TSG-RAN Working Group 3 meeting #6
Sophia Antipolis, 23-27 August 1999

TSG R3#6(99)962

Agenda Item: 16.3, (21)

Source: Siemens / Italtel

Title: NBAP Procedures for object oriented logical O&M

Document for: Approval

1 Introduction

This contribution is based on the contribution 'Info Model and State Management Functions for NodeB logical O&M', which introduces an information model for the 'Manageable Logical NodeB. With this contribution the transport protocol at the object oriented interface is described.
For realisation of the transport protocol it is proposed within this document to enhance the already defined NBAP procedures for the transport mechanism of operation requests, operation responses and notifications between NodeB and RNC.
The support of the info model by a standardised protocol like SNMP or IIOP is presently not required but should be for further study. The reasons therefore are described within this document.

2 Description

2.1 Distributed Objects and Data Copies

Logical resource objects are distributed over the RNS. The NodeB logical resource objects are 'duplicated', they have one representation on NodeB itself and the other representation on RNC as part of RNS logical resources (see **Figure 1**). E.g. the logical resource object *CellXY* can be found on RNC and a copy is located on NodeB itself. The managed object representation of NodeB logical resource on RNC acts in two roles. From point of view of the RNC Management System it acts as an management agent. From point of view of NodeB it acts as a manager.

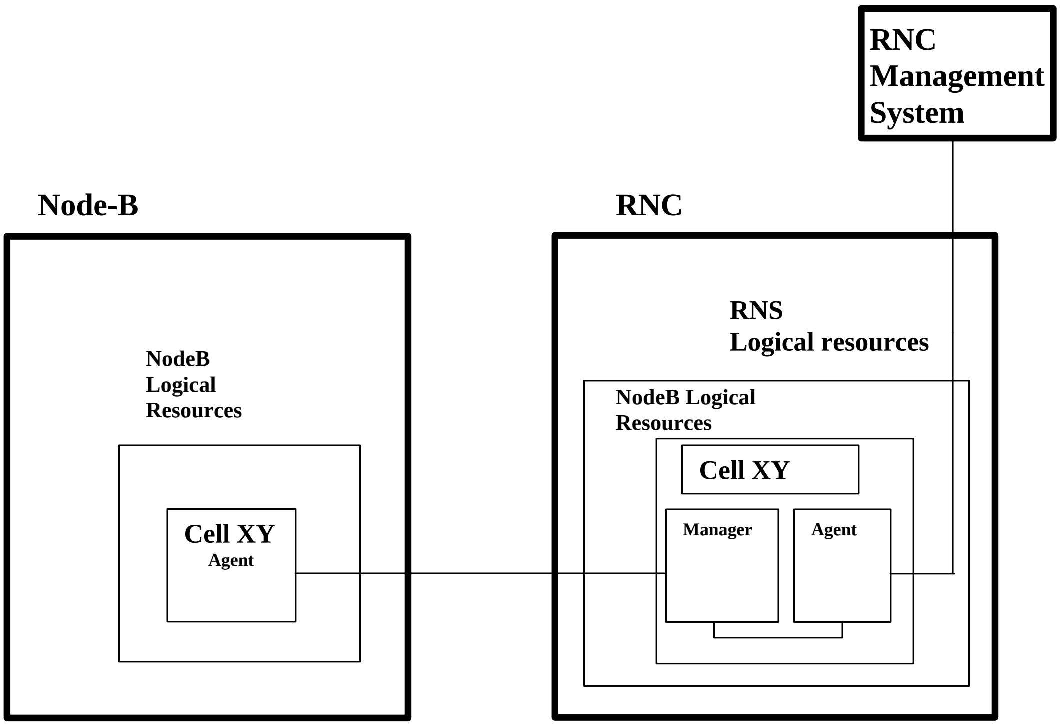

Figure 1:: Distributed Objects

A protocol is needed which provides a communication mechanism between the managed object on NodeB and its Manager on RNC.
The protocol has to:

- send operation requests from an object representation on RNC to its associated object representation on NodeB and to return the result of the operation back to the RNC object (see **Figure 2**)
- send the notifications from object representation on NodeB to its associated object representation on RNC via an event handler (see **Figure 3**)

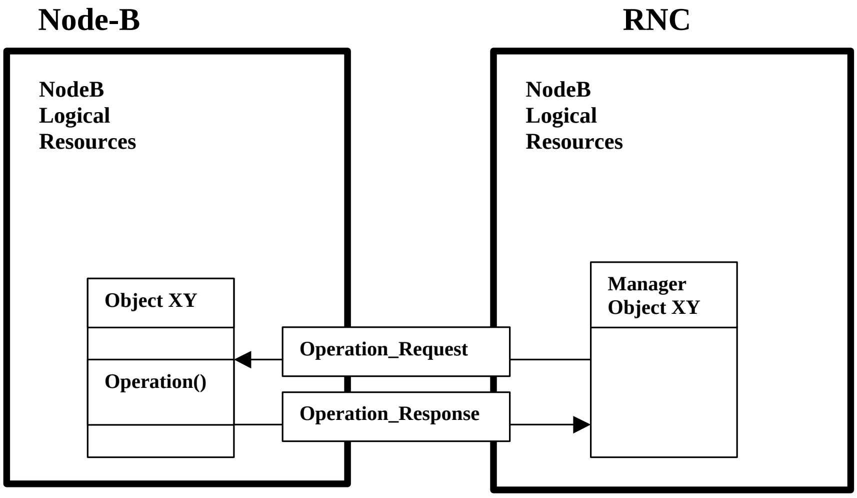

Figure 2: Transport of an Operation Request and Operation Response

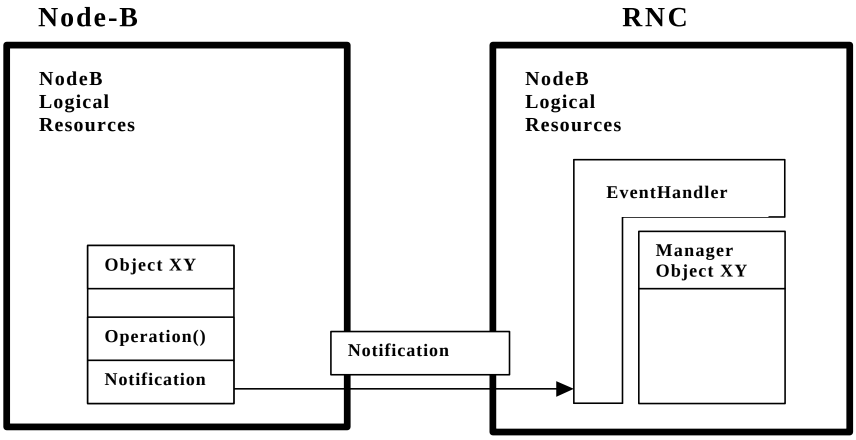

Figure 3: Transport of a Notification

.
Within this contribution we propose to enhance the NBAP management procedures for O&M to provide an object oriented communication mechanism.

Usage of SNMP or IIOP (CORBA):
When a standardized management protocol like SNMP or when the CORBA protocol IIOP for distributed objects is used at the Iub interface, an IP protocol stack below the SNMP / IIOP protocol is required. This can not be done on top of NBAP procedures so the already defined architecture has to be re-defined.
In addition, the usage of ITU-T X.7xx standards like ITU-T X.731 requires an object model according GDMO / ASN.1. How such an info model can be supported by an SNMP / IIOP protocol is FFS.

2.2 Derived NBAP O&M messages from Information Model

If the NBAP O&M procedures are used for an object oriented interface, the messages must be enhanced by some parameters:

1. The source object and the destination object must be specified. If NodeB and RNC use the same numbering scheme for object identifiers, the destination can be derived from the source.
2. The relationship between a message transporting an operation request and the message transporting the related operation result must be given in the messages.
3. The operation which is requested from the source object on the destination object or the notification which is send by the source object to the destination object must be defined.

2.2.1 Object oriented O&M NBAP message format

An object oriented enhanced O&M NBAP message comprises the following parameters:

1. Message Discriminator
2. The MessageType parameter for object oriented O&M NBAP procedures has the following values:
   OperationRequest
   OperationResponse
   Notification
3. TransactionID

4. The *Source* specifies the source object and its class of the operation request, operation response or notification unambiguously.
   Note: NodeB and RNC can use the same scheme for associating the object identifiers to the objects. In this case *Source* is not necessary, because the *Source* is given by the *Destination*.
5. The *Destination* specifies the destination object and its class for the operation request, operation response or notification unambiguously. If the operation is a create() operation, *Destination* specifies the superior (containing) object.
6. The *JobNo.* identifies the relationship between an operation request and an operation result message. It is not present for *MessageType*: notification.
7. The *ContentID* specifies the operation / notification which is transported within the NBAP message unambiguously within the managed object class, which is defined in *Destination*.
   Note: If the *ContentID* contains a *create()* operation, the parameter *Destination* specifies the superior (containing) object and its class. The class of the *create()* operation itself must be specified separately in *ClassID*.
8. *ClassID* is only present if *ContentID* specifies a create() operation. In this case ClassID identifies the object class for the create() operation.
9. The parameters of the operation / notifications as defined in the information model

Information Element	**Reference**	**Type**
Message Discriminator		M
MessageType		M
Transaction ID		M
Source		M
Destination		M
Job No.		Conditional
ContentID		M
ClassID		Conditional
Operation / Notification Parameter 1		M / O
...		M / O
...		M / O
Operation / Notification Parameter n		M / O

Table 1: Object Oriented O&M NBAP message format

ClassID, Operation and Notification and their parameter are defined within the Information Model.

2.3 Comparison currently defined NBAP messages with oo O&M NBAP messages

Below the current NBAP SETUP FACH message is compared as example with a message of an object oriented interface which transports the FACH.create() operation.

OO NBAP message	Fach. Create()		**Current NBAP message**	Setup FACH
Information Element	**Type**		**Information Element**	**Type**
Message Discriminator	M		Message Discriminator	M
MessageType: *OperationRequest*	M		Message Type	M
Transaction ID	M		Transaction ID	M
Source: *Carrier*	M		Cell ID	M
Destination: *Carrier*	M		Cell carrier ID	M
Job No.	M		DL scrambling code ID -FFS	M
Operation ID: *Create()*	M		DL common transport channel ID	M
ClassID: *FACH*	M		DL common transport channel type	M
DL scrambling code ID –FFS	M		DL channelisation code number	M
DL channelisation code number	M		DL channelisation code spreading factor	M
DL channelisation code spreading factor	M			
AdminState	O			

3 Conclusion

It is proposed to insert the content of chapter 2.2.1 in this contribution to chapter 9.1.x in TS25.433. The object oriented logical O&M procedures can be derived from the Information Model using the presented NBAP transport mechanism.