The Concepts and Principles to Design a Performance Management Framework for the ODP-NGN

D. Magaya and H.E. Hanrahan
Centre for Telecommunications Access and Services
University of the Witwatersrand, Johannesburg.
DMagaya@CellC.net

Abstract: The provision of information and communication anytime, anywhere and in any form becomes a reality through the introduction of Next Generation Networks (NGN). The NGN is an evolving reference architecture, which defines a QoS enabled and integrated network to carry real-time, multimedia, multiparty and distributed services. We refer to a NGN that uses RM-ODP concepts and principles as an ODP-NGN. A weakness of the ODP-NGN is the complexity of performance management. Unfortunately, there is a lack of framework that simplify the design and development of ODP-NGN performance management applications. This paper is part of an on-going project which aims to develop a performance management framework for the ODP-NGN. This paper presents the concepts and principles that can be used to design a performance management framework for the ODP-NGN. The RM-ODP enterprise, information and computational view points are employed for this purpose. An experiment in which performance management application developed using the PMF is used to manage parts of the SATINA platform will be carried out.

Keywords: TINA, DPE, RM-ODP, Concepts, Principles, Performance Management, Framework, Objects, Architecture.

I. INTRODUCTION

The next generation network (NGN) is an evolving reference architecture, which uses a QoS enabled and integrated transport network to carry real-time, multimedia, multiparty and distributed services. The NGN service architecture comprises a set of concepts, principles and guidelines for constructing, deploying, operating services in the NGN environment. Although the NGN provides an advanced architecture, its disadvantage is the heterogeneity of services and transport networks. The NGN addresses this problem through the use of Open Distributed Processing (ODP) concepts to implement a distributed processing environment (DPE) that abstracts complexities of the underlying heterogeneous networks. A NGN that uses ODP concepts is referred to as an ODP-NGN. Although the ODP-NGN employs better and advanced concepts in its service architecture (SA), unfortunately, it introduces complex Performance Management (PM) problems. For instance, the use of ODP leads to PM problems inherent in both open systems and distributed computing environments.

A companion paper presents a framework that is used to simplify the design and development of ODP-NGN PM applications for the ODP-NGN [105]. A performance management framework (PMF) is a reusable design of all or part of a system that provides some generic guidelines and basic functionality in a specific domain such that applications developers can customize and increase functionality by extending the framework. The ODP-NGN PMF is made up of a hierarchy of abstraction levels as shown in figure 1. A detailed explanation of figure 1 is found in [105].

The second level comprises of the concepts, policies and principles that are employed to define a PMF. The RM-ODP View Points and the ODP-NGN system definition are used for this purpose. The benefits of using viewpoints as an abstraction method are explained in [105]. The ODP-NGN PMF utilises the first three viewpoints of the RM-ODP namely, the Enterprise, Information and Computational viewpoints. The other view points are outside the scope of the project.

The rest of the paper is organised as follows. Sections II, III and IV present the enterprise, information and computational view points respectively. Section V provides the ODP-NGN network definition i.e. the details of the CA, NA and SA. Section VI presents the conclusions.

II. THE PMF ENTERPRISE VIEW POINT

The Enterprise View Point (EVP) defines PM objectives, rules, roles and policies from an enterprise perspective. The scope refers to the extend to which
the managed system is influenced by the performance management system. The roles refer to the task performed by the performance management system to ensure that the elements specified in the scope meet the performance management objectives. Policies refer to the broad generic rules that must be met in the process of achieving the performance management objectives. Enterprise (PM) specifications can be placed on the enterprise objects and/or their interactions. The EVP of the PMF shows the functions that are provided without prescribing how these functions are implemented. An EVP is the at the most abstract level of detail appropriate for modelling the ODP-NGN PM system concerns.

A performance system is a grouping of objects intended to achieve a specified performance management purpose. Roles of objects within the performance system are governed by policies where a policy is permission, prohibition or obligations. The EVP is concerned with performative activities that influence the PMF policy. These are activities that result in obligations, prohibition or permission. The ODP-NGN PMF EVP is effected by designing a template which is populated according to the PM requirements (table 1). The table variables can be changed to suit the particular ODP-NGN PM circumstances.

### The EVP PM Modelling Template

<table>
<thead>
<tr>
<th>The EVP Variable</th>
<th>User Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. Performance system members</td>
<td></td>
</tr>
<tr>
<td>ii. PM objective for system</td>
<td></td>
</tr>
<tr>
<td>iii. PM scope for system and members</td>
<td></td>
</tr>
<tr>
<td>iv. PM role for system and members</td>
<td></td>
</tr>
<tr>
<td>v. PM policy for system and members</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Obligations</td>
</tr>
<tr>
<td></td>
<td>Permissions</td>
</tr>
<tr>
<td></td>
<td>Prohibitions</td>
</tr>
</tbody>
</table>

Table 1: The EVP PM Modelling template

The EVP may also specify management domains. A domain gives the means for describing management policy for a group of managed objects rather than having to do this for each object.

### III. THE INFORMATION VIEW POINT

The Information View Point (IVP) models performance relevant information that flows between service components, users, network providers and service providers. Essentially, the IVP seeks to define how performance management is specified relative to information flow in the system. This viewpoint specification captures the enterprise level PM requirements and information required to support PM policy (table 2). The IVP allows the modelling of performance information in an application, technology and system independent manner. It facilitates the production of a PM information model for the ODP-NGN. Once the information model is available, the data can be manipulated in an application specific manner. Table 3 relates a PM objective to the corresponding information model. For instance, an objective to reduce access delay requires an information model for delay response time. An analysis of the ODP-NGN PMF IVP results in the definition of the information objects. There are many ODP-NGN information objects that can be identified and the relationship between them established using concepts such as GDMO, GRM and OMT[80].

<table>
<thead>
<tr>
<th>PM Objective</th>
<th>IVP information Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce Access Delay</td>
<td>Delay</td>
</tr>
<tr>
<td>Increase Service Speed</td>
<td>Throughput</td>
</tr>
<tr>
<td>Reduce Service Variability</td>
<td>Jitter</td>
</tr>
</tbody>
</table>

Table 2: The Information View Point Objectives

The key elements of the ODP-NGN PMF information view points are the performance relevant component, the performance relevant event and the performance relevant packet unit.

#### 1. The Performance Relevant Component

The Performance Relevant Component (PrC) represents the ODP-NGN component that is affected by performance commands. Table 3 shows the typical PrC in the service architecture and computational architecture. Monitoring will be done on the PrC. The PrC represents a managed object.

<table>
<thead>
<tr>
<th>PrC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer</td>
<td>Retailer</td>
</tr>
<tr>
<td>SA</td>
<td>Consumer Domain</td>
</tr>
<tr>
<td>Terminal</td>
<td>Retailer Domain</td>
</tr>
<tr>
<td>CA</td>
<td>Node</td>
</tr>
<tr>
<td>Capsule</td>
<td>Capsule</td>
</tr>
<tr>
<td>Service COG</td>
<td>Service COG</td>
</tr>
</tbody>
</table>

Table 3: The performance relevant components

#### 2. The performance relevant event

We propose to measure the PrC behaviour in terms of events, which we generically call a Performance Relevant Event (PrE). The PrE reflects changes that are of significance to performance management. Event driven monitoring is based on obtaining information about the occurrences of PrEs which provide a dynamic view of the ODP-NGN performance activities since only data relating to changes in the system are collected. The PrE is used to model and calculate time-based specifications e.g. the response time of a service access request. We acknowledge that an approach based on defining a Performance relevant Status (PrS) and using the status to measure PrC behaviour can also be employed. A more elaborate solution can be developed by combining the PrE and PrS to determine PrC behaviour. However we decided to use PrE only for two major reasons. Firstly, it is simpler since we only have to deal with a change in status (PrE occurs whenever there is a change in status). Secondly, the tools (PEPA and fuzzy logic) used in the PMF Facilities modelling can be easily related to the PrE.

#### 2. The Performance relevant Packet Unit (PrPU)
We introduce the concept of a PrPU (Performance relevant Packet Unit). A PrPU is any packet of data e.g. cell, frame, byte, message or octet which possesses some performance relevance information. The size of the PrPU is undefined because it depends on the data type, the application type, the architecture etc. The PM facilities such as monitoring are performed on PrPU. The PrPU is used to model and calculate flow-based performance specifications e.g. throughput. The performance information obtained from the PrPU is used to develop a performance specification (figure 1).

Figure 1: The Performance specification hierarchy

Figure 1 shows a model for the PM profiles, signatures and measurement. The input/output layer shows how the user specifies performance using linguistic variables such as audio, colour, and speed and waiting time. These specifications are made at the input/output interface or the terminal equipment. The softer control layer maps the user specifications into performance characteristics. The soft control layer maps the profile information into performance characteristics. The hard control layer maps the characteristics into real PrPU measurements.

- **Measurement**: A measurement is an identifiable or measurable aspect of a PrE. Several measurements such as response time, frequencies etc constitute the hard PM layer of information.
- **Characteristic**: A characteristic is a set calculated or derived from one or more measurements. Characteristics such as delay, throughput etc form the soft PM layer
- **Profile**: A profile is a set of one or more characteristics per given application or data stream. Profiles such as video throughput, video and delay form the softer PM of information.

**IV. The Computational View Point**

The Computational View Point (CVP) is a progression of the EVP and the IVP described in sections (II) and (III) respectively. The CVP addresses the functional decomposition of the PM system into units of distribution and the interactions required to manage performance. The units of functional decomposition are called objects and are used to develop PM applications. The three object definitions that we explore are:

1. **The PMF object model**

   The PMF object model refers to three object categories which are; Managed Objects (MdO), Manager Objects (MrO) and Management Objects (MtO) (figure 2). This project primarily focuses on defining MtO and MdO objects.

   - Managed objects (MtO) : The MtO is an abstraction of the CA, NA or SA objects from a PMF point of view. It represents the resources of interest in the aspects relevant for performance management purposes. For an example, the objects of a service implementation in the SA, the network elements in the NA and the capsule in the CA. Each MtO has got a service interface that sends performance management operations on to the underlying resource and that receives management information such as attribute values and notifications. A managed object is an entity to which a management policy applies and whose behaviour can be monitored or changed by a manager object.
   - Manager Objects MrO : The MrO is an abstraction of the management application as seen from a PMF point of view. Manager objects administer one or more MtOs through their management interfaces. Manager objects monitor the activities of managed objects, make management decisions based on that information, and perform control actions on managed objects. A manager may be human or automated. Managers may also be managed objects for higher level managers. Specific manager objects are the monitoring, control and analysis facilities [105].
   - Management Objects MtO : They represent the PMF from an implementation perspective. They give the possibility to intervene in the management process by the authorized user. To this end, management objects provide graphical user interface for the human user.
and mediate user input to manager object as well as information emitted from the manager.

Figure 3: The PMF Computational Objects

2. The PMF Object Behaviour and interface Specification

We use the template object model (figure 3) to describe the ODP-NGN PMF object behaviour and interface specifications. Interfaces in the computational viewpoint consist of a signature, a profile, behaviour and an environment contract. An interface signature and profile represent the syntactic aspects of the functionality (behaviour) found in the interface. The behaviour specification represents what the effects are of invoking the actions described in the signature. The environment contract represents aspects associated with the interface that its signature and behaviour specification alone do not capture. These might relate to management contracts.

Figure 4: The Performance Management Framework Object template

The object behaviour and interface specification determines whether an object behaves as a $M,O$ or as a $M,O$. We describe the management interface, operational interface, stream interface and binding interface.

The design of management applications requires the specification of management interfaces. The level of specification granularity is dependent on the requirements of the management application. For instance, the management application may require behaviour modification which entails, invoking management operations on individual services, the objects that build the applications, the cluster that contains the object, the capsule that contain the cluster and the node that supports the capsule. Figure 4 shows the three types of operations provided by a management interface which are:

- Control commands
- Requests for status information
- Monitoring information generated by the object

The control commands are directed by the manager object to the managed object; requests for information are issued by the managing object and result in a reply from the managed object. Monitoring information generated by the object allows the transfer of unsolicited notifications from the managed object to the manager.

Figure 5: The managed Object Interfaces

The second interface shown in figure 5 is an operational interface. The Operational Interface supports the normal information processing operations, fulfilling the main purpose of the service provided by the object.

3. The $M,O$ PMF Contract

PM is defined and managed using environment contracts of the object behaviour and its interface specifications. The environment contract expresses the requirements that have to be fulfilled and it addresses issues such as performance of objects and indications on how behaviour can invalidate the contract. Every $M,O$ has got a management contract that is a definition of the access rights for attributes, executable operations and imitable notifications. Thus a contract object controls the transparency of management services interface with respect to a certain user or user role and a dedicated service. There are a few things that form part of a management contract and these are, the relation, the requirements, the capabilities and the offers. The contracts are based on signatures and profile i.e. a contract to meet given signatures or profiles (figure 1).

V. THE ODP-NGN DEFINITION

An essential part of the concepts, principles and roles implementation layer (figure 1) is a framework that defines the ODP-NGN. A detailed definition of the ODP-NGN is provided in [105]. The ODP-NGN can be sub-divided into three architectural components which are, the computational architecture (CA), network architecture (NA) and service architecture (SA).
The purpose and of scope of performance in the CA

The purpose and of scope of performance in the SA

The purpose and of scope of performance in the NA

<table>
<thead>
<tr>
<th>Enterprise Model</th>
<th>Information Model</th>
<th>Computation model</th>
</tr>
</thead>
<tbody>
<tr>
<td>The purpose and of scope of performance in the CA</td>
<td>A model of the performance information for the CA.</td>
<td>The computational objects required to manage the CA performance.</td>
</tr>
<tr>
<td>The purpose and of scope of performance in the SA</td>
<td>A model of the performance information for the NA.</td>
<td>The computational objects required to manage the SA performance.</td>
</tr>
<tr>
<td>The purpose and of scope of performance in the NA</td>
<td>A model of the performance information for the SA.</td>
<td>The computational objects required to manage the NA performance.</td>
</tr>
</tbody>
</table>

Table 4: The PMF view points and the ODP-NGN architecture components

Table 4 illustrates how the RM-ODP view points are applied to the components of the ODP-NGN from a PMF concepts, policies and principles perspective. The view points are applied to the ODP-NGN’s CA, NA and SA.

### 1. The SA, CA and NA Performance management

**The SA performance management**

The ODP-NGN and TINA service architectures are similar in that they provide a framework to offer a generic set of functionalities in a stake-holder and service independent manner. This is achieved by defining service components [1]. Performance management in the SA is achieved through the management of service components to meet the required performance specifications. The SA specifies components that provide a framework for subdividing the functionality of ODP-NGN systems. The service components are high-level abstractions which can be decomposed into COs and COGs.

The CO is defined as a unit of distribution over the DPE node. In cases where the service component completely maps to a CO, it means a single DPE node supports the service component functionality. The CO can be grouped into Computational Object Group (COG) which is shown in figure 15. COGS are not a unit of DPE distribution but are defined in terms of their internal COs and/or COGs. Describing the CO types that form the group specifies a COG type. However, the COG do not encapsulate their internal structure. Service components and their interfaces map into COs and COGs and CO interfaces and COG group contracts respectively (figure 6). There are several CO mappings of a service component thus allowing for flexibility in design. Each mapping may provide some benefit to the designer e.g. improving system performance by allowing a special type of deployment of COs. Figure 15 shows an example of a COG. The SA performance management is effectively achieved by managing the COG performance provided that the COG maps in the SA service components which are of performance relevance i.e. PrCs.

**CA Performance Management:**

The CA management is subdivided into the management of the software, DPE, computing systems and the kernel transport network. DPE management includes the management of the components of the DPE model i.e. the management of nodes, clusters, capsules and eCO. Computing systems management includes the management of computing systems that are supporting the DPE and that provide the communications and computing capabilities used by the DPE, generically called the Native Computing and Communications Environment (NCCE). The kernel transport network deals with the management of the telecommunication network that is interconnected the DPE nodes. The software management deals with activities such as deployment, configuration and instantiation of software. The ODP-NGN PMF is limited to the performance of the nodes, clusters and capsules.

**The NA performance management**

The network architecture comprises of the resources required to transport, transmit, switch route and relay the ODP-NGN traffic. The TINA network resource architecture provides a comprehensive description of the NA. Performance management in the NA involves the setting up of appropriate network elements, network layers and network domains to achieve the specified QoS.

### VI. Conclusion

The concepts, principles and policies required to design an ODP-NGN PMF have been presented. The RM-ODP enterprise view point, information view point and computational view point are used to achieve this objective. The principles that define the ODP-NGN components are presented. The TINA computational architecture, the network architecture and the service architecture are employed. Experimental work to verify the ODP-NGN PMF concepts and principles is in progress.

### VII. References


[27] Nahorstedt K., Li B., A Control-Based Middleware Framework for Quality of Service Adaptations


[50] Znaty S., Arsenis S., Sclavos J., ODP Viewpoints of QoS Management Application


