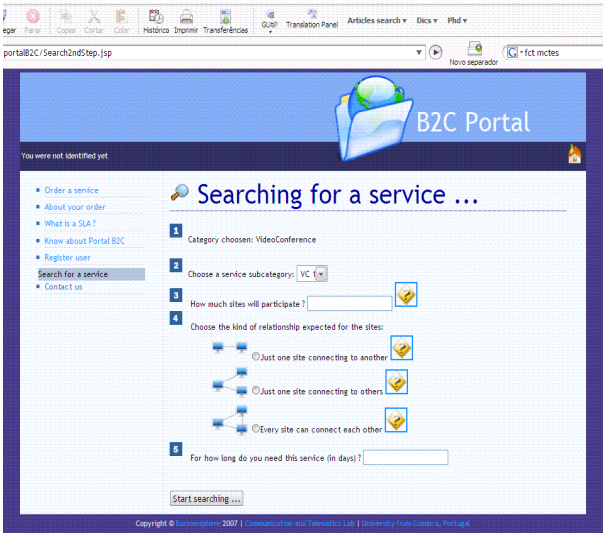

Figure 7: B2C Portal

During our tests we verified that the architecture can deal with the service life cycle for the VPN use case. Therefore, it is expected the same behavior for different types of services, since during service management specific service information is not considered.

In the prototype, we could also confirm the architecture capability to configure the VPN service in a dynamic fashion, right after customer requisition. The service was assembled considering the requirements imposed by the customer. However, for an initial implementation phase we used a basic assembly approach based on the lower price requirement. As the work progresses we intend to develop a more complete service assembly mechanism, taking into account other service parameters, such as jitter, bandwidth, packet loss, and so forth.

We also have in mind to study ways to improve the utilization of UDDI as service directory. During implementation we realized that UDDI is not completely suitable for service selection. It has a limited search capability, due to the restricted range of criteria used in selection. Other topic we intend to investigate is how to use federation of UDDIs, and consequently methods to perform synchronization between them.

6 Conclusion

In this paper we proposed an architecture (based on NGN principles) to dynamically provide services in inter-domain environments. This architecture aims to facilitate the service provisioning among providers through the use of a service management function independent from the network management. This feature allows the service developer to not worry about network details, such as topology or available resources.

Another advantage of the architecture is the possibility to manage different types of services, since the information exchanged during negotiation does not consider specific service details. It concerns about the establishment of an end-to-end connection, assuring that the service requirements (such as jitter and packet loss) are fulfilled.

We implemented a prototype to validate the architecture. This prototype manages an inter-domain VPN service provisioning, since its publishing until its execution. During prototype implementation was verified that assembling of elements between domains it is not an easy task, due to the several involved service requirements. For demonstration purposes, we applied a basic assembly mechanism based solely on the lower price requirement.

For future works we intend to design a more complex service assembly mechanism, taking into account other service requirements and probably use some well-know graph search algorithms to optimize the paths selection.

An analysis on optimization and scalability problems on the Business Layer will also be performed. In addition we plan to verify QoS and security mechanisms to apply on the architecture, allowing a better service experience to customers, approximating from the NGN promises.

Acknowledgements

This work was partially funded by FCT (scholarship contracts SFRH/BD/16261/2004 and SFRH/BD/29103/2006) and supported by the IST FP6 OpenNet Project.

7 References

[1] I. Grida, B. Yahia, E. Bertin, N. Crespi. “Next/New Generation Networks Services and Management”. Proc of the International conference on Networking and Services, 2006. ICNS '06. July 2006.

[2] A. R. Modarressi, S. Mohan. “Control and management in next-generation networks: challenges and opportunities”. In IEEE Communications Magazine, vol 38, pg 94-102.

[3] Y. Dafeng, Y. Fangchun, “A Universal Service Layer Management Model in NGN”. Proc. of the 9th International Conference on Advanced Communication Technology (ICACT2007), Phoenix Park, Republic of Korea, February 2007.

[4] M. Li, K. Sandrasegaran. “Network Management Challenges for Next Generation Networks”. In Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary (LCN05). Sydney, Australia, Nov. 2005.

[5] A. Matos, F. Matos, P. Simões, E. Monteiro. “A Framework for the Establishment Inter-Domain, On – Demand VPNs”. Accepted for publication at IEEE/IFIP Network Operations and Management Symposium (NOMS 2008), Salvador, April 2008.

[6] B. M. Gupta, M. Sarkar. “Business Integration Architecture for Next Generation OSS (NGOSS)”, Infosys Whitepaper, January 2006.

[7] C. R. Gallen, J. S. Reeve, “Using Open Source to realise a NGOSS Proof of Concept”. Proc. of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006), Vancouver, April 2006

[8] TMF, http://www.tmforum.org/browse.aspx.

[9] M. B. Kelly, “The TeleManagement Forum’s Enhanced Telecom Operations Map (eTOM)”. Journal of Nework and Systems Management, March 2003, vol. 11, nº 1, pp. 109–119.

[10] S. Fratini. “Multi Tehcnology Operations Systems Interface (MTOSI) Business Case”. Telcordia White Paper.
[11] TISPAN. ETSI 282 001 Ver 1.1.1. NGN Functional Architecture Release 1 Overall architecture.
[12] OSE – OMA Service Environment, http://www.openmobilealliance.org/.
[13] T. Nolle, “A New Business Layer for IP Networks”, Business Communications Review, July 2005.
[14] J. Alateras, “Ipsphere Framework Technical Specification – Release 1”, 2007. Available at http://www.ipsphereforum.org/R1Spec.html.
[15] T. M. Nguyen and N. Boukhatem. “Service Level Negotiation and COPS-SLS Protocol”. Proceedings of Annals of Telecommunications, vol. 59, issue 1-2, pp. 37-51.
[16] Yin, Z.M.; Yang, F.C.; Liu, Y.Z.; Service management architecture and information model for next generation network with dynamic service level agreement management. Proceedings of the 12th IEEE International Conference on Networks, (ICON 2004). Volume 1, 16-19 Nov. 2004 Page(s):437 - 441, vol.1.
[17] Acimovic-Raspopovic, Vladanka S.; Stojanovic, Mirjana D.; Teodorovic, Dusan B.; Quality of Service Negotiation in Next Generation Networks.8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS 2007). 26-28 Sept. 2007 Page(s):77 - 86.
[18] UDDI, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec.
[19] Web Services, http://www.w3.org/2002/ws/.
[20] Dynamips, http://www.ipflow.utc.fr/index.php/Cisco_7200_Simulator.
[21] Dynagen, http://dynagen.org/.
[22] E. Rosen, E. and Rekhter, Y., BGP/MPLS IP Virtual Private Networks (VPNs) - RFC 4364. 2006.