Integrated Network Management Agents of IMT-2000 based on ASIB Component

Soo-Hyun Park

School of Information Management, Kookmin University,
861-1, Chungnung-dong, Sungbuk-ku, Seoul, 136-702, Korea,
Tel : +82-2-910-4559, Fax : +82-2-910-4519,
E-mail: shpark21@kookmin.ac.kr

Abstract: *By managing various communication networks collectively, Telecommunication Management Network (TMN) has appeared as a concept to aim for the unified and effective communication network operation and maintenance. IMT-2000 and other networks such as PSTN, cellular networks and packet switching network that are integrated with IMT-2000 have been developed in different platforms - hardware and operating system, and a same circumstance will happen in the maintenance in the future. Also, it is difficult to approach to the standard for Q3 interface implementation of the agent in TMN system that may occur in the development or the maintenance for the different platforms of IMT-2000 and other networks that are integrated with IMT-2000. In order to solve this kind of problems, this paper suggests integrated network management agent of IMT-2000 based on the NTS model and Advanced Intelligent Network(AIN).*[1]

1. Introduction

By managing various communication networks collectively, Telecommunication Management Network (TMN) has appeared as a concept to aim for the unified and effective communication network operation and maintenance. Since TMN has been developed by different operating systems and different versions of hardware platforms in the implementation process, several problems have been found in the step of developing and maintaining the class of TMN system agent[1][2][3]. Especially, in the case of IMT-2000 which is the mobile communication system that connects the various kinds of communication services that are supported by the fixed network and other mobile communication services such as Public Switching Telephony Network (PSTN), Personal Communication Network(PCN), Digital Cellular Network(DCN), and B-ISDN with various wireless links by the request of the mobile communication users, these kind of problems are more complicated. The main problem is that the support of the multi-platform becomes impossible, and eventually, it is hard to provide the consistent interface in case of developing Q3 interface between Data Communication Network(DCN), Operation System(OS), Mediation Device(MD), and Network Element(NE)[4][5]. By constructing IMT-2000 network based on Advanced Intelligent Network(AIN)[6][7], it becomes possible to combine and integrate with the components of the intelligent network, and it can easily provide the services of the wired intelligent networks that have been developed in the past or that will be developed in the future. However, IMT-2000 and other networks such as PSTN and PCN that are integrated with IMT-2000 have been developed in different hardware and operating system, and a same circumstance will happen in the further maintenance in the future. One of the main problems is that the agents in TMN cannot have the standard for Q3 interface implementation, and they cannot support the multi-platform. Furthermore, the compatibility for the maintenance & administration system for the different networks that are integrated with IMT-2000 is not guaranteed[1][2].

Third, since the different networks are maintaining the different network management system, the compatibility between the operation and maintenance systems is not guaranteed in case of implementing TMN that is the concentrated network management system.

In order to solve this kind of problems, this paper suggests integrated network management agent of IMT-2000 based on the NTS model and Advanced Intelligent Network(AIN). The NTS model is a model that integrates *Applicable SIB Repository(ASR)* based on Platform Independent Class Repository(PICR) and the concept of Service Creation Sequence Model(SCSM) that is the intelligent service generation model in AIN)[1][2][3].

By using this model, we can create the new services on the communication network by using Service Independent Building Blocks(SIBs) that is already defined, and if it is impossible to create the new communication service for the previously defined SIB, it includes the function to store the newly made SIB into the database of Service Management Part(SMP) / Service Creation Environment Point(SCEP)[6][7]. Not only for the new communication service, but in order to create the application program that composes the network components such as Network Element Network Management System(NE NMS) Agent, the concept of Applicable SIB(ASIB) is needed. By applying this concept, the applications such as NMS agent

[1] This paper is supported by the internal research fund of Kookmin University, Korea.

designed by the NTS model by using ASIBs that is stored in SMP ASR can be composed[6][7].

2. Applicable Service Independent Building Block Componentware

2.1 Basic Definitions

It is very easy approach to define ASIB componentware near from the the formalism of the NTS model which is composed by the definitions and the descriptions written below. In the NTS model, the objective and abstract objects of the real world can be described as the entity nodes, and they can be defined as follows. In order to describe the real world object, the elements such as the name of the object, that attribute that the object has, and the view point of the object, must be described, and in order to reflect the concept of the "Intelligence", the attribute for the loading type is needed. That is why the entity node is defined as a structure. The structure mentioned here is not the structure that is described in the predicate logic or the first order logic.

【 Definition 1 】 Aspect Node Structure

If S is ***the set of the aspect structure***, it can be defined as the following.

$\forall s \in S, s = <$ ASPid, OWNER $>$

where, ASPid : Name of aspect s

OWNER : Set of the entity node name that has s as the aspect □

【 Definition 2】 Attribute Set of Entity Node Structure

The *attribute set* A of E can be defined as the following structure.

$\forall \forall a \in A, a = <$ Aid, AT $>$

where, Aid : Attribute name

AT : Attribute type set

AT $\in$ { char, string, integer, real, boolean } □

【 Definition 3 】 Entity Node Structure

If E is ***the set of the entity node structure***, E can be defined as follows.

$\forall e \in E,\ e = <$ Eid, A, S, LT $>$

where, Eid : Name of the entity node e

A : Attribute Set

S : It is the view that e has, and it signifies the set of the aspect node name(id)

LT : It is the loading type of the entity

LT $\in$ { Dynamic, Static, none } □

The concept of the attribute used on the entity node structure of the **Definition 3** can be defined as the following. The attribute set of the entity node structure manages the main role on the specialization abstraction concept definition of the NTS model. The reason is that the entity nodes that have been abstracted by the specialization link have inherited the attribute set o the upper node(class). The NTS model defines the attribute(A) simply among the elements that compose the entity node structure. The reason is that the element that reflects the Loading Type(LT) attribute which reflects the integration concept about the outsourcing this entity either dynamically or statically from the independent class repository becomes independent.

The algorithm related to the creation and the deletion of the entity used in the NTS model is like the following,

[Algorithm] create_entity(*Entity*) // Creation of new entity

1 For the set of the entity structures $E = \{ e_i \mid 1 \le i \le n \}$, **FOR** i = 1 **TO** n

1.1 **IF** ***Entity***.Eid = e_i.Eid **THEN**

1.1.1 **IF** ***Entity*** is uniformity entity of e_i **THEN**
Create Entity as the uniformity entity of ei.

1.1.2 **ELSE**
error_print(" Already exist... ")
Return

1.1.3 **END IF**

1.2 **END IF**

END FOR

// Assignment of the attribute

2 e_{n+1}.Eid ← ***Entity***.Eid
e_{n+1}.A ← ***Entity***.A
e_{n+1}.S ← ***Entity***.S
e_{n+1}.LT ← ***Entity***.LT

3 $E \leftarrow E \cup e_{n+1}$

4 Return □

[Algorithm] delete_entity(*Entity*)

// Delete the current entity

// If the entity is deleted, delete the attribute that the entity has and the relationship of the entity.

1 For the set of the entity structures $E = \{ e_i \mid 1 \le i \le n \}$

1.1 **IF** there_exist(***Entity***) = FALSE **THEN**

1.1.1 error_print(" There does not exist..... ");

1.1.2 exit;

1.2 **END IF**

2 $E \leftarrow E - \{$ ***Entity*** $\}$

3 Delete the subtree of ***Entity***.

4 Delete the uniformity entity of ***Entity***.

5 Return. □

The aggregation/specialization abstraction concept between the entity nodes must be made through the aspect node. The aspect node doesn't effect the abstraction concept

between these entity nodes. When the aggregation / specialization abstraction concept is applied to the super class node, the viewing point of this node is provided, so that the standard to process the abstraction can be determined. By applying the aspect concept to the abstraction concept of the entity node, the following theorem can be defined.

The representative entity defined in the abstraction concept of the multiplicity can have its instances. These instances are IM-ASIB componentware type entity node and OM-ASIB componentware type entity node, and they can be connected to the representative entity by using the multiplicity instance link. The definition for the multiplicity instance link is as follows.

[Definition 4] Multiplicity Instance Link

The multiplicity instance link, L_M, can be defined by the following structure.

$$L_M = < E_M, E_C, \phi >$$

where,

E_M : Entity node set that is abstracted by the multiplicity link

E_C : It signifies the entity node set that has only the elements that have the following characteristics among the elements of the entity node set $E = \{ e_i \mid 1 \le i \le n \}$.

1) for $\forall e \in E$, e.LT = Dynamic $\vee$ e.LT = Static
(It signifies the set of the elements that have dynamic or static loading type, that is, the set of OM-Component type entity node and IM-Component entity type node.)

2) em = super_class_of(Ec)
(for $\forall e_c \in E_M$, the super class entity node of e_c is e_m)

$\phi : E_M \rightarrow 2^{Ec}$, such that,
$\phi(em) = \{ e_c \mid$ *Aspect-Object* e_c *uses the component* $e_m \}$ □

By the multiplicity abstraction concept, the algorithm hich carries out mapping between representative entity and OM/IM-Component type entity node through multiplicity instance link is like the following. It is trivial to prove the correctness of the multiplicity conversion algorithm.

[Algorithm] multiconversion() // Multiplicity Conversion Algorithm

1 Let e be an entity class in super class AO.

2 Check abstraction type and OWNER to be applied to e.

3 **IF** abstraction_type_of(e) = multiplicity **THEN**
// Generate representative class to reflect multiplicity abstraction concept.

3.1 Create Erep which is new AO to have properties like the following.
3.1.1 E_{rep}.Eid ← e.Eid // Use entity id of entity class e according to that
3.1.2 E_{rep}.A ← e.A // Inherit attributes of super class e
3.1.2.1 E_{rep}.A.Aid ← e.A.Aid
3.1.2.2 E_{rep}.A.AT ← e.A.AT
3.1.3 E_{rep}.LT ← none

3.1.4 **IF** Erep has any ASPECT **THEN**
3.1.4.1 Create aspect node set which has views of E_{rep}.
3.1.5 **END IF**

// Generate IM-Component type entity node and OM-Component type entity node connected to Multiplicity Instance Link
// n : number of IM/OM-Component type entity node to be created

Let $E = \{ e_i \mid 1 \le i \le n \}$ be set of IM/OM-Component type entity structure
3.1.6 **FOR** i = 1 **TO** n
3.1.6.1 e_i.Eid ← new entity id // Assign new entity id

3.1.6.2 e_i.A ← e.A // Inherit attributes of super class e
3.1.6.2.1 e_i.A.Aid ← e.A.Aid
3.1.6.2.2 e_i.A.AT ← e.A.AT
3.1.6.3 e_i.LT ← none
3.1.7 **END FOR**

4 **ELSE**
4.1 **exception_handling();**

5 **END IF** □

2.2 Definition of ASIB

The NTS model has the following type concept by focusing on SIB that is the core concept of SCSM. SIB has the standardized interface, ASIB formal structure is defined by the following. ASIB formal structure is defined as the structure defined by predicate logic.

[Definition 5] Applicable Service Independent Building Block(ASIB) Formal Structure

If S_{Ib} is *the set of SIB structures*, it is defined as the following.

$\forall b \in S_{Ib}$, $b = < Sid, S_f, S_v, I, O, \gamma_n, \Gamma_n, L_p >$

where,

Sid : name of SIB

S_f : Service Feature Set
It signifies the service characteristics defined by AIN service plane, and the components are different depending on the case of AIN CS-x.

S_f = { ABD, ATT, AUTC, AUTZ, ACB, ... , TCS, TDR }

S_v : Set of services to be applied

Set of AIN services that this SIB will be applied to

S_v = { ABD, ACC, AAB, CD, CF, CRD, CCBS, CON, ... , MAS, APP }

I : It is the set of input data, and it has the following elements.

I = { Service_Support_Data, Call_Instance_Data, C_{lsp} }

O : It is the set of output data, and it has the following elements.

O = { Call_Instance_Data, C_{lep} }

γ_n: $S_v \rightarrow S_f$

In order to create each service, it defines the relationship between the service features defined in the service plane and the service.

From $S_v = \{ v_i \mid 1 \leq i \leq n \}$ and $S_f = \{ f_j \mid 1 \leq j \leq n \}$, For $\exists v_i \in S_v$, $v_i \leftarrow SSP(\exists f_j \in S_f)$

Where, SSP : Service Logic Program

Γ_n: $S_f \rightarrow$ **Sib**

It defines the relationship between the service features defined in the service plane and SIBs of Global Function Plane (GFP) that are needed to support these services features.

From $S_f = \{ f_i \mid 1 \leq i \leq n \}$ and $S_{lb} = \{ b_j \mid 1 \leq j \leq n \}$, For $\exists f_i \in S_f$, $f_i \leftarrow GSL(\exists b_j \in S_{lb})$

Where, GSL : Global Service Logic

L_p : Logical Point Constant Set

Logical Point Constant Set is defined by the following.

L_p = { C_{lsp}, C_{lep} }

where, C_{lsp} : Logical Start Point Constant

C_{lep} : Logical End Point Constant

From this defintion, *Applicable SIB (ASIB)* is defined as SIB that has the following conditions.

1) If A_{sib} is *the set of Applicable SIB (ASIB)*, then

$A_{sib} \subset S_{lb}$

2) For $\forall b \in S_{lb}$, $\forall a \in A_{sib}$, iff {APP} $\in$ b. S_f □

ASIB does not signify SIB that is used for service creation, but it signifies SIB that is used to create application programs used in TMN agent[8][9][][10] or in switching system. ASIB is created changing the intermediate entity node that is defined by the NTS model and the leaf nodes such as ILB and OLB by using NTS algorithm. ASIB is defined as SIB elements that satisfy a lot of conditions among the class of SIBs.

3. Conclusion

Component-based Development (CBD) is the proper embodiment tool for implementation of systems that are designed by the NTS model supporting ILB/OLB componentware concept practically.

As transformation from the concepts defined in the NTS model to the interfaces of the Interface Specification Model(ISM) which is proposed in the CBD, we can have versatile implementation methodologies of the NTS model design through CORBA and JAVA packages. Furthermore, we can maximize reusability of the ASIB regarding ILB/OLB entity type nodes through the mapping from the loding blocks to interfaces. Also.

This paper suggests some defintintions and algorithm for NTS model which is accustomed to design network element agents of TMN system regarding IMT-2000 related network.

IMT-2000 network based on AIN CS-3 can easily apply the services of the wired intelligent network that have been developed in the past or that will be developed in the future by combining and integrating with the components of intelligent network. It is difficult to develop the standard for Q3 interface implementation of the agent in TMN system that may occur in the development or the maintenance for the different platforms of IMT-2000 and other networks that are integrated with IMT-2000. In order to solve this kind of problems, this paper suggest the NTS model based on AIN ASIB. The NTS model signifies the front step for developing the network management system and application system based on TINA in the future.

References

[1] Soo-Hyun Park, "Applicable SIBs of ASR for Network Management Agents Design", In Proceedings of The 2001 International Technical Conference on Circuit/Systems, Computers and Communications (ITC-CSCC'2001), pp. 494 - 497, Tokushima, Japan, 2001

[2] Soo-Hyun Park, "Integrating Heterogeneous Network Elements via IMT-2000 by Using ASIB of AIN", *In Proceeding of The 2001 International Symposium on Future Software Technology (ISFST'2001)*, pp.124 - 127 Ghengzou, China, 2001

[3] Soo-Hyun Park, Sung-Gi Min, "ISM Conversion Algorithm for the Definition of Distributed Objects IDL", *In Proceeding of The 2000 International Symposium on Future Software Technology (ISFST'2000)*, pp.315 - 317, Guiyang, China, 2000

[4] ITU-T, Draft Recommendation Q.FNA, "Network Functional Model for IMT-2000", USA, Sep., 1998

[5] ITU-T, Draft Recommendation Q.FIF, "Information Flows for IMT-2000", Geneva, Sep., 1997

[6] ITU-T Recommendations - AIN Q.1210 ~ Q.1229

[7] K.B. Choi, et al, "AIN Technology", Hong-Neung Press Korea, 1997

[8] ITU-T Recommendation M.3010, "Principles for a TMN", 1992.

[9] ITU-T Recommendation M.3020, "TMN Interface Specification Methodology", 1992.

[10] ITU-T Recommendation M.3100, "Generic Network Information Model", 1992.